FORMALIZED MATHEMATICS
Volume 3, Number 2, 1992
Université Catholique de Louvain

On a Mathematical Model of Programs

Yatsuka Nakamura
Shinshu University
Nagano

Andrzej Trybulec
Warsaw University
Białystok

Summary. We continue the work on mathematical modeling of hardware and software started in [17]. The main objective of this paper is the definition of a program. We start with the concept of partial product, i.e. the set of all partial functions f from I to $\bigcup_{i\in I} A_i$, fulfilling the condition $f.i \in A_i$ for $i \in dom f$. The computation and the result of a computation are defined in usual way. A finite partial state is called autonomic if the result of a computation starting with it does not depend on the remaining memory and an AMI is called programmable if it has a non empty autonomic partial finite state. We prove the consistency of the following set of properties of an AMI: data-oriented, halting, steady-programmed, realistic and programmable. For this purpose we define a trivial AMI. It has only the instruction counter and one instruction location. The only instruction of it is the halt instruction. A preprogram is a finite partial state that halts. We conclude with the definition of a program of a partial function F mapping the set of the finite partial states into itself. It is a finite partial state s such that for every finite partial state $s' \in dom F$ the result of any computation starting with $s+s'$ includes $F.s'$.

MML Identifier: AMI_2.

The papers [24], [22], [28], [6], [7], [23], [14], [1], [19], [26], [25], [10], [3], [5], [15], [29], [21], [2], [20], [8], [18], [4], [9], [12], [13], [27], [11], [16], and [17] provide the notation and terminology for this paper.

1. Preliminaries

For simplicity we follow the rules: A, B, C will denote sets, f, g, h will denote functions, x, y, z will be arbitrary, and i, j, k will denote natural numbers. The scheme *UniqSet* concerns a set $\mathcal{A}$, a set $\mathcal{B}$, and a unary predicate $\mathcal{P}$, and states that:

$\mathcal{A} = \mathcal{B}$

ISSN 0777–4028

provided the following requirements are met:

- for every x holds $x \in \mathcal{A}$ if and only if $\mathcal{P}[x]$,
- for every x holds $x \in \mathcal{B}$ if and only if $\mathcal{P}[x]$.

The following propositions are true:

(1) A misses $B \setminus C$ if and only if B misses $A \setminus C$.

(2) For every function f holds $\pi_1(\operatorname{dom} f \times \operatorname{rng} f) \circ f = \operatorname{dom} f$.

(3) If $f \approx g$ and $\langle x, y \rangle \in f$ and $\langle x, z \rangle \in g$, then $y = z$.

(4) If for every x such that $x \in A$ holds x is a function and for all functions f, g such that $f \in A$ and $g \in A$ holds $f \approx g$, then $\bigcup A$ is a function.

(5) If $\operatorname{dom} f \subseteq A \cup B$, then $f \restriction A +\cdot f \restriction B = f$.

(6) $\operatorname{dom} f \subseteq \operatorname{dom}(f +\cdot g)$ and $\operatorname{dom} g \subseteq \operatorname{dom}(f +\cdot g)$.

(7) For arbitrary x_1, x_2, y_1, y_2 holds $[x_1 \longmapsto y_1, x_2 \longmapsto y_2] = (x_1 \dot{\longmapsto} y_1) +\cdot (x_2 \dot{\longmapsto} y_2)$.

(8) For all x, y holds $x \dot{\longmapsto} y = \{\langle x, y \rangle\}$.

(9) For arbitrary a, b, c holds $[a \longmapsto b, a \longmapsto c] = a \dot{\longmapsto} c$.

(10) For every function f holds $\operatorname{dom} f$ is finite if and only if f is finite.

(11) If $x \in \prod f$, then x is a function.

2. Partial products

Let f be a function. The functor $\prod^{\cdot} f$ yields a non-empty set of functions and is defined by:

(Def.1) $x \in \prod^{\cdot} f$ if and only if there exists g such that $x = g$ and $\operatorname{dom} g \subseteq \operatorname{dom} f$ and for every x such that $x \in \operatorname{dom} g$ holds $g(x) \in f(x)$.

Next we state a number of propositions:

(12) $x \in \prod^{\cdot} f$ if and only if there exists g such that $x = g$ and $\operatorname{dom} g \subseteq \operatorname{dom} f$ and for every x such that $x \in \operatorname{dom} g$ holds $g(x) \in f(x)$.

