FORMALIZED MATHEMATICS
Volume 11, Number 4, 2003
University of Białystok

SCMPDS Is Not Standard

Artur Korniłowicz[1]
University of Białystok

Yasunari Shidama
Shinshu University
Nagano

Summary. The aim of the paper is to show that SCMPDS ([8]) does not belong to the class of standard computers ([16]).

MML Identifier: SCMPDS_9.

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

1. Preliminaries

In this paper r, s are real numbers.

We now state several propositions:

(1) $0 \leqslant r + |r|$.

(2) $0 \leqslant -r + |r|$.

(3) If $|r| = |s|$, then $r = s$ or $r = -s$.

(4) For all natural numbers i, j such that $i < j$ and $i \neq 0$ holds $\frac{i}{j}$ is not integer.

(5) $\{2 \cdot k; k$ ranges over natural numbers: $k > 1\}$ is infinite.

(6) For every function f and for all sets a, b, c such that $a \neq c$ holds $(f {+\cdot} (a \dot{\longmapsto} b))(c) = f(c)$.

(7) For every function f and for all sets a, b, c, d such that $a \neq b$ holds $(f {+\cdot} [a \longmapsto c, b \longmapsto d])(a) = c$ and $(f {+\cdot} [a \longmapsto c, b \longmapsto d])(b) = d$.

[1]This paper was written during the first author's post-doctoral fellowship granted by Shinshu University, Japan.

ISSN 1426–2630

2. SCMPDS

For simplicity, we adopt the following rules: a, b are Int positions, i is an instruction of SCMPDS, l is an instruction-location of SCMPDS, and k, k_1, k_2 are integers.

Let l_1, l_2 be Int positions and let a, b be integers. Then $[l_1 \longmapsto a, l_2 \longmapsto b]$ is a finite partial state of SCMPDS.

One can verify that SCMPDS has non trivial instruction locations.

Let l be an instruction-location of SCMPDS. The functor $\mathrm{locnum}(l)$ yields a natural number and is defined by:

(Def. 1) $\mathbf{i}_{\mathrm{locnum}(l)} = l$.

Let l be an instruction-location of SCMPDS. Then $\mathrm{locnum}(l)$ is an element of $\mathbb{N}$.

We now state a number of propositions:

(8) $l = 2 \cdot \mathrm{locnum}(l) + 2$.

(9) For all instruction-locations l_3, l_4 of SCMPDS such that $l_3 \neq l_4$ holds $\mathrm{locnum}(l_3) \neq \mathrm{locnum}(l_4)$.

(10) For all instruction-locations l_3, l_4 of SCMPDS such that $l_3 \neq l_4$ holds $\mathrm{Next}(l_3) \neq \mathrm{Next}(l_4)$.

(11) Let N be a set with non empty elements, S be an IC-Ins-separated definite non empty non void AMI over N, i be an instruction of S, and l be an instruction-location of S. Then $\mathrm{JUMP}(i) \subseteq \mathrm{NIC}(i, l)$.

(12) If for every state s of SCMPDS such that $\mathbf{IC}_s = l$ and $s(l) = i$ holds $(\mathrm{Exec}(i, s))(\mathbf{IC}_{\mathrm{SCMPDS}}) = \mathrm{Next}(\mathbf{IC}_s)$, then $\mathrm{NIC}(i, l) = \{\mathrm{Next}(l)\}$.

(13) If for every instruction-location l of SCMPDS holds $\mathrm{NIC}(i, l) = \{\mathrm{Next}(l)\}$, then $\mathrm{JUMP}(i)$ is empty.

(14) $\mathrm{NIC}(\mathrm{goto}\ k, l) = \{2 \cdot |k + \mathrm{locnum}(l)| + 2\}$.

(15) $\mathrm{NIC}(\mathrm{return}\, a, l) = \{2 \cdot k; k$ ranges over natural numbers: $k > 1\}$.

(16) $\mathrm{NIC}(\mathrm{saveIC}(a, k_1), l) = \{\mathrm{Next}(l)\}$.

(17) $\mathrm{NIC}(a{:=}k_1, l) = \{\mathrm{Next}(l)\}$.

(18) $\mathrm{NIC}(a_{k_1}{:=}k_2, l) = \{\mathrm{Next}(l)\}$.

