JOURNAL OF FORMALIZED MATHEMATICS
Volume 2, Released 1990, Published 2003
Inst. of Computer Science, Univ. of Białystok

A Classical First Order Language

Czesław Byliński
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 [7]. Well-formed formulas of this language are defined and some usual connectives and quantifiers of [7], [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 [6]. For such purpose some auxiliary notation and propositions are introduced.

MML Identifier: CQC_LANG.

WWW: http://mizar.org/JFM/Vol2/cqc_lang.html

The articles [9], [11], [10], [12], [3], [4], [8], [2], [7], [1], and [5] provide the notation and terminology for this paper.

In this paper i, j, k are natural numbers.

Let x, y, a, b be sets. The functor $(x = y \rightarrow a, b)$ yielding a set is defined as follows:

(Def. 1) $$(x = y \rightarrow a, b) = \begin{cases} a, & \text{if } x = y, \\ b, & \text{otherwise.} \end{cases}$$

Let D be a non empty set, let x, y be sets, and let a, b be elements of D. Then $(x = y \rightarrow a, b)$ is an element of D.

Let x, y be sets. The functor $x \dot{\longmapsto} y$ yielding a set is defined by:

(Def. 2) $x \dot{\longmapsto} y = \{x\} \longmapsto y$.

Let x, y be sets. One can verify that $x \dot{\longmapsto} y$ is function-like and relation-like.

Next we state two propositions:

(5)[1] For all sets x, y holds $\mathrm{dom}(x \dot{\longmapsto} y) = \{x\}$ and $\mathrm{rng}(x \dot{\longmapsto} y) = \{y\}$.

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

For simplicity, we adopt the following rules: x, y denote bound variables, a denotes a free variable, p, q denote elements of WFF, and l, l_1 denote finite sequences of elements of Var.

The following proposition is true

(7) For every set x holds $x \in \mathrm{Var}$ iff $x \in \mathrm{FixedVar}$ or $x \in \mathrm{FreeVar}$ or $x \in \mathrm{BoundVar}$.

A substitution is a partial function from FreeVar to Var.

In the sequel f is a substitution.

Let us consider l, f. The functor $l[f]$ yields a finite sequence of elements of Var and is defined by:

[1] The propositions (1)–(4) have been removed.

(Def. 3) $\operatorname{len}(l[f]) = \operatorname{len} l$ and for every k such that $1 \leq k$ and $k \leq \operatorname{len} l$ holds if $l(k) \in \operatorname{dom} f$, then $l[f](k) = f(l(k))$ and if $l(k) \notin \operatorname{dom} f$, then $l[f](k) = l(k)$.

Let us consider k, 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.

Next we state the proposition

(10)[2] $a \dot{\longmapsto} x$ is a substitution.

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

Next we state the proposition

(11) If $f = a \dot{\longmapsto} x$ and $l_1 = l[f]$ and $1 \leq k$ and $k \leq \operatorname{len} l$, then if $l(k) = a$, then $l_1(k) = x$ and if $l(k) \neq a$, then $l_1(k) = l(k)$.

The subset CQC-WFF of WFF is defined as follows:

(Def. 4) $\text{CQC-WFF} = \{s; s \text{ ranges over formulae: } \operatorname{Fixed} s = \emptyset \wedge \operatorname{Free} s = \emptyset\}$.

Let us note that CQC-WFF is non empty.

The following proposition is true

(13)[3] p is an element of CQC-WFF iff $\operatorname{Fixed} p = \emptyset$ and $\operatorname{Free} p = \emptyset$.

Let us consider k and let I_1 be a list of variables of the length k. We say that I_1 is variables list-like if and only if:

(Def. 5) $\operatorname{rng} I_1 \subseteq \operatorname{BoundVar}$.

Let us consider k. One can verify that there exists a list of variables of the length k which is variables list-like.

Let us consider k. A variables list of k is a variables list-like list of variables of the length k.

