FORMALIZED MATHEMATICS
Volume 4, Number 1, 1993
Université Catholique de Louvain

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 [6]. 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`.

The notation and terminology used in this paper have been introduced in the following articles: [10], [1], [13], [11], [9], [4], [5], [2], [3], [8], [6], [7], and [12].

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

One can prove the following propositions:

(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 $\mathrm{Result}(s) = (\mathrm{Computation}(s))(k)$.

(5) For all natural numbers k, l such that $k \neq l$ holds $\mathbf{d}_k \neq \mathbf{d}_l$.

(6) For all natural numbers k, l such that $k \neq l$ holds $\mathbf{i}_k \neq \mathbf{i}_l$.

(7) 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 $\mathbf{i}_n \neq \mathbf{d}_m$.

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

ISSN 0777-4028

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 said to be 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) $\mathbf{IC}_{\rm it} = \mathbf{i}_{(i_1)}$,

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

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

One can prove the following propositions:

(8) Let x_1, x_2, x_3, x_4 be arbitrary and let 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 arbitrary and let 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 arbitrary and let 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 arbitrary and let 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 arbitrary and let 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 arbitrary and let 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**, and let i_2, i_3, i_4, i_5 be integers, and let i_1 be a natural number, and let 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

(i) $\mathbf{IC}_s = \mathbf{i}_{(i_1)}$,

(ii) $s(\mathbf{i}_0) = I_1$,

(iii) $s(\mathbf{i}_1) = I_2$,

(iv) $s(\mathbf{i}_2) = I_3$,

(v) $s(\mathbf{i}_3) = I_4$,

(vi) $s(\mathbf{i}_4) = I_5$,

(vii) $s(\mathbf{i}_5) = I_6$,

(viii) $s(\mathbf{i}_6) = I_7$,
(ix) $s(\mathbf{i}_7) = I_8$,
(x) $s(\mathbf{i}_8) = I_9$,
(xi) $s(\mathbf{d}_0) = i_2$,
(xii) $s(\mathbf{d}_1) = i_3$,
(xiii) $s(\mathbf{d}_2) = i_4$, and
(xiv) $s(\mathbf{d}_3) = i_5$.

(15) Let I_1, I_2 be instructions of **SCM**, and let i_2, i_3 be integers, and let i_1 be a natural number, and let 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 a, b be data-locations. Then $a{:=}b$ is an instruction of **SCM**. Then $\mathrm{AddTo}(a, b)$ is an instruction of **SCM**. Then $\mathrm{SubFrom}(a, b)$ is an instruction of **SCM**. Then $\mathrm{MultBy}(a, b)$ is an instruction of **SCM**. Then $\mathrm{Divide}(a, b)$ is an instruction of **SCM**.

Let l_1 be an instruction-location of **SCM**. Then goto l_1 is an instruction of **SCM**. Let a be a data-location. Then **if** $a = 0$ **goto** l_1 is an instruction of **SCM**. Then **if** $a > 0$ **goto** l_1 is an instruction of **SCM**.

Let s be a state of **SCM**. 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) $\mathrm{CurInstr}((\mathrm{Computation}(s))(\text{the complexity of } s)) = \mathbf{halt}_{\mathbf{SCM}}$, and
(ii) for every natural number k such that $\mathrm{CurInstr}((\mathrm{Computation}(s))(k)) = \mathbf{halt}_{\mathbf{SCM}}$ holds the complexity of $s \leq k$.

We now state a number of propositions:

(16) Let s be a state of **SCM** and let 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 let 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, and let s be a state of **SCM**, and let 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, and let s be a state of **SCM**, and let 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, and let s be a state of **SCM**, and let a, b be data-locations. Suppose $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} =$

$\mathbf{i}_n$ and $s(\mathbf{i}_n) = \text{SubFrom}(a, b)$. Then $\mathbf{IC}_{(\text{Computation}(s))(k+1)} = \mathbf{i}_{n+1}$ and $(\text{Computation}(s))(k+1)(a) = (\text{Computation}(s))(k)(a) - (\text{Computation}(s))(k)(b)$ and for every data-location d such that $d \neq a$ holds $(\text{Computation}(s))(k+1)(d) = (\text{Computation}(s))(k)(d)$.

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

(26) (i) $(\mathbf{halt}_{\mathbf{SCM}})_1 = 0$,

(ii) for all data-locations a, b holds $(a{:=}b)_1 = 1$,

(iii) for all data-locations a, b holds $(\text{AddTo}(a, b))_1 = 2$,

(iv) for all data-locations a, b holds $(\text{SubFrom}(a, b))_1 = 3$,

(v) for all data-locations a, b holds $(\text{MultBy}(a, b))_1 = 4$,

(vi) for all data-locations a, b holds $(\text{Divide}(a, b))_1 = 5$,

(vii) for every instruction-location i of **SCM** holds $(\text{goto } i)_1 = 6$,

(viii) for every data-location a and for every instruction-location i of **SCM** holds $(\mathbf{if}\ a = 0\ \mathbf{goto}\ i)_1 = 7$, and

(ix) for every data-location a and for every instruction-location i of **SCM** holds $(\mathbf{if}\ a > 0\ \mathbf{goto}\ i)_1 = 8$.

