FORMALIZED MATHEMATICS
Vol. 20, No. 2, Pages 147–155, 2012
DOI: 10.2478/v10037-012-0017-x

VERSITA
versita.com/fm/

Semantics of MML Query[1]

Grzegorz Bancerek
Białystok Technical University
Poland

Summary. In the paper the semantics of MML Query queries is given. The formalization is done according to [4].

MML identifier: `MMLQUERY`, version: `7.12.02 4.181.1147`

The notation and terminology used here have been introduced in the following papers: [1], [5], [11], [8], [10], [6], [2], [3], [15], [13], [14], [9], [12], and [7].

1. Elementary Queries

Let X be a set. A list of X is a subset of X. An operation of X is a binary relation on X.

Let x, y, R be sets. The predicate $x, y \in R$ is defined by:

(Def. 1) $\langle x, y \rangle \in R$.

Let x, y, R be sets. We introduce $x, y \notin R$ as an antonym of $x, y \in R$.

For simplicity, we use the following convention: X, Y, z, s denote sets, L, L_1, L_2, A denote lists of X, x denotes an element of X, O, O_2, O_3 denote operations of X, and m denotes a natural number.

The following proposition is true

(1) For all binary relations R_1, R_2 holds $R_1 \subseteq R_2$ iff for every z holds $R_1{}^\circ z \subseteq R_2{}^\circ z$.

Let us consider X, O, x. We introduce $x\ O$ as a synonym of $O^\circ x$.

Let us consider X, O, x. Then $x\ O$ is a list of X.

One can prove the following proposition

[1]This work has been supported by the Polish Ministry of Science and Higher Education project "Managing a Large Repository of Computer-verified Mathematical Knowledge" (N N519 385136).

ISSN 1426–2630(p), 1898-9934(e)

(2) $x, y \in O$ iff $y \in x\ O$.

Let us consider X, O, L. We introduce $L|O$ as a synonym of $O^\circ L$.

Let us consider X, O, L. Then $L|O$ is a list of X and it can be characterized by the condition:

(Def. 2) $L|O = \bigcup\{x\ O : x \in L\}$.

The functor $L\&O$ yielding a list of X is defined as follows:

(Def. 3) $L\&O = \bigcap\{x\ O : x \in L\}$.

The functor $L\,\texttt{where}\,O$ yielding a list of X is defined as follows:

(Def. 4) $L\,\texttt{where}\,O = \{x : \bigvee_y\ (x, y \in O\ \wedge\ x \in L)\}$.

Let O_2 be an operation of X. The functor $L\,\texttt{where}\,O = O_2$ yielding a list of X is defined as follows:

(Def. 5) $L\,\texttt{where}\,O = O_2 = \{x : \overline{\overline{x\ O}} = \overline{\overline{x\ O_2}}\ \wedge\ x \in L\}$.

The functor $L\,\texttt{where}\,O \leq O_2$ yielding a list of X is defined by:

(Def. 6) $L\,\texttt{where}\,O \leq O_2 = \{x : \overline{\overline{x\ O}} \subseteq \overline{\overline{x\ O_2}}\ \wedge\ x \in L\}$.

The functor $L\,\texttt{where}\,O \geq O_2$ yields a list of X and is defined by:

(Def. 7) $L\,\texttt{where}\,O \geq O_2 = \{x : \overline{\overline{x\ O_2}} \subseteq \overline{\overline{x\ O}}\ \wedge\ x \in L\}$.

The functor $L\,\texttt{where}\,O < O_2$ yielding a list of X is defined as follows:

(Def. 8) $L\,\texttt{where}\,O < O_2 = \{x : \overline{\overline{x\ O}} \in \overline{\overline{x\ O_2}}\ \wedge\ x \in L\}$.

