JOURNAL OF FORMALIZED MATHEMATICS
Volume 5, Released 1993, Published 2003
Inst. of Computer Science, Univ. of Białystok

Development of Terminology for SCM[1]

Grzegorz Bancerek
Polish Academy of Sciences
Institute of Mathematics
Warsaw

Piotr Rudnicki
University of Alberta
Department of Computing Science
Edmonton

Summary. We develop a higher level terminology for the **SCM** machine defined by Nakamura and Trybulec in [5]. Among numerous technical definitions and lemmas we define a complexity measure of a halting state of **SCM** and a loader for **SCM** for arbitrary finite sequence of instructions. In order to test the introduced terminology we discuss properties of eight shortest halting programs, one for each instruction.

MML Identifier: SCM_1.

WWW: http://mizar.org/JFM/Vol5/scm_1.html

The articles [6], [11], [8], [1], [10], [7], [12], [3], [4], [2], [5], and [9] provide the notation and terminology for this paper.

Let i be an integer. Then $\langle i \rangle$ is a finite sequence of elements of $\mathbb{Z}$.

The following propositions are true:

(1) For every state s of **SCM** holds $\mathbf{IC}_s = s(0)$ and $\mathrm{CurInstr}(s) = s(s(0))$.

(2) For every state s of **SCM** and for every natural number k holds $\mathrm{CurInstr}((\mathrm{Computation}(s))(k)) = s(\mathbf{IC}_{(\mathrm{Computation}(s))(k)})$ and $\mathrm{CurInstr}((\mathrm{Computation}(s))(k)) = s((\mathrm{Computation}(s))(k)(0))$.

(3) For every state s of **SCM** such that there exists a natural number k such that $s(\mathbf{IC}_{(\mathrm{Computation}(s))(k)}) = \mathbf{halt}_{\mathbf{SCM}}$ holds s is halting.

(4) For every state s of **SCM** and for every natural number k such that $s(\mathbf{IC}_{(\mathrm{Computation}(s))(k)}) = \mathbf{halt}_{\mathbf{SCM}}$ holds $\mathrm{Result}(s) = (\mathrm{Computation}(s))(k)$.

(7)[1] For all natural numbers n, m holds $\mathbf{IC}_{\mathbf{SCM}} \neq \mathbf{i}_n$ and $\mathbf{IC}_{\mathbf{SCM}} \neq \mathbf{d}_n$ and $\mathbf{i}_n \neq \mathbf{d}_m$.

Let I be a finite sequence of elements of the instructions of **SCM**, let D be a finite sequence of elements of $\mathbb{Z}$, and let i_1, p_1, d_1 be natural numbers. A state of **SCM** is called a state with instruction counter on i_1, with I located from p_1, and D from d_1 if it satisfies the conditions (Def. 1).

(Def. 1)(i) $\mathbf{IC}_{\mathrm{it}} = \mathbf{i}_{(i_1)}$,

(ii) for every natural number k such that $k < \mathrm{len}\, I$ holds $\mathrm{it}(\mathbf{i}_{p_1+k}) = I(k+1)$, and

(iii) for every natural number k such that $k < \mathrm{len}\, D$ holds $\mathrm{it}(\mathbf{d}_{d_1+k}) = D(k+1)$.

The following propositions are true:

[1]This work was partially supported by NSERC Grant OGP9207 while the first author visited University of Alberta, May–June 1993.

[1] The propositions (5) and (6) have been removed.

© Association of Mizar Users

(8) Let x_1, x_2, x_3, x_4 be sets and p be a finite sequence. If $p = \langle x_1 \rangle \frown \langle x_2 \rangle \frown \langle x_3 \rangle \frown \langle x_4 \rangle$, then $\operatorname{len} p = 4$ and $p(1) = x_1$ and $p(2) = x_2$ and $p(3) = x_3$ and $p(4) = x_4$.

