JOURNAL OF FORMALIZED MATHEMATICS
Volume 5, Released 1993, Published 2003
Inst. of Computer Science, Univ. of Białystok

Two Programs for SCM. Part II - Programs[1]

Grzegorz Bancerek
Polish Academy of Sciences
Institute of Mathematics
Warsaw

Piotr Rudnicki
University of Alberta
Department of Computing Science
Edmonton

Summary. We prove the correctness of two short programs for the **SCM** machine: one computes Fibonacci numbers and the other computes the *fusc* function of Dijkstra [5]. The formal definitions of these functions can be found in [4]. We prove the total correctness of the programs in two ways: by conducting inductions on computations and inductions on input data. In addition we characterize the concrete complexity of the programs as defined in [3].

MML Identifier: `FIB_FUSC`.

WWW: `http://mizar.org/JFM/Vol5/fib_fusc.html`

The articles [9], [1], [7], [2], [6], [3], [8], and [4] provide the notation and terminology for this paper.

The program computing Fib is a finite sequence of elements of the instructions of **SCM** and is defined by:

(Def. 1) The program computing Fib $= \langle \textbf{if } \mathbf{d}_1 > 0 \textbf{ goto } \mathbf{i}_2 \rangle \frown \langle \mathbf{halt_{SCM}} \rangle \frown \langle \mathbf{d}_3 {:=} \mathbf{d}_0 \rangle \frown \langle \mathrm{SubFrom}(\mathbf{d}_1, \mathbf{d}_0) \rangle \frown \langle \textbf{if } \mathbf{d}_1 = 0 \textbf{ goto } \mathbf{i}_1 \rangle \frown \langle \mathbf{d}_4 {:=} \mathbf{d}_2 \rangle \frown \langle \mathbf{d}_2 {:=} \mathbf{d}_3 \rangle \frown \langle \mathrm{AddTo}(\mathbf{d}_3, \mathbf{d}_4) \rangle \frown \langle \mathrm{goto}\ (\mathbf{i}_3) \rangle$.

We now state the proposition

(1) Let N be a natural number and s be a state with instruction counter on 0, with the program computing Fib located from 0, and $\langle 1 \rangle \frown \langle N \rangle \frown \langle 0 \rangle \frown \langle 0 \rangle$ from 0. Then

(i) s is halting,

(ii) if $N = 0$, then the complexity of $s = 1$,

(iii) if $N > 0$, then the complexity of $s = 6 \cdot N - 2$, and

(iv) $(\mathrm{Result}(s))(\mathbf{d}_3) = \mathrm{Fib}(N)$.

Let i be an integer. The functor $\mathrm{Fusc}'(i)$ yields a natural number and is defined as follows:

(Def. 2) There exists a natural number n such that $i = n$ and $\mathrm{Fusc}'(i) = \mathrm{Fusc}(n)$ or i is not a natural number and $\mathrm{Fusc}'(i) = 0$.

The program computing Fusc is a finite sequence of elements of the instructions of **SCM** and is defined by:

(Def. 3) The program computing Fusc $= \langle \textbf{if } \mathbf{d}_1 = 0 \textbf{ goto } \mathbf{i}_8 \rangle \frown \langle \mathbf{d}_4 {:=} \mathbf{d}_0 \rangle \frown \langle \mathrm{Divide}(\mathbf{d}_1, \mathbf{d}_4) \rangle \frown \langle \textbf{if } \mathbf{d}_4 = 0 \textbf{ goto } \mathbf{i}_6 \rangle \frown \langle \mathrm{AddTo}(\mathbf{d}_3, \mathbf{d}_2) \rangle \frown \langle \mathrm{goto}\ (\mathbf{i}_0) \rangle \frown \langle \mathrm{AddTo}(\mathbf{d}_2, \mathbf{d}_3) \rangle \frown \langle \mathrm{goto}\ (\mathbf{i}_0) \rangle \frown \langle \mathbf{halt_{SCM}} \rangle$.

[1] This work was partially supported by NSERC Grant OGP9207 while the first author visited University of Alberta, May–June 1993.

The following propositions are true:

(2) Let N be a natural number. Suppose $N > 0$. Let s be a state with instruction counter on 0, with the program computing Fusc located from 0, and $\langle 2\rangle \frown \langle N\rangle \frown \langle 1\rangle \frown \langle 0\rangle$ from 0. Then s is halting and $(\mathrm{Result}(s))(\mathbf{d}_3) = \mathrm{Fusc}(N)$ and the complexity of $s = 6\cdot(\lfloor \log_2 N\rfloor + 1) + 1$.

(3) Let N be a natural number, k, F_1, F_2 be natural numbers, and s be a state with instruction counter on 3, with the program computing Fib located from 0, and $\langle 1\rangle \frown \langle N\rangle \frown \langle F_1\rangle \frown \langle F_2\rangle$ from 0. Suppose $N > 0$ and $F_1 = \mathrm{Fib}(k)$ and $F_2 = \mathrm{Fib}(k+1)$. Then

(i) s is halting,

(ii) the complexity of $s = 6\cdot N - 4$, and

(iii) there exists a natural number m such that $m = (k+N) - 1$ and $(\mathrm{Result}(s))(\mathbf{d}_2) = \mathrm{Fib}(m)$ and $(\mathrm{Result}(s))(\mathbf{d}_3) = \mathrm{Fib}(m+1)$.

(5)[1] Let n be a natural number, N, A, B be natural numbers, and s be a state with instruction counter on 0, with the program computing Fusc located from 0, and $\langle 2\rangle \frown \langle n\rangle \frown \langle A\rangle \frown \langle B\rangle$ from 0. Suppose $N > 0$ and $\mathrm{Fusc}(N) = A\cdot \mathrm{Fusc}(n) + B\cdot \mathrm{Fusc}(n+1)$. Then

(i) s is halting,

(ii) $(\mathrm{Result}(s))(\mathbf{d}_3) = \mathrm{Fusc}(N)$,

(iii) if $n = 0$, then the complexity of $s = 1$, and

(iv) if $n > 0$, then the complexity of $s = 6\cdot(\lfloor \log_2 n\rfloor + 1) + 1$.

(6) Let N be a natural number. Suppose $N > 0$. Let s be a state with instruction counter on 0, with the program computing Fusc located from 0, and $\langle 2\rangle \frown \langle N\rangle \frown \langle 1\rangle \frown \langle 0\rangle$ from 0. Then

(i) s is halting,

(ii) $(\mathrm{Result}(s))(\mathbf{d}_3) = \mathrm{Fusc}(N)$,

(iii) if $N = 0$, then the complexity of $s = 1$, and

(iv) if $N > 0$, then the complexity of $s = 6\cdot(\lfloor \log_2 N\rfloor + 1) + 1$.

References

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

[2] 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`.

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

[4] Grzegorz Bancerek and Piotr Rudnicki. Two programs for **scm**. Part I - preliminaries. *Journal of Formalized Mathematics*, 5, 1993. `http://mizar.org/JFM/Vol5/pre_ff.html`.

[5] Edsger W. Dijkstra. *Selected Writings on Computing, a Personal Perspective*.

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

[7] Konrad Raczkowski and Andrzej Nędzusiak. Real exponents and logarithms. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/power.html`.

[8] 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`.

[1] The proposition (4) has been removed.

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

Received October 8, 1993

Published January 2, 2004