(13) If $\operatorname{dom} g \subseteq \operatorname{dom} f$ and for every x such that $x \in \operatorname{dom} g$ holds $g(x) \in f(x)$, then $g \in \prod^{\cdot} f$.

(14) If $g \in \prod^{\cdot} f$, then $\operatorname{dom} g \subseteq \operatorname{dom} f$ and for every x such that $x \in \operatorname{dom} g$ holds $g(x) \in f(x)$.

(15) $\square \in \prod^{\cdot} f$.

(16) $\prod f \subseteq \prod^{\cdot} f$.

(17) If $x \in \prod^{\cdot} f$, then x is a partial function from $\operatorname{dom} f$ to $\bigcup \operatorname{rng} f$.

(18) If $g \in \prod f$ and $h \in \prod^{\cdot} f$, then $g +\cdot h \in \prod f$.

(19) If $\prod f \neq \emptyset$, then $g \in \prod^{\cdot} f$ if and only if there exists h such that $h \in \prod f$ and $g \leq h$.

(20) $\prod^{\cdot} f \subseteq \operatorname{dom} f \dot{\rightarrow} \bigcup \operatorname{rng} f$.

(21) If $f \subseteq g$, then $\prod^{\cdot} f \subseteq \prod^{\cdot} g$.

(22) $\prod^{\cdot} \square = \{\square\}$.

(23) $A \dot{\to} B = \prod^{\cdot}(A \longmapsto B)$.

(24) For all non-empty sets A, B and for every function f from A into B holds $\prod^{\cdot} f = \prod^{\cdot}(f \upharpoonright \{x : f(x) \neq \emptyset\})$, where x ranges over elements of A.

(25) If $x \in \operatorname{dom} f$ and $y \in f(x)$, then $x \dot{\longmapsto} y \in \prod^{\cdot} f$.

(26) $\prod^{\cdot} f = \{\Box\}$ if and only if for every x such that $x \in \operatorname{dom} f$ holds $f(x) = \emptyset$.

(27) If $A \subseteq \prod^{\cdot} f$ and for all functions h_1, h_2 such that $h_1 \in A$ and $h_2 \in A$ holds $h_1 \approx h_2$, then $\bigcup A \in \prod^{\cdot} f$.

(28) If $g \approx h$ and $g \in \prod^{\cdot} f$ and $h \in \prod^{\cdot} f$, then $g \cup h \in \prod^{\cdot} f$.

(29) If $g \subseteq h$ and $h \in \prod^{\cdot} f$, then $g \in \prod^{\cdot} f$.

(30) If $g \in \prod^{\cdot} f$, then $g \upharpoonright A \in \prod^{\cdot} f$.

(31) If $g \in \prod^{\cdot} f$, then $g \upharpoonright A \in \prod^{\cdot}(f \upharpoonright A)$.

(32) If $h \in \prod^{\cdot}(f +\cdot g)$, then there exist functions f', g' such that $f' \in \prod^{\cdot} f$ and $g' \in \prod^{\cdot} g$ and $h = f' +\cdot g'$.

(33) For all functions f', g' such that $\operatorname{dom} g$ misses $\operatorname{dom} f' \setminus \operatorname{dom} g'$ and $f' \in \prod^{\cdot} f$ and $g' \in \prod^{\cdot} g$ holds $f' +\cdot g' \in \prod^{\cdot}(f +\cdot g)$.

(34) For all functions f', g' such that $\operatorname{dom} f'$ misses $\operatorname{dom} g \setminus \operatorname{dom} g'$ and $f' \in \prod^{\cdot} f$ and $g' \in \prod^{\cdot} g$ holds $f' +\cdot g' \in \prod^{\cdot}(f +\cdot g)$.

(35) If $g \in \prod^{\cdot} f$ and $h \in \prod^{\cdot} f$, then $g +\cdot h \in \prod^{\cdot} f$.

(36) For arbitrary x_1, x_2, y_1, y_2 such that $x_1 \in \operatorname{dom} f$ and $y_1 \in f(x_1)$ and $x_2 \in \operatorname{dom} f$ and $y_2 \in f(x_2)$ holds $[x_1 \longmapsto y_1, x_2 \longmapsto y_2] \in \prod^{\cdot} f$.

