JOURNAL OF FORMALIZED MATHEMATICS
Volume 12, Released 2000, Published 2003
Inst. of Computer Science, Univ. of Białystok

Quick Sort on SCMPDS[1]

Jing-Chao Chen
Shanghai Jiaotong University / China Bell Labs

Summary. Proving the correctness of quick sort is much more complicated than proving the correctness of the insert sort. Its difficulty is determined mainly by the following points:

- Quick sort needs to use a push-down stack.
- It contains three nested loops.
- A subroutine of this algorithm, "Partition", has no loop-invariant.

This means we cannot justify the correctness of the "Partition" subroutine by the Hoare's axiom on program verification. This article is organized as follows. First, we present several fundamental properties of "while" program and finite sequence. Second, we define the "Partition" subroutine on SCMPDS, the task of which is to split a sequence into a smaller and a larger subsequence. The definition of quick sort on SCMPDS follows. Finally, we describe the basic property of the "Partition" and quick sort, and prove their correctness.

MML Identifier: `SCPQSORT`.

WWW: `http://mizar.org/JFM/Vol12/scpqsort.html`

The articles [23], [8], [24], [27], [7], [9], [26], [2], [19], [20], [25], [22], [6], [15], [10], [1], [13], [5], [11], [12], [14], [3], [4], [16], [21], [18], and [17] provide the notation and terminology for this paper.

1. The Several Properties of "while" Program and Finite Sequence

In this paper n, p_0 are natural numbers.

Let I, J be shiftable Program-blocks, let a be an Int position, and let k_1 be an integer. One can check that **if** $a > k_1$ **then** I **else** J is shiftable.

We now state the proposition

(1) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, J be a shiftable Program-block, a, b be Int positions, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a),k_1)) > 0$ and I is closed on s and halting on s. Then $(\mathrm{IExec}(\mathbf{if}\ a > k_1\ \mathbf{then}\ I\ \mathbf{else}\ J,s))(b) = (\mathrm{IExec}(I,s))(b)$.

The following propositions are true:

(2) Let s, s_1 be states of SCMPDS, I be a No-StopCode shiftable Program-block, a be an Int position, i be an integer, and m be a natural number. Suppose $\mathrm{card}\, I > 0$ and I is closed on s and halting on s and $s(\mathrm{DataLoc}(s(a),i)) > 0$ and $m = \mathrm{LifeSpan}(s +\cdot \mathrm{Initialized}(\mathrm{stop}\, I)) + 2$ and $s_1 = (\mathrm{Computation}(s +\cdot \mathrm{Initialized}(\mathrm{stop\ while} > 0(a,i,I))))(m)$. Then $s_1 \restriction \mathrm{Data\text{-}Loc}_{\mathrm{SCM}} = \mathrm{IExec}(I,s) \restriction \mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$ and $s_1 +\cdot \mathrm{Initialized}(\mathrm{stop\ while} > 0(a,i,I)) = s_1$.

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

(3) Let s be a state of SCMPDS and I be a Program-block. Suppose that for every state t of SCMPDS such that $t \restriction \text{Data-Loc}_{\text{SCM}} = s \restriction \text{Data-Loc}_{\text{SCM}}$ holds I is halting on t. Then I is closed on s.

(4) For all instructions i_1, i_2, i_3, i_4 of SCMPDS holds $\text{card}(i_1; i_2; i_3; i_4) = 4$.

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

(i) $\text{card} I > 0$,

(ii) $s(x) \geq c + s(\text{DataLoc}(s(a), i))$, and

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

Then while $> 0(a, i, I)$ is closed on s and while $> 0(a, i, I)$ is halting on s and if $s(\text{DataLoc}(s(a), i)) > 0$, then $\text{IExec}(\text{while} > 0(a, i, I), s) = \text{IExec}(\text{while} > 0(a, i, I), \text{IExec}(I, s))$.