The functor $L\,\texttt{where}\,O > O_2$ yields a list of X and is defined by:

(Def. 9) $L\,\texttt{where}\,O > O_2 = \{x : \overline{\overline{x\ O_2}} \in \overline{\overline{x\ O}}\ \wedge\ x \in L\}$.

Let us consider X, L, O, n. The functor $L\,\texttt{where}\,O = n$ yielding a list of X is defined as follows:

(Def. 10) $L\,\texttt{where}\,O = n = \{x : \overline{\overline{x\ O}} = n\ \wedge\ x \in L\}$.

The functor $L\,\texttt{where}\,O \leq n$ yielding a list of X is defined by:

(Def. 11) $L\,\texttt{where}\,O \leq n = \{x : \overline{\overline{x\ O}} \subseteq n\ \wedge\ x \in L\}$.

The functor $L\,\texttt{where}\,O \geq n$ yielding a list of X is defined as follows:

(Def. 12) $L\,\texttt{where}\,O \geq n = \{x : n \subseteq \overline{\overline{x\ O}}\ \wedge\ x \in L\}$.

The functor $L\,\texttt{where}\,O < n$ yields a list of X and is defined as follows:

(Def. 13) $L\,\texttt{where}\,O < n = \{x : \overline{\overline{x\ O}} \in n\ \wedge\ x \in L\}$.

The functor $L\,\texttt{where}\,O > n$ yields a list of X and is defined by:

(Def. 14) $L\,\texttt{where}\,O > n = \{x : n \in \overline{\overline{x\ O}}\ \wedge\ x \in L\}$.

One can prove the following propositions:

(3) $x \in L\,\texttt{where}\,O$ iff $x \in L$ and $x\ O \neq \emptyset$.

(4) $L\,\texttt{where}\,O \subseteq L$.

(5) If $L \subseteq \operatorname{dom} O$, then $L\,\texttt{where}\,O = L$.

(6) If $n \neq 0$ and $L_1 \subseteq L_2$, then $L_1\,\texttt{where}\,O \geq n \subseteq L_2\,\texttt{where}\,O$.

(7) $L\,\texttt{where}\,O \geq 1 = L\,\texttt{where}\,O$.

(8) If $L_1 \subseteq L_2$, then $L_1\,\mathtt{where}\,O > n \subseteq L_2\,\mathtt{where}\,O$.

(9) $L\,\mathtt{where}\,O > 0 = L\,\mathtt{where}\,O$.

(10) If $n \neq 0$ and $L_1 \subseteq L_2$, then $L_1\,\mathtt{where}\,O = n \subseteq L_2\,\mathtt{where}\,O$.

(11) $L\,\mathtt{where}\,O \geq n+1 = L\,\mathtt{where}\,O > n$.

(12) $L\,\mathtt{where}\,O \leq n = L\,\mathtt{where}\,O < n+1$.

(13) If $n \leq m$ and $L_1 \subseteq L_2$ and $O_1 \subseteq O_2$, then $L_1\,\mathtt{where}\,O_1 \geq m \subseteq L_2\,\mathtt{where}\,O_2 \geq n$.

(14) If $n \leq m$ and $L_1 \subseteq L_2$ and $O_1 \subseteq O_2$, then $L_1\,\mathtt{where}\,O_1 > m \subseteq L_2\,\mathtt{where}\,O_2 > n$.

(15) If $n \leq m$ and $L_1 \subseteq L_2$ and $O_1 \subseteq O_2$, then $L_1\,\mathtt{where}\,O_2 \leq n \subseteq L_2\,\mathtt{where}\,O_1 \leq m$.

(16) If $n \leq m$ and $L_1 \subseteq L_2$ and $O_1 \subseteq O_2$, then $L_1\,\mathtt{where}\,O_2 < n \subseteq L_2\,\mathtt{where}\,O_1 < m$.

(17) If $O_1 \subseteq O_2$ and $L_1 \subseteq L_2$ and $O \subseteq O_3$, then $L_1\,\mathtt{where}\,O \geq O_2 \subseteq L_2\,\mathtt{where}\,O_3 \geq O_1$.