(9) Let x_1, x_2, x_3, x_4, x_5 be sets and p be a finite sequence. Suppose $p = \langle x_1 \rangle \frown \langle x_2 \rangle \frown \langle x_3 \rangle \frown \langle x_4 \rangle \frown \langle x_5 \rangle$. Then $\operatorname{len} p = 5$ and $p(1) = x_1$ and $p(2) = x_2$ and $p(3) = x_3$ and $p(4) = x_4$ and $p(5) = x_5$.

(10) Let x_1, x_2, x_3, x_4, x_5, x_6 be sets and p be a finite sequence. Suppose $p = \langle x_1 \rangle \frown \langle x_2 \rangle \frown \langle x_3 \rangle \frown \langle x_4 \rangle \frown \langle x_5 \rangle \frown \langle x_6 \rangle$. Then $\operatorname{len} p = 6$ and $p(1) = x_1$ and $p(2) = x_2$ and $p(3) = x_3$ and $p(4) = x_4$ and $p(5) = x_5$ and $p(6) = x_6$.

(11) Let x_1, x_2, x_3, x_4, x_5, x_6, x_7 be sets and p be a finite sequence. Suppose $p = \langle x_1 \rangle \frown \langle x_2 \rangle \frown \langle x_3 \rangle \frown \langle x_4 \rangle \frown \langle x_5 \rangle \frown \langle x_6 \rangle \frown \langle x_7 \rangle$. Then $\operatorname{len} p = 7$ and $p(1) = x_1$ and $p(2) = x_2$ and $p(3) = x_3$ and $p(4) = x_4$ and $p(5) = x_5$ and $p(6) = x_6$ and $p(7) = x_7$.

(12) Let x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8 be sets and p be a finite sequence. Suppose $p = \langle x_1 \rangle \frown \langle x_2 \rangle \frown \langle x_3 \rangle \frown \langle x_4 \rangle \frown \langle x_5 \rangle \frown \langle x_6 \rangle \frown \langle x_7 \rangle \frown \langle x_8 \rangle$. Then $\operatorname{len} p = 8$ and $p(1) = x_1$ and $p(2) = x_2$ and $p(3) = x_3$ and $p(4) = x_4$ and $p(5) = x_5$ and $p(6) = x_6$ and $p(7) = x_7$ and $p(8) = x_8$.

(13) Let x_1, x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9 be sets and p be a finite sequence. Suppose $p = \langle x_1 \rangle \frown \langle x_2 \rangle \frown \langle x_3 \rangle \frown \langle x_4 \rangle \frown \langle x_5 \rangle \frown \langle x_6 \rangle \frown \langle x_7 \rangle \frown \langle x_8 \rangle \frown \langle x_9 \rangle$. Then $\operatorname{len} p = 9$ and $p(1) = x_1$ and $p(2) = x_2$ and $p(3) = x_3$ and $p(4) = x_4$ and $p(5) = x_5$ and $p(6) = x_6$ and $p(7) = x_7$ and $p(8) = x_8$ and $p(9) = x_9$.

(14) Let I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9 be instructions of **SCM**, i_2, i_3, i_4, i_5 be integers, i_1 be a natural number, and s be a state with instruction counter on i_1, with $\langle I_1 \rangle \frown \langle I_2 \rangle \frown \langle I_3 \rangle \frown \langle I_4 \rangle \frown \langle I_5 \rangle \frown \langle I_6 \rangle \frown \langle I_7 \rangle \frown \langle I_8 \rangle \frown \langle I_9 \rangle$ located from 0, and $\langle i_2 \rangle \frown \langle i_3 \rangle \frown \langle i_4 \rangle \frown \langle i_5 \rangle$ from 0. Then $\mathbf{IC}_s = \mathbf{i}_{(i_1)}$ and $s(\mathbf{i}_0) = I_1$ and $s(\mathbf{i}_1) = I_2$ and $s(\mathbf{i}_2) = I_3$ and $s(\mathbf{i}_3) = I_4$ and $s(\mathbf{i}_4) = I_5$ and $s(\mathbf{i}_5) = I_6$ and $s(\mathbf{i}_6) = I_7$ and $s(\mathbf{i}_7) = I_8$ and $s(\mathbf{i}_8) = I_9$ and $s(\mathbf{d}_0) = i_2$ and $s(\mathbf{d}_1) = i_3$ and $s(\mathbf{d}_2) = i_4$ and $s(\mathbf{d}_3) = i_5$.