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

(i) $\text{card} I > 0$,

(ii) $s(x) \geq c$, and

(iii) for every state t of SCMPDS such that $t(x) \geq c$ and $t(y) = s(y)$ and $t(a) = s(a)$ and $t(\text{DataLoc}(s(a), i)) > 0$ holds $(\text{IExec}(I, t))(a) = t(a)$ and I is closed on t and halting on t and $(\text{IExec}(I, t))(\text{DataLoc}(s(a), i)) < t(\text{DataLoc}(s(a), i))$ and $(\text{IExec}(I, t))(x) \geq c$ and $(\text{IExec}(I, t))(y) = t(y)$.

Then while $> 0(a, i, I)$ is closed on s and while $> 0(a, i, I)$ is halting on s and if $s(\text{DataLoc}(s(a), i)) > 0$, then $\text{IExec}(\text{while} > 0(a, i, I), s) = \text{IExec}(\text{while} > 0(a, i, I), \text{IExec}(I, s))$.

(7) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a, x_1, x_2, x_3, x_4 be Int positions, and i, c, m_1 be integers. Suppose that

(i) $\text{card} I > 0$,

(ii) $s(x_4) = (s(x_3) - c) + s(x_1)$,

(iii) $m_1 \leq s(x_3) - c$, and

(iv) for every state t of SCMPDS such that $t(x_4) = (t(x_3) - c) + t(x_1)$ and $m_1 \leq t(x_3) - c$ and $t(x_2) = s(x_2)$ and $t(a) = s(a)$ and $t(\text{DataLoc}(s(a), i)) > 0$ holds $(\text{IExec}(I, t))(a) = t(a)$ and I is closed on t and halting on t and $(\text{IExec}(I, t))(\text{DataLoc}(s(a), i)) < t(\text{DataLoc}(s(a), i))$ and $(\text{IExec}(I, t))(x_4) = ((\text{IExec}(I, t))(x_3) - c) + (\text{IExec}(I, t))(x_1)$ and $m_1 \leq (\text{IExec}(I, t))(x_3) - c$ and $(\text{IExec}(I, t))(x_2) = t(x_2)$.

Then while $> 0(a, i, I)$ is closed on s and while $> 0(a, i, I)$ is halting on s and if $s(\text{DataLoc}(s(a), i)) > 0$, then $\text{IExec}(\text{while} > 0(a, i, I), s) = \text{IExec}(\text{while} > 0(a, i, I), \text{IExec}(I, s))$.

(8) Let f be a finite sequence of elements of $\mathbb{Z}$ and m, k_1, k, n be natural numbers. Suppose that $m \leq k$ and $k \leq n$ and $k_1 = k - 1$ and f is non decreasing on m, k_1 and f is non decreasing on $k + 1$, n and for every natural number i such that $m \leq i$ and $i < k$ holds $f(i) \leq f(k)$ and for every natural number i such that $k < i$ and $i \leq n$ holds $f(k) \leq f(i)$. Then f is non decreasing on m, n.

(9) Let f, g be finite sequences and x be a set. Suppose $x \in \text{dom} g$ and f and g are fiberwise equipotent. Then there exists a set y such that $y \in \text{dom} g$ and $f(x) = g(y)$.

(10) Let f, g, h be finite sequences. Then f and g are fiberwise equipotent if and only if $h^\frown f$ and $h^\frown g$ are fiberwise equipotent.

(11) Let f, g be finite sequences and m, n, j be natural numbers. Suppose that f and g are fiberwise equipotent and $m \leq n$ and $n \leq \operatorname{len} f$ and for every natural number i such that $1 \leq i$ and $i \leq m$ holds $f(i) = g(i)$ and for every natural number i such that $n < i$ and $i \leq \operatorname{len} f$ holds $f(i) = g(i)$ and $m < j$ and $j \leq n$. Then there exists a natural number k such that $m < k$ and $k \leq n$ and $f(j) = g(k)$.

