JOURNAL OF FORMALIZED MATHEMATICS
Volume 12, Released 2000, Published 2003
Inst. of Computer Science, Univ. of Białystok

The Properties of Instructions of SCM over Ring

Artur Korniłowicz
University of Białystok

MML Identifier: SCMRING3.

WWW: http://mizar.org/JFM/Vol12/scmring3.html

The articles [22], [29], [12], [23], [19], [30], [8], [9], [7], [2], [3], [24], [1], [27], [10], [5], [11], [20], [4], [6], [16], [28], [17], [18], [25], [21], [14], [13], [26], and [15] provide the notation and terminology for this paper.

For simplicity, we follow the rules: R is a good ring, r is an element of R, a, b are Data-Locations of R, i_1, i_2, i_3 are instruction-locations of $\mathbf{SCM}(R)$, I is an instruction of $\mathbf{SCM}(R)$, s_1, s_2 are states of $\mathbf{SCM}(R)$, T is an instruction type of $\mathbf{SCM}(R)$, and k is a natural number.

Let us mention that $\mathbb{Z}$ is infinite.

Let us mention that INT.Ring is infinite and good.

Let us note that there exists a 1-sorted structure which is strict and infinite.

Let us observe that there exists a ring which is strict, infinite, and good.

Next we state the proposition

(1) ObjectKind(a) = the carrier of R.

Let R be a good ring, let l_1, l_2 be Data-Locations of R, and let a, b be elements of R. Then $[l_1 \longmapsto a, l_2 \longmapsto b]$ is a finite partial state of $\mathbf{SCM}(R)$.

We now state a number of propositions:

(2) $a \notin$ the instruction locations of $\mathbf{SCM}(R)$.

(3) $a \neq \mathbf{IC}_{\mathbf{SCM}(R)}$.

(4) $\text{Data-Loc}_{\text{SCM}} \neq$ the instruction locations of $\mathbf{SCM}(R)$.

(5) For every object o of $\mathbf{SCM}(R)$ holds $o = \mathbf{IC}_{\mathbf{SCM}(R)}$ or $o \in$ the instruction locations of $\mathbf{SCM}(R)$ or o is a Data-Location of R.

(6) If $i_2 \neq i_3$, then $\text{Next}(i_2) \neq \text{Next}(i_3)$.

(7) If s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}(R)$, then $s_1(a) = s_2(a)$.

(8) $\text{InsCode}(\mathbf{halt}_{\mathbf{SCM}(R)}) = 0$.

(9) $\text{InsCode}(a{:=}b) = 1$.

(10) $\text{InsCode}(\text{AddTo}(a,b)) = 2$.

(11) $\text{InsCode}(\text{SubFrom}(a,b)) = 3$.

(12) $\text{InsCode}(\text{MultBy}(a,b)) = 4$.

(13) $\text{InsCode}(a{:=}r) = 5$.

(14) $\mathrm{InsCode}(\mathrm{goto}\ i_2) = 6$.

(15) $\mathrm{InsCode}(\mathbf{if}\ a = 0\ \mathbf{goto}\ i_2) = 7$.

(16) If $\mathrm{InsCode}(I) = 0$, then $I = \mathbf{halt}_{\mathbf{SCM}(R)}$.

(17) If $\mathrm{InsCode}(I) = 1$, then there exist a, b such that $I = a{:=}b$.

(18) If $\mathrm{InsCode}(I) = 2$, then there exist a, b such that $I = \mathrm{AddTo}(a, b)$.

(19) If $\mathrm{InsCode}(I) = 3$, then there exist a, b such that $I = \mathrm{SubFrom}(a, b)$.

(20) If $\mathrm{InsCode}(I) = 4$, then there exist a, b such that $I = \mathrm{MultBy}(a, b)$.

(21) If $\mathrm{InsCode}(I) = 5$, then there exist a, r such that $I = a{:=}r$.

(22) If $\mathrm{InsCode}(I) = 6$, then there exists i_3 such that $I = \mathrm{goto}\ i_3$.

(23) If $\mathrm{InsCode}(I) = 7$, then there exist a, i_2 such that $I = \mathbf{if}\ a = 0\ \mathbf{goto}\ i_2$.

(24) $\mathrm{AddressPart}(\mathbf{halt}_{\mathbf{SCM}(R)}) = \emptyset$.

(25) $\mathrm{AddressPart}(a{:=}b) = \langle a, b \rangle$.

(26) $\mathrm{AddressPart}(\mathrm{AddTo}(a, b)) = \langle a, b \rangle$.

(27) $\mathrm{AddressPart}(\mathrm{SubFrom}(a, b)) = \langle a, b \rangle$.

(28) $\mathrm{AddressPart}(\mathrm{MultBy}(a, b)) = \langle a, b \rangle$.

