FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Computation of Two Consecutive Program Blocks for SCMPDS[1]

Jing-Chao Chen
Shanghai Jiaotong University

Summary. In this article, a program block without halting instructions is called No-StopCode program block. If a program consists of two blocks, where the first block is parahalting (i.e. halt for all states) and No-StopCode, and the second block is parahalting and shiftable, it can be computed by combining the computation results of the two blocks. For a program which consists of a instruction and a block, we obtain a similar conclusion. For a large amount of programs, the computation method given in the article is useful, but it is not suitable to recursive programs.

MML Identifier: SCMPDS_5.

The terminology and notation used here have been introduced in the following articles: [16], [20], [11], [21], [5], [6], [18], [2], [12], [13], [17], [14], [4], [10], [9], [19], [7], [1], [15], [8], and [3].

1. Preliminaries

For simplicity, we use the following convention: x denotes a set, m, n denote natural numbers, a, b denote Int position, i denotes an instruction of SCMPDS, s, s_1, s_2 denote states of SCMPDS, k_1, k_2 denote integers, l_1 denotes an instruction-location of SCMPDS, I, J denote Program-block, and N denotes a set with non empty elements.

One can prove the following propositions:

[1]This research is partially supported by the National Natural Science Foundation of China Grant No. 69873033.

ISSN 1426–2630

(1) Let S be a halting von Neumann definite AMI over N and s be a state of S. If $s = \mathrm{Following}(s)$, then for every n holds $(\mathrm{Computation}(s))(n) = s$.

(2) $x \in \mathrm{dom}\,\mathrm{Load}(i)$ iff $x = \mathrm{inspos}\,0$.

(3) If $l_1 \in \mathrm{dom}\,\mathrm{stop}\,I$ and $(\mathrm{stop}\,I)(l_1) \neq \mathbf{halt}_{\mathrm{SCMPDS}}$, then $l_1 \in \mathrm{dom}\,I$.

(4) $\mathrm{dom}\,\mathrm{Load}(i) = \{\mathrm{inspos}\,0\}$ and $(\mathrm{Load}(i))(\mathrm{inspos}\,0) = i$.

(5) $\mathrm{inspos}\,0 \in \mathrm{dom}\,\mathrm{Load}(i)$.

(6) $\mathrm{card}\,\mathrm{Load}(i) = 1$.

(7) $\mathrm{card}\,\mathrm{stop}\,I = \mathrm{card}\,I + 1$.

(8) $\mathrm{card}\,\mathrm{stop}\,\mathrm{Load}(i) = 2$.

(9) $\mathrm{inspos}\,0 \in \mathrm{dom}\,\mathrm{stop}\,\mathrm{Load}(i)$ and $\mathrm{inspos}\,1 \in \mathrm{dom}\,\mathrm{stop}\,\mathrm{Load}(i)$.

(10) $(\mathrm{stop}\,\mathrm{Load}(i))(\mathrm{inspos}\,0) = i$ and $(\mathrm{stop}\,\mathrm{Load}(i))(\mathrm{inspos}\,1) = \mathbf{halt}_{\mathrm{SCMPDS}}$.

(11) $x \in \mathrm{dom}\,\mathrm{stop}\,\mathrm{Load}(i)$ iff $x = \mathrm{inspos}\,0$ or $x = \mathrm{inspos}\,1$.

(12) $\mathrm{dom}\,\mathrm{stop}\,\mathrm{Load}(i) = \{\mathrm{inspos}\,0, \mathrm{inspos}\,1\}$.

(13) $\mathrm{inspos}\,0 \in \mathrm{dom}\,\mathrm{Initialized}(\mathrm{stop}\,\mathrm{Load}(i))$ and $\mathrm{inspos}\,1 \in \mathrm{dom}\,\mathrm{Initialized}$ $(\mathrm{stop}\,\mathrm{Load}(i))$ and $(\mathrm{Initialized}(\mathrm{stop}\,\mathrm{Load}(i)))(\mathrm{inspos}\,0) = i$ and $(\mathrm{Initialized}$ $(\mathrm{stop}\,\mathrm{Load}(i)))(\mathrm{inspos}\,1) = \mathbf{halt}_{\mathrm{SCMPDS}}$.