2. Program Partition is to Split a Sequence into a Smaller and a Larger Subsequence

The Program-block Partition is defined by the condition (Def. 1).

(Def. 1) $\mathrm{Partition} = ((\mathrm{GBP},5) := (\mathrm{GBP},4))$; $\mathrm{SubFrom}(\mathrm{GBP},5,\mathrm{GBP},2)$; $((\mathrm{GBP},3) := (\mathrm{GBP},2))$; $\mathrm{AddTo}(\mathrm{GBP},3,1)$; while $>$ $0(\mathrm{GBP},5,$while $>$ $0(\mathrm{GBP},5,((\mathrm{GBP},7) := (\mathrm{GBP},5))$; $\mathrm{AddTo}(\mathrm{GBP},5,-1)$; $((\mathrm{GBP},6) := (\mathrm{intpos}\,4,0))$; $\mathrm{SubFrom}(\mathrm{GBP},6,\mathrm{intpos}\,2,0)$; (**if** $\mathrm{GBP} > 6$ **then** $\mathrm{AddTo}(\mathrm{GBP},4,-1)$; $\mathrm{AddTo}(\mathrm{GBP},7,-1)$ **else** $\mathrm{Load}((\mathrm{GBP})_5{:=}0)))$; while $>$ $0(\mathrm{GBP},7,((\mathrm{GBP},5) := (\mathrm{GBP},7))$; $\mathrm{AddTo}(\mathrm{GBP},7,-1)$; $((\mathrm{GBP},6) := (\mathrm{intpos}\,2,0))$; $\mathrm{SubFrom}(\mathrm{GBP},6,\mathrm{intpos}\,3,0)$; (**if** $\mathrm{GBP} >$ 6 **then** $\mathrm{AddTo}(\mathrm{GBP},3,1)$; $\mathrm{AddTo}(\mathrm{GBP},5,-1)$ **else** $\mathrm{Load}((\mathrm{GBP})_7{:=}0)))$; (**if** $\mathrm{GBP} > 0$ **then** 5 **else** $(((\mathrm{GBP},6) := (\mathrm{intpos}\,4,0))$; $((\mathrm{intpos}\,4,0) := (\mathrm{intpos}\,3,0))$; $((\mathrm{intpos}\,3,0) := (\mathrm{GBP},6))$; $\mathrm{AddTo}(\mathrm{GBP},5,-2)$; $\mathrm{AddTo}(\mathrm{GBP},3,1)$; $\mathrm{AddTo}(\mathrm{GBP},4,-1))))$; $((\mathrm{GBP},6) := (\mathrm{intpos}\,4,0))$; $((\mathrm{intpos}\,4,0) := (\mathrm{intpos}\,2,0))$; $((\mathrm{intpos}\,2,0) := (\mathrm{GBP},6))$.

3. The Construction of Quick Sort

Let n, p_0 be natural numbers. The functor $\mathrm{QuickSort}(n,p_0)$ yielding a Program-block is defined by the condition (Def. 2).

(Def. 2) $\mathrm{QuickSort}(n,p_0) = (\mathrm{GBP}{:=}0)$; $(\mathrm{SBP}{:=}1)$; $((\mathrm{SBP})_{p_1}{:=}p_0+1)$; $((\mathrm{SBP})_{p_1+1}{:=}p_1)$; while $>$ $0(\mathrm{GBP},1,((\mathrm{GBP},2) := (\mathrm{SBP},p_1+1))$; $\mathrm{SubFrom}(\mathrm{GBP},2,\mathrm{SBP},p_1)$; (**if** $\mathrm{GBP} > 2$ **then** $((\mathrm{GBP},2) := (\mathrm{SBP},p_1))$; $((\mathrm{GBP},4) := (\mathrm{SBP},p_1+1))$; Partition; $(((\mathrm{SBP},p_1+3) := (\mathrm{SBP},p_1+1))$; $((\mathrm{SBP},p_1+1) := (\mathrm{GBP},4))$; $((\mathrm{SBP},p_1+2) := (\mathrm{GBP},4))$; $\mathrm{AddTo}(\mathrm{SBP},p_1+1,-1)$; $\mathrm{AddTo}(\mathrm{SBP},p_1+2,1)$; $\mathrm{AddTo}(\mathrm{GBP},1,2))$ **else** $\mathrm{Load}(\mathrm{AddTo}(\mathrm{GBP},1,-2))))$, where $p_1 = p_0 + n$.

