FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

# On the Composition of Macro Instructions. Part II [1]

Noriko Asamoto
Ochanomizu University
Tokyo

Yatsuka Nakamura
Shinshu University
Nagano

Piotr Rudnicki
University of Alberta
Edmonton

Andrzej Trybulec
Warsaw University
Białystok

**Summary.** We define the semantics of macro instructions (introduced in [26]) in terms of executions of $\mathbf{SCM}_{\rm FSA}$. In a similar way, we define the semantics of macro composition. Several attributes of macro instructions are introduced (paraclosed, parahalting, keeping 0) and their usage enables a systematic treatment of the composition of macro intructions. This article is continued in [1].

MML Identifier: `SCMFSA6B`.

The notation and terminology used in this paper are introduced in the following articles: [20], [30], [14], [3], [28], [31], [9], [10], [4], [21], [8], [29], [12], [2], [19], [7], [13], [11], [15], [16], [25], [5], [18], [6], [27], [22], [23], [24], [26], and [17].

## 1. Preliminaries

The following propositions are true:

(1) For all functions $f$, $g$ and for all sets $x$, $y$ such that $x \notin \operatorname{dom} f$ and $f \subseteq g$ holds $f \subseteq g +\cdot (x, y)$.

(2) For every function $f$ and for all sets $x$, $y$, $A$ such that $x \notin A$ holds $f \restriction A = (f +\cdot (x, y)) \restriction A$.

(3) For all functions $f$, $g$ and for every set $A$ such that $A \cap \operatorname{dom} f \subseteq A \cap \operatorname{dom} g$ holds $(f +\cdot g \restriction A) \restriction A = g \restriction A$.

[1]This work was partially supported by NSERC Grant OGP9207 and NATO CRG 951368.



© 1997 Warsaw University - Białystok
ISSN 1426–2630

## 2. Properties of Start-At

For simplicity we follow a convention: $m$, $n$ will denote natural numbers, $x$ will denote a set, $i$ will denote an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, $I$, $J$ will denote macro instructions, $a$ will denote an integer location, $f$ will denote a finite sequence location, $l$, $l_1$ will denote instructions-locations of $\mathbf{SCM}_{\mathrm{FSA}}$, and $s$, $s_1$, $s_2$ will denote states of $\mathbf{SCM}_{\mathrm{FSA}}$.

We now state a number of propositions:

(4) $\text{Start-At}(\text{insloc}(0)) \subseteq \text{Initialized}(I)$.

(5) If $I+\cdot\, \text{Start-At}(\text{insloc}(n)) \subseteq s$, then $I \subseteq s$.

(6) $(I+\cdot\, \text{Start-At}(\text{insloc}(n))) \restriction (\text{the instruction locations of } \mathbf{SCM}_{\mathrm{FSA}}) = I$.

(7) If $x \in \operatorname{dom} I$, then $I(x) = (I+\cdot\, \text{Start-At}(\text{insloc}(n)))(x)$.

(8) If $\text{Initialized}(I) \subseteq s$, then $I+\cdot\, \text{Start-At}(\text{insloc}(0)) \subseteq s$.

(9) $a \notin \operatorname{dom} \text{Start-At}(l)$.

(10) $f \notin \operatorname{dom} \text{Start-At}(l)$.

(11) $l_1 \notin \operatorname{dom} \text{Start-At}(l)$.

(12) $a \notin \operatorname{dom}(I+\cdot\, \text{Start-At}(l))$.

(13) $f \notin \operatorname{dom}(I+\cdot\, \text{Start-At}(l))$.

(14) $s+\cdot I+\cdot\, \text{Start-At}(\text{insloc}(0)) = s+\cdot\, \text{Start-At}(\text{insloc}(0))+\cdot I$.

## 3. Properties of AMI structures

In the sequel $N$ will denote a non empty set with non empty elements.

Next we state two propositions:

(15) If $s = \text{Following}(s)$, then for every $n$ holds $(\text{Computation}(s))(n) = s$.

(16) Let $S$ be a halting von Neumann definite AMI over $N$ and let $s$ be a state of $S$. If $s$ is halting, then $\text{Result}(s) = (\text{Computation}(s))(\text{LifeSpan}(s))$.

