FORMALIZED MATHEMATICS
Volume 5, Number 1, 1996
Warsaw University - Białystok

A Compiler of Arithmetic Expressions for SCM [1]

Grzegorz Bancerek
Polish Academy of Sciences
Institute of Mathematics
Warsaw

Piotr Rudnicki
University of Alberta
Department of Computing Science
Edmonton

Summary. We define a set of binary arithmetic expressions with the following operations: $+$, $-$, $\cdot$, `mod`, and `div` and formalize the common meaning of the expressions in the set of integers. Then, we define a compile function that for a given expression results in a program for the **SCM** machine defined by Nakamura and Trybulec in [13]. We prove that the generated program when loaded into the machine and executed computes the value of the expression. The program uses additional memory and runs in time linear in length of the expression.

MML Identifier: `SCM_COMP`.

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

The following two propositions are true:

(1) Let I_1, I_2 be finite sequences of elements of the instructions of **SCM**, and let D be a finite sequence of elements of $\mathbb{Z}$, and let i_1, p_1, d_1 be natural numbers. Then every state with instruction counter on i_1, with $I_1 \frown I_2$ located from p_1, and D from d_1 is a state with instruction counter on i_1, with I_1 located from p_1, and D from d_1 and a state with instruction counter on i_1, with I_2 located from $p_1 + \operatorname{len} I_1$, and D from d_1.

(2) Let I_1, I_2 be finite sequences of elements of the instructions of **SCM**, and let i_1, p_1, d_1, k, i_2 be natural numbers, and let s be a state with instruction counter on i_1, with $I_1 \frown I_2$ located from p_1, and $\varepsilon_{\mathbb{Z}}$ from d_1, and let u be a state of **SCM**. Suppose $u = (\text{Computation}(s))(k)$ and

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

ISSN 0777–4028

$\mathbf{i}_{(i_2)} = \mathbf{IC}_u$. Then u is a state with instruction counter on i_2, with I_2 located from $p_1 + \operatorname{len} I_1$, and $\varepsilon_{\mathbb{Z}}$ from d_1.

The binary strict non empty tree construction structure $\mathrm{AE}_{\mathrm{SCM}}$ with terminals, nonterminals, and useful nonterminals is defined by the conditions (Def.1).

(Def.1) (i) The terminals of $\mathrm{AE}_{\mathrm{SCM}} = \text{Data-Loc}_{\mathrm{SCM}}$,

(ii) the nonterminals of $\mathrm{AE}_{\mathrm{SCM}} = [\!: 1, 5 :\!]$, and

(iii) for all symbols x, y, z of $\mathrm{AE}_{\mathrm{SCM}}$ holds $x \Rightarrow \langle y, z \rangle$ iff $x \in [\!: 1, 5 :\!]$.

A binary term is an element of $\mathrm{TS}(\mathrm{AE}_{\mathrm{SCM}})$.

Let n_1 be a nonterminal of $\mathrm{AE}_{\mathrm{SCM}}$ and let t_1, t_2 be binary terms. Then n_1-tree(t_1, t_2) is a binary term.

Let t be a terminal of $\mathrm{AE}_{\mathrm{SCM}}$. Then the root tree of t is a binary term.

Let t be a terminal of $\mathrm{AE}_{\mathrm{SCM}}$. The functor ${}^{@}t$ yielding a data-location is defined as follows:

(Def.2) ${}^{@}t = t$.

One can prove the following propositions:

(3) For every nonterminal n_1 of $\mathrm{AE}_{\mathrm{SCM}}$ holds $n_1 = \langle 0, 0 \rangle$ or $n_1 = \langle 0, 1 \rangle$ or $n_1 = \langle 0, 2 \rangle$ or $n_1 = \langle 0, 3 \rangle$ or $n_1 = \langle 0, 4 \rangle$.

(4) (i) $\langle 0, 0 \rangle$ is a nonterminal of $\mathrm{AE}_{\mathrm{SCM}}$,

(ii) $\langle 0, 1 \rangle$ is a nonterminal of $\mathrm{AE}_{\mathrm{SCM}}$,

(iii) $\langle 0, 2 \rangle$ is a nonterminal of $\mathrm{AE}_{\mathrm{SCM}}$,

(iv) $\langle 0, 3 \rangle$ is a nonterminal of $\mathrm{AE}_{\mathrm{SCM}}$, and

(v) $\langle 0, 4 \rangle$ is a nonterminal of $\mathrm{AE}_{\mathrm{SCM}}$.

Let t_3, t_4 be binary terms. The functor $t_3 + t_4$ yields a binary term and is defined as follows:

(Def.3) $t_3 + t_4 = \langle 0, 0 \rangle\text{-tree}(t_3, t_4)$.

The functor $t_3 - t_4$ yielding a binary term is defined as follows:

(Def.4) $t_3 - t_4 = \langle 0, 1 \rangle\text{-tree}(t_3, t_4)$.

The functor $t_3 \cdot t_4$ yields a binary term and is defined by:

(Def.5) $t_3 \cdot t_4 = \langle 0, 2 \rangle\text{-tree}(t_3, t_4)$.

