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

Conditional Branch Macro Instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Part I

Noriko Asamoto
Ochanomizu University
Tokyo

MML Identifier: `SCMFSA8A`.

WWW: `http://mizar.org/JFM/Vol8/scmfsa8a.html`

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

One can prove the following propositions:

(2)[1] For all functions f, g and for every set D such that $\mathrm{dom}\, g$ misses D holds $(f+\cdot g)\restriction D = f\restriction D$, where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

(3) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{dom}(s\restriction$the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}) =$ the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(4) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ such that s is halting and for every natural number k such that $\mathrm{LifeSpan}(s) \leq k$ holds $\mathrm{CurInstr}((\mathrm{Computation}(s))(k)) = \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(5) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ such that s is halting and for every natural number k such that $\mathrm{LifeSpan}(s) \leq k$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s))}$.

(6) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Then s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$ if and only if $\mathbf{IC}_{(s_1)} = \mathbf{IC}_{(s_2)}$ and $s_1\restriction(\text{Int-Locations} \cup \text{FinSeq-Locations}) = s_2\restriction(\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(7) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every macro instruction I holds $\mathbf{IC}_{\mathrm{IExec}(I,s)} = \mathbf{IC}_{\mathrm{Result}(s+\cdot\, \mathrm{Initialized}(I))}$.

(8) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every macro instruction I holds $\mathrm{Initialize}(s)+\cdot\, \mathrm{Initialized}(I) = s+\cdot\, \mathrm{Initialized}(I)$.

(9) For every macro instruction I and for every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $I \subseteq I+\cdot\, \text{Start-At}(l)$.

(10) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $s\restriction(\text{Int-Locations} \cup \text{FinSeq-Locations}) = (s+\cdot\, \text{Start-At}(l))\restriction(\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(11) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I be a macro instruction, and l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $s\restriction(\text{Int-Locations} \cup \text{FinSeq-Locations}) = (s+\cdot(I+\cdot\, \text{Start-At}(l)))\restriction(\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(12) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $\mathrm{dom}(s\restriction$the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}})$ misses $\mathrm{dom}\, \text{Start-At}(l)$.

[1] The proposition (1) has been removed.

(13) For every state s of $\mathbf{SCM}_{\rm FSA}$ and for every macro instruction I holds $s+\cdot\,{\rm Initialized}(I) = {\rm Initialize}(s)+\cdot(I+\cdot\,{\rm Start\text{-}At}({\rm insloc}(0)))$.

(14) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I_1, I_2 be macro instructions, and l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. Then $s+\cdot(I_1+\cdot\,{\rm Start\text{-}At}(l))$ and $s+\cdot(I_2+\cdot\,{\rm Start\text{-}At}(l))$ are equal outside the instruction locations of $\mathbf{SCM}_{\rm FSA}$.

(15) ${\rm dom}({\rm Stop}_{\mathbf{SCM}_{\rm FSA}}) = \{{\rm insloc}(0)\}$.

(16) ${\rm insloc}(0) \in {\rm dom}({\rm Stop}_{\mathbf{SCM}_{\rm FSA}})$ and ${\rm Stop}_{\mathbf{SCM}_{\rm FSA}}({\rm insloc}(0)) = \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}$.

(17) ${\rm card}({\rm Stop}_{\mathbf{SCM}_{\rm FSA}}) = 1$.

Let P be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. The functor ${\rm Directed}(P,l)$ yields a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ and is defined by:

(Def. 1) ${\rm Directed}(P,l) = ({\rm id}_{\text{the instructions of } \mathbf{SCM}_{\rm FSA}}+\cdot(\mathbf{halt}_{\mathbf{SCM}_{\rm FSA}} \dot{\longmapsto} {\rm goto}\ l))\cdot P$.

One can prove the following proposition

(18) For every programmed finite partial state I of $\mathbf{SCM}_{\rm FSA}$ holds ${\rm Directed}(I) = {\rm Directed}(I, {\rm insloc}({\rm card}\, I))$.

Let P be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. One can check that ${\rm Directed}(P,l)$ is halt-free.

Let P be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$. Note that ${\rm Directed}(P)$ is halt-free.