(18) If $O_1 \subseteq O_2$ and $L_1 \subseteq L_2$ and $O \subseteq O_3$, then $L_1\,\mathtt{where}\,O > O_2 \subseteq L_2\,\mathtt{where}\,O_3 > O_1$.

(19) If $O_1 \subseteq O_2$ and $L_1 \subseteq L_2$ and $O \subseteq O_3$, then $L_1\,\mathtt{where}\,O_3 \leq O_1 \subseteq L_2\,\mathtt{where}\,O \leq O_2$.

(20) If $O_1 \subseteq O_2$ and $L_1 \subseteq L_2$ and $O \subseteq O_3$, then $L_1\,\mathtt{where}\,O_3 < O_1 \subseteq L_2\,\mathtt{where}\,O < O_2$.

(21) $L\,\mathtt{where}\,O > O_1 \subseteq L\,\mathtt{where}\,O$.

(22) If $O_1 \subseteq O_2$ and $L_1 \subseteq L_2$, then $L_1\,\mathtt{where}\,O_1 \subseteq L_2\,\mathtt{where}\,O_2$.

(23) $a \in L|O$ iff there exists b such that $a \in b\ O$ and $b \in L$.

Let us consider X, A, B. We introduce $A\,\mathtt{and}\,B$ as a synonym of $A \cap B$. We introduce $A\,\mathtt{or}\,B$ as a synonym of $A \cup B$. We introduce $A\,\mathtt{butnot}\,B$ as a synonym of $A \setminus B$.

Let us consider X, A, B. Then $A\,\mathtt{and}\,B$ is a list of X. Then $A\,\mathtt{or}\,B$ is a list of X. Then $A\,\mathtt{butnot}\,B$ is a list of X.

We now state several propositions:

(24) If $L_1 \neq \emptyset$ and $L_2 \neq \emptyset$, then $(L_1\,\mathtt{or}\,L_2)\&O = (L_1\&O)\,\mathtt{and}(L_2\&O)$.

(25) If $L_1 \subseteq L_2$ and $O_1 \subseteq O_2$, then $L_1|O_1 \subseteq L_2|O_2$.

(26) If $O_1 \subseteq O_2$, then $L\&O_1 \subseteq L\&O_2$.

(27) $L\&(O_1\,\mathtt{and}\,O_2) = (L\&O_1)\,\mathtt{and}(L\&O_2)$.

(28) If $L_1 \neq \emptyset$ and $L_1 \subseteq L_2$, then $L_2\&O \subseteq L_1\&O$.

2. Operations

One can prove the following two propositions:

(29) For all operations O_1, O_2 of X such that for every x holds $x\ O_1 = x\ O_2$ holds $O_1 = O_2$.

(30) For all operations O_1, O_2 of X such that for every L holds $L|O_1 = L|O_2$ holds $O_1 = O_2$.

The functor $\mathtt{not}\,O$ yielding an operation of X is defined as follows:

(Def. 15) For every L holds $L|\,\mathtt{not}\,O = \bigcup\{(x\ O = \emptyset \to \{x\}, \emptyset) : x \in L\}$.

Let us consider X and let O_1, O_2 be operations of X. We introduce $O_1\,\mathtt{and}\,O_2$ as a synonym of $O_1 \cap O_2$. We introduce $O_1\,\mathtt{or}\,O_2$ as a synonym of $O_1 \cup O_2$. We introduce $O_1\,\mathtt{butnot}\,O_2$ as a synonym of $O_1 \setminus O_2$. We introduce $O_1|O_2$ as a synonym of $O_1 \cdot O_2$.

Let us consider X and let O_1, O_2 be operations of X. Then $O_1\,\mathtt{and}\,O_2$ is an operation of X and it can be characterized by the condition:

(Def. 16) For every L holds $L|(O_1\,\mathtt{and}\,O_2) = \bigcup\{(x\ O_1)\,\mathtt{and}(x\ O_2) : x \in L\}$.