Let us consider $N$, let $S$ be a von Neumann definite AMI over $N$, let $s$ be a state of $S$, let $l$ be an instruction-location of $S$, and let $i$ be an instruction of $S$. Then $s +\cdot (l, i)$ is a state of $S$.

Let $s$ be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, let $l_2$ be an integer location, and let $k$ be an integer. Then $s +\cdot (l_2, k)$ is a state of $\mathbf{SCM}_{\mathrm{FSA}}$.

We now state the proposition

(17) Let $S$ be a steady-programmed von Neumann definite AMI over $N$, and let $s$ be a state of $S$, and given $n$. Then $s \restriction (\text{the instruction locations of } S) = (\text{Computation}(s))(n) \restriction (\text{the instruction locations of } S)$.

## 4. Execution of macro instructions

Let $I$ be a macro instruction and let $s$ be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. The functor IExec$(I, s)$ yielding a state of $\mathbf{SCM}_{\mathrm{FSA}}$ is defined as follows:

(Def. 1) $\mathrm{IExec}(I, s) = \mathrm{Result}(s +\cdot \mathrm{Initialized}(I)) +\cdot s \restriction$ (the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$).

Let $I$ be a macro instruction. We say that $I$ is paraclosed if and only if:

(Def. 2) For every state $s$ of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every natural number $n$ such that $I +\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)) \subseteq s$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(n)} \in \mathrm{dom}\, I$.

We say that $I$ is parahalting if and only if:

(Def. 3) $I +\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0))$ is halting.

We say that $I$ is keeping 0 if and only if:

(Def. 4) For every state $s$ of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $I +\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)) \subseteq s$ and for every natural number $k$ holds $(\mathrm{Computation}(s))(k)(\mathrm{intloc}(0)) = s(\mathrm{intloc}(0))$.

Let us note that there exists a macro instruction which is parahalting.

Next we state two propositions:

(18) For every parahalting macro instruction $I$ such that $I +\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)) \subseteq s$ holds $s$ is halting.

(19) For every parahalting macro instruction $I$ such that $\mathrm{Initialized}(I) \subseteq s$ holds $s$ is halting.

Let $I$ be a parahalting macro instruction. One can verify that Initialized$(I)$ is halting.

We now state two propositions:

(20) $s_2 +\cdot (\mathbf{IC}_{(s_2)}, \mathrm{goto}\ (\mathbf{IC}_{(s_2)}))$ is not halting.

(21) Suppose that

(i) $s_1$ and $s_2$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$,

(ii) $I \subseteq s_1$,

(iii) $I \subseteq s_2$, and

(iv) for every $m$ such that $m < n$ holds $\mathbf{IC}_{(\mathrm{Computation}(s_2))(m)} \in \mathrm{dom}\, I$.

Given $m$. Suppose $m \leq n$. Then $(\mathrm{Computation}(s_1))(m)$ and $(\mathrm{Computation}(s_2))(m)$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

One can check that every macro instruction which is parahalting is also paraclosed and every macro instruction which is keeping 0 is also paraclosed.

The following propositions are true:

(22) Let $I$ be a parahalting macro instruction and let $a$ be a read-write integer location. If $a \notin \mathrm{UsedIntLoc}(I)$, then $(\mathrm{IExec}(I, s))(a) = s(a)$.

(23) For every parahalting macro instruction $I$ such that $f \notin \mathrm{UsedInt}^{*}\,\mathrm{Loc}(I)$ holds $(\mathrm{IExec}(I, s))(f) = s(f)$.

(24) If $\mathbf{IC}_s = l$ and $s(l) = \mathrm{goto}\ l$, then $s$ is not halting.

One can verify that every macro instruction which is parahalting is also non empty.

One can prove the following propositions:

(25) For every parahalting macro instruction $I$ holds $\operatorname{dom} I \neq \emptyset$.

(26) For every parahalting macro instruction $I$ holds $\operatorname{insloc}(0) \in \operatorname{dom} I$.