4. The Basic Property of Partition Program

Next we state four propositions:

(12) $\operatorname{card}\mathrm{Partition} = 38$.

(13) Let s be a state of SCMPDS and m_1, p_0 be natural numbers. Suppose $s(\mathrm{GBP}) = 0$ and $s(\mathrm{intpos}\,4) - s(\mathrm{intpos}\,2) > 0$ and $s(\mathrm{intpos}\,2) = m_1$ and $m_1 \geq p_0 + 1$ and $p_0 \geq 7$. Then Partition is closed on s and Partition is halting on s.

(14) Let s be a state of SCMPDS, m_1, p_0, n be natural numbers, and f, f_1 be finite sequences of elements of $\mathbb{Z}$. Suppose that $s(\mathrm{GBP}) = 0$ and $s(\mathrm{intpos}\,4) - s(\mathrm{intpos}\,2) > 0$ and $s(\mathrm{intpos}\,2) = m_1$ and $m_1 \geq p_0 + 1$ and $s(\mathrm{intpos}\,4) \leq p_0 + n$ and $p_0 \geq 7$ and f is FinSequence on s, p_0 and $\operatorname{len} f = n$ and f_1 is FinSequence on $\mathrm{IExec}(\mathrm{Partition},s)$, p_0 and $\operatorname{len} f_1 = n$. Then

(i) $(\mathrm{IExec}(\mathrm{Partition},s))(\mathrm{GBP}) = 0$,

(ii) $(\mathrm{IExec}(\mathrm{Partition},s))(\mathrm{intpos}\,1) = s(\mathrm{intpos}\,1)$,

(iii) f and f_1 are fiberwise equipotent, and

(iv) there exists a natural number m_4 such that $m_4 = (\mathrm{IExec}(\mathrm{Partition},s))(\mathrm{intpos}\,4)$ and $m_1 \leq m_4$ and $m_4 \leq s(\mathrm{intpos}\,4)$ and for every natural number i such that $m_1 \leq i$ and $i < m_4$ holds $(\mathrm{IExec}(\mathrm{Partition},s))(\mathrm{intpos}\,m_4) \geq (\mathrm{IExec}(\mathrm{Partition},s))(\mathrm{intpos}\,i)$ and for every natural number i such that $m_4 < i$ and $i \leq s(\mathrm{intpos}\,4)$ holds $(\mathrm{IExec}(\mathrm{Partition},s))(\mathrm{intpos}\,m_4) \leq (\mathrm{IExec}(\mathrm{Partition},s))(\mathrm{intpos}\,i)$ and for every natural number i such that $i \geq p_0 + 1$ but $i < s(\mathrm{intpos}\,2)$ or $i > s(\mathrm{intpos}\,4)$ holds $(\mathrm{IExec}(\mathrm{Partition},s))(\mathrm{intpos}\,i) = s(\mathrm{intpos}\,i)$.

(15) Partition is No-StopCode and shiftable.

5. THE BASIC PROPERTY OF QUICK SORT AND ITS CORRECTNESS

Next we state three propositions:

(16) $\operatorname{card} \operatorname{QuickSort}(n, p_0) = 57$.

(17) For all natural numbers p_0, n such that $p_0 \geq 7$ holds $\operatorname{QuickSort}(n, p_0)$ is parahalting.