Then $O_1\,\mathtt{or}\,O_2$ is an operation of X and it can be characterized by the condition:

(Def. 17) For every L holds $L|(O_1\,\mathtt{or}\,O_2) = \bigcup\{(x\ O_1)\,\mathtt{or}(x\ O_2) : x \in L\}$.

Then $O_1\,\mathtt{butnot}\,O_2$ is an operation of X and it can be characterized by the condition:

(Def. 18) For every L holds $L|(O_1\,\mathtt{butnot}\,O_2) = \bigcup\{(x\ O_1)\,\mathtt{butnot}(x\ O_2) : x \in L\}$.

Then $O_1|O_2$ is an operation of X and it can be characterized by the condition:

(Def. 19) For every L holds $L|(O_1|O_2) = L|O_1|O_2$.

The functor $O_1\&O_2$ yielding an operation of X is defined as follows:

(Def. 20) For every L holds $L|(O_1\&O_2) = \bigcup\{(x\ O_1)\&O_2 : x \in L\}$.

We now state a number of propositions:

(31) $x\ (O_1\,\mathtt{and}\,O_2) = (x\ O_1)\,\mathtt{and}(x\ O_2)$.

(32) $x\ (O_1\,\mathtt{or}\,O_2) = (x\ O_1)\,\mathtt{or}(x\ O_2)$.

(33) $x\ (O_1\,\mathtt{butnot}\,O_2) = (x\ O_1)\,\mathtt{butnot}(x\ O_2)$.

(34) $x\ (O_1|O_2) = (x\ O_1)|O_2$.

(35) $x\ (O_1\&O_2) = (x\ O_1)\&O_2$.

(36) $z, s \in \mathtt{not}\,O$ iff $z = s$ and $z \in X$ and $z \notin \mathrm{dom}\,O$.

(37) $\mathtt{not}\,O = \mathrm{id}_{X \setminus \mathrm{dom}\,O}$.

(38) $\mathrm{dom}\,\mathtt{not}\,\mathtt{not}\,O = \mathrm{dom}\,O$.

(39) $L\,\mathtt{where}\,\mathtt{not}\,\mathtt{not}\,O = L\,\mathtt{where}\,O$.

(40) $L\,\mathtt{where}\,O = 0 = L\,\mathtt{where}\,\mathtt{not}\,O$.

(41) $\mathtt{not}\,\mathtt{not}\,\mathtt{not}\,O = \mathtt{not}\,O$.

(42) $\mathtt{not}\,O_1\,\mathtt{or}\,\mathtt{not}\,O_2 \subseteq \mathtt{not}(O_1\,\mathtt{and}\,O_2)$.

(43) $\mathtt{not}(O_1 \mathbin{\mathtt{or}} O_2) = \mathtt{not}\, O_1 \mathbin{\mathtt{and}} \mathtt{not}\, O_2$.

(44) If $\mathrm{dom}\, O_1 = X$ and $\mathrm{dom}\, O_2 = X$, then $(O_1 \mathbin{\mathtt{or}} O_2) \& O = (O_1 \& O) \mathbin{\mathtt{and}} (O_2 \& O)$.

Let us consider X, O. We say that O is filtering if and only if:

(Def. 21) $O \subseteq \mathrm{id}_X$.

Next we state the proposition

(45) O is filtering iff $O = \mathrm{id}_{\mathrm{dom}\, O}$.

Let us consider X, O. Note that $\mathtt{not}\, O$ is filtering.

Let us consider X. Note that there exists an operation of X which is filtering.

In the sequel F_1, F_2 denote filtering operations of X.

Let us consider X, F, O. One can check the following observations:

* $F \mathbin{\mathtt{and}} O$ is filtering,
* $O \mathbin{\mathtt{and}} F$ is filtering, and
* $F \mathbin{\mathtt{butnot}} O$ is filtering.