(27) Let $J$ be a parahalting macro instruction. Suppose $J +\cdot$ Start-At(insloc(0)) $\subseteq s_1$. Let $n$ be a natural number. Suppose ProgramPart(Relocated$(J, n)$) $\subseteq s_2$ and $\mathbf{IC}_{(s_2)} = \operatorname{insloc}(n)$ and $s_1 \restriction$ (Int-Locations $\cup$ FinSeq-Locations) $= s_2 \restriction$ (Int-Locations $\cup$ FinSeq-Locations). Let $i$ be a natural number. Then $\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)} + n = \mathbf{IC}_{(\mathrm{Computation}(s_2))(i)}$ and IncAddr(CurInstr((Computation$(s_1)$)$(i)$), $n$) $=$ CurInstr((Computation$(s_2)$)$(i)$) and (Computation$(s_1)$)$(i) \restriction$ (Int-Locations $\cup$ FinSeq-Locations) $=$ (Computation$(s_2)$)$(i) \restriction$ (Int-Locations $\cup$ FinSeq-Locations).

(28) Let $I$ be a parahalting macro instruction. Suppose $I +\cdot$ Start-At(insloc(0)) $\subseteq s_1$ and $I +\cdot$ Start-At(insloc(0)) $\subseteq s_2$ and $s_1$ and $s_2$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$. Let $k$ be a natural number. Then (Computation$(s_1)$)$(k)$ and (Computation$(s_2)$)$(k)$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$ and CurInstr((Computation$(s_1)$)$(k)$) $=$ CurInstr((Computation$(s_2)$)$(k)$).

(29) Let $I$ be a parahalting macro instruction. Suppose $I +\cdot$ Start-At(insloc(0)) $\subseteq s_1$ and $I +\cdot$ Start-At(insloc(0)) $\subseteq s_2$ and $s_1$ and $s_2$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$. Then LifeSpan$(s_1)$ $=$ LifeSpan$(s_2)$ and Result$(s_1)$ and Result$(s_2)$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(30) For every parahalting macro instruction $I$ holds $\mathbf{IC}_{\mathrm{IExec}(I,s)} = \mathbf{IC}_{\mathrm{Result}(s +\cdot \mathrm{Initialized}(I))}$.

(31) For every non empty macro instruction $I$ holds $\operatorname{insloc}(0) \in \operatorname{dom} I$ and $\operatorname{insloc}(0) \in \operatorname{dom} \mathrm{Initialized}(I)$ and $\operatorname{insloc}(0) \in \operatorname{dom}(I +\cdot$ Start-At(insloc(0))).

(32) $x \in \operatorname{dom} \mathrm{Macro}(i)$ iff $x = \operatorname{insloc}(0)$ or $x = \operatorname{insloc}(1)$.

(33) (Macro$(i)$)(insloc(0)) $= i$ and (Macro$(i)$)(insloc(1)) $= \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$ and (Initialized(Macro$(i)$))(insloc(0)) $= i$ and (Initialized(Macro$(i)$))(insloc(1)) $= \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$ and (Macro$(i) +\cdot$ Start-At(insloc(0)))(insloc(0)) $= i$.

(34) If Initialized$(I) \subseteq s$, then $\mathbf{IC}_s = \operatorname{insloc}(0)$.

Let us observe that there exists a macro instruction which is keeping 0 and parahalting.

One can prove the following proposition

(35) For every keeping 0 parahalting macro instruction $I$ holds (IExec$(I, s)$)(intloc(0)) $= 1$.

## 5. The composition of macro instructions

We now state several propositions:

(36) Let $I$ be a paraclosed macro instruction and let $J$ be a macro instruction. Suppose $I+\cdot$ Start-At(insloc(0)) $\subseteq s$ and $s$ is halting. Given $m$. Suppose $m \leq$ LifeSpan($s$). Then (Computation($s$))($m$) and (Computation($s+\cdot(I;J)$))($m$) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(37) For every paraclosed macro instruction $I$ such that $s+\cdot I$ is halting and Directed($I$) $\subseteq s$ and Start-At(insloc(0)) $\subseteq s$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s+\cdot I)+1)} =$ insloc(card $I$).

(38) Let $I$ be a paraclosed macro instruction. If $s+\cdot I$ is halting and Directed($I$) $\subseteq s$ and Start-At(insloc(0)) $\subseteq s$, then (Computation($s$))(LifeSpan($s+\cdot I$))$\restriction$(Int-Locations $\cup$ FinSeq-Locations) = (Computation($s$))(LifeSpan($s+\cdot I$)+1)$\restriction$(Int-Locations $\cup$ FinSeq-Locations).