One can prove the following propositions:

(19) For every programmed finite partial state P of $\mathbf{SCM}_{\rm FSA}$ and for every instruction-location l of $\mathbf{SCM}_{\rm FSA}$ holds ${\rm dom}\,{\rm Directed}(P,l) = {\rm dom}\, P$.

(20) Let P be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ and l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. Then ${\rm Directed}(P,l) = P+\cdot(\mathbf{halt}_{\mathbf{SCM}_{\rm FSA}} \dot{\longmapsto} {\rm goto}\ l)\cdot P$.

(21) Let P be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$, l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$, and x be a set such that $x \in {\rm dom}\, P$. Then

(i) if $P(x) = \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}$, then $({\rm Directed}(P,l))(x) = {\rm goto}\ l$, and

(ii) if $P(x) \neq \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}$, then $({\rm Directed}(P,l))(x) = P(x)$.

(22) Let i be an instruction of $\mathbf{SCM}_{\rm FSA}$, a be an integer location, and n be a natural number. If i does not destroy a, then ${\rm IncAddr}(i,n)$ does not destroy a.

(23) Let P be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$, n be a natural number, and a be an integer location. If P does not destroy a, then ${\rm ProgramPart}({\rm Relocated}(P,n))$ does not destroy a.

(24) For every good programmed finite partial state P of $\mathbf{SCM}_{\rm FSA}$ and for every natural number n holds ${\rm ProgramPart}({\rm Relocated}(P,n))$ is good.

(25) Let I, J be programmed finite partial states of $\mathbf{SCM}_{\rm FSA}$ and a be an integer location. Suppose I does not destroy a and J does not destroy a. Then $I+\cdot J$ does not destroy a.

(26) For all good programmed finite partial states I, J of $\mathbf{SCM}_{\rm FSA}$ holds $I+\cdot J$ is good.

(27) Let I be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$, l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$, and a be an integer location. If I does not destroy a, then ${\rm Directed}(I,l)$ does not destroy a.

Let I be a good programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. Observe that $\mathrm{Directed}(I,l)$ is good.

Let I be a good macro instruction. One can verify that $\mathrm{Directed}(I)$ is good.

Let I be a macro instruction and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. Note that $\mathrm{Directed}(I,l)$ is initial.

Let I, J be good macro instructions. Observe that $I;J$ is good.

Let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. The functor $\mathrm{Goto}(l)$ yields a halt-free good macro instruction and is defined as follows:

(Def. 2) $\mathrm{Goto}(l) = \mathrm{insloc}(0) \dot{\longmapsto} \mathrm{goto}\ l$.

Let s be a state of $\mathbf{SCM}_{\rm FSA}$ and let P be an initial finite partial state of $\mathbf{SCM}_{\rm FSA}$. We say that P is pseudo-closed on s if and only if the condition (Def. 3) is satisfied.

(Def. 3) There exists a natural number k such that $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot(P+\cdot\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)} = \mathrm{insloc}(\mathrm{card}\,\mathrm{ProgramPart}(P))$ and for every natural number n such that $n < k$ holds $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot(P+\cdot\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(n)} \in \mathrm{dom}\,P$.

Let P be an initial finite partial state of $\mathbf{SCM}_{\rm FSA}$. We say that P is pseudo-paraclosed if and only if:

(Def. 4) For every state s of $\mathbf{SCM}_{\rm FSA}$ holds P is pseudo-closed on s.

Let us note that there exists a macro instruction which is pseudo-paraclosed.

Let s be a state of $\mathbf{SCM}_{\rm FSA}$ and let P be an initial finite partial state of $\mathbf{SCM}_{\rm FSA}$. Let us assume that P is pseudo-closed on s. The functor $\mathrm{pseudo}-\mathrm{LifeSpan}(s,P)$ yields a natural number and is defined by:

(Def. 5) $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot(P+\cdot\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{pseudo}-\mathrm{LifeSpan}(s,P))} = \mathrm{insloc}(\mathrm{card}\,\mathrm{ProgramPart}(P))$ and for every natural number n such that $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot(P+\cdot\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(n)} \notin \mathrm{dom}\,P$ holds $\mathrm{pseudo}-\mathrm{LifeSpan}(s,P) \leq n$.

One can prove the following propositions:

(28) For all macro instructions I, J and for every set x such that $x \in \mathrm{dom}\,I$ holds $(I;J)(x) = (\mathrm{Directed}(I))(x)$.