(15) Let I_1, I_2 be instructions of **SCM**, i_2, i_3 be integers, i_1 be a natural number, and s be a state with instruction counter on i_1, with $\langle I_1 \rangle \frown \langle I_2 \rangle$ located from 0, and $\langle i_2 \rangle \frown \langle i_3 \rangle$ from 0. Then $\mathbf{IC}_s = \mathbf{i}_{(i_1)}$ and $s(\mathbf{i}_0) = I_1$ and $s(\mathbf{i}_1) = I_2$ and $s(\mathbf{d}_0) = i_2$ and $s(\mathbf{d}_1) = i_3$.

Let N be a set with non empty elements, let S be a halting IC-Ins-separated definite non empty non void AMI over N, and let s be a state of S. Let us assume that s is halting. The complexity of s is a natural number and is defined by the conditions (Def. 2).

(Def. 2)(i) CurInstr((Computation(s))(the complexity of s)) $= \mathbf{halt}_S$, and

(ii) for every natural number k such that CurInstr((Computation(s))(k)) $= \mathbf{halt}_S$ holds the complexity of $s \leq k$.

We introduce LifeSpan(s) as a synonym of the complexity of s.

We now state a number of propositions:

(16) Let s be a state of **SCM** and k be a natural number. Then $s(\mathbf{IC}_{(\mathrm{Computation}(s))(k)}) \neq \mathbf{halt}_{\mathbf{SCM}}$ and $s(\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)}) = \mathbf{halt}_{\mathbf{SCM}}$ if and only if the complexity of $s = k+1$ and s is halting.

(17) Let s be a state of **SCM** and k be a natural number. If $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} \neq \mathbf{IC}_{(\mathrm{Computation}(s))(k+1)}$ and $s(\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)}) = \mathbf{halt}_{\mathbf{SCM}}$, then the complexity of $s = k+1$.

(18) Let k, n be natural numbers, s be a state of **SCM**, and a, b be data-locations. Suppose $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{i}_n$ and $s(\mathbf{i}_n) = a{:=}b$. Then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = \mathbf{i}_{n+1}$ and $(\mathrm{Computation}(s))(k+1)(a) = (\mathrm{Computation}(s))(k)(b)$ and for every data-location d such that $d \neq a$ holds $(\mathrm{Computation}(s))(k+1)(d) = (\mathrm{Computation}(s))(k)(d)$.

(19) Let k, n be natural numbers, s be a state of **SCM**, and a, b be data-locations. Suppose $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{i}_n$ and $s(\mathbf{i}_n) = \mathrm{AddTo}(a,b)$. Then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = \mathbf{i}_{n+1}$ and $(\mathrm{Computation}(s))(k+1)(a) = (\mathrm{Computation}(s))(k)(a) + (\mathrm{Computation}(s))(k)(b)$ and for every data-location d such that $d \neq a$ holds $(\mathrm{Computation}(s))(k+1)(d) = (\mathrm{Computation}(s))(k)(d)$.

(20) Let k, n be natural numbers, s be a state of **SCM**, and a, b be data-locations. Suppose $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{i}_n$ and $s(\mathbf{i}_n) = \mathrm{SubFrom}(a,b)$. Then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = \mathbf{i}_{n+1}$ and $(\mathrm{Computation}(s))(k+1)(a) = (\mathrm{Computation}(s))(k)(a) - (\mathrm{Computation}(s))(k)(b)$ and for every data-location d such that $d \neq a$ holds $(\mathrm{Computation}(s))(k+1)(d) = (\mathrm{Computation}(s))(k)(d)$.

