JOURNAL OF FORMALIZED MATHEMATICS
Volume 10, Released 1998, Published 2003
Inst. of Computer Science, Univ. of Białystok

The Basic Properties of SCM over Ring

Artur Korniłowicz
University of Białystok

MML Identifier: SCMRING2.
WWW: http://mizar.org/JFM/Vol10/scmring2.html

The articles [13], [12], [17], [18], [3], [4], [11], [16], [8], [14], [15], [1], [2], [5], [6], [9], [10], and [7] provide the notation and terminology for this paper.

1. **SCM** over Ring

In this paper I is an element of $\mathbb{Z}_8$, S is a non empty 1-sorted structure, t is an element of S, and x is a set.

Let R be a good ring. The functor $\mathbf{SCM}(R)$ yields a strict AMI over $\{$the carrier of $R\}$ and is defined by the conditions (Def. 1).

(Def. 1) The carrier of $\mathbf{SCM}(R) = \mathbb{N}$ and the instruction counter of $\mathbf{SCM}(R) = 0$ and the instruction locations of $\mathbf{SCM}(R) = \text{Instr-Loc}_{\text{SCM}}$ and the instruction codes of $\mathbf{SCM}(R) = \mathbb{Z}_8$ and the instructions of $\mathbf{SCM}(R) = \text{Instr}_{\text{SCM}}(R)$ and the object kind of $\mathbf{SCM}(R) = \text{OK}_{\text{SCM}}(R)$ and the execution of $\mathbf{SCM}(R) = \text{Exec}_{\text{SCM}}(R)$.

Let R be a good ring. One can check that $\mathbf{SCM}(R)$ is non empty and non void.

Let R be a good ring, let s be a state of $\mathbf{SCM}(R)$, and let a be an element of $\text{Data-Loc}_{\text{SCM}}$. Then $s(a)$ is an element of R.

Let R be a good ring. An object of $\mathbf{SCM}(R)$ is called a Data-Location of R if:

(Def. 2) It $\in$ (the carrier of $\mathbf{SCM}(R)) \setminus (\text{Instr-Loc}_{\text{SCM}} \cup \{0\})$.

For simplicity, we follow the rules: R denotes a good ring, r denotes an element of R, a, b, c, d_1, d_2 denote Data-Locations of R, and i_1 denotes an instruction-location of $\mathbf{SCM}(R)$.

We now state the proposition

(1) x is a Data-Location of R iff $x \in \text{Data-Loc}_{\text{SCM}}$.

Let R be a good ring, let s be a state of $\mathbf{SCM}(R)$, and let a be a Data-Location of R. Then $s(a)$ is an element of R.

The following propositions are true:

(2) $\langle 0, \emptyset \rangle \in \text{Instr}_{\text{SCM}}(S)$.

(3) $\langle 0, \emptyset \rangle$ is an instruction of $\mathbf{SCM}(R)$.

(4) If $x \in \{1, 2, 3, 4\}$, then $\langle x, \langle d_1, d_2 \rangle \rangle \in \text{Instr}_{\text{SCM}}(S)$.

(5) $\langle 5, \langle d_1, t \rangle \rangle \in \text{Instr}_{\text{SCM}}(S)$.

© Association of Mizar Users

(6) $\langle 6, \langle i_1 \rangle \rangle \in \mathrm{Instr}_{\mathrm{SCM}}(S)$.

(7) $\langle 7, \langle i_1, d_1 \rangle \rangle \in \mathrm{Instr}_{\mathrm{SCM}}(S)$.

Let R be a good ring and let a, b be Data-Locations of R. The functor $a{:=}b$ yielding an instruction of **SCM**(R) is defined by:

(Def. 3) $a{:=}b = \langle 1, \langle a, b \rangle \rangle$.

The functor $\mathrm{AddTo}(a,b)$ yields an instruction of **SCM**(R) and is defined as follows:

(Def. 4) $\mathrm{AddTo}(a,b) = \langle 2, \langle a, b \rangle \rangle$.