(29) For every instruction-location l of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{card}\,\mathrm{Goto}(l) = 1$.

(30) Let P be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ and x be a set such that $x \in \mathrm{dom}\,P$. Then

(i) if $P(x) = \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}$, then $(\mathrm{Directed}(P))(x) = \mathrm{goto}\ \mathrm{insloc}(\mathrm{card}\,P)$, and

(ii) if $P(x) \neq \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}$, then $(\mathrm{Directed}(P))(x) = P(x)$.

(31) Let s be a state of $\mathbf{SCM}_{\rm FSA}$ and P be an initial finite partial state of $\mathbf{SCM}_{\rm FSA}$. Suppose P is pseudo-closed on s. Let n be a natural number. If $n < \mathrm{pseudo}-\mathrm{LifeSpan}(s,P)$, then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot(P+\cdot\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(n)} \in \mathrm{dom}\,P$ and $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot(P+\cdot\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(n)) \neq \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}$.

(32) Let s be a state of $\mathbf{SCM}_{\rm FSA}$ and I, J be macro instructions. Suppose I is pseudo-closed on s. Let k be a natural number. Suppose $k \leq \mathrm{pseudo}-\mathrm{LifeSpan}(s,I)$. Then $(\mathrm{Computation}(s+\cdot(I+\cdot\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)$ and $(\mathrm{Computation}(s+\cdot((I;J)+\cdot\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)$ are equal outside the instruction locations of $\mathbf{SCM}_{\rm FSA}$.

(33) For every programmed finite partial state I of $\mathbf{SCM}_{\rm FSA}$ and for every instruction-location l of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{card}\,\mathrm{Directed}(I,l) = \mathrm{card}\,I$.

(34) For every macro instruction I holds $\mathrm{card}\,\mathrm{Directed}(I) = \mathrm{card}\,I$.

(35) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. Suppose I is closed on s and halting on s. Let k be a natural number. Suppose $k \leq \mathrm{LifeSpan}(s+\cdot(I+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0))))$. Then $(\mathrm{Computation}(s+\cdot(I+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)$ and $(\mathrm{Computation}(s+\cdot(\mathrm{Directed}(I)+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$ and $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot(\mathrm{Directed}(I)+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)) \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(36) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. Suppose I is closed on s and halting on s. Then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot(\mathrm{Directed}(I)+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s+\cdot(I+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0))))+1)} = \mathrm{insloc}(\mathrm{card}\, I)$ and $(\mathrm{Computation}(s+\cdot(I+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s+\cdot(I+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))\restriction(\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations}) = (\mathrm{Computation}(s+\cdot(\mathrm{Directed}(I)+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s+\cdot(I+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0))))+1)\restriction(\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations})$.

(37) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. If I is closed on s and halting on s, then $\mathrm{Directed}(I)$ is pseudo-closed on s.

(38) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. If I is closed on s and halting on s, then $\mathrm{pseudo} - \mathrm{LifeSpan}(s, \mathrm{Directed}(I)) = \mathrm{LifeSpan}(s+\cdot(I+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))) + 1$.

(39) Let I be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. If I is halt-free, then $\mathrm{Directed}(I, l) = I$.

(40) For every macro instruction I such that I is halt-free holds $\mathrm{Directed}(I) = I$.

(41) For all macro instructions I, J holds $\mathrm{Directed}(I); J = I; J$.

(42) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I, J be macro instructions. Suppose I is closed on s and halting on s. Then

(i) for every natural number k such that $k \leq \mathrm{LifeSpan}(s+\cdot(I+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0))))$ holds $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot(\mathrm{Directed}(I)+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)} = \mathbf{IC}_{(\mathrm{Computation}(s+\cdot((I;\, J)+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)}$ and $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot(\mathrm{Directed}(I)+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)) = \mathrm{CurInstr}((\mathrm{Computation}(s+\cdot((I;\, J)+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k))$,

(ii) $(\mathrm{Computation}(s+\cdot(\mathrm{Directed}(I)+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s+\cdot(I+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0))))+1)\restriction(\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations}) = (\mathrm{Computation}(s+\cdot((I;\, J)+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s+\cdot(I+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0))))+1)\restriction(\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations})$, and

(iii) $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot(\mathrm{Directed}(I)+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s+\cdot(I+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0))))+1)} = \mathbf{IC}_{(\mathrm{Computation}(s+\cdot((I;\, J)+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s+\cdot(I+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0))))+1)}$.