(14) For all Program-block I, J holds $\mathrm{Initialized}(\mathrm{stop}\,I;J) =$ $(I;(J;\mathrm{SCMPDS} - \mathrm{Stop}))+\cdot\,\mathrm{Start\text{-}At}(\mathrm{inspos}\,0)$.

(15) For all Program-block I, J holds $\mathrm{Initialized}(I) \subseteq \mathrm{Initialized}(\mathrm{stop}\,I;J)$.

(16) $\mathrm{dom}\,\mathrm{stop}\,I \subseteq \mathrm{dom}\,\mathrm{stop}\,I;J$.

(17) For all Program-block I, J holds $\mathrm{Initialized}(\mathrm{stop}\,I)+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I;J)$ $= \mathrm{Initialized}(\mathrm{stop}\,I;J)$.

(18) If $\mathrm{Initialized}(I) \subseteq s$, then $\mathbf{IC}_s = \mathrm{inspos}\,0$.

(19) $(s+\cdot\,\mathrm{Initialized}(I))(a) = s(a)$.

(20) Let I be a parahalting Program-block. Suppose $\mathrm{Initialized}(\mathrm{stop}\,I) \subseteq s_1$ and $\mathrm{Initialized}(\mathrm{stop}\,I) \subseteq s_2$ and s_1 and s_2 are equal outside the instruction locations of SCMPDS. Let k be a natural number. Then $(\mathrm{Computation}(s_1))(k)$ and $(\mathrm{Computation}(s_2))(k)$ are equal outside the instruction locations of SCMPDS and $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(k)) =$ $\mathrm{CurInstr}((\mathrm{Computation}(s_2))(k))$.

(21) Let I be a parahalting Program-block. Suppose $\mathrm{Initialized}(\mathrm{stop}\,I) \subseteq s_1$ and $\mathrm{Initialized}(\mathrm{stop}\,I) \subseteq s_2$ and s_1 and s_2 are equal outside the instruction locations of SCMPDS. Then $\mathrm{LifeSpan}(s_1) = \mathrm{LifeSpan}(s_2)$ and $\mathrm{Result}(s_1)$ and $\mathrm{Result}(s_2)$ are equal outside the instruction locations of SCMPDS.

(22) For every Program-block I holds $\mathbf{IC}_{\mathrm{IExec}(I,s)} = \mathbf{IC}_{\mathrm{Result}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I))}$.

(23) Let I be a parahalting Program-block and J be a Program-block. Suppose $\mathrm{Initialized}(\mathrm{stop}\,I) \subseteq s$. Let given m. Suppose $m \leqslant \mathrm{LifeSpan}(s)$. Then $(\mathrm{Computation}(s))(m)$ and $(\mathrm{Computation}(s+\cdot(I;J)))(m)$ are equal outside the instruction locations of SCMPDS.

(24) Let I be a parahalting Program-block and J be a Program-block. Suppose Initialized(stop I) $\subseteq s$. Let given m. Suppose $m \leqslant$ LifeSpan(s). Then (Computation(s))(m) and (Computation(s+· Initialized(stop I;J)))(m) are equal outside the instruction locations of SCMPDS.

2. Non Halting Instructions and Parahalting Instructions

Let i be an instruction of SCMPDS. We say that i is No-StopCode if and only if:

(Def. 1) $i \neq \mathbf{halt}_{\mathrm{SCMPDS}}$.

Let i be an instruction of SCMPDS. We say that i is parahalting if and only if:

(Def. 2) Load(i) is parahalting.

One can verify that there exists an instruction of SCMPDS which is No-StopCode, shiftable, and parahalting.

One can prove the following proposition

(25) If $k_1 \neq 0$, then goto k_1 is No-StopCode.

Let us consider a. Observe that return a is No-StopCode.

Let us consider a, k_1. Note that a:=k_1 is No-StopCode and saveIC(a, k_1) is No-StopCode.