(21) Let k, n be natural numbers, s be a state of **SCM**, and a, b be data-locations. Suppose $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{i}_n$ and $s(\mathbf{i}_n) = \mathrm{MultBy}(a,b)$. Then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = \mathbf{i}_{n+1}$ and $(\mathrm{Computation}(s))(k+1)(a) = (\mathrm{Computation}(s))(k)(a) \cdot (\mathrm{Computation}(s))(k)(b)$ and for every data-location d such that $d \neq a$ holds $(\mathrm{Computation}(s))(k+1)(d) = (\mathrm{Computation}(s))(k)(d)$.

(22) Let k, n be natural numbers, s be a state of **SCM**, and a, b be data-locations. Suppose $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{i}_n$ and $s(\mathbf{i}_n) = \mathrm{Divide}(a,b)$ and $a \neq b$. Then

(i) $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = \mathbf{i}_{n+1}$,

(ii) $(\mathrm{Computation}(s))(k+1)(a) = (\mathrm{Computation}(s))(k)(a) \div (\mathrm{Computation}(s))(k)(b)$,

(iii) $(\mathrm{Computation}(s))(k+1)(b) = (\mathrm{Computation}(s))(k)(a) \bmod (\mathrm{Computation}(s))(k)(b)$, and

(iv) for every data-location d such that $d \neq a$ and $d \neq b$ holds $(\mathrm{Computation}(s))(k+1)(d) = (\mathrm{Computation}(s))(k)(d)$.

(23) Let k, n be natural numbers, s be a state of **SCM**, and i_1 be an instruction-location of **SCM**. Suppose $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{i}_n$ and $s(\mathbf{i}_n) = \mathrm{goto}\ i_1$. Then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = i_1$ and for every data-location d holds $(\mathrm{Computation}(s))(k+1)(d) = (\mathrm{Computation}(s))(k)(d)$.

(24) Let k, n be natural numbers, s be a state of **SCM**, a be a data-location, and i_1 be an instruction-location of **SCM**. Suppose $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{i}_n$ and $s(\mathbf{i}_n) = \mathbf{if}\ a = 0\ \mathbf{goto}\ i_1$. Then

(i) if $(\mathrm{Computation}(s))(k)(a) = 0$, then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = i_1$,

(ii) if $(\mathrm{Computation}(s))(k)(a) \neq 0$, then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = \mathbf{i}_{n+1}$, and

(iii) for every data-location d holds $(\mathrm{Computation}(s))(k+1)(d) = (\mathrm{Computation}(s))(k)(d)$.

(25) Let k, n be natural numbers, s be a state of **SCM**, a be a data-location, and i_1 be an instruction-location of **SCM**. Suppose $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{i}_n$ and $s(\mathbf{i}_n) = \mathbf{if}\ a > 0\ \mathbf{goto}\ i_1$. Then

(i) if $(\mathrm{Computation}(s))(k)(a) > 0$, then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = i_1$,

(ii) if $(\mathrm{Computation}(s))(k)(a) \leq 0$, then $\mathbf{IC}_{(\mathrm{Computation}(s))(k+1)} = \mathbf{i}_{n+1}$, and

(iii) for every data-location d holds $(\mathrm{Computation}(s))(k+1)(d) = (\mathrm{Computation}(s))(k)(d)$.

(26) $(\mathbf{halt}_{\mathbf{SCM}})_{\mathbf{1}} = 0$ and for all data-locations a, b holds $(a{:=}b)_{\mathbf{1}} = 1$ and for all data-locations a, b holds $(\mathrm{AddTo}(a,b))_{\mathbf{1}} = 2$ and for all data-locations a, b holds $(\mathrm{SubFrom}(a,b))_{\mathbf{1}} = 3$ and for all data-locations a, b holds $(\mathrm{MultBy}(a,b))_{\mathbf{1}} = 4$ and for all data-locations a, b holds $(\mathrm{Divide}(a,b))_{\mathbf{1}} = 5$ and for every instruction-location i of **SCM** holds $(\mathrm{goto}\ i)_{\mathbf{1}} = 6$ and for every data-location a and for every instruction-location i of **SCM** holds $(\mathbf{if}\ a = 0\ \mathbf{goto}\ i)_{\mathbf{1}} = 7$ and for every data-location a and for every instruction-location i of **SCM** holds $(\mathbf{if}\ a > 0\ \mathbf{goto}\ i)_{\mathbf{1}} = 8$.