(43) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I, J be macro instructions. Suppose I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$. Then

(i) for every natural number k such that $k \leq \mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(I))$ holds $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{Directed}(I))))(k)} = \mathbf{IC}_{(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(I;\, J)))(k)}$ and $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{Directed}(I))))(k)) = \mathrm{CurInstr}((\mathrm{Computation}(s+\cdot \mathrm{Initialized}(I;\, J)))(k))$,

(ii) $(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{Directed}(I))))(\mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(I))+1)\restriction(\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations}) = (\mathrm{Computation}(s+\cdot \mathrm{Initialized}(I;\, J)))(\mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(I))+1)\restriction(\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations})$, and

(iii) $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{Directed}(I))))(\mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(I))+1)} = \mathbf{IC}_{(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(I;\, J)))(\mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(I))+1)}$.

(44) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. Suppose I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$. Let k be a natural number. Suppose $k \leq \mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(I))$. Then $(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(I)))(k)$ and $(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{Directed}(I))))(k)$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$ and $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{Directed}(I))))(k)) \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(45) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. Suppose I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$. Then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{Directed}(I))))(\mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(I))+1)} = \mathrm{insloc}(\mathrm{card}\, I)$ and $(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(I)))(\mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(I)))\restriction(\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations}) = (\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{Directed}(I))))(\mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(I))+1)\restriction(\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations})$.

(46) Let I be a macro instruction and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose I is closed on s and halting on s. Then $I; \mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}}$ is closed on s and $I; \mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}}$ is halting on s.

(47) For every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{insloc}(0) \in \mathrm{dom}\,\mathrm{Goto}(l)$ and $(\mathrm{Goto}(l))(\mathrm{insloc}(0)) = \mathrm{goto}\ l$.

(48) Let N be a set with non empty elements, S be a definite non empty non void AMI over N, I be a programmed finite partial state of S, and x be a set. If $x \in \mathrm{dom}\,I$, then $I(x)$ is an instruction of S.

(49) Let I be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, x be a set, and k be a natural number. If $x \in \mathrm{dom}\,\mathrm{ProgramPart}(\mathrm{Relocated}(I,k))$, then $(\mathrm{ProgramPart}(\mathrm{Relocated}(I,k)))(x) = (\mathrm{Relocated}(I,k))(x)$.

(50) For every programmed finite partial state I of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every natural number k holds $\mathrm{ProgramPart}(\mathrm{Relocated}(\mathrm{Directed}(I),k)) = \mathrm{Directed}(\mathrm{ProgramPart}(\mathrm{Relocated}(I,k)),\mathrm{insloc}(\mathrm{card}\,I + k))$.

(51) Let I, J be programmed finite partial states of $\mathbf{SCM}_{\mathrm{FSA}}$ and l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $\mathrm{Directed}(I {+}\cdot J, l) = \mathrm{Directed}(I,l) {+}\cdot \mathrm{Directed}(J,l)$.

(52) For all macro instructions I, J holds $\mathrm{Directed}(I;\, J) = I;\, \mathrm{Directed}(J)$.

(53) Let I be a macro instruction and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. If I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$, then $\mathbf{IC}_{(\mathrm{Computation}(s {+}\cdot \mathrm{Initialized}(I;\, \mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}})))(\mathrm{LifeSpan}(s {+}\cdot \mathrm{Initialized}(I))+1)} = \mathrm{insloc}(\mathrm{card}\,I)$.

(54) Let I be a macro instruction and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$. Then $(\mathrm{Computation}(s {+}\cdot \mathrm{Initialized}(I)))(\mathrm{LifeSpan}(s {+}\cdot \mathrm{Initialized}(I)))\restriction(\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations}) = (\mathrm{Computation}(s {+}\cdot \mathrm{Initialized}(I;\, \mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}})))(\mathrm{LifeSpan}(s {+}\cdot \mathrm{Initialized}(I))+1)\restriction(\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations})$.

(55) Let I be a macro instruction and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. If I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$, then $s {+}\cdot \mathrm{Initialized}(I;\, \mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}})$ is halting.

