FORMALIZED MATHEMATICS
Volume 9, Number 2, 2001
University of Białystok

Insert Sort on SCMPDS[1]

Jing-Chao Chen
Shanghai Jiaotong University

Summary. The goal of this article is to examine the effectiveness of "for-loop" and "while-loop" statements on SCMPDS by insert sort. In this article, first of all, we present an approach to compute the execution result of "for-loop" program by "loop-invariant", based on Hoare's axioms for program verification. Secondly, we extend the fundamental properties of the finite sequence and complex instructions of SCMPDS. Finally, we prove the correctness of the insert sort program described in the article.

MML Identifier: SCPISORT.

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

1. Preliminaries

In this paper n, p_0 are natural numbers.

Let f be a finite sequence of elements of $\mathbb{Z}$, let s be a state of SCMPDS, and let m be a natural number. We say that f is FinSequence on s, m if and only if:

(Def. 1) For every natural number i such that $1 \leqslant i$ and $i \leqslant \operatorname{len} f$ holds $f(i) = s(\operatorname{intpos} m + i)$.

We now state four propositions:

(1) Let f be a finite sequence of elements of $\mathbb{Z}$ and m, n be natural numbers. If $m \geqslant n$, then f is non decreasing on m, n.

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

ISSN 1426–2630

(2) Let s be a state of SCMPDS and n, m be natural numbers. Then there exists a finite sequence f of elements of $\mathbb{Z}$ such that $\operatorname{len} f = n$ and for every natural number i such that $1 \leqslant i$ and $i \leqslant \operatorname{len} f$ holds $f(i) = s(\operatorname{intpos} m+i)$.

(3) Let s be a state of SCMPDS and n, m be natural numbers. Then there exists a finite sequence f of elements of $\mathbb{Z}$ such that $\operatorname{len} f = n$ and f is FinSequence on s, m.

(4) Let f, g be finite sequences of elements of $\mathbb{Z}$ and m, n be natural numbers. Suppose that $1 \leqslant n$ and $n \leqslant \operatorname{len} f$ and $1 \leqslant m$ and $m \leqslant \operatorname{len} f$ and $\operatorname{len} f = \operatorname{len} g$ and $f(m) = g(n)$ and $f(n) = g(m)$ and for every natural number k such that $k \neq m$ and $k \neq n$ and $1 \leqslant k$ and $k \leqslant \operatorname{len} f$ holds $f(k) = g(k)$. Then f and g are fiberwise equipotent.

The following propositions are true:

(5) For all states s_1, s_2 of SCMPDS such that for every Int position a holds $s_1(a) = s_2(a)$ holds $\operatorname{Dstate} s_1 = \operatorname{Dstate} s_2$.

(6) Let s be a state of SCMPDS, I be a No-StopCode Program-block, and j be a parahalting shiftable instruction of SCMPDS. Suppose I is closed on s and halting on s. Then $I;\ j$ is closed on s and $I;\ j$ is halting on s.

(7) Let s be a state of SCMPDS, I be a No-StopCode Program-block, J be a shiftable parahalting Program-block, and a be an Int position. If I is closed on s and halting on s, then $(\operatorname{IExec}(I;\ J, s))(a) = (\operatorname{IExec}(J, \operatorname{IExec}(I, s)))(a)$.

(8) Let s be a state of SCMPDS, I be a No-StopCode parahalting Program-block, J be a shiftable Program-block, and a be an Int position. If J is closed on $\operatorname{IExec}(I, s)$ and halting on $\operatorname{IExec}(I, s)$, then $(\operatorname{IExec}(I;\ J, s))(a) = (\operatorname{IExec}(J, \operatorname{IExec}(I, s)))(a)$.

(9) Let s be a state of SCMPDS, I be a Program-block, and J be a shiftable parahalting Program-block. Suppose I is closed on s and halting on s. Then $I;\ J$ is closed on s and $I;\ J$ is halting on s.

(10) Let s be a state of SCMPDS, I be a parahalting Program-block, and J be a shiftable Program-block. Suppose J is closed on $\operatorname{IExec}(I, s)$ and halting on $\operatorname{IExec}(I, s)$. Then $I;\ J$ is closed on s and $I;\ J$ is halting on s.

(11) Let s be a state of SCMPDS, I be a Program-block, and j be a parahalting shiftable instruction of SCMPDS. Suppose I is closed on s and halting on s. Then $I;\ j$ is closed on s and $I;\ j$ is halting on s.