The functor $t_3 \div t_4$ yields a binary term and is defined by:

(Def.6) $t_3 \div t_4 = \langle 0, 3 \rangle\text{-tree}(t_3, t_4)$.

The functor $t_3 \bmod t_4$ yielding a binary term is defined as follows:

(Def.7) $t_3 \bmod t_4 = \langle 0, 4 \rangle\text{-tree}(t_3, t_4)$.

We now state the proposition

(5) Let t_5 be a binary term. Then

(i) there exists a terminal t of $\mathrm{AE}_{\mathrm{SCM}}$ such that $t_5 =$ the root tree of t, or

(ii) there exist binary terms t_1, t_2 such that $t_5 = t_1 + t_2$ or $t_5 = t_1 - t_2$ or $t_5 = t_1 \cdot t_2$ or $t_5 = t_1 \div t_2$ or $t_5 = t_1 \bmod t_2$.

Let o be a nonterminal of $\mathrm{AE}_{\mathrm{SCM}}$ and let i, j be integers. The functor $o(i, j)$ yielding an integer is defined as follows:

(Def.8) (i) $o(i, j) = i + j$ if $o = \langle 0, 0 \rangle$,

(ii) $o(i, j) = i - j$ if $o = \langle 0, 1 \rangle$,

(iii) $o(i, j) = i \cdot j$ if $o = \langle 0, 2 \rangle$,

(iv) $o(i, j) = i \div j$ if $o = \langle 0, 3 \rangle$,

(v) $o(i, j) = i \bmod j$ if $o = \langle 0, 4 \rangle$.

Let s be a state of **SCM** and let t be a terminal of $\mathrm{AE}_{\mathrm{SCM}}$. Then $s(t)$ is an integer.

$\mathbb{Z}$ is a non empty subset of $\mathbb{R}$.

One can verify that every element of $\mathbb{Z}$ is integer.

Let D be a non empty set, let f be a function from $\mathbb{Z}$ into D, and let x be an integer. Then $f(x)$ is an element of D.

Let s be a state of **SCM** and let t_5 be a binary term. The functor $t_5 \ ^{@} \ s$ yields an integer and is defined by the condition (Def.9).

(Def.9) There exists a function f from $\mathrm{TS}(\mathrm{AE}_{\mathrm{SCM}})$ into $\mathbb{Z}$ such that

(i) $t_5 \ ^{@} \ s = f(t_5)$,

(ii) for every terminal t of $\mathrm{AE}_{\mathrm{SCM}}$ holds $f(\text{the root tree of } t) = s(t)$, and

(iii) for every nonterminal n_1 of $\mathrm{AE}_{\mathrm{SCM}}$ and for all binary terms t_1, t_2 and for all symbols r_1, r_2 of $\mathrm{AE}_{\mathrm{SCM}}$ such that $r_1 =$ the root label of t_1 and $r_2 =$ the root label of t_2 and $n_1 \Rightarrow \langle r_1, r_2 \rangle$ and for all elements x_1, x_2 of $\mathbb{Z}$ such that $x_1 = f(t_1)$ and $x_2 = f(t_2)$ holds $f(n_1\text{-tree}(t_1, t_2)) = n_1(x_1, x_2)$.

One can prove the following three propositions:

(6) For every state s of **SCM** and for every terminal t of $\mathrm{AE}_{\mathrm{SCM}}$ holds (the root tree of t) $^{@} \ s = s(t)$.

(7) For every state s of **SCM** and for every nonterminal n_1 of $\mathrm{AE}_{\mathrm{SCM}}$ and for all binary terms t_1, t_2 holds $(n_1\text{-tree}(t_1, t_2)) \ ^{@} \ s = n_1(t_1 \ ^{@} \ s, t_2 \ ^{@} \ s)$.

(8) Let s be a state of **SCM** and let t_1, t_2 be binary terms. Then $(t_1 + t_2) \ ^{@} \ s = (t_1 \ ^{@} \ s) + (t_2 \ ^{@} \ s)$ and $(t_1 - t_2) \ ^{@} \ s = (t_1 \ ^{@} \ s) - (t_2 \ ^{@} \ s)$ and $t_1 \cdot t_2 \ ^{@} \ s = (t_1 \ ^{@} \ s) \cdot (t_2 \ ^{@} \ s)$ and $(t_1 \div t_2) \ ^{@} \ s = (t_1 \ ^{@} \ s) \div (t_2 \ ^{@} \ s)$ and $(t_1 \bmod t_2) \ ^{@} \ s = (t_1 \ ^{@} \ s) \bmod (t_2 \ ^{@} \ s)$.