We now state the proposition

(15)[4] Let l be a list of variables of the length k. Then l is a variables list of k if and only if $\{l(i) : 1 \leq i \wedge i \leq \operatorname{len} l \wedge l(i) \in \operatorname{FreeVar}\} = \emptyset$ and $\{l(j) : 1 \leq j \wedge j \leq \operatorname{len} l \wedge l(j) \in \operatorname{FixedVar}\} = \emptyset$.

In the sequel r, s denote elements of CQC-WFF.

One can prove the following propositions:

(16) VERUM is an element of CQC-WFF.

(17) Let P be a k-ary predicate symbol and l be a list of variables of the length k. Then $P[l]$ is an element of CQC-WFF if and only if the following conditions are satisfied:

(i) $\{l(i) : 1 \leq i \wedge i \leq \operatorname{len} l \wedge l(i) \in \operatorname{FreeVar}\} = \emptyset$, and

(ii) $\{l(j) : 1 \leq j \wedge j \leq \operatorname{len} l \wedge l(j) \in \operatorname{FixedVar}\} = \emptyset$.

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

One can prove the following two propositions:

(18) $\neg p$ is an element of CQC-WFF iff p is an element of CQC-WFF.

(19) $p \wedge q$ is an element of CQC-WFF iff p is an element of CQC-WFF and q is an element of CQC-WFF.

[2] The propositions (8) and (9) have been removed.
[3] The proposition (12) has been removed.
[4] The proposition (14) has been removed.

VERUM is an element of CQC-WFF. Let us consider r. Then $\neg r$ is an element of CQC-WFF. Let us consider s. Then $r \wedge s$ is an element of CQC-WFF.

We now state three propositions:

(20) $r \Rightarrow s$ is an element of CQC-WFF.

(21) $r \vee s$ is an element of CQC-WFF.

(22) $r \Leftrightarrow s$ is an element of CQC-WFF.

Let us consider r, s. Then $r \Rightarrow s$ is an element of CQC-WFF. Then $r \vee s$ is an element of CQC-WFF. Then $r \Leftrightarrow s$ is an element of CQC-WFF.

Next we state the proposition

(23) $\forall_x p$ is an element of CQC-WFF iff p is an element of CQC-WFF.

Let us consider x, r. Then $\forall_x r$ is an element of CQC-WFF.

The following proposition is true

(24) $\exists_x r$ is an element of CQC-WFF.

Let us consider x, r. Then $\exists_x r$ is an element of CQC-WFF.

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 following condition is satisfied:

- Let given r, s, x, k, l be a variables list of k, and P be a k-ary predicate symbol. Then $\mathcal{P}[\text{VERUM}]$ and $\mathcal{P}[P[l]]$ and if $\mathcal{P}[r]$, then $\mathcal{P}[\neg r]$ and if $\mathcal{P}[r]$ and $\mathcal{P}[s]$, then $\mathcal{P}[r \wedge s]$ and if $\mathcal{P}[r]$, then $\mathcal{P}[\forall_x r]$.

The scheme *CQC Func Ex* deals with 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 CQC-WFF into $\mathcal{A}$ such that

(i) $F(\text{VERUM}) = \mathcal{B}$, and

(ii) for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P holds $F(P[l]) = \mathcal{F}(k,P,l)$ and $F(\neg r) = \mathcal{G}(F(r))$ and $F(r \wedge s) = \mathcal{H}(F(r),F(s))$ and $F(\forall_x r) = \mathcal{I}(x,F(r))$

for all values of the parameters.

The scheme *CQC Func Uniq* deals with a non empty set $\mathcal{A}$, a function $\mathcal{B}$ from CQC-WFF into $\mathcal{A}$, a function $\mathcal{C}$ from CQC-WFF 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:

- (i) $\mathcal{B}(\text{VERUM}) = \mathcal{D}$, and
  (ii) for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P holds $\mathcal{B}(P[l]) = \mathcal{F}(k,P,l)$ and $\mathcal{B}(\neg r) = \mathcal{G}(\mathcal{B}(r))$ and $\mathcal{B}(r \wedge s) = \mathcal{H}(\mathcal{B}(r),\mathcal{B}(s))$ and $\mathcal{B}(\forall_x r) = \mathcal{I}(x,\mathcal{B}(r))$,
  and
- (i) $\mathcal{C}(\text{VERUM}) = \mathcal{D}$, and
  (ii) for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P holds $\mathcal{C}(P[l]) = \mathcal{F}(k,P,l)$ and $\mathcal{C}(\neg r) = \mathcal{G}(\mathcal{C}(r))$ and $\mathcal{C}(r \wedge s) = \mathcal{H}(\mathcal{C}(r),\mathcal{C}(s))$ and $\mathcal{C}(\forall_x r) = \mathcal{I}(x,\mathcal{C}(r))$.