The functor $\mathrm{SubFrom}(a,b)$ yields an instruction of **SCM**(R) and is defined as follows:

(Def. 5) $\mathrm{SubFrom}(a,b) = \langle 3, \langle a, b \rangle \rangle$.

The functor $\mathrm{MultBy}(a,b)$ yielding an instruction of **SCM**(R) is defined by:

(Def. 6) $\mathrm{MultBy}(a,b) = \langle 4, \langle a, b \rangle \rangle$.

Let R be a good ring, let a be a Data-Location of R, and let r be an element of R. The functor $a{:=}r$ yielding an instruction of **SCM**(R) is defined by:

(Def. 7) $a{:=}r = \langle 5, \langle a, r \rangle \rangle$.

Let R be a good ring and let l be an instruction-location of **SCM**(R). The functor goto l yields an instruction of **SCM**(R) and is defined as follows:

(Def. 8) goto $l = \langle 6, \langle l \rangle \rangle$.

Let R be a good ring, let l be an instruction-location of **SCM**(R), and let a be a Data-Location of R. The functor **if** $a = 0$ **goto** l yields an instruction of **SCM**(R) and is defined as follows:

(Def. 9) **if** $a = 0$ **goto** $l = \langle 7, \langle l, a \rangle \rangle$.

One can prove the following proposition

(8) Let I be a set. Then I is an instruction of **SCM**(R) if and only if one of the following conditions is satisfied:

$I = \langle 0, \emptyset \rangle$ or there exist a, b such that $I = a{:=}b$ or there exist a, b such that $I = \mathrm{AddTo}(a,b)$ or there exist a, b such that $I = \mathrm{SubFrom}(a,b)$ or there exist a, b such that $I = \mathrm{MultBy}(a,b)$ or there exists i_1 such that $I =$ goto i_1 or there exist a, i_1 such that $I =$ **if** $a = 0$ **goto** i_1 or there exist a, r such that $I = a{:=}r$.

In the sequel s denotes a state of **SCM**(R).

Let us consider R. One can check that **SCM**(R) is IC-Ins-separated.

The following two propositions are true:

(9) $\mathbf{IC}_{\mathbf{SCM}(R)} = 0$.

(10) For every **SCM**-state S over R such that $S = s$ holds $\mathbf{IC}_s = \mathbf{IC}_S$.

Let R be a good ring and let i_1 be an instruction-location of **SCM**(R). The functor $\mathrm{Next}(i_1)$ yielding an instruction-location of **SCM**(R) is defined as follows:

(Def. 10) There exists an element m_1 of $\mathrm{Instr\text{-}Loc}_{\mathrm{SCM}}$ such that $m_1 = i_1$ and $\mathrm{Next}(i_1) = \mathrm{Next}(m_1)$.

We now state two propositions:

(11) For every instruction-location i_1 of **SCM**(R) and for every element m_1 of $\mathrm{Instr\text{-}Loc}_{\mathrm{SCM}}$ such that $m_1 = i_1$ holds $\mathrm{Next}(m_1) = \mathrm{Next}(i_1)$.

(12) Let I be an instruction of **SCM**(R) and i be an element of $\mathrm{Instr}_{\mathrm{SCM}}(R)$. If $i = I$, then for every **SCM**-state S over R such that $S = s$ holds $\mathrm{Exec}(I, s) = \mathrm{Exec\text{-}Res}_{\mathrm{SCM}}(i, S)$.

2. USERS GUIDE

We now state several propositions:

(13) $(\mathrm{Exec}(a{:=}b,s))(\mathbf{IC}_{\mathbf{SCM}(R)}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(a{:=}b,s))(a) = s(b)$ and for every c such that $c \neq a$ holds $(\mathrm{Exec}(a{:=}b,s))(c) = s(c)$.

(14) $(\mathrm{Exec}(\mathrm{AddTo}(a,b),s))(\mathbf{IC}_{\mathbf{SCM}(R)}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(\mathrm{AddTo}(a,b),s))(a) = s(a) + s(b)$ and for every c such that $c \neq a$ holds $(\mathrm{Exec}(\mathrm{AddTo}(a,b),s))(c) = s(c)$.