2. Computing the Execution Result of For-Loop Program by Loop-Invariant

In this article we present several logical schemes. The scheme *ForDownHalt* deals with a state $\mathcal{A}$ of SCMPDS, a No-StopCode shiftable Program-block $\mathcal{B}$, an Int position $\mathcal{C}$, an integer $\mathcal{D}$, a natural number $\mathcal{E}$, and a unary predicate $\mathcal{P}$, and states that:

$\mathcal{P}[\mathcal{A}]$ or not $\mathcal{P}[\mathcal{A}]$ but for-down$(\mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{B})$ is closed on $\mathcal{A}$ but for-down$(\mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{B})$ is halting on $\mathcal{A}$

provided the following requirements are met:

- $\mathcal{E} > 0$,
- $\mathcal{P}[\text{Dstate}\,\mathcal{A}]$, and
- Let t be a state of SCMPDS. Suppose $\mathcal{P}[\text{Dstate}\,t]$ and $t(\mathcal{C}) = \mathcal{A}(\mathcal{C})$ and $t(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) > 0$. Then $(\text{IExec}(\mathcal{B};\ \text{AddTo}(\mathcal{C}, \mathcal{D}, -\mathcal{E}), t))(\mathcal{C}) = t(\mathcal{C})$ and $(\text{IExec}(\mathcal{B};\ \text{AddTo}(\mathcal{C}, \mathcal{D}, -\mathcal{E}), t))(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) = t(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) - \mathcal{E}$ and $\mathcal{B}$ is closed on t and $\mathcal{B}$ is halting on t and $\mathcal{P}[\text{Dstate}\,\text{IExec}(\mathcal{B};\ \text{AddTo}(\mathcal{C}, \mathcal{D}, -\mathcal{E}), t)]$.

The scheme *ForDownExec* deals with a state $\mathcal{A}$ of SCMPDS, a No-StopCode shiftable Program-block $\mathcal{B}$, an Int position $\mathcal{C}$, an integer $\mathcal{D}$, a natural number $\mathcal{E}$, and a unary predicate $\mathcal{P}$, and states that:

$\mathcal{P}[\mathcal{A}]$ or not $\mathcal{P}[\mathcal{A}]$ but $\text{IExec}(\text{for-down}(\mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{B}), \mathcal{A}) = \text{IExec}(\text{for-down}(\mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{B}), \text{IExec}(\mathcal{B};\ \text{AddTo}(\mathcal{C}, \mathcal{D}, -\mathcal{E}), \mathcal{A}))$

provided the parameters meet the following conditions:

- $\mathcal{E} > 0$,
- $\mathcal{A}(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) > 0$,
- $\mathcal{P}[\text{Dstate}\,\mathcal{A}]$, and
- Let t be a state of SCMPDS. Suppose $\mathcal{P}[\text{Dstate}\,t]$ and $t(\mathcal{C}) = \mathcal{A}(\mathcal{C})$ and $t(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) > 0$. Then $(\text{IExec}(\mathcal{B};\ \text{AddTo}(\mathcal{C}, \mathcal{D}, -\mathcal{E}), t))(\mathcal{C}) = t(\mathcal{C})$ and $(\text{IExec}(\mathcal{B};\ \text{AddTo}(\mathcal{C}, \mathcal{D}, -\mathcal{E}), t))(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) = t(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) - \mathcal{E}$ and $\mathcal{B}$ is closed on t and $\mathcal{B}$ is halting on t and $\mathcal{P}[\text{Dstate}\,\text{IExec}(\mathcal{B};\ \text{AddTo}(\mathcal{C}, \mathcal{D}, -\mathcal{E}), t)]$.

The scheme *ForDownEnd* deals with a state $\mathcal{A}$ of SCMPDS, a No-StopCode shiftable Program-block $\mathcal{B}$, an Int position $\mathcal{C}$, an integer $\mathcal{D}$, a natural number $\mathcal{E}$, and a unary predicate $\mathcal{P}$, and states that:

$\mathcal{P}[\mathcal{A}]$ or not $\mathcal{P}[\mathcal{A}]$ but $(\text{IExec}(\text{for-down}(\mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{B}), \mathcal{A}))(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) \leqslant 0$ but $\mathcal{P}[\text{Dstate}\,\text{IExec}(\text{for-down}(\mathcal{C}, \mathcal{D}, \mathcal{E}, \mathcal{B}), \mathcal{A})]$

provided the parameters have the following properties:

- $\mathcal{E} > 0$,
- $\mathcal{P}[\text{Dstate}\,\mathcal{A}]$, and
- Let t be a state of SCMPDS. Suppose $\mathcal{P}[\text{Dstate}\,t]$ and $t(\mathcal{C}) = \mathcal{A}(\mathcal{C})$ and $t(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) > 0$. Then $(\text{IExec}(\mathcal{B};\ \text{AddTo}(\mathcal{C}, \mathcal{D}, -\mathcal{E}), t))(\mathcal{C}) = t(\mathcal{C})$ and $(\text{IExec}(\mathcal{B};\ \text{AddTo}(\mathcal{C}, \mathcal{D}, -\mathcal{E}), t))(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) = t(\text{DataLoc}(\mathcal{A}(\mathcal{C}), \mathcal{D})) - \mathcal{E}$ and $\mathcal{B}$ is closed on t and $\mathcal{B}$ is halting on t and $\mathcal{P}[\text{Dstate}\,\text{IExec}(\mathcal{B};\ \text{AddTo}(\mathcal{C}, \mathcal{D}, -\mathcal{E}), t)]$.