Let us consider a, k_1, k_2. One can check the following observations:

* $(a, k_1) <> 0\_goto k_2$ is No-StopCode,
* $(a, k_1) <= 0\_goto k_2$ is No-StopCode,
* $(a, k_1) >= 0\_goto k_2$ is No-StopCode, and
* a_{k_1}:=k_2 is No-StopCode.

Let us consider a, k_1, k_2. Note that AddTo(a, k_1, k_2) is No-StopCode.

Let us consider a, b, k_1, k_2. One can verify the following observations:

* AddTo(a, k_1, b, k_2) is No-StopCode,
* SubFrom(a, k_1, b, k_2) is No-StopCode,
* MultBy(a, k_1, b, k_2) is No-StopCode,
* Divide(a, k_1, b, k_2) is No-StopCode, and
* $(a, k_1) := (b, k_2)$ is No-StopCode.

Let us note that $\mathbf{halt}_{\mathrm{SCMPDS}}$ is parahalting.

Let i be a parahalting instruction of SCMPDS. Observe that Load(i) is parahalting.

Let us consider a, k_1. Observe that a:=k_1 is parahalting.

Let us consider a, k_1, k_2. Note that a_{k_1}:=k_2 is parahalting and AddTo(a, k_1, k_2) is parahalting.

Let us consider a, b, k_1, k_2. One can check the following observations:

* AddTo(a, k_1, b, k_2) is parahalting,
* SubFrom(a, k_1, b, k_2) is parahalting,
* MultBy(a, k_1, b, k_2) is parahalting,
* Divide(a, k_1, b, k_2) is parahalting, and
* $(a, k_1) := (b, k_2)$ is parahalting.

Next we state the proposition

(26) If InsCode$(i) = 1$, then i is not parahalting.

Let I_1 be a finite partial state of SCMPDS. We say that I_1 is No-StopCode if and only if:

(Def. 3) For every instruction-location x of SCMPDS such that $x \in \operatorname{dom} I_1$ holds $I_1(x) \neq \mathbf{halt}_{\mathrm{SCMPDS}}$.

Let us observe that there exists a Program-block which is parahalting, shiftable, and No-StopCode.

Let I, J be No-StopCode Program-block. Observe that $I;J$ is No-StopCode.

Let i be a No-StopCode instruction of SCMPDS. Observe that Load(i) is No-StopCode.

Let i be a No-StopCode instruction of SCMPDS and let J be a No-StopCode Program-block. Note that $i;J$ is No-StopCode.

Let I be a No-StopCode Program-block and let j be a No-StopCode instruction of SCMPDS. Observe that $I;j$ is No-StopCode.

Let i, j be No-StopCode instructions of SCMPDS. Observe that $i;j$ is No-StopCode.

Next we state several propositions:

(27) For every parahalting No-StopCode Program-block I such that $\mathrm{Initialized}(\mathrm{stop}\, I) \subseteq s$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s+\cdot\, \mathrm{Initialized}(\mathrm{stop}\, I)))} = \mathrm{inspos}\, \mathrm{card}\, I$.

(28) For every parahalting Program-block I and for every natural number k such that $k < \mathrm{LifeSpan}(s+\cdot\, \mathrm{Initialized}(\mathrm{stop}\, I))$ holds $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\, \mathrm{Initialized}(\mathrm{stop}\, I)))(k)} \in \operatorname{dom} I$.

(29) Let I be a parahalting Program-block and k be a natural number. Suppose $\mathrm{Initialized}(I) \subseteq s$ and $k \leqslant \mathrm{LifeSpan}(s+\cdot\, \mathrm{Initialized}(\mathrm{stop}\, I))$. Then $(\mathrm{Computation}(s))(k)$ and $(\mathrm{Computation}(s+\cdot\, \mathrm{Initialized}(\mathrm{stop}\, I)))(k)$ are equal outside the instruction locations of SCMPDS.

(30) For every parahalting No-StopCode Program-block I such that $\mathrm{Initialized}(I) \subseteq s$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s+\cdot\, \mathrm{Initialized}(\mathrm{stop}\, I)))} = \mathrm{inspos}\, \mathrm{card}\, I$.