(39) Let $I$ be a parahalting macro instruction. Suppose Initialized($I$) $\subseteq s$. Let $k$ be a natural number. If $k \leq$ LifeSpan($s$), then CurInstr((Computation($s+\cdot$ Directed($I$)))($k$)) $\neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(40) Let $I$ be a paraclosed macro instruction. Suppose $s+\cdot(I+\cdot$ Start-At (insloc(0))) is halting. Let $J$ be a macro instruction and let $k$ be a natural number. Suppose $k \leq$ LifeSpan($s+\cdot(I+\cdot$ Start-At(insloc(0)))). Then (Computation($s+\cdot(I+\cdot$ Start-At(insloc(0)))))($k$) and (Computation($s+\cdot$ $((I;J)+\cdot$ Start-At(insloc(0)))))($k$) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

Let $I$, $J$ be parahalting macro instructions. Note that $I$;$J$ is parahalting.

Next we state two propositions:

(41) Let $I$ be a keeping 0 macro instruction. Suppose $s+\cdot(I+\cdot$ Start-At(insloc (0))) is not halting. Let $J$ be a macro instruction and let $k$ be a natural number. Then (Computation($s+\cdot(I+\cdot$ Start-At(insloc(0)))))($k$) and (Computation($s+\cdot((I;J)+\cdot$ Start-At(insloc(0)))))($k$) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(42) Let $I$ be a keeping 0 macro instruction. Suppose $s+\cdot I$ is halting. Let $J$ be a paraclosed macro instruction. Suppose $(I;J)+\cdot$ Start-At(insloc(0)) $\subseteq s$. Let $k$ be a natural number. Then (Computation(Result($s+\cdot I$)$+\cdot(J+\cdot$ Start-At(insloc(0)))))($k$)$+\cdot$ Start-At ($\mathbf{IC}_{(\mathrm{Computation}(\mathrm{Result}(s+\cdot I)+\cdot(J+\cdot \mathrm{Start\text{-}At(insloc(0)))))(k)}}$ + card $I$) and (Computation($s+\cdot(I;J)$))(LifeSpan($s+\cdot I$) + 1 + $k$) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

Let $I$, $J$ be keeping 0 macro instructions. Note that $I$;$J$ is keeping 0.

The following two propositions are true:

(43) Let $I$ be a keeping 0 parahalting macro instruction and let $J$ be a parahalting macro instruction. Then LifeSpan($s+\cdot$ Initialized($I$;$J$)) =

LifeSpan($s+\cdot$ Initialized$(I)$) $+ 1 +$ LifeSpan(Result($s+\cdot$ Initialized$(I)$)$+\cdot$ Initialized$(J)$)).

(44) Let $I$ be a keeping 0 parahalting macro instruction and let $J$ be a parahalting macro instruction. Then $\mathrm{IExec}(I;J,s) = \mathrm{IExec}(J,\mathrm{IExec}(I,s)) +\cdot \text{Start-At}(\mathbf{IC}_{\mathrm{IExec}(J,\mathrm{IExec}(I,s))} + \mathrm{card}\, I)$.

## References


[1] Noriko Asamoto, Yatsuka Nakamura, Piotr Rudnicki, and Andrzej Trybulec. On the composition of macro instructions. Part III. *Formalized Mathematics*, 6(**1**):53–57, 1997.

[2] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.

[3] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.

[4] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.

[5] Grzegorz Bancerek and Piotr Rudnicki. Development of terminology for **SCM**. *Formalized Mathematics*, 4(**1**):61–67, 1993.

[6] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.

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

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

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

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

[11] 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.

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

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

[14] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 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] Piotr Rudnicki and Andrzej Trybulec. Memory handling for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 6(**1**):29–36, 1997.

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

[19] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.

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

[21] Andrzej Trybulec and Agata Darmochwał. Boolean domains. *Formalized Mathematics*, 1(**1**):187–190, 1990.

[22] Andrzej Trybulec and Yatsuka Nakamura. Computation in $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):537–542, 1996.

[23] Andrzej Trybulec and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):571–576, 1996.

[24] Andrzej Trybulec and Yatsuka Nakamura. Relocability for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):583–586, 1996.

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

[26] Andrzej Trybulec, Yatsuka Nakamura, and Noriko Asamoto. On the compositions of macro instructions. Part I. *Formalized Mathematics*, 6(**1**):21–27, 1997.
[27] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\rm FSA}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[28] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[29] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[30] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[31] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

*Received July 22, 1996*