We now state three propositions:

(12) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a, x, y be Int positions, i, c be integers, and n be a natural number.

Suppose that

(i) $n > 0$,

(ii) $s(x) \geqslant s(y) + c$, and

(iii) for every state t of SCMPDS such that $t(x) \geqslant t(y) + c$ and $t(a) = s(a)$ and $t(\mathrm{DataLoc}(s(a), i)) > 0$ holds $(\mathrm{IExec}(I;\ \mathrm{AddTo}(a, i, -n), t))(a) = t(a)$ and $(\mathrm{IExec}(I;\ \mathrm{AddTo}(a, i, -n), t))(\mathrm{DataLoc}(s(a), i)) = t(\mathrm{DataLoc}(s(a), i)) - n$ and I is closed on t and halting on t and $(\mathrm{IExec}(I;\ \mathrm{AddTo}(a, i, -n), t))(x) \geqslant (\mathrm{IExec}(I;\ \mathrm{AddTo}(a, i, -n), t))(y) + c$.

Then for-down(a, i, n, I) is closed on s and for-down(a, i, n, I) is halting on s.

(13) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a, x, y be Int positions, i, c be integers, and n be a natural number. Suppose that

(i) $n > 0$,

(ii) $s(x) \geqslant s(y) + c$,

(iii) $s(\mathrm{DataLoc}(s(a), i)) > 0$, and

(iv) for every state t of SCMPDS such that $t(x) \geqslant t(y) + c$ and $t(a) = s(a)$ and $t(\mathrm{DataLoc}(s(a), i)) > 0$ holds $(\mathrm{IExec}(I;\ \mathrm{AddTo}(a, i, -n), t))(a) = t(a)$ and $(\mathrm{IExec}(I;\ \mathrm{AddTo}(a, i, -n), t))(\mathrm{DataLoc}(s(a), i)) = t(\mathrm{DataLoc}(s(a), i)) - n$ and I is closed on t and halting on t and $(\mathrm{IExec}(I;\ \mathrm{AddTo}(a, i, -n), t))(x) \geqslant (\mathrm{IExec}(I;\ \mathrm{AddTo}(a, i, -n), t))(y) + c$.

Then IExec(for-down$(a, i, n, I), s)$ = IExec(for-down(a, i, n, I), $\mathrm{IExec}(I;\ \mathrm{AddTo}(a, i, -n), s))$.

(14) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, i be an integer, and n be a natural number. Suppose that

(i) $s(\mathrm{DataLoc}(s(a), i)) > 0$,

(ii) $n > 0$,

(iii) $\mathrm{card}\, I > 0$,

(iv) $a \neq \mathrm{DataLoc}(s(a), i)$, and

(v) for every state t of SCMPDS such that $t(a) = s(a)$ holds $(\mathrm{IExec}(I, t))(a) = t(a)$ and $(\mathrm{IExec}(I, t))(\mathrm{DataLoc}(s(a), i)) = t(\mathrm{DataLoc}(s(a), i))$ and I is closed on t and halting on t.

Then for-down(a, i, n, I) is closed on s and for-down(a, i, n, I) is halting on s.

3. A Program for Insert Sort

Let n, p_0 be natural numbers. The functor insert-sort(n, p_0) yielding a Program-block is defined by the condition (Def. 2).

(Def. 2) $\text{insert-sort}(n, p_0) = (\text{GBP} := 0);\ ((\text{GBP})_1 := 0);\ ((\text{GBP})_2 := n - 1);$ $((\text{GBP})_3 := p_0);\ \text{for-down}(\text{GBP}, 2, 1, \text{AddTo}(\text{GBP}, 3, 1);$ $((\text{GBP}, 4) := (\text{GBP}, 3));\ \text{AddTo}(\text{GBP}, 1, 1);\ ((\text{GBP}, 6) := (\text{GBP}, 1));$ $\text{while} > 0(\text{GBP}, 6, ((\text{GBP}, 5) := (\text{intpos}\,4, -1));$ $\text{SubFrom}(\text{GBP}, 5, \text{intpos}\,4, 0);\ (\textbf{if}\ \text{GBP} > 5\ \textbf{then}$ $((\text{GBP}, 5) := (\text{intpos}\,4, -1));\ ((\text{intpos}\,4, -1) := (\text{intpos}\,4, 0));$ $((\text{intpos}\,4, 0) := (\text{GBP}, 5));\ \text{AddTo}(\text{GBP}, 4, -1);\ \text{AddTo}(\text{GBP}, 6, -1)$ $\textbf{else}\ \text{Load}((\text{GBP})_6 := 0))))$.