(31) For every parahalting Program-block I such that $\mathrm{Initialized}(I) \subseteq s$ holds $\mathrm{CurInstr}((\mathrm{Computation}(s))(\mathrm{LifeSpan}(s+\cdot\, \mathrm{Initialized}(\mathrm{stop}\, I)))) = \mathbf{halt}_{\mathrm{SCMPDS}}$ or $\mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s+\cdot\, \mathrm{Initialized}(\mathrm{stop}\, I)))} = \mathrm{inspos}\, \mathrm{card}\, I$.

(32) Let I be a parahalting No-StopCode Program-block and k be a natural number. If Initialized$(I) \subseteq s$ and $k <$ LifeSpan(s+· Initialized(stop I)), then CurInstr((Computation(s))(k)) $\neq$ $\mathbf{halt}_{\rm SCMPDS}$.

(33) Let I be a parahalting Program-block, J be a Program-block, and k be a natural number. Suppose $k \leqslant$ LifeSpan(s+· Initialized(stop I)). Then (Computation(s+· Initialized(stop I)))(k) and (Computation(s+·((I;J)+· Start-At(inspos 0))))(k) are equal outside the instruction locations of SCMPDS.

(34) Let I be a parahalting Program-block, J be a Program-block, and k be a natural number. Suppose $k \leqslant$ LifeSpan(s+· Initialized(stop I)). Then (Computation(s+· Initialized(stop I)))(k) and (Computation(s+· Initialized (stop I;J)))(k) are equal outside the instruction locations of SCMPDS.

Let I be a parahalting Program-block and let J be a parahalting shiftable Program-block. One can verify that I;J is parahalting.

Let i be a parahalting instruction of SCMPDS and let J be a parahalting shiftable Program-block. Note that i;J is parahalting.

Let I be a parahalting Program-block and let j be a parahalting shiftable instruction of SCMPDS. Observe that I;j is parahalting.

Let i be a parahalting instruction of SCMPDS and let j be a parahalting shiftable instruction of SCMPDS. One can check that i;j is parahalting.

Next we state the proposition

(35) Let s, s_1 be states of SCMPDS and J be a parahalting shiftable Program-block. If s = (Computation(s_1+· Initialized(stop J)))(m), then Exec(CurInstr(s), s+· Start-At($\mathbf{IC}_s + n$)) = Following(s)+· Start-At($\mathbf{IC}_{\text{Following}(s)} + n$).

3. Computation of two Consecutive Program Blocks

The following propositions are true:

(36) Let I be a parahalting No-StopCode Program-block, J be a parahalting shiftable Program-block, and k be a natural number. Suppose Initialized(stop I;J) $\subseteq s$. Then (Computation(Result(s+· Initialized (stop I))+· Initialized(stop J)))(k)+· Start-At ($\mathbf{IC}_{(\text{Computation}(\text{Result}(s+\cdot\,\text{Initialized}(\text{stop}\,I))+\cdot\,\text{Initialized}(\text{stop}\,J)))(k)}$ + card I) and (Computation(s+· Initialized(stop I;J)))(LifeSpan(s+· Initialized(stop I))+ k) are equal outside the instruction locations of SCMPDS.

(37) Let I be a parahalting No-StopCode Program-block and J be a parahalting shiftable Program-block. Then LifeSpan(s+· Initialized(stop I;J)) = LifeSpan(s+· Initialized(stop I))+LifeSpan(Result(s+· Initialized(stop I))+· Initialized(stop J)).

(38) Let I be a parahalting No-StopCode Program-block and J be a parahalting shiftable Program-block. Then $\mathrm{IExec}(I;J,s) = \mathrm{IExec}(J,\mathrm{IExec}(I,s)) +\cdot \text{Start-At}(\mathbf{IC}_{\mathrm{IExec}(J,\mathrm{IExec}(I,s))} + \mathrm{card}\, I)$.

(39) Let I be a parahalting No-StopCode Program-block and J be a parahalting shiftable Program-block. Then $(\mathrm{IExec}(I;J,s))(a) = (\mathrm{IExec}(J,\mathrm{IExec}(I,s)))(a)$.