(19) $\mathrm{NIC}((a, k_1) := (b, k_2), l) = \{\mathrm{Next}(l)\}$.

(20) $\mathrm{NIC}(\mathrm{AddTo}(a, k_1, k_2), l) = \{\mathrm{Next}(l)\}$.

(21) $\mathrm{NIC}(\mathrm{AddTo}(a, k_1, b, k_2), l) = \{\mathrm{Next}(l)\}$.

(22) $\mathrm{NIC}(\mathrm{SubFrom}(a, k_1, b, k_2), l) = \{\mathrm{Next}(l)\}$.

(23) $\mathrm{NIC}(\mathrm{MultBy}(a, k_1, b, k_2), l) = \{\mathrm{Next}(l)\}$.

(24) $\mathrm{NIC}(\mathrm{Divide}(a, k_1, b, k_2), l) = \{\mathrm{Next}(l)\}$.

(25) $\mathrm{NIC}((a, k_1) <> 0\_\mathrm{goto}\, k_2, l) = \{\mathrm{Next}(l), |2 \cdot (k_2 + \mathrm{locnum}(l))| + 2\}$.

(26) $\mathrm{NIC}((a, k_1) <= 0\_\mathrm{goto}\, k_2, l) = \{\mathrm{Next}(l), |2 \cdot (k_2 + \mathrm{locnum}(l))| + 2\}$.

(27) $\mathrm{NIC}((a, k_1) >= 0\_\mathrm{goto}\, k_2, l) = \{\mathrm{Next}(l), |2 \cdot (k_2 + \mathrm{locnum}(l))| + 2\}$.

Let us consider k. Observe that JUMP(goto k) is empty.

Next we state the proposition

(28) $\mathrm{JUMP}(\mathrm{return}\, a) = \{2 \cdot k; k$ ranges over natural numbers: $k > 1\}$.

Let us consider a. Note that JUMP(return a) is infinite.

Let us consider a, k_1. One can verify that JUMP(saveIC(a, k_1)) is empty.

Let us consider a, k_1. Observe that JUMP(a:=k_1) is empty.

Let us consider a, k_1, k_2. Note that JUMP(a_{k_1}:=k_2) is empty.

Let us consider a, b, k_1, k_2. One can check that JUMP($(a, k_1) := (b, k_2)$) is empty.

Let us consider a, k_1, k_2. One can verify that JUMP(AddTo(a, k_1, k_2)) is empty.

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

* JUMP(AddTo(a, k_1, b, k_2)) is empty,
* JUMP(SubFrom(a, k_1, b, k_2)) is empty,
* JUMP(MultBy(a, k_1, b, k_2)) is empty, and
* JUMP(Divide(a, k_1, b, k_2)) is empty.

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

* JUMP((a, k_1) <> 0_goto k_2) is empty,
* JUMP((a, k_1) <= 0_goto k_2) is empty, and
* JUMP((a, k_1) >= 0_goto k_2) is empty.

Next we state two propositions:

(29) SUCC(l) = the instruction locations of SCMPDS.

(30) Let N be a set with non empty elements, S be an IC-Ins-separated definite non empty non void AMI over N, and l_3, l_4 be instruction-locations of S. If SUCC(l_3) = the instruction locations of S, then $l_3 \leqslant l_4$.

Let us mention that SCMPDS is non InsLoc-antisymmetric.

One can verify that SCMPDS is non standard.

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[2] Grzegorz Bancerek. The ordinal numbers. *Formalized Mathematics*, 1(**1**):91–96, 1990.
[3] Grzegorz Bancerek. Sequences of ordinal numbers. *Formalized Mathematics*, 1(**2**):281–290, 1990.
[4] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[5] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.
[6] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[7] 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.

[8] Jing-Chao Chen. The SCMPDS computer and the basic semantics of its instructions. *Formalized Mathematics*, 8(**1**):183–191, 1999.
[9] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[10] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[11] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[12] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.
[13] Andrzej Trybulec. Subsets of complex numbers. *To appear in Formalized Mathematics.*
[14] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[15] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[16] Andrzej Trybulec, Piotr Rudnicki, and Artur Korniłowicz. Standard ordering of instruction locations. *Formalized Mathematics*, 9(**2**):291–301, 2001.
[17] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[18] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.
[19] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[20] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received September 27, 2003