4. The Property of Insert Sort and Its Correctness

We now state two propositions:

(15) $\text{card}\,\text{insert-sort}(n, p_0) = 23$.

(16) If $p_0 \geqslant 7$, then $\text{insert-sort}(n, p_0)$ is parahalting.

One can prove the following propositions:

(17) Let s be a state of SCMPDS, f, g be finite sequences of elements of $\mathbb{Z}$, and k_0, k be natural numbers. Suppose that $s(a_4) \geqslant 7 + s(a_6)$ and $s(\text{GBP}) = 0$ and $k = s(a_6)$ and $k_0 = s(a_4) - s(a_6) - 1$ and f is FinSequence on s, k_0 and g is FinSequence on $\text{IExec}(I_2, s)$, k_0 and $\text{len}\, f = \text{len}\, g$ and $\text{len}\, f > k$ and f is non decreasing on 1, k. Then

(i) f and g are fiberwise equipotent,

(ii) g is non decreasing on 1, $k + 1$,

(iii) for every natural number i such that $i > k + 1$ and $i \leqslant \text{len}\, f$ holds $f(i) = g(i)$, and

(iv) for every natural number i such that $1 \leqslant i$ and $i \leqslant k + 1$ there exists a natural number j such that $1 \leqslant j$ and $j \leqslant k + 1$ and $g(i) = f(j)$,

where $a_4 = \text{intpos}\,4$, $a_6 = \text{intpos}\,6$, $I_2 = W_1$, $W_1 = \text{while} > 0(\text{GBP}, 6, B_1)$, $B_1 = k_1; k_2; I_1$, $k_1 = (\text{GBP}, 5) := (\text{intpos}\,4, -1)$, $k_2 = \text{SubFrom}(\text{GBP}, 5, \text{intpos}\,4, 0)$, $I_1 = \textbf{if}\ \text{GBP} > 5\ \textbf{then}\ T_1\ \textbf{else}\ F_1$, $T_1 = k_3; k_4; k_5; k_6; k_7$, $k_3 = (\text{GBP}, 5) := (\text{intpos}\,4, -1)$, $k_4 = (\text{intpos}\,4, -1) := (\text{intpos}\,4, 0)$, $k_5 = (\text{intpos}\,4, 0) := (\text{GBP}, 5)$, $k_6 = \text{AddTo}(\text{GBP}, 4, -1)$, $k_7 = \text{AddTo}(\text{GBP}, 6, -1)$, and $F_1 = \text{Load}((\text{GBP})_6 := 0)$.

(18) Let s be a state of SCMPDS, f, g be finite sequences of elements of $\mathbb{Z}$, and p_0, n be natural numbers. Suppose $p_0 \geqslant 6$ and $\text{len}\, f = n$ and $\text{len}\, g = n$ and f is FinSequence on s, p_0 and g is FinSequence on $\text{IExec}(\text{insert-sort}(n, p_0 + 1), s)$, p_0. Then f and g are fiberwise equipotent and g is non decreasing on 1, n.

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. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[4] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[5] Jing-Chao Chen. Computation and program shift in the SCMPDS computer. *Formalized Mathematics*, 8(**1**):193–199, 1999.
[6] Jing-Chao Chen. Computation of two consecutive program blocks for SCMPDS. *Formalized Mathematics*, 8(**1**):211–217, 1999.
[7] Jing-Chao Chen. The construction and computation of conditional statements for SCMPDS. *Formalized Mathematics*, 8(**1**):219–234, 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. The construction and computation of while-loop programs for SCMPDS. *Formalized Mathematics*, 9(**2**):397–405, 2001.
[11] Jing-Chao Chen. Recursive Euclide algorithm. *Formalized Mathematics*, 9(**1**):1–4, 2001.
[12] Jing-Chao Chen and Piotr Rudnicki. The construction and computation of for-loop programs for SCMPDS. *Formalized Mathematics*, 9(**1**):209–219, 2001.
[13] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[14] Andrzej Kondracki. The Chinese Remainder Theorem. *Formalized Mathematics*, 6(**4**):573–577, 1997.
[15] Jarosław Kotowicz. Functions and finite sequences of real numbers. *Formalized Mathematics*, 3(**2**):275–278, 1992.
[16] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[17] Piotr Rudnicki. The `for` (going up) macro instruction. *Formalized Mathematics*, 7(**1**):107–114, 1998.
[18] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[19] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[20] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[21] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.

Received June 14, 2000