3. Computations

In the sequel N is a non-empty set with non-empty elements.

We now define five new constructions. Let us consider N, and let S be a von Neumann definite AMI over N, and let s be a state of S. The functor $\operatorname{CurInstr}(s)$ yields an instruction of S and is defined as follows:

(Def.2) $\operatorname{CurInstr}(s) = s(\mathbf{IC}_s)$.

Let us consider N, and let S be a von Neumann definite AMI over N, and let s be a state of S. The functor $\operatorname{Following}(s)$ yielding a state of S is defined by:

(Def.3) $\operatorname{Following}(s) = \operatorname{Exec}(\operatorname{CurInstr}(s), s)$.

Let us consider N, and let S be a von Neumann definite AMI over N, and let s be a state of S. The functor $\operatorname{Computation}(s)$ yielding a function from $\mathbb{N}$ into $\prod$(the object kind of S) **qua** a non-empty set is defined by:

(Def.4) $(\operatorname{Computation}(s))(0) = s$ **qua** an element of $\prod$(the object kind of S) **qua** a non-empty set and for every i and for every element x of $\prod$(the object kind of S) **qua** a non-empty set such that $x = (\operatorname{Computation}(s))(i)$ holds $(\operatorname{Computation}(s))(i+1) = \operatorname{Following}(x)$.

Let us consider N, and let S be a von Neumann definite AMI over N. A state of S is halting if:

(Def.5) there exists k such that CurInstr((Computation(it))(k)) = $\mathbf{halt}_S$.

Let us consider N, and let S be an AMI over N, and let f be a function from $\mathbb{N}$ into $\prod$(the object kind of S) **qua** a non-empty set, and let us consider k. Then $f(k)$ is a state of S. Let us consider N. An AMI over N is realistic if:

(Def.6) the instructions of it $\neq$ the instruction locations of it.

One can prove the following proposition

(37) For every S being a von Neumann definite AMI over N such that S is realistic holds for no instruction-location l of S holds $\mathbf{IC}_S = l$.

In the sequel S denotes a von Neumann definite AMI over N and s denotes a state of S. One can prove the following propositions:

(38) (Computation(s))(0) = s.

(39) (Computation(s))($k+1$) = Following((Computation(s))(k)).

(40) For every k holds
(Computation(s))($i+k$) = (Computation((Computation(s))(i)))(k).

(41) If $i \leq j$, then for every N and for every S being a halting von Neumann definite AMI over N and for every state s of S such that
CurInstr((Computation(s))(i)) = $\mathbf{halt}_S$
holds (Computation(s))(j) = (Computation(s))(i).

Let us consider N, and let S be a halting von Neumann definite AMI over N, and let s be a state of S satisfying the condition: s is halting. The functor Result(s) yields a state of S and is defined as follows:

(Def.7) there exists k such that Result(s) = (Computation(s))(k) and CurInstr(Result(s)) = $\mathbf{halt}_S$.

Next we state the proposition

(42) For every N and for every S being a steady-programmed von Neumann definite AMI over N and for every state s of S and for every instruction-location i of S holds $s(i)$ = (Following(s))(i).

Let us consider N, and let S be a definite AMI over N, and let s be a state of S, and let l be an instruction-location of S. Then $s(l)$ is an instruction of S.

Next we state several propositions:

(43) For every N and for every S being a steady-programmed von Neumann definite AMI over N and for every state s of S and for every instruction-location i of S and for every k holds $s(i)$ = (Computation(s))(k)(i).

(44) For every N and for every S being a steady-programmed von Neumann definite AMI over N and for every state s of S holds (Computation(s))($k+1$) = Exec($s(\mathbf{IC}_{(\mathrm{Computation}(s))(k)})$, (Computation($s$))($k$)).

(45) For every N and for every S being a steady-programmed von Neumann halting definite AMI over N and for every state s of S and for every k such that $s(\mathbf{IC}_{(\mathrm{Computation}(s))(k)}) = \mathbf{halt}_S$
holds Result(s) = (Computation(s))(k).

(46) For every N and for every S being a steady-programmed von Neumann halting definite AMI over N and for every state s of S such that there exists k such that $s(\mathbf{IC}_{(\mathrm{Computation}(s))(k)}) = \mathbf{halt}_S$ and for every i holds $\mathrm{Result}(s) = \mathrm{Result}((\mathrm{Computation}(s))(i))$.