(15) $(\mathrm{Exec}(\mathrm{SubFrom}(a,b),s))(\mathbf{IC}_{\mathbf{SCM}(R)}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(\mathrm{SubFrom}(a,b),s))(a) = s(a) - s(b)$ and for every c such that $c \neq a$ holds $(\mathrm{Exec}(\mathrm{SubFrom}(a,b),s))(c) = s(c)$.

(16) $(\mathrm{Exec}(\mathrm{MultBy}(a,b),s))(\mathbf{IC}_{\mathbf{SCM}(R)}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(\mathrm{MultBy}(a,b),s))(a) = s(a) \cdot s(b)$ and for every c such that $c \neq a$ holds $(\mathrm{Exec}(\mathrm{MultBy}(a,b),s))(c) = s(c)$.

(17) $(\mathrm{Exec}(\mathrm{goto}\ i_1,s))(\mathbf{IC}_{\mathbf{SCM}(R)}) = i_1$ and $(\mathrm{Exec}(\mathrm{goto}\ i_1,s))(c) = s(c)$.

(18) If $s(a) = 0_R$, then $(\mathrm{Exec}(\mathbf{if}\ a = 0\ \mathbf{goto}\ i_1,s))(\mathbf{IC}_{\mathbf{SCM}(R)}) = i_1$ and if $s(a) \neq 0_R$, then $(\mathrm{Exec}(\mathbf{if}\ a = 0\ \mathbf{goto}\ i_1,s))(\mathbf{IC}_{\mathbf{SCM}(R)}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(\mathbf{if}\ a = 0\ \mathbf{goto}\ i_1,s))(c) = s(c)$.

(19) $(\mathrm{Exec}(a{:=}r,s))(\mathbf{IC}_{\mathbf{SCM}(R)}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(a{:=}r,s))(a) = r$ and for every c such that $c \neq a$ holds $(\mathrm{Exec}(a{:=}r,s))(c) = s(c)$.

3. HALT INSTRUCTION

The following two propositions are true:

(20) For every instruction I of $\mathbf{SCM}(R)$ such that there exists s such that $(\mathrm{Exec}(I,s))(\mathbf{IC}_{\mathbf{SCM}(R)}) = \mathrm{Next}(\mathbf{IC}_s)$ holds I is non halting.

(21) For every instruction I of $\mathbf{SCM}(R)$ such that $I = \langle 0, \emptyset \rangle$ holds I is halting.

Let us consider R, a, b. One can check the following observations:

* $a{:=}b$ is non halting,
* $\mathrm{AddTo}(a,b)$ is non halting,
* $\mathrm{SubFrom}(a,b)$ is non halting, and
* $\mathrm{MultBy}(a,b)$ is non halting.

Let us consider R, i_1. One can check that goto i_1 is non halting.
Let us consider R, a, i_1. Observe that **if** $a = 0$ **goto** i_1 is non halting.
Let us consider R, a, r. Note that $a{:=}r$ is non halting.
Let us consider R. Observe that $\mathbf{SCM}(R)$ is halting, definite, data-oriented, steady-programmed, and realistic.
The following two propositions are true:

(29)[1] For every instruction I of $\mathbf{SCM}(R)$ such that I is halting holds $I = \mathbf{halt}_{\mathbf{SCM}(R)}$.

(30) $\mathbf{halt}_{\mathbf{SCM}(R)} = \langle 0, \emptyset \rangle$.

[1] The propositions (22)–(28) have been removed.

REFERENCES

[1] Grzegorz Bancerek. König's theorem. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/card_3.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] Czesław Byliński. A classical first order language. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/cqc_lang.html.

[6] 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.

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

[8] 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.

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

[10] 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.

[11] 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.

[12] Andrzej Trybulec. Enumerated sets. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/enumset1.html.

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

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

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

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

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

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

Received November 29, 1998

Published January 2, 2004