(29) $\mathrm{AddressPart}(a{:=}r) = \langle a, r \rangle$.

(30) $\mathrm{AddressPart}(\mathrm{goto}\ i_2) = \langle i_2 \rangle$.

(31) $\mathrm{AddressPart}(\mathbf{if}\ a = 0\ \mathbf{goto}\ i_2) = \langle i_2, a \rangle$.

(32) If $T = 0$, then $\mathrm{AddressParts}\, T = \{0\}$.

Let us consider R, T. One can check that $\mathrm{AddressParts}\, T$ is non empty.
The following propositions are true:

(33) If $T = 1$, then $\mathrm{dom} \prod_{\mathrm{AddressParts}\, T} = \{1, 2\}$.

(34) If $T = 2$, then $\mathrm{dom} \prod_{\mathrm{AddressParts}\, T} = \{1, 2\}$.

(35) If $T = 3$, then $\mathrm{dom} \prod_{\mathrm{AddressParts}\, T} = \{1, 2\}$.

(36) If $T = 4$, then $\mathrm{dom} \prod_{\mathrm{AddressParts}\, T} = \{1, 2\}$.

(37) If $T = 5$, then $\mathrm{dom} \prod_{\mathrm{AddressParts}\, T} = \{1, 2\}$.

(38) If $T = 6$, then $\mathrm{dom} \prod_{\mathrm{AddressParts}\, T} = \{1\}$.

(39) If $T = 7$, then $\mathrm{dom} \prod_{\mathrm{AddressParts}\, T} = \{1, 2\}$.

(40) $\prod_{\mathrm{AddressParts}\, \mathrm{InsCode}(a{:=}b)}(1) = \mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$.

(41) $\prod_{\mathrm{AddressParts}\, \mathrm{InsCode}(a{:=}b)}(2) = \mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$.

(42) $\prod_{\mathrm{AddressParts}\, \mathrm{InsCode}(\mathrm{AddTo}(a,b))}(1) = \mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$.

(43) $\prod_{\mathrm{AddressParts}\, \mathrm{InsCode}(\mathrm{AddTo}(a,b))}(2) = \mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$.

(44) $\prod_{\mathrm{AddressParts}\, \mathrm{InsCode}(\mathrm{SubFrom}(a,b))}(1) = \mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$.

(45) $\prod_{\mathrm{AddressParts}\, \mathrm{InsCode}(\mathrm{SubFrom}(a,b))}(2) = \mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$.

(46) $\prod_{\mathrm{AddressParts\,InsCode(MultBy}(a,b))}(1) = \text{Data-Loc}_{\mathrm{SCM}}$.

(47) $\prod_{\mathrm{AddressParts\,InsCode(MultBy}(a,b))}(2) = \text{Data-Loc}_{\mathrm{SCM}}$.

(48) $\prod_{\mathrm{AddressParts\,InsCode}(a:=r)}(1) = \text{Data-Loc}_{\mathrm{SCM}}$.

(49) $\prod_{\mathrm{AddressParts\,InsCode}(a:=r)}(2) =$ the carrier of R.

(50) $\prod_{\mathrm{AddressParts\,InsCode(goto}\ i_2)}(1) =$ the instruction locations of $\mathbf{SCM}(R)$.

(51) $\prod_{\mathrm{AddressParts\,InsCode}(\mathbf{if}\ a=0\ \mathbf{goto}\ i_2)}(1) =$ the instruction locations of $\mathbf{SCM}(R)$.

(52) $\prod_{\mathrm{AddressParts\,InsCode}(\mathbf{if}\ a=0\ \mathbf{goto}\ i_2)}(2) = \text{Data-Loc}_{\mathrm{SCM}}$.

(53) $\mathrm{NIC}(\mathbf{halt}_{\mathbf{SCM}(R)}, i_1) = \{i_1\}$.

Let us consider R. Note that $\mathrm{JUMP}(\mathbf{halt}_{\mathbf{SCM}(R)})$ is empty.
Next we state the proposition

(54) $\mathrm{NIC}(a{:=}b, i_1) = \{\mathrm{Next}(i_1)\}$.

Let us consider R, a, b. Note that $\mathrm{JUMP}(a{:=}b)$ is empty.
Next we state the proposition

(55) $\mathrm{NIC}(\mathrm{AddTo}(a,b), i_1) = \{\mathrm{Next}(i_1)\}$.

Let us consider R, a, b. Observe that $\mathrm{JUMP}(\mathrm{AddTo}(a,b))$ is empty.
Next we state the proposition

(56) $\mathrm{NIC}(\mathrm{SubFrom}(a,b), i_1) = \{\mathrm{Next}(i_1)\}$.

Let us consider R, a, b. Note that $\mathrm{JUMP}(\mathrm{SubFrom}(a,b))$ is empty.
The following proposition is true

(57) $\mathrm{NIC}(\mathrm{MultBy}(a,b), i_1) = \{\mathrm{Next}(i_1)\}$.