Let us consider X, F_1, F_2. One can verify that $F_1 \mathbin{\mathtt{or}} F_2$ is filtering.

(46) If $z \in x\ F$, then $z = x$.

(47) $L|F = L \mathbin{\mathtt{where}} F$.

(48) $\mathtt{not}\,\mathtt{not}\, F = F$.

(49) $\mathtt{not}(F_1 \mathbin{\mathtt{and}} F_2) = \mathtt{not}\, F_1 \mathbin{\mathtt{or}} \mathtt{not}\, F_2$.

(50) $\mathrm{dom}(O \mathbin{\mathtt{or}} \mathtt{not}\, O) = X$.

(51) $F \mathbin{\mathtt{or}} \mathtt{not}\, F = \mathrm{id}_X$.

(52) $O \mathbin{\mathtt{and}} \mathtt{not}\, O = \emptyset$.

(53) $(O_1 \mathbin{\mathtt{or}} O_2) \mathbin{\mathtt{and}} \mathtt{not}\, O_1 \subseteq O_2$.

3. Rough Queries

Let A be a finite sequence and let a be a set. The functor $\#\mathrm{occurrences}(a, A)$ yielding a natural number is defined as follows:

(Def. 22) $\#\mathrm{occurrences}(a, A) = \overline{\overline{\{i : i \in \mathrm{dom}\, A \ \wedge\ a \in A(i)\}}}$.

We now state two propositions:

(54) For every finite sequence A and for every set a holds $\#\mathrm{occurrences}(a, A) \leq \mathrm{len}\, A$.

(55) For every finite sequence A and for every set a holds $A \neq \emptyset$ and $\#\mathrm{occurrences}(a, A) = \mathrm{len}\, A$ iff $a \in \bigcap \mathrm{rng}\, A$.

The functor $\max\#\, A$ yielding a natural number is defined as follows:

(Def. 23) For every set a holds $\#\mathrm{occurrences}(a, A) \leq \max\#\, A$ and for every n such that for every set a holds $\#\mathrm{occurrences}(a, A) \leq n$ holds $\max\#\, A \leq n$.

(56) For every finite sequence A holds $\max\# A \leq \operatorname{len} A$.

(57) For every finite sequence A and for every set a such that $\#\text{occurrences}(a, A) = \operatorname{len} A$ holds $\max\# A = \operatorname{len} A$.

Let us consider X, let A be a finite sequence of elements of 2^X, and let n be a natural number. The functor $\texttt{rough}\, n(A)$ yields a list of X and is defined as follows:

(Def. 24) $\texttt{rough}\, n(A) = \{x : n \leq \#\text{occurrences}(x, A)\}$ if $X \neq \emptyset$.

Let m be a natural number. The functor $\texttt{rough}\, n\text{-}m(A)$ yields a list of X and is defined by:

(Def. 25) $\texttt{rough}\, n\text{-}m(A) = \{x : n \leq \#\text{occurrences}(x, A) \ \wedge\ \#\text{occurrences}(x, A) \leq m\}$ if $X \neq \emptyset$.

Let us consider X and let A be a finite sequence of elements of 2^X. The functor $\texttt{rough}(A)$ yielding a list of X is defined by:

(Def. 26) $\texttt{rough}(A) = \texttt{rough}\, \max\# A(A)$.

Next we state several propositions:

(58) For every finite sequence A of elements of 2^X holds $\texttt{rough}\, n\text{-}\operatorname{len} A(A) = \texttt{rough}\, n(A)$.

(59) For every finite sequence A of elements of 2^X such that $n \leq m$ holds $\texttt{rough}\, m(A) \subseteq \texttt{rough}\, n(A)$.