(47) For every S being an AMI over N and for every object o of S holds $\mathrm{ObjectKind}(o)$ is non-empty.

4. Finite partial states

We now define five new constructions. Let us consider N, and let S be an AMI over N. The functor $\mathrm{FinPartSt}(S)$ yielding a subset of $\prod^{\cdot}$ (the object kind of S) is defined by:

(Def.8) $\mathrm{FinPartSt}(S) = \{p : p \text{ is finite}\}$, where p ranges over elements of $\prod^{\cdot}$ (the object kind of S).

Let us consider N, and let S be an AMI over N. An element of $\prod^{\cdot}$ (the object kind of S) is called a finite partial state of S if:

(Def.9) it is finite.

Let us consider N, and let S be a von Neumann definite AMI over N. A finite partial state of S is autonomic if:

(Def.10) for all states s_1, s_2 of S such that it $\subseteq s_1$ and it $\subseteq s_2$ and for every i holds $(\mathrm{Computation}(s_1))(i) \restriction \mathrm{dom}\,\mathrm{it} = (\mathrm{Computation}(s_2))(i) \restriction \mathrm{dom}\,\mathrm{it}$.

A finite partial state of S is halting if:

(Def.11) for every state s of S such that it $\subseteq s$ holds s is halting.

Let us consider N. A von Neumann definite AMI over N is programmable if:

(Def.12) there exists a finite partial state of it which is non-empty and autonomic.

We now state two propositions:

(48) For every S being a von Neumann definite AMI over N and for all non-empty sets A, B and for all objects l_1, l_2 of S such that $\mathrm{ObjectKind}(l_1) = A$ and $\mathrm{ObjectKind}(l_2) = B$ and for every element a of A and for every element b of B holds $[l_1 \longmapsto a, l_2 \longmapsto b]$ is a finite partial state of S.

(49) For every S being a von Neumann definite AMI over N and for every non-empty set A and for every object l_1 of S such that $\mathrm{ObjectKind}(l_1) = A$ and for every element a of A holds $l_1 \dot{\longmapsto} a$ is a finite partial state of S.

Let us consider N, and let S be a von Neumann definite AMI over N, and let l_1 be an object of S, and let a be an element of $\mathrm{ObjectKind}(l_1)$. Then $l_1 \dot{\longmapsto} a$ is a finite partial state of S. Let us consider N, and let S be a von Neumann definite AMI over N, and let l_1, l_2 be objects of S, and let a be an element of $\mathrm{ObjectKind}(l_1)$, and let b be an element of $\mathrm{ObjectKind}(l_2)$. Then $[l_1 \longmapsto a, l_2 \longmapsto b]$ is a finite partial state of S.

5. Trivial AMI

Let us consider N. The functor $\mathbf{AMI}_{\rm t}$ yields a strict AMI over N and is defined by the conditions (Def.13).

(Def.13) (i) The objects of $\mathbf{AMI}_{\rm t} = \{0, 1\}$,

(ii) the instruction counter of $\mathbf{AMI}_{\rm t} = 0$,

(iii) the instruction locations of $\mathbf{AMI}_{\rm t} = \{1\}$,

(iv) the instruction codes of $\mathbf{AMI}_{\rm t} = \{0\}$,

(v) the halt instruction of $\mathbf{AMI}_{\rm t} = 0$,

(vi) the instructions of $\mathbf{AMI}_{\rm t} = \{\langle 0, \varepsilon \rangle\}$,

(vii) the object kind of $\mathbf{AMI}_{\rm t} = [0 \longmapsto \{1\}, 1 \longmapsto \{\langle 0, \varepsilon \rangle\}]$,

(viii) the execution of $\mathbf{AMI}_{\rm t} = \{\langle 0, \varepsilon \rangle\} \longmapsto {\rm id}_{\prod [0 \longmapsto \{1\}, 1 \longmapsto \{\langle 0, \varepsilon \rangle\}]}$.

Next we state several propositions:

(50) $\mathbf{AMI}_{\rm t}$ is von Neumann.

(51) $\mathbf{AMI}_{\rm t}$ is data-oriented.

