JOURNAL OF FORMALIZED MATHEMATICS
Volume 8, Released 1996, Published 2003
Inst. of Computer Science, Univ. of Białystok

Computation in $\mathbf{SCM}_{\mathrm{FSA}}$

Andrzej Trybulec
Warsaw University
Białystok

Yatsuka Nakamura
Shinshu University
Nagano

Summary. The properties of computations in $\mathbf{SCM}_{\mathrm{FSA}}$ are investigated.

MML Identifier: `SCMFSA_3`.

WWW: `http://mizar.org/JFM/Vol8/scmfsa_3.html`

The articles [14], [19], [1], [12], [20], [4], [7], [17], [5], [9], [2], [6], [18], [8], [10], [11], [15], [13], [3], and [16] provide the notation and terminology for this paper.

1. Preliminaries

The following propositions are true:

(1) $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \notin$ Int-Locations.

(2) $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \notin$ FinSeq-Locations.

(3) Let i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be an instruction of $\mathbf{SCM}$. Suppose $i = I$. Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and S be a state of $\mathbf{SCM}$. Suppose $S = s\restriction$the carrier of $\mathbf{SCM}{+\cdot}$((the instruction locations of $\mathbf{SCM}) \longmapsto I)$. Then $\mathrm{Exec}(i,s) = s{+\cdot}\mathrm{Exec}(I,S){+\cdot}s\restriction$the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(4) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $s_1\restriction(\text{Int-Locations} \cup \text{FinSeq-Locations} \cup \{\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\}) = s_2\restriction(\text{Int-Locations} \cup \text{FinSeq-Locations} \cup \{\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\})$. Let l be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $\mathrm{Exec}(l,s_1)\restriction(\text{Int-Locations} \cup \text{FinSeq-Locations} \cup \{\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\}) = \mathrm{Exec}(l,s_2)\restriction(\text{Int-Locations} \cup \text{FinSeq-Locations} \cup \{\mathbf{IC}$

(5) Let N be a set with non empty elements, S be a steady-programmed non empty non void AMI over N, i be an instruction of S, and s be a state of S. Then $\mathrm{Exec}(i,s)\restriction$the instruction locations of $S = s\restriction$the instruction locations of S.

2. Finite partial states of $\mathbf{SCM}_{\mathrm{FSA}}$

The following propositions are true:

(6) For every finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{DataPart}(p) = p\restriction(\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(7) For every finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ holds p is data-only iff $\mathrm{dom}\,p \subseteq \text{Int-Locations} \cup \text{FinSeq-Locations}$.

(8) For every finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{dom}\,\mathrm{DataPart}(p) \subseteq \text{Int-Locations} \cup \text{FinSeq-Locations}$.

(9) For every finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{dom}\,\mathrm{ProgramPart}(p) \subseteq$ the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(10) Let i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and p be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $\mathrm{Exec}(i, s+\cdot p) = \mathrm{Exec}(i,s)+\cdot p$.

(11) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, i_1 be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$, and a be an integer location. Then $s(a) = (s+\cdot \text{Start-At}(i_1))(a)$.

(12) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, i_1 be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$, and a be a finite sequence location. Then $s(a) = (s+\cdot \text{Start-At}(i_1))(a)$.

(13) For all states s, t of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $s+\cdot t\restriction(\text{Int-Locations}\cup\text{FinSeq-Locations})$ is a state of $\mathbf{SCM}_{\mathrm{FSA}}$.

3. Autonomic finite partial states of $\mathbf{SCM}_{\mathrm{FSA}}$

Let l_1 be an integer location and let a be an integer. Then $l_1 \dot{\longmapsto} a$ is a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$.

The following proposition is true

(14) For every autonomic finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $\mathrm{DataPart}(p) \neq \emptyset$ holds $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\, p$.

Let us note that there exists a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ which is autonomic and non programmed.

Next we state a number of propositions:

(15) For every autonomic non programmed finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\, p$.

(16) For every autonomic finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\, p$ holds $\mathbf{IC}_p \in \mathrm{dom}\, p$.

(17) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. If $p \subseteq s$, then for every natural number i holds $\mathbf{IC}_{(\mathrm{Computation}(s))(i)} \in \mathrm{dom}\,\mathrm{ProgramPart}(p)$.

(18) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number. Then $\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)} = \mathbf{IC}_{(\mathrm{Computation}(s_2))(i)}$ and $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{CurInstr}((\mathrm{Computation}(s_2))(i))$.

(19) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number and d_1, d_2 be integer locations. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = d_1{:=}d_2$ and $d_1 \in \mathrm{dom}\, p$, then $(\mathrm{Computation}(s_1))(i)(d_2) = (\mathrm{Computation}(s_2))(i)(d_2)$.

(20) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number and d_1, d_2 be integer locations. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{AddTo}(d_1, d_2)$ and $d_1 \in \mathrm{dom}\, p$, then $(\mathrm{Computation}(s_1))(i)(d_1) + (\mathrm{Computation}(s_1))(i)(d_2) = (\mathrm{Computation}(s_2))(i)(d_1) + (\mathrm{Computation}(s_2))(i)(d_2)$.