Let us consider R, a, b. Note that $\mathrm{JUMP}(\mathrm{MultBy}(a,b))$ is empty.
We now state the proposition

(58) $\mathrm{NIC}(a{:=}r, i_1) = \{\mathrm{Next}(i_1)\}$.

Let us consider R, a, r. One can verify that $\mathrm{JUMP}(a{:=}r)$ is empty.
Next we state two propositions:

(59) $\mathrm{NIC}(\mathrm{goto}\ i_2, i_1) = \{i_2\}$.

(60) $\mathrm{JUMP}(\mathrm{goto}\ i_2) = \{i_2\}$.

Let us consider R, i_2. Observe that $\mathrm{JUMP}(\mathrm{goto}\ i_2)$ is non empty and trivial.
Next we state three propositions:

(61) $i_2 \in \mathrm{NIC}(\mathbf{if}\ a = 0\ \mathbf{goto}\ i_2, i_1)$ and $\mathrm{NIC}(\mathbf{if}\ a = 0\ \mathbf{goto}\ i_2, i_1) \subseteq \{i_2, \mathrm{Next}(i_1)\}$.

(62) Let R be a non trivial good ring, a be a Data-Location of R, and i_1, i_2 be instruction-locations of $\mathbf{SCM}(R)$. Then $\mathrm{NIC}(\mathbf{if}\ a = 0\ \mathbf{goto}\ i_2, i_1) = \{i_2, \mathrm{Next}(i_1)\}$.

(63) $\mathrm{JUMP}(\mathbf{if}\ a = 0\ \mathbf{goto}\ i_2) = \{i_2\}$.

Let us consider R, a, i_2. One can check that $\mathrm{JUMP}(\mathbf{if}\ a = 0\ \mathbf{goto}\ i_2)$ is non empty and trivial.
We now state two propositions:

(64) $\mathrm{SUCC}(i_1) = \{i_1, \mathrm{Next}(i_1)\}$.

(65) Let f be a function from $\mathbb{N}$ into the instruction locations of $\mathbf{SCM}(R)$. Suppose that for every natural number k holds $f(k) = \mathbf{i}_k$. Then

(i) f is bijective, and

(ii) for every natural number k holds $f(k+1) \in \mathrm{SUCC}(f(k))$ and for every natural number j such that $f(j) \in \mathrm{SUCC}(f(k))$ holds $k \leq j$.

Let us consider R. One can check that $\mathbf{SCM}(R)$ is standard.
Next we state three propositions:

(66) $\mathrm{il}_{\mathbf{SCM}(R)}(k) = \mathbf{i}_k$.

(67) $\mathrm{Next}(\mathrm{il}_{\mathbf{SCM}(R)}(k)) = \mathrm{il}_{\mathbf{SCM}(R)}(k+1)$.

(68) $\mathrm{Next}(i_1) = \mathrm{NextLoc}\, i_1$.

Let R be a good ring and let k be a natural number. The functor $\mathrm{dl}_R(k)$ yields a Data-Location of R and is defined by:

(Def. 1) $\mathrm{dl}_R(k) = \mathbf{d}_k$.

Let us consider R. Observe that $\mathrm{InsCode}(\mathbf{halt}_{\mathbf{SCM}(R)})$ is jump-only.
Let us consider R. Observe that $\mathbf{halt}_{\mathbf{SCM}(R)}$ is jump-only.
Let us consider R, i_2. Observe that $\mathrm{InsCode}(\mathrm{goto}\ i_2)$ is jump-only.
Let us consider R, i_2. Observe that goto i_2 is jump-only.
Let us consider R, a, i_2. Note that $\mathrm{InsCode}(\mathbf{if}\ a = 0\ \mathbf{goto}\ i_2)$ is jump-only.
Let us consider R, a, i_2. Note that $\mathbf{if}\ a = 0\ \mathbf{goto}\ i_2$ is jump-only.
In the sequel S denotes a non trivial good ring, p, q denote Data-Locations of S, and w denotes an element of S.
Let us consider S, p, q. One can verify that $\mathrm{InsCode}(p{:=}q)$ is non jump-only.
Let us consider S, p, q. Observe that $p{:=}q$ is non jump-only.
Let us consider S, p, q. One can verify that $\mathrm{InsCode}(\mathrm{AddTo}(p,q))$ is non jump-only.
Let us consider S, p, q. One can check that $\mathrm{AddTo}(p,q)$ is non jump-only.
Let us consider S, p, q. Note that $\mathrm{InsCode}(\mathrm{SubFrom}(p,q))$ is non jump-only.
Let us consider S, p, q. Note that $\mathrm{SubFrom}(p,q)$ is non jump-only.
Let us consider S, p, q. Observe that $\mathrm{InsCode}(\mathrm{MultBy}(p,q))$ is non jump-only.
Let us consider S, p, q. Note that $\mathrm{MultBy}(p,q)$ is non jump-only.
Let us consider S, p, w. One can verify that $\mathrm{InsCode}(p{:=}w)$ is non jump-only.
Let us consider S, p, w. One can check that $p{:=}w$ is non jump-only.
Let us consider R, a, b. Note that $a{:=}b$ is sequential.
Let us consider R, a, b. Note that $\mathrm{AddTo}(a,b)$ is sequential.
Let us consider R, a, b. One can verify that $\mathrm{SubFrom}(a,b)$ is sequential.
Let us consider R, a, b. One can check that $\mathrm{MultBy}(a,b)$ is sequential.
Let us consider R, a, r. Note that $a{:=}r$ is sequential.
Let us consider R, i_2. Note that goto i_2 is non sequential.
Let us consider R, a, i_2. One can check that $\mathbf{if}\ a = 0\ \mathbf{goto}\ i_2$ is non sequential.
Let us consider R, i_2. One can verify that goto i_2 is non instruction location free.
Let us consider R, a, i_2. One can verify that $\mathbf{if}\ a = 0\ \mathbf{goto}\ i_2$ is non instruction location free.
Let us consider R. Note that $\mathbf{SCM}(R)$ is homogeneous and has explicit jumps and no implicit jumps.
Let us consider R. One can check that $\mathbf{SCM}(R)$ is regular.
The following two propositions are true:

(69) $\mathrm{IncAddr}(\mathrm{goto}\ i_2,k) = \mathrm{goto}\ \mathrm{il}_{\mathbf{SCM}(R)}(\mathrm{locnum}(i_2)+k)$.

(70) $\mathrm{IncAddr}(\mathbf{if}\ a = 0\ \mathbf{goto}\ i_2,k) = \mathbf{if}\ a = 0\ \mathbf{goto}\ \mathrm{il}_{\mathbf{SCM}(R)}(\mathrm{locnum}(i_2)+k)$.

Let us consider R. Note that $\mathbf{SCM}(R)$ is IC-good and Exec-preserving.

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. The ordinal numbers. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/ordinal1.html.

[3] Grzegorz Bancerek. Sequences of ordinal numbers. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/ordinal2.html.

[4] Grzegorz Bancerek. König's theorem. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/card_3.html.

[5] 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.

[6] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Journal of Formalized Mathematics*, 8, 1996. http://mizar.org/JFM/Vol8/funct_7.html.

[7] Józef Białas. Group and field definitions. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/realset1.html.

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

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

[10] Czesław Byliński. A classical first order language. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/cqc_lang.html.

[11] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/funct_4.html.

[12] Agata Darmochwał. Finite sets. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/finset_1.html.

[13] Artur Korniłowicz. The basic properties of **SCM** over ring. *Journal of Formalized Mathematics*, 10, 1998. http://mizar.org/JFM/Vol10/scmring2.html.

[14] Artur Korniłowicz. The construction of **SCM** over ring. *Journal of Formalized Mathematics*, 10, 1998. http://mizar.org/JFM/Vol10/scmring1.html.

[15] Artur Korniłowicz. On the composition of macro instructions of standard computers. *Journal of Formalized Mathematics*, 12, 2000. http://mizar.org/JFM/Vol12/amistd_2.html.

[16] Eugeniusz Kusak, Wojciech Leończuk, and Michał Muzalewski. Abelian groups, fields and vector spaces. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/vectsp_1.html.

[17] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Journal of Formalized Mathematics*, 4, 1992. http://mizar.org/JFM/Vol4/ami_1.html.

[18] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Journal of Formalized Mathematics*, 4, 1992. http://mizar.org/JFM/Vol4/ami_2.html.

[19] Beata Padlewska. Families of sets. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/setfam_1.html.

[20] Dariusz Surowik. Cyclic groups and some of their properties — part I. *Journal of Formalized Mathematics*, 3, 1991. http://mizar.org/JFM/Vol3/gr_cy_1.html.

[21] Yasushi Tanaka. On the decomposition of the states of SCM. *Journal of Formalized Mathematics*, 5, 1993. http://mizar.org/JFM/Vol5/ami_5.html.

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

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

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

[25] 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.

[26] Andrzej Trybulec, Piotr Rudnicki, and Artur Korniłowicz. Standard ordering of instruction locations. *Journal of Formalized Mathematics*, 12, 2000. http://mizar.org/JFM/Vol12/amistd_1.html.

[27] Wojciech A. Trybulec. Vectors in real linear space. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/rlvect_1.html.

[28] Wojciech A. Trybulec. Groups. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/group_1.html.

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

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

Received April 14, 2000

Published January 2, 2004