(56) Let I be a macro instruction and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. If I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$, then $\mathrm{LifeSpan}(s {+}\cdot \mathrm{Initialized}(I;\, \mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}})) = \mathrm{LifeSpan}(s {+}\cdot \mathrm{Initialized}(I)) + 1$.

(57) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. If I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$, then $\mathrm{IExec}(I;\, \mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}}, s) = \mathrm{IExec}(I,s) {+}\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(\mathrm{card}\,I))$.

(58) Let I, J be macro instructions and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose I is closed on s and halting on s. Then $I;\, \mathrm{Goto}(\mathrm{insloc}(\mathrm{card}\,J+1));\, J;\, \mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}}$ is closed on s and $I;\, \mathrm{Goto}(\mathrm{insloc}(\mathrm{card}\,J+1));\, J;\, \mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}}$ is halting on s.

(59) Let I, J be macro instructions and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. If I is closed on s and halting on s, then $s {+}\cdot ((I;\, \mathrm{Goto}(\mathrm{insloc}(\mathrm{card}\,J+1));\, J;\, \mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}}) {+}\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))$ is halting.

(60) Let I, J be macro instructions and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. If I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$, then $s {+}\cdot \mathrm{Initialized}(I;\, \mathrm{Goto}(\mathrm{insloc}(\mathrm{card}\,J+1));\, J;\, \mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}})$ is halting.

(61) Let I, J be macro instructions and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. If I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$, then $\mathbf{IC}_{\mathrm{IExec}(I;\, \mathrm{Goto}(\mathrm{insloc}(\mathrm{card}\,J+1));\, J;\, \mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}}, s)} = \mathrm{insloc}(\mathrm{card}\,I + \mathrm{card}\,J + 1)$.

(62) Let I, J be macro instructions and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$. Then $\mathrm{IExec}(I;\, \mathrm{Goto}(\mathrm{insloc}(\mathrm{card}\,J+1));\, J;\, \mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}}, s) = \mathrm{IExec}(I,s) {+}\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(\mathrm{card}\,I + \mathrm{card}\,J + 1))$.

REFERENCES

[1] Noriko Asamoto. Constant assignment macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Part II. *Journal of Formalized Mathematics*, 8, 1996. http://mizar.org/JFM/Vol8/scmfsa7b.html.

[2] Noriko Asamoto, Yatsuka Nakamura, Piotr Rudnicki, and Andrzej Trybulec. On the composition of macro instructions. Part II. *Journal of Formalized Mathematics*, 8, 1996. http://mizar.org/JFM/Vol8/scmfsa6b.html.

[3] Noriko Asamoto, Yatsuka Nakamura, Piotr Rudnicki, and Andrzej Trybulec. On the composition of macro instructions. Part III. *Journal of Formalized Mathematics*, 8, 1996. http://mizar.org/JFM/Vol8/scmfsa6c.html.

[4] Grzegorz Bancerek. Cardinal numbers. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/card_1.html.

[5] Grzegorz Bancerek. The fundamental properties of natural numbers. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/nat_1.html.

[6] 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.

[7] Grzegorz Bancerek and Piotr Rudnicki. Development of terminology for **scm**. *Journal of Formalized Mathematics*, 5, 1993. http://mizar.org/JFM/Vol5/scm_1.html.

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

[9] Czesław Byliński. Functions and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/funct_1.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] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Journal of Formalized Mathematics*, 4, 1992. http://mizar.org/JFM/Vol4/ami_1.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 and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Journal of Formalized Mathematics*, 8, 1996. http://mizar.org/JFM/Vol8/scmfsa_4.html.

[17] Andrzej Trybulec and Yatsuka Nakamura. Relocability for $\mathbf{SCM}_{\mathrm{FSA}}$. *Journal of Formalized Mathematics*, 8, 1996. http://mizar.org/JFM/Vol8/scmfsa_5.html.

[18] Andrzej Trybulec, Yatsuka Nakamura, and Noriko Asamoto. On the compositions of macro instructions. Part I. *Journal of Formalized Mathematics*, 8, 1996. http://mizar.org/JFM/Vol8/scmfsa6a.html.

[19] 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.

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

Received August 27, 1996

Published January 2, 2004