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

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

Noriko Asamoto
Ochanomizu University
Tokyo

MML Identifier: `SCMFSA8A`.

The terminology and notation used in this paper are introduced in the following papers: [16], [22], [6], [10], [23], [11], [12], [9], [5], [7], [13], [19], [15], [21], [17], [18], [2], [8], [20], [14], [4], [3], and [1].

One can prove the following propositions:

(1) For all functions f, g such that $\mathrm{dom}\, f$ misses $\mathrm{dom}\, g$ holds $f{+\cdot}g = g{+\cdot}f$.

(2) For all functions f, g and for every set D such that $\mathrm{dom}\, g$ misses D holds $(f{+\cdot}g) \restriction D = f \restriction D$.

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

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

(5) For every state s of $\mathbf{SCM}_{\rm FSA}$ 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 $\mathbf{SCM}_{\rm FSA}$. Then s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\rm FSA}$ 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 $\mathbf{SCM}_{\rm FSA}$ 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 $\mathbf{SCM}_{\rm FSA}$ 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 $\mathbf{SCM}_{\rm FSA}$ holds $I \subseteq I{+\cdot}\,\text{Start-At}(l)$.

ISSN 1426–2630

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

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

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

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

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

(17) $\mathrm{card}(\text{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}}) = 1$.

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

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

One can prove the following proposition

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

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

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

Next we state several propositions:

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

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

(21) Let P be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$, and let x be arbitrary. Suppose $x \in \mathrm{dom}\, P$. Then if $P(x) = \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$, then $(\text{Directed}(P, l))(x) = \text{goto } l$ and if $P(x) \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$, then $(\text{Directed}(P, l))(x) = P(x)$.

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

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

(25) Let I, J be programmed finite partial states of $\mathbf{SCM}_{\rm FSA}$ and let 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 $\mathbf{SCM}_{\rm FSA}$, and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$, and let a be an integer location. If I does not destroy a, then $\mathrm{Directed}(I, l)$ 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 $\mathbf{SCM}_{\rm FSA}$. Note that $\mathrm{Directed}(I, l)$ is good.

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

Let I be a macro instruction and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. One can verify 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 by:

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

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

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

Let I be a finite partial state of $\mathbf{SCM}_{\rm FSA}$. We say that I is psuedo-paraclosed if and only if:

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

Let us observe that there exists a macro instruction which is psuedo-paraclosed.

Let s be a state of $\mathbf{SCM}_{\rm FSA}$ and let I be a macro instruction. Let us assume that I is psuedo-closed on s. The functor $\mathrm{psuedo}-\mathrm{LifeSpan}(s, I)$ yielding a natural number is defined by:

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

We now state a number of propositions:

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

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

(30) Let P be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let x be arbitrary. Suppose $x \in \operatorname{dom} P$. Then if $P(x) = \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$, then $(\mathrm{Directed}(P))(x) = \mathrm{goto}\ \mathrm{insloc}(\operatorname{card} P)$ and if $P(x) \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$, then $(\mathrm{Directed}(P))(x) = P(x)$.

(31) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let I be a macro instruction. Suppose I is psuedo-closed on s. Let n be a natural number. If $n < \mathrm{psuedo}-\mathrm{LifeSpan}(s, I)$, then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot(I+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(n)} \in \operatorname{dom} I$ and $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot(I+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(n)) \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(32) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let I, J be macro instructions. Suppose I is psuedo-closed on s. Let k be a natural number. Suppose $k \leq \mathrm{psuedo}-\mathrm{LifeSpan}(s, I)$. 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 $\mathbf{SCM}_{\mathrm{FSA}}$.

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

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

(35) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let 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 let I be a macro instruction. Suppose I is closed on s and halting on s.
Then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot(\mathrm{Directed}(I)+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s+\cdot(I+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0))))+1)} = \mathrm{insloc}(\operatorname{card} I)$ 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 let I be a macro instruction. If I is closed on s and halting on s, then $\mathrm{Directed}(I)$ is psuedo-closed on s.

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

(39) Let I be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let 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 let 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 let 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 let 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 let 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}

$\cup$ FinSeq-Locations).

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

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

(48) Let I be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ and let x be arbitrary. If $x \in \mathrm{dom}\,I$, then $I(x)$ is an instruction of $\mathbf{SCM}_{\rm FSA}$.

(49) Let I be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$, and let x be arbitrary, and let 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 $\mathbf{SCM}_{\rm FSA}$ 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 $\mathbf{SCM}_{\rm FSA}$ and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. 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 let s be a state of $\mathbf{SCM}_{\rm FSA}$. If I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$, then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(I;\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}})))(\mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(I))+1)} = \mathrm{insloc}(\mathrm{card}\,I)$.

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

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

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

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

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

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

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

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

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

References

[1] Noriko Asamoto. Constant assignment macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Part II. *Formalized Mathematics*, 6(**1**):59–63, 1997.

[2] Noriko Asamoto. Some multi instructions defined by sequence of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):615–619, 1996.

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

[4] Noriko Asamoto, Yatsuka Nakamura, Piotr Rudnicki, and Andrzej Trybulec. On the composition of macro instructions. Part II. *Formalized Mathematics*, 6(**1**):41–47, 1997.

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

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

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

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

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

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

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

[12] 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.

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

[14] Piotr Rudnicki and Andrzej Trybulec. Memory handling for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 6(**1**):29–36, 1997.

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

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

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

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

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

[20] Andrzej Trybulec, Yatsuka Nakamura, and Noriko Asamoto. On the compositions of macro instructions. Part I. *Formalized Mathematics*, 6(**1**):21–27, 1997.

[21] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\rm FSA}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.

[22] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.

[23] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received August 27, 1996