FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

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

Andrzej Trybulec
Warsaw University
Białystok

Yatsuka Nakamura
Shinshu University
Nagano

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

MML Identifier: SCMFSA_3.

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

1. Preliminaries

One can prove the following propositions:

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

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

(3) Let i be an instruction of $\mathbf{SCM}_{\rm FSA}$ and let I be an instruction of $\mathbf{SCM}$. Suppose $i = I$. Let s be a state of $\mathbf{SCM}_{\rm FSA}$ and let S be a state of $\mathbf{SCM}$. Suppose $S = s \restriction$ (the objects 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 $\mathbf{SCM}_{\rm FSA}$).

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

(5) Let N be a non empty set with non empty elements, and let S be a steady-programmed AMI over N, and let i be an instruction of S, and let s

ISSN 1426–2630

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 $\mathbf{SCM}_{\rm FSA}$

One can prove the following two propositions:

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

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

Let us observe that there exists a finite partial state of $\mathbf{SCM}_{\rm FSA}$ which is data-only.

We now state two propositions:

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

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

Let I_1 be a partial function from $\mathrm{FinPartSt}(\mathbf{SCM}_{\rm FSA})$ to $\mathrm{FinPartSt}(\mathbf{SCM}_{\rm FSA})$. We say that I_1 is data-only if and only if the condition (Def. 1) is satisfied.

(Def. 1) Let p be a finite partial state of $\mathbf{SCM}_{\rm FSA}$. Suppose $p \in \mathrm{dom}\, I_1$. Then p is data-only and for every finite partial state q of $\mathbf{SCM}_{\rm FSA}$ such that $q = I_1(p)$ holds q is data-only.

One can verify that there exists a partial function from $\mathrm{FinPartSt}(\mathbf{SCM}_{\rm FSA})$ to $\mathrm{FinPartSt}(\mathbf{SCM}_{\rm FSA})$ which is data-only.

One can prove the following four propositions:

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

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

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

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

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

We now 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 let 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 let 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 let 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 let 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 let 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 let 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 let 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 let 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 let 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 let d_1, d_2 be integer locations. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{MultBy}(d_1, d_2)$ and

$d_1 \in \operatorname{dom} p$, then $(\text{Computation}(s_1))(i)(d_1) \cdot (\text{Computation}(s_1))(i)(d_2) = (\text{Computation}(s_2))(i)(d_1) \cdot (\text{Computation}(s_2))(i)(d_2)$.

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

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

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

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

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

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

(29) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let 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 let d_1 be an integer location, and let 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 let 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 let d_1 be an integer location, and let 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. 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] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.

[5] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.

[6] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.

[7] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.

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

[9] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.

[10] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.

[11] Czesław Byliński. Products and coproducts in categories. *Formalized Mathematics*, 2(**5**):701–709, 1991.

[12] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.

[13] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.

[14] Rafał Kwiatek and Grzegorz Zwara. The divisibility of integers and integer relative primes. *Formalized Mathematics*, 1(**5**):829–832, 1990.

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

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

[17] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.

[18] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.

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

[20] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.

[21] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.
[22] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[23] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[24] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[25] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. An extension of **SCM**. *Formalized Mathematics*, 5(**4**):507–512, 1996.
[26] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\rm FSA}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[27] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[28] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[29] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[30] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received February 7, 1996