(21) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number and d_1, d_2 be integer locations. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{SubFrom}(d_1, d_2)$ and $d_1 \in \mathrm{dom}\, p$, then $(\mathrm{Computation}(s_1))(i)(d_1) - (\mathrm{Computation}(s_1))(i)(d_2) = (\mathrm{Computation}(s_2))(i)(d_1) - (\mathrm{Computation}(s_2))(i)(d_2)$.

(22) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number and d_1, d_2 be integer locations. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{MultBy}(d_1, d_2)$ and $d_1 \in \mathrm{dom}\, p$, then $(\mathrm{Computation}(s_1))(i)(d_1) \cdot (\mathrm{Computation}(s_1))(i)(d_2) = (\mathrm{Computation}(s_2))(i)(d_1) \cdot (\mathrm{Computation}(s_2))(i)(d_2)$.

(23) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number and d_1, d_2 be integer locations. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{Divide}(d_1, d_2)$ and $d_1 \in \mathrm{dom}\, p$ and $d_1 \neq d_2$, then $(\mathrm{Computation}(s_1))(i)(d_1) \div (\mathrm{Computation}(s_1))(i)(d_2) = (\mathrm{Computation}(s_2))(i)(d_1) \div (\mathrm{Computation}(s_2))(i)(d_2)$.

(24) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number and d_1, d_2 be integer locations. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{Divide}(d_1, d_2)$ and $d_2 \in \mathrm{dom}\, p$ and $d_1 \neq d_2$, then $(\mathrm{Computation}(s_1))(i)(d_1) \bmod (\mathrm{Computation}(s_1))(i)(d_2) = (\mathrm{Computation}(s_2))(i)(d_1) \bmod (\mathrm{Computation}(s_2))(i)(d_2)$.

(25) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, d_1 be an integer location, and l_2 be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathbf{if}\ d_1 = 0\ \mathbf{goto}\ l_2$ and $l_2 \neq \mathrm{Next}(\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)})$, then $(\mathrm{Computation}(s_1))(i)(d_1) = 0$ iff $(\mathrm{Computation}(s_2))(i)(d_1) = 0$.

(26) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, d_1 be an integer location, and l_2 be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathbf{if}\ d_1 > 0\ \mathbf{goto}\ l_2$ and $l_2 \neq \mathrm{Next}(\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)})$, then $(\mathrm{Computation}(s_1))(i)(d_1) > 0$ iff $(\mathrm{Computation}(s_2))(i)(d_1) > 0$.

(27) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, d_1, d_2 be integer locations, and f be a finite sequence location. Suppose $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = d_1{:=}f_{d_2}$ and $d_1 \in \mathrm{dom}\, p$. Let k_1, k_2 be natural numbers. If $k_1 = |(\mathrm{Computation}(s_1))(i)(d_2)|$ and $k_2 = |(\mathrm{Computation}(s_2))(i)(d_2)|$, then $(\mathrm{Computation}(s_1))(i)(f)_{k_1} = (\mathrm{Computation}(s_2))(i)(f)_{k_2}$.

(28) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, d_1, d_2 be integer locations, and f be a finite sequence location. Suppose $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = f_{d_2}{:=}d_1$ and $f \in \mathrm{dom}\, p$. Let k_1, k_2 be natural numbers. If $k_1 = |(\mathrm{Computation}(s_1))(i)(d_2)|$ and $k_2 = |(\mathrm{Computation}(s_2))(i)(d_2)|$, then $(\mathrm{Computation}(s_1))(i)(f) +\cdot (k_1, (\mathrm{Computation}(s_1))(i)(d_1)) = (\mathrm{Computation}(s_2))(i)(f) +\cdot (k_2, (\mathrm{Computation}(s_2))(i)(d_1))$.

(29) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, d_1 be an integer location, and f be a finite sequence location. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = d_1{:=}\mathrm{len} f$ and $d_1 \in \mathrm{dom}\, p$, then $\mathrm{len}(\mathrm{Computation}(s_1))(i)(f) = \mathrm{len}(\mathrm{Computation}(s_2))(i)(f)$.

(30) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, d_1 be an integer location, and f be a finite sequence location. Suppose $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = f{:=}\langle \underbrace{0, \ldots, 0}_{d_1} \rangle$ and $f \in \mathrm{dom}\, p$. Let k_1, k_2 be natural numbers. If $k_1 = |(\mathrm{Computation}(s_1))(i)(d_1)|$ and $k_2 = |(\mathrm{Computation}(s_2))(i)(d_1)|$, then $k_1 \mapsto 0 = k_2 \mapsto 0$.

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

[4] Czesław Byliński. Functions and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/funct_1.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. Finite sequences and tuples of elements of a non-empty sets. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/finseq_2.html`.

[7] 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`.

[8] Czesław Byliński. Products and coproducts in categories. *Journal of Formalized Mathematics*, 4, 1992. `http://mizar.org/JFM/Vol4/cat_3.html`.

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

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

[11] 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`.

[12] Jan Popiołek. Some properties of functions modul and signum. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/absvalue.html`.

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

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

[15] 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`.

[16] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\mathrm{FSA}}$ computer. *Journal of Formalized Mathematics*, 8, 1996. `http://mizar.org/JFM/Vol8/scmfsa_2.html`.

[17] Michał J. Trybulec. Integers. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/int_1.html`.

[18] Wojciech A. Trybulec. Pigeon hole principle. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/finseq_4.html`.

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

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

Received February 7, 1996

Published January 2, 2004