(27) Let N be a non empty set with non empty elements, S be an IC-Ins-separated definite halting non empty non void AMI over N, s be a state of S, and m be a natural number. Then s is halting if and only if $(\mathrm{Computation}(s))(m)$ is halting.

(28) Let s_1, s_2 be states of **SCM** and k, c be natural numbers. Suppose $s_2 = (\mathrm{Computation}(s_1))(k)$ and the complexity of $s_2 = c$ and s_2 is halting and $0 < c$. Then the complexity of $s_1 = k + c$.

(29) For all states s_1, s_2 of **SCM** and for every natural number k such that $s_2 = (\mathrm{Computation}(s_1))(k)$ and s_2 is halting holds $\mathrm{Result}(s_2) = \mathrm{Result}(s_1)$.

(30) Let I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9 be instructions of **SCM**, i_2, i_3, i_4, i_5 be integers, i_1 be a natural number, and s be a state of **SCM**. Suppose that $\mathbf{IC}_s = \mathbf{i}_{(i_1)}$ and $s(\mathbf{i}_0) = I_1$ and $s(\mathbf{i}_1) = I_2$ and $s(\mathbf{i}_2) = I_3$ and $s(\mathbf{i}_3) = I_4$ and $s(\mathbf{i}_4) = I_5$ and $s(\mathbf{i}_5) = I_6$ and $s(\mathbf{i}_6) = I_7$ and $s(\mathbf{i}_7) = I_8$ and $s(\mathbf{i}_8) = I_9$ and $s(\mathbf{d}_0) = i_2$ and $s(\mathbf{d}_1) = i_3$ and $s(\mathbf{d}_2) = i_4$ and $s(\mathbf{d}_3) = i_5$. Then s is a state with instruction counter on i_1, with $\langle I_1 \rangle \frown \langle I_2 \rangle \frown \langle I_3 \rangle \frown \langle I_4 \rangle \frown \langle I_5 \rangle \frown \langle I_6 \rangle \frown \langle I_7 \rangle \frown \langle I_8 \rangle \frown \langle I_9 \rangle$ located from 0, and $\langle i_2 \rangle \frown \langle i_3 \rangle \frown \langle i_4 \rangle \frown \langle i_5 \rangle$ from 0.

(31) Let s be a state with instruction counter on 0, with $\langle \mathbf{halt_{SCM}} \rangle$ located from 0, and $\varepsilon_{\mathbb{Z}}$ from 0. Then s is halting and the complexity of $s = 0$ and $\mathrm{Result}(s) = s$.

(32) Let i_2, i_3 be integers and s be a state with instruction counter on 0, with $\langle \mathbf{d}_0{:=}\mathbf{d}_1 \rangle \frown \langle \mathbf{halt_{SCM}} \rangle$ located from 0, and $\langle i_2 \rangle \frown \langle i_3 \rangle$ from 0. Then

(i) s is halting,

(ii) the complexity of $s = 1$,

(iii) $(\mathrm{Result}(s))(\mathbf{d}_0) = i_3$, and

(iv) for every data-location d such that $d \neq \mathbf{d}_0$ holds $(\mathrm{Result}(s))(d) = s(d)$.

(33) Let i_2, i_3 be integers and s be a state with instruction counter on 0, with $\langle \mathrm{AddTo}(\mathbf{d}_0, \mathbf{d}_1) \rangle \frown \langle \mathbf{halt_{SCM}} \rangle$ located from 0, and $\langle i_2 \rangle \frown \langle i_3 \rangle$ from 0. Then

(i) s is halting,

(ii) the complexity of $s = 1$,

(iii) $(\mathrm{Result}(s))(\mathbf{d}_0) = i_2 + i_3$, and

(iv) for every data-location d such that $d \neq \mathbf{d}_0$ holds $(\mathrm{Result}(s))(d) = s(d)$.