(60) Let A be a finite sequence of elements of 2^X and n_1, n_2, m_1, m_2 be natural numbers. If $n_1 \leq m_1$ and $n_2 \leq m_2$, then $\texttt{rough}\, m_1\text{-}n_2(A) \subseteq \texttt{rough}\, n_1\text{-}m_2(A)$.

(61) For every finite sequence A of elements of 2^X holds $\texttt{rough}\, n\text{-}m(A) \subseteq \texttt{rough}\, n(A)$.

(62) For every finite sequence A of elements of 2^X such that $A \neq \emptyset$ holds $\texttt{rough}\, \operatorname{len} A(A) = \bigcap \operatorname{rng} A$.

(63) For every finite sequence A of elements of 2^X holds $\texttt{rough}\, 1(A) = \bigcup A$.

(64) For all lists L_1, L_2 of X holds $\texttt{rough}\, 2(\langle L_1, L_2 \rangle) = L_1\ \texttt{and}\ L_2$.

(65) For all lists L_1, L_2 of X holds $\texttt{rough}\, 1(\langle L_1, L_2 \rangle) = L_1\ \texttt{or}\ L_2$.

4. Constructor Database

We introduce constructor databases which are extensions of 1-sorted structures and are systems

⟨ a carrier, constructors, a ref-operation ⟩,

where the carrier is a set, the constructors constitute a list of the carrier, and the ref-operation is a relation between the carrier and the constructors.

Let X be a 1-sorted structure. A list of X is a list of the carrier of X. An operation of X is an operation of the carrier of X.

Let us consider X, let S be a subset of X, and let R be a relation between X and S. The functor ${}^{@}R$ yields a binary relation on X and is defined by:

(Def. 27) ${}^{@}R = R$.

Let X be a constructor database and let a be an element of X. The functor $a\,\mathtt{ref}$ yielding a list of X is defined as follows:

(Def. 28) $a\,\mathtt{ref} = a$ ${}^{@}$the ref-operation of X.

The functor $a\,\mathtt{occur}$ yields a list of X and is defined as follows:

(Def. 29) $a\,\mathtt{occur} = a$ (${}^{@}$the ref-operation of X)$^{\smile}$.

The following proposition is true

(66) For every constructor database X and for all elements x, y of X holds $x \in y\,\mathtt{ref}$ iff $y \in x\,\mathtt{occur}$.

Let X be a constructor database. We say that X is ref-finite if and only if:

(Def. 30) For every element x of X holds $x\,\mathtt{ref}$ is finite.

One can verify that every constructor database which is finite is also ref-finite.

Let us note that there exists a constructor database which is finite and non empty.

Let X be a ref-finite constructor database and let x be an element of X. Observe that $x\,\mathtt{ref}$ is finite.

Let X be a constructor database and let A be a finite sequence of elements of the constructors of X. The functor $\mathtt{atleast}(A)$ yielding a list of X is defined by:

(Def. 31) $\mathtt{atleast}(A) = \{x \in X\colon \operatorname{rng} A \subseteq x\,\mathtt{ref}\}$ if the carrier of $X \neq \emptyset$.

The functor $\mathtt{atmost}(A)$ yielding a list of X is defined as follows:

(Def. 32) $\mathtt{atmost}(A) = \{x \in X\colon x\,\mathtt{ref} \subseteq \operatorname{rng} A\}$ if the carrier of $X \neq \emptyset$.

The functor $\mathtt{exactly}(A)$ yields a list of X and is defined by:

(Def. 33) $\mathtt{exactly}(A) = \{x \in X\colon x\,\mathtt{ref} = \operatorname{rng} A\}$ if the carrier of $X \neq \emptyset$.

Let n be a natural number. The functor $\mathtt{atleast\ minus}\,n(A)$ yields a list of X and is defined by:

(Def. 34) $\mathtt{atleast\ minus}\,n(A) = \{x \in X\colon \overline{\overline{\operatorname{rng} A \setminus x\,\mathtt{ref}}} \leq n\}$ if the carrier of $X \neq \emptyset$.