4. Computation of the Program Consisting of a Instruction and a Block

Let s be a state of SCMPDS. The functor Initialized(s) yields a state of SCMPDS and is defined by:

(Def. 4) $\mathrm{Initialized}(s) = s +\cdot \text{Start-At}(\mathrm{inspos}\, 0)$.

Next we state several propositions:

(40) $\mathbf{IC}_{\mathrm{Initialized}(s)} = \mathrm{inspos}\, 0$ and $(\mathrm{Initialized}(s))(a) = s(a)$ and $(\mathrm{Initialized}(s))(l_1) = s(l_1)$.

(41) s_1 and s_2 are equal outside the instruction locations of SCMPDS iff $s_1 \restriction (\text{Data-Loc}_{\mathrm{SCM}} \cup \{\mathbf{IC}_{\mathrm{SCMPDS}}\}) = s_2 \restriction (\text{Data-Loc}_{\mathrm{SCM}} \cup \{\mathbf{IC}_{\mathrm{SCMPDS}}\})$.

(42) If $s_1 \restriction \text{Data-Loc}_{\mathrm{SCM}} = s_2 \restriction \text{Data-Loc}_{\mathrm{SCM}}$, then $s_1(\mathrm{DataLoc}(s_1(a), k_1)) = s_2(\mathrm{DataLoc}(s_2(a), k_1))$.

(43) If $s_1 \restriction \text{Data-Loc}_{\mathrm{SCM}} = s_2 \restriction \text{Data-Loc}_{\mathrm{SCM}}$ and $\mathrm{InsCode}(i) \neq 3$, then $\mathrm{Exec}(i, s_1) \restriction \text{Data-Loc}_{\mathrm{SCM}} = \mathrm{Exec}(i, s_2) \restriction \text{Data-Loc}_{\mathrm{SCM}}$.

(44) For every shiftable instruction i of SCMPDS such that $s_1 \restriction \text{Data-Loc}_{\mathrm{SCM}} = s_2 \restriction \text{Data-Loc}_{\mathrm{SCM}}$ holds $\mathrm{Exec}(i, s_1) \restriction \text{Data-Loc}_{\mathrm{SCM}} = \mathrm{Exec}(i, s_2) \restriction \text{Data-Loc}_{\mathrm{SCM}}$.

(45) For every parahalting instruction i of SCMPDS holds $\mathrm{Exec}(i, \mathrm{Initialized}(s)) = \mathrm{IExec}(\mathrm{Load}(i), s)$.

(46) Let I be a parahalting No-StopCode Program-block and j be a parahalting shiftable instruction of SCMPDS. Then $(\mathrm{IExec}(I;j,s))(a) = (\mathrm{Exec}(j,\mathrm{IExec}(I,s)))(a)$.

(47) Let i be a No-StopCode parahalting instruction of SCMPDS and j be a shiftable parahalting instruction of SCMPDS. Then $(\mathrm{IExec}(i;j,s))(a) = (\mathrm{Exec}(j,\mathrm{Exec}(i,\mathrm{Initialized}(s))))(a)$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.
[2] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[3] Grzegorz Bancerek and Piotr Rudnicki. Development of terminology for **scm**. *Formalized Mathematics*, 4(**1**):61–67, 1993.

[4] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.
[5] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[6] 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.
[7] Jing-Chao Chen. Computation and program shift in the SCMPDS computer. *Formalized Mathematics*, 8(**1**):193–199, 1999.
[8] Jing-Chao Chen. The construction and shiftability of program blocks for SCMPDS. *Formalized Mathematics*, 8(**1**):201–210, 1999.
[9] Jing-Chao Chen. The SCMPDS computer and the basic semantics of its instructions. *Formalized Mathematics*, 8(**1**):183–191, 1999.
[10] Jing-Chao Chen. A small computer model with push-down stack. *Formalized Mathematics*, 8(**1**):175–182, 1999.
[11] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[12] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[13] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.
[14] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[15] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[16] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[17] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[18] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[19] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.
[20] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[21] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received June 15, 1999