The scheme *CQC Def correctness* deals with a non empty set $\mathcal{A}$, an element $\mathcal{B}$ of CQC-WFF, 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 CQC-WFF into $\mathcal{A}$ such that $d = F(\mathcal{B})$ and $F(\mathrm{VERUM}) = \mathcal{C}$ and for all r, s, x, k and for every variables list l of k and for every k-ary predicate symbol P holds $F(P[l]) = \mathcal{F}(k,P,l)$ and $F(\neg r) = \mathcal{G}(F(r))$ and $F(r \wedge s) = \mathcal{H}(F(r),F(s))$ and $F(\forall_x r) = \mathcal{I}(x,F(r))$, and

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

for all values of the parameters.

The scheme *CQC Def VERUM* 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 $\mathcal{A}$, 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 meet the following requirement:

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

The scheme *CQC Def atomic* deals with 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 parameters satisfy the following condition:

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

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 CQC-WFF, 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 following condition is satisfied:

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

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

The scheme *QC Def universal* 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 $\mathcal{A}$, 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 CQC-WFF, and states that:

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

provided the following condition is met:

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

Let us consider p, x. The functor $p(x)$ yields an element of WFF and is defined by the condition (Def. 6).

(Def. 6) There exists a function F from WFF into WFF such that

(i) $p(x) = F(p)$, and

(ii) for every q holds $F(\text{VERUM}) = \text{VERUM}$ and if q is atomic, then $F(q) = \text{PredSym}(q)[\text{Args}(q)[\mathbf{a}_0 \dot{\longmapsto} x]]$ and if q is negative, then $F(q) = \neg F(\text{Arg}(q))$ and if q is conjunctive, then $F(q) = F(\text{LeftArg}(q)) \wedge F(\text{RightArg}(q))$ and if q is universal, then $F(q) = (\text{Bound}(q) = x \rightarrow q, \forall_{\text{Bound}(q)} F(\text{Scope}(q)))$.

Next we state a number of propositions:

(28)[5] $\text{VERUM}(x) = \text{VERUM}$.

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

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

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

[5] The propositions (25)–(27) have been removed.

REFERENCES

[1] Grzegorz Bancerek. Connectives and subformulae of the first order language. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/qc_lang2.html.

[2] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/finseq_1.html.

[3] Czesław Byliński. Functions and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/funct_1.html.

[4] Czesław Byliński. Functions from a set to a set. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/funct_2.html.

[5] Czesław Byliński and Grzegorz Bancerek. Variables in formulae of the first order language. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/qc_lang3.html.

[6] Witold A. Pogorzelski and Tadeusz Prucnal. The substitution rule for predicate letters in the first-order predicate calculus. *Reports on Mathematical Logic*, (5):77–90, 1975.

[7] Piotr Rudnicki and Andrzej Trybulec. A first order language. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/qc_lang1.html.

[8] Andrzej Trybulec. Binary operations applied to functions. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/funcop_1.html.

[9] Andrzej Trybulec. Tarski Grothendieck set theory. *Journal of Formalized Mathematics*, Axiomatics, 1989. http://mizar.org/JFM/Axiomatics/tarski.html.

[10] Andrzej Trybulec. Subsets of real numbers. *Journal of Formalized Mathematics*, Addenda, 2003. http://mizar.org/JFM/Addenda/numbers.html.

[11] Zinaida Trybulec. Properties of subsets. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/subset_1.html.

[12] Edmund Woronowicz. Relations and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/relat_1.html.

Received May 11, 1990

Published January 2, 2004