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

(28) Let s_1, s_2 be states of **SCM** and let k, c be natural numbers. Suppose $s_2 = (\text{Computation}(s_1))(k)$ 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 $s_2 = (\text{Computation}(s_1))(k)$ and s_2 is halting holds $\text{Result}(s_2) = \text{Result}(s_1)$.

(30) Let I_1, I_2, I_3, I_4, I_5, I_6, I_7, I_8, I_9 be instructions of **SCM**, and let i_2, i_3, i_4, i_5 be integers, and let i_1 be a natural number, and let s be a state of **SCM**. Suppose that

(i) $\mathbf{IC}_s = \mathbf{i}_{(i_1)}$,

(ii) $s(\mathbf{i}_0) = I_1$,

(iii) $s(\mathbf{i}_1) = I_2$,

(iv) $s(\mathbf{i}_2) = I_3$,

(v) $s(\mathbf{i}_3) = I_4$,

(vi) $s(\mathbf{i}_4) = I_5$,

(vii) $s(\mathbf{i}_5) = I_6$,

(viii) $s(\mathbf{i}_6) = I_7$,

(ix) $s(\mathbf{i}_7) = I_8$,

(x) $s(\mathbf{i}_8) = I_9$,

(xi) $s(\mathbf{d}_0) = i_2$,

(xii) $s(\mathbf{d}_1) = i_3$,

(xiii) $s(\mathbf{d}_2) = i_4$, and

(xiv) $s(\mathbf{d}_3) = i_5$.

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 $\langle \mathbf{halt}_{\mathbf{SCM}}\rangle$ located from 0, and $\varepsilon_{\mathbb{Z}}$ from 0. Then s is halting and the complexity of $s = 0$ and $\text{Result}(s) = s$.

(32) Let i_2, i_3 be integers and let s be a state with instruction counter on 0, with $\langle \mathbf{d}_0{:=}\mathbf{d}_1\rangle \frown \langle \mathbf{halt}_{\mathbf{SCM}}\rangle$ 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) $(\text{Result}(s))(\mathbf{d}_0) = i_3$, and

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

(33) Let i_2, i_3 be integers and let s be a state with instruction counter on 0, with $\langle \mathrm{AddTo}(\mathbf{d}_0, \mathbf{d}_1) \rangle \frown \langle \mathbf{halt}_{\mathbf{SCM}} \rangle$ 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 let s be a state with instruction counter on 0, with $\langle \mathrm{SubFrom}(\mathbf{d}_0, \mathbf{d}_1) \rangle \frown \langle \mathbf{halt}_{\mathbf{SCM}} \rangle$ 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 let s be a state with instruction counter on 0, with $\langle \mathrm{MultBy}(\mathbf{d}_0, \mathbf{d}_1) \rangle \frown \langle \mathbf{halt}_{\mathbf{SCM}} \rangle$ 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 let s be a state with instruction counter on 0, with $\langle \mathrm{Divide}(\mathbf{d}_0, \mathbf{d}_1) \rangle \frown \langle \mathbf{halt}_{\mathbf{SCM}} \rangle$ 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) $(\mathrm{Result}(s))(\mathbf{d}_1) = i_2 \bmod i_3$, 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 let s be a state with instruction counter on 0, with $\langle \mathrm{goto}\ (\mathrm{i}_1) \rangle \frown \langle \mathbf{halt}_{\mathbf{SCM}} \rangle$ 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 let s be a state with instruction counter on 0, with $\langle \mathbf{if}\ \mathbf{d}_0 = 0\ \mathbf{goto}\ \mathbf{i}_1 \rangle \frown \langle \mathbf{halt}_{\mathbf{SCM}} \rangle$ 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 let s be a state with instruction counter on 0, with $\langle \mathbf{if}\ \mathbf{d}_0 > 0\ \mathbf{goto}\ \mathbf{i}_1 \rangle \frown \langle \mathbf{halt}_{\mathbf{SCM}} \rangle$ 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. *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. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.

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

[6] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.

[7] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.

[8] Dariusz Surowik. Cyclic groups and some of their properties - part I. *Formalized Mathematics*, 2(**5**):623–627, 1991.

[9] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.

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

[11] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.

[12] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.

[13] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.

Received October 8, 1993