Let X be a ref-finite constructor database, let A be a finite sequence of elements of the constructors of X, and let n be a natural number. The functor $\mathtt{atmost\ plus}\,n(A)$ yields a list of X and is defined by:

(Def. 35) $\mathtt{atmost\ plus}\,n(A) = \{x \in X\colon \overline{\overline{x\,\mathtt{ref} \setminus \operatorname{rng} A}} \leq n\}$ if the carrier of $X \neq \emptyset$.

Let m be a natural number. The functor $\mathtt{exactly\ plus}\,n\,\mathtt{minus}\,m(A)$ yielding a list of X is defined by:

(Def. 36) $\underline{\texttt{exactly plus}\, n\, \texttt{minus}\, m(A)} = \{x \in X\colon \overline{\overline{x\,\texttt{ref} \setminus \operatorname{rng} A}} \leq n \wedge \overline{\overline{\operatorname{rng} A \setminus x\,\texttt{ref}}} \leq m\}$ if the carrier of $X \neq \emptyset$.

In the sequel X denotes a constructor database, x denotes an element of X, B denotes a finite sequence of elements of the constructors of Y, and y denotes an element of Y.

The following propositions are true:

(67) $\texttt{atleast minus}\, 0(A) = \texttt{atleast}(A)$.

(68) $\texttt{atmost plus}\, 0(B) = \texttt{atmost}(B)$.

(69) $\texttt{exactly plus}\, 0\, \texttt{minus}\, 0(B) = \texttt{exactly}(B)$.

(70) If $n \leq m$, then $\texttt{atleast minus}\, n(A) \subseteq \texttt{atleast minus}\, m(A)$.

(71) If $n \leq m$, then $\texttt{atmost plus}\, n(B) \subseteq \texttt{atmost plus}\, m(B)$.

(72) For all natural numbers n_1, n_2, m_1, m_2 such that $n_1 \leq m_1$ and $n_2 \leq m_2$ holds $\texttt{exactly plus}\, n_1\, \texttt{minus}\, n_2(B) \subseteq \texttt{exactly plus}\, m_1\, \texttt{minus}\, m_2(B)$.

(73) $\texttt{atleast}(A) \subseteq \texttt{atleast minus}\, n(A)$.

(74) $\texttt{atmost}(B) \subseteq \texttt{atmost plus}\, n(B)$.

(75) $\texttt{exactly}(B) \subseteq \texttt{exactly plus}\, n\, \texttt{minus}\, m(B)$.

(76) $\texttt{exactly}(A) = \texttt{atleast}(A)\, \texttt{and}\, \texttt{atmost}(A)$.

(77) $\texttt{exactly plus}\, n\, \texttt{minus}\, m(B) = \texttt{atleast minus}\, m(B)\, \texttt{and}\, \texttt{atmost plus}\, n(B)$.

(78) If $A \neq \emptyset$, then $\texttt{atleast}(A) = \bigcap\{x\, \texttt{occur} : x \in \operatorname{rng} A\}$.

(79) For all elements c_1, c_2 of X such that $A = \langle c_1, c_2 \rangle$ holds $\texttt{atleast}(A) = c_1\, \texttt{occur}\, \texttt{and}\, c_2\, \texttt{occur}$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.
[2] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[3] Grzegorz Bancerek. The ordinal numbers. *Formalized Mathematics*, 1(**1**):91–96, 1990.
[4] Grzegorz Bancerek. Information retrieval and rendering with MML query. *LNCS*, 4108:266–279, 2006.
[5] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[6] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[7] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[8] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[9] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[10] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[11] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[12] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[13] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[14] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

[15] Bo Zhang, Hiroshi Yamazaki, and Yatsuka Nakamura. Set sequences and monotone class. *Formalized Mathematics*, 13(**4**):435–441, 2005.

Received December 18, 2011