(52) $\mathbf{AMI}_{\rm t}$ is halting.

(53) For all states s_1, s_2 of $\mathbf{AMI}_{\rm t}$ holds $s_1 = s_2$.

(54) $\mathbf{AMI}_{\rm t}$ is steady-programmed.

(55) $\mathbf{AMI}_{\rm t}$ is definite.

(56) $\mathbf{AMI}_{\rm t}$ is realistic.

Let us consider N. Then $\mathbf{AMI}_{\rm t}$ is a von Neumann definite strict AMI over N.

One can prove the following proposition

(57) $\mathbf{AMI}_{\rm t}$ is programmable.

Let us consider N. Note that there exists a von Neumann definite strict AMI over N which is data-oriented halting steady-programmed realistic and programmable.

One can prove the following two propositions:

(58) For every S being an AMI over N and for every state s of S and for every finite partial state p of S holds $s \restriction {\rm dom}\, p$ is a finite partial state of S.

(59) For every S being an AMI over N holds $\emptyset$ is a finite partial state of S.

Let us consider N, and let S be a von Neumann definite AMI over N. Observe that there exists a non-empty autonomic finite partial state of S.

Let us consider N, and let S be an AMI over N, and let f, g be finite partial states of S. Then $f +\cdot g$ is a finite partial state of S.

6. Autonomic finite partial states

We now state four propositions:

(60) For every S being a realistic von Neumann definite AMI over N and for every instruction-location l_3 of S and for every element l of ObjectKind($\mathbf{IC}_S$) such that $l = l_3$ and for every element h of ObjectKind(l_3) such that $h = \mathbf{halt}_S$ and for every state s of S such that $[\mathbf{IC}_S \longmapsto l, l_3 \longmapsto h] \subseteq s$ holds CurInstr$(s) = \mathbf{halt}_S$.

(61) For every S being a realistic von Neumann definite AMI over N and for every instruction-location l_3 of S and for every element l of ObjectKind($\mathbf{IC}_S$) such that $l = l_3$ and for every element h of ObjectKind(l_3) such that $h = \mathbf{halt}_S$ holds $[\mathbf{IC}_S \longmapsto l, l_3 \longmapsto h]$ is halting.

(62) Let S be a realistic halting von Neumann definite AMI over N. Then for every instruction-location l_3 of S and for every element l of ObjectKind($\mathbf{IC}_S$) such that $l = l_3$ and for every element h of ObjectKind(l_3) such that $h = \mathbf{halt}_S$ and for every state s of S such that $[\mathbf{IC}_S \longmapsto l, l_3 \longmapsto h] \subseteq s$ and for every i holds $(\text{Computation}(s))(i) = s$.

(63) For every S being a realistic halting von Neumann definite AMI over N and for every instruction-location l_3 of S and for every element l of ObjectKind($\mathbf{IC}_S$) such that $l = l_3$ and for every element h of ObjectKind(l_3) such that $h = \mathbf{halt}_S$ holds $[\mathbf{IC}_S \longmapsto l, l_3 \longmapsto h]$ is autonomic.

We now define two new constructions. Let us consider N, and let S be a realistic halting von Neumann definite AMI over N. One can check that there exists a finite partial state of S which is autonomic and halting.

Let us consider N, and let S be a realistic halting von Neumann definite AMI over N. A pre-program of S is an autonomic halting finite partial state of S.

Let us consider N, and let S be a realistic halting von Neumann definite AMI over N, and let s be a finite partial state of S. Let us assume that s is a pre-program of S. The functor Result(s) yields a finite partial state of S and is defined as follows:

(Def.14) for every state s' of S such that $s \subseteq s'$ holds $\text{Result}(s) = \text{Result}(s') \restriction \text{dom}\, s$.

7. Pre-programs and programs

Let us consider N, and let S be a realistic halting von Neumann definite AMI over N, and let p be a finite partial state of S, and let F be a function. We say that p computes F if and only if:

(Def.15) for an arbitrary x such that $x \in \text{dom}\, F$ there exists a finite partial state s of S such that $x = s$ and $p +\cdot s$ is a pre-program of S and $F(s) \subseteq \text{Result}(p +\cdot s)$.

The following three propositions are true:

(64) For every S being a realistic halting von Neumann definite AMI over N and for every finite partial state p of S holds p computes $\square$.