(34) Let i_2, i_3 be integers and s be a state with instruction counter on 0, with $\langle \mathrm{SubFrom}(\mathbf{d}_0, \mathbf{d}_1) \rangle \frown \langle \mathbf{halt_{SCM}} \rangle$ located from 0, and $\langle i_2 \rangle \frown \langle i_3 \rangle$ from 0. Then

(i) s is halting,

(ii) the complexity of $s = 1$,

(iii) $(\mathrm{Result}(s))(\mathbf{d}_0) = i_2 - i_3$, and

(iv) for every data-location d such that $d \neq \mathbf{d}_0$ holds $(\mathrm{Result}(s))(d) = s(d)$.

(35) Let i_2, i_3 be integers and s be a state with instruction counter on 0, with $\langle \mathrm{MultBy}(\mathbf{d}_0, \mathbf{d}_1) \rangle \frown \langle \mathbf{halt_{SCM}} \rangle$ located from 0, and $\langle i_2 \rangle \frown \langle i_3 \rangle$ from 0. Then

(i) s is halting,

(ii) the complexity of $s = 1$,

(iii) $(\mathrm{Result}(s))(\mathbf{d}_0) = i_2 \cdot i_3$, and

(iv) for every data-location d such that $d \neq \mathbf{d}_0$ holds $(\mathrm{Result}(s))(d) = s(d)$.

(36) Let i_2, i_3 be integers and s be a state with instruction counter on 0, with $\langle \mathrm{Divide}(\mathbf{d}_0, \mathbf{d}_1) \rangle \frown \langle \mathbf{halt_{SCM}} \rangle$ located from 0, and $\langle i_2 \rangle \frown \langle i_3 \rangle$ from 0. Then

(i) s is halting,

(ii) the complexity of $s = 1$,

(iii) $(\mathrm{Result}(s))(\mathbf{d}_0) = i_2 \div i_3$,

(iv) $(\mathrm{Result}(s))(\mathbf{d}_1) = i_2 \mod i_3$, and

(v) for every data-location d such that $d \neq \mathbf{d}_0$ and $d \neq \mathbf{d}_1$ holds $(\mathrm{Result}(s))(d) = s(d)$.

(37) Let i_2, i_3 be integers and s be a state with instruction counter on 0, with $\langle$goto $(\mathbf{i}_1)\rangle ^\frown \langle\mathbf{halt_{SCM}}\rangle$ located from 0, and $\langle i_2\rangle ^\frown \langle i_3\rangle$ from 0. Then s is halting and the complexity of $s = 1$ and for every data-location d holds $(\mathrm{Result}(s))(d) = s(d)$.

(38) Let i_2, i_3 be integers and s be a state with instruction counter on 0, with $\langle\mathbf{if}\ \mathbf{d}_0 = 0\ \mathbf{goto}\ \mathbf{i}_1\rangle ^\frown \langle\mathbf{halt_{SCM}}\rangle$ located from 0, and $\langle i_2\rangle ^\frown \langle i_3\rangle$ from 0. Then s is halting and the complexity of $s = 1$ and for every data-location d holds $(\mathrm{Result}(s))(d) = s(d)$.

(39) Let i_2, i_3 be integers and s be a state with instruction counter on 0, with $\langle\mathbf{if}\ \mathbf{d}_0 > 0\ \mathbf{goto}\ \mathbf{i}_1\rangle ^\frown \langle\mathbf{halt_{SCM}}\rangle$ located from 0, and $\langle i_2\rangle ^\frown \langle i_3\rangle$ from 0. Then s is halting and the complexity of $s = 1$ and for every data-location d holds $(\mathrm{Result}(s))(d) = s(d)$.

REFERENCES

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/nat_1.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] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Journal of Formalized Mathematics*, 4, 1992. http://mizar.org/JFM/Vol4/ami_1.html.

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

[7] Andrzej Trybulec. Tuples, projections and Cartesian products. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/mcart_1.html.

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

[9] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Journal of Formalized Mathematics*, 5, 1993. http://mizar.org/JFM/Vol5/ami_3.html.

[10] Michał J. Trybulec. Integers. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/int_1.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 October 8, 1993

Published January 2, 2004