(18) Let s be a state of SCMPDS and p_0, n be natural numbers. Suppose $p_0 \geq 7$. Then there exist finite sequences f, g of elements of $\mathbb{Z}$ such that

(i) $\operatorname{len} f = n$,

(ii) f is FinSequence on s, p_0,

(iii) $\operatorname{len} g = n$,

(iv) g is FinSequence on $\operatorname{IExec}(\operatorname{QuickSort}(n, p_0), s)$, p_0,

(v) f and g are fiberwise equipotent, and

(vi) g is non decreasing on 1, n.

REFERENCES

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

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

[3] Grzegorz Bancerek. König's theorem. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/card_3.html`.

[4] 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`.

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

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

[7] Czesław Byliński. Functions and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/funct_1.html`.

[8] Czesław Byliński. Functions from a set to a set. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/funct_2.html`.

[9] 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`.

[10] Jing-Chao Chen. Computation and program shift in the SCMPDS computer. *Journal of Formalized Mathematics*, 11, 1999. `http://mizar.org/JFM/Vol11/scmpds_3.html`.

[11] Jing-Chao Chen. Computation of two consecutive program blocks for SCMPDS. *Journal of Formalized Mathematics*, 11, 1999. `http://mizar.org/JFM/Vol11/scmpds_5.html`.

[12] Jing-Chao Chen. The construction and computation of conditional statements for SCMPDS. *Journal of Formalized Mathematics*, 11, 1999. `http://mizar.org/JFM/Vol11/scmpds_6.html`.

[13] Jing-Chao Chen. The construction and shiftability of program blocks for SCMPDS. *Journal of Formalized Mathematics*, 11, 1999. `http://mizar.org/JFM/Vol11/scmpds_4.html`.

[14] Jing-Chao Chen. Recursive Euclide algorithm. *Journal of Formalized Mathematics*, 11, 1999. `http://mizar.org/JFM/Vol11/scmp_gcd.html`.

[15] Jing-Chao Chen. The SCMPDS computer and the basic semantics of its instructions. *Journal of Formalized Mathematics*, 11, 1999. `http://mizar.org/JFM/Vol11/scmpds_2.html`.

[16] Jing-Chao Chen. The construction and computation of while-loop programs for SCMPDS. *Journal of Formalized Mathematics*, 12, 2000. `http://mizar.org/JFM/Vol12/scmpds_8.html`.

[17] Jing-Chao Chen. Insert sort on SCMPDS. *Journal of Formalized Mathematics*, 12, 2000. `http://mizar.org/JFM/Vol12/scpisort.html`.

[18] Jarosław Kotowicz. Functions and finite sequences of real numbers. *Journal of Formalized Mathematics*, 5, 1993. `http://mizar.org/JFM/Vol5/rfinseq.html`.

[19] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Journal of Formalized Mathematics*, 4, 1992. `http://mizar.org/JFM/Vol4/ami_1.html`.

[20] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Journal of Formalized Mathematics*, 4, 1992. `http://mizar.org/JFM/Vol4/ami_2.html`.

[21] Piotr Rudnicki. The `for` (going up) macro instruction. *Journal of Formalized Mathematics*, 10, 1998. `http://mizar.org/JFM/Vol10/sfmastr3.html`.

[22] Yasushi Tanaka. On the decomposition of the states of SCM. *Journal of Formalized Mathematics*, 5, 1993. `http://mizar.org/JFM/Vol5/ami_5.html`.

[23] Andrzej Trybulec. Tarski Grothendieck set theory. *Journal of Formalized Mathematics*, Axiomatics, 1989. `http://mizar.org/JFM/Axiomatics/tarski.html`.

[24] Andrzej Trybulec. Subsets of real numbers. *Journal of Formalized Mathematics*, Addenda, 2003. `http://mizar.org/JFM/Addenda/numbers.html`.

[25] 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`.

[26] Michał J. Trybulec. Integers. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/int_1.html`.

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

Received June 14, 2000

Published January 2, 2004