(65) For every S being a realistic halting von Neumann definite AMI over N and for every finite partial state p of S holds p is a pre-program of S if and only if p computes $\emptyset \dot{\longmapsto} \mathrm{Result}(p)$.

(66) For every S being a realistic halting von Neumann definite AMI over N and for every finite partial state p of S holds p is a pre-program of S if and only if p computes $\emptyset \dot{\longmapsto} \emptyset$.

Let us consider N, and let S be a realistic halting von Neumann definite AMI over N. A partial function from $\mathrm{FinPartSt}(S)$ to $\mathrm{FinPartSt}(S)$ is computable if:

(Def.16) there exists a finite partial state p of S such that p computes it.

Next we state three propositions:

(67) For every N and for every S being a realistic halting von Neumann definite AMI over N and for every partial function F from $\mathrm{FinPartSt}(S)$ to $\mathrm{FinPartSt}(S)$ such that $F = \square$ holds F is computable.

(68) For every N and for every S being a realistic halting von Neumann definite AMI over N and for every partial function F from $\mathrm{FinPartSt}(S)$ to $\mathrm{FinPartSt}(S)$ such that $F = \emptyset \dot{\longmapsto} \emptyset$ holds F is computable.

(69) For every N and for every S being a realistic halting von Neumann definite AMI over N and for every pre-program p of S and for every partial function F from $\mathrm{FinPartSt}(S)$ to $\mathrm{FinPartSt}(S)$ such that $F = \emptyset \dot{\longmapsto} \mathrm{Result}(p)$ holds F is computable.

Let us consider N, and let S be a realistic halting von Neumann definite AMI over N, and let F be a partial function from $\mathrm{FinPartSt}(S)$ to $\mathrm{FinPartSt}(S)$ satisfying the condition: F is computable. A finite partial state of S is called a program of F if:

(Def.17) it computes F.

The following propositions are true:

(70) For every N and for every S being a realistic halting von Neumann definite AMI over N and for every partial function F from $\mathrm{FinPartSt}(S)$ to $\mathrm{FinPartSt}(S)$ such that $F = \square$ every finite partial state of S is a program of F.

(71) For every N and for every S being a realistic halting von Neumann definite AMI over N and for every partial function F from $\mathrm{FinPartSt}(S)$ to $\mathrm{FinPartSt}(S)$ such that $F = \emptyset \dot{\longmapsto} \emptyset$ every pre-program of S is a program of F.

(72) For every N and for every S being a realistic halting von Neumann definite AMI over N and for every pre-program p of S and for every partial function F from $\mathrm{FinPartSt}(S)$ to $\mathrm{FinPartSt}(S)$ such that $F = \emptyset \dot{\longmapsto} \mathrm{Result}(p)$ holds p is a program of F.

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[2] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[3] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[4] Czesław Byliński. Basic functions and operations on functions. *Formalized Mathematics*, 1(**1**):245–254, 1990.
[5] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.
[6] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[7] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[8] Czesław Byliński. Graphs of functions. *Formalized Mathematics*, 1(**1**):169–173, 1990.
[9] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[10] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.
[11] Czesław Byliński. Products and coproducts in categories. *Formalized Mathematics*, 2(**5**):701–709, 1991.
[12] Czesław Byliński. Subcategories and products of categories. *Formalized Mathematics*, 1(**4**):725–732, 1990.
[13] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[14] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[15] Rafał Kwiatek and Grzegorz Zwara. The divisibility of integers and integer relative primes. *Formalized Mathematics*, 1(**5**):829–832, 1990.
[16] Michał Muzalewski. Rings and modules - part II. *Formalized Mathematics*, 2(**4**):579–585, 1991.
[17] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[18] Henryk Oryszczyszyn and Krzysztof Prażmowski. Real functions spaces. *Formalized Mathematics*, 1(**3**):555–561, 1990.
[19] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.
[20] Dariusz Surowik. Cyclic groups and some of their properties - part I. *Formalized Mathematics*, 2(**5**):623–627, 1991.
[21] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[22] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[23] Andrzej Trybulec. Function domains and Frænkel operator. *Formalized Mathematics*, 1(**3**):495–500, 1990.
[24] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[25] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[26] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[27] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[28] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[29] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received December 29, 1992