Let n_1 be a nonterminal of $\mathrm{AE}_{\mathrm{SCM}}$ and let n be a natural number. The functor $\mathrm{Selfwork}(n_1, n)$ yielding an element of (the instructions of **SCM qua** set)* is defined as follows:

(Def.10) (i) $\mathrm{Selfwork}(n_1, n) = \langle \mathrm{AddTo}(\mathbf{d}_n, \mathbf{d}_{n+1}) \rangle$ if $n_1 = \langle 0, 0 \rangle$,

(ii) $\mathrm{Selfwork}(n_1, n) = \langle \mathrm{SubFrom}(\mathbf{d}_n, \mathbf{d}_{n+1}) \rangle$ if $n_1 = \langle 0, 1 \rangle$,

(iii) $\mathrm{Selfwork}(n_1, n) = \langle \mathrm{MultBy}(\mathbf{d}_n, \mathbf{d}_{n+1}) \rangle$ if $n_1 = \langle 0, 2 \rangle$,

(iv) $\mathrm{Selfwork}(n_1, n) = \langle \mathrm{Divide}(\mathbf{d}_n, \mathbf{d}_{n+1}) \rangle$ if $n_1 = \langle 0, 3 \rangle$,

(v) $\mathrm{Selfwork}(n_1, n) = \langle \mathrm{Divide}(\mathbf{d}_n, \mathbf{d}_{n+1}), \mathbf{d}_n {:=} \mathbf{d}_{n+1} \rangle$ if $n_1 = \langle 0, 4 \rangle$.

Let t_5 be a binary term and let a_1 be a natural number. The functor $\mathrm{Compile}(t_5, a_1)$ yielding a finite sequence of elements of the instructions of **SCM** is defined by the condition (Def.11).

(Def.11) There exists a function f from $\mathrm{TS}(\mathrm{AE}_{\mathrm{SCM}})$ into ((the instructions of **SCM qua** set)*)$^{\mathbb{N}}$ such that

(i) $\mathrm{Compile}(t_5, a_1) = (f(t_5)$ **qua** element of ((the instructions of **SCM**

qua set)*)$^{\mathbb{N}}$)(a_1),

(ii) for every terminal t of $\mathrm{AE_{SCM}}$ there exists a function g from $\mathbb{N}$ into (the instructions of **SCM qua** set)* such that $g = f$(the root tree of t) and for every natural number n holds $g(n) = \langle \mathbf{d}_n{:=}^{@}t \rangle$, and

(iii) for every nonterminal n_1 of $\mathrm{AE_{SCM}}$ and for all binary terms t_3, t_4 and for all symbols r_1, r_2 of $\mathrm{AE_{SCM}}$ such that $r_1 =$ the root label of t_3 and $r_2 =$ the root label of t_4 and $n_1 \Rightarrow \langle r_1, r_2 \rangle$ there exist functions g, f_1, f_2 from $\mathbb{N}$ into (the instructions of **SCM qua** set)* such that $g = f(n_1\text{-tree}(t_3, t_4))$ and $f_1 = f(t_3)$ and $f_2 = f(t_4)$ and for every natural number n holds $g(n) = f_1(n) \frown f_2(n+1) \frown \mathrm{Selfwork}(n_1, n)$.

One can prove the following propositions:

(9) For every terminal t of $\mathrm{AE_{SCM}}$ and for every natural number n holds Compile(the root tree of t, n) $= \langle \mathbf{d}_n{:=}^{@}t \rangle$.

(10) Let n_1 be a nonterminal of $\mathrm{AE_{SCM}}$, and let t_3, t_4 be binary terms, and let n be a natural number, and let r_1, r_2 be symbols of $\mathrm{AE_{SCM}}$. Suppose $r_1 =$ the root label of t_3 and $r_2 =$ the root label of t_4 and $n_1 \Rightarrow \langle r_1, r_2 \rangle$. Then $\mathrm{Compile}(n_1\text{-tree}(t_3, t_4), n) = (\mathrm{Compile}(t_3, n)) \frown \mathrm{Compile}(t_4, n+1) \frown \mathrm{Selfwork}(n_1, n)$.

Let t be a terminal of $\mathrm{AE_{SCM}}$. The functor $\mathbf{d}^{-1}(t)$ yielding a natural number is defined as follows:

(Def.12) $\mathbf{d}_{\mathbf{d}^{-1}(t)} = t$.

Let n_2, n_3 be natural numbers. Then $\max(n_2, n_3)$ is a natural number.

Let t_5 be a binary term. The functor $\max_{\mathrm{DL}}(t_5)$ yielding a natural number is defined by the condition (Def.13).

(Def.13) There exists a function f from $\mathrm{TS}(\mathrm{AE_{SCM}})$ into $\mathbb{N}$ such that

(i) $\max_{\mathrm{DL}}(t_5) = f(t_5)$,

(ii) for every terminal t of $\mathrm{AE_{SCM}}$ holds f(the root tree of t) $= \mathbf{d}^{-1}(t)$, and

(iii) for every nonterminal n_1 of $\mathrm{AE_{SCM}}$ and for all binary terms t_1, t_2 and for all symbols r_1, r_2 of $\mathrm{AE_{SCM}}$ such that $r_1 =$ the root label of t_1 and $r_2 =$ the root label of t_2 and $n_1 \Rightarrow \langle r_1, r_2 \rangle$ and for all natural numbers x_1, x_2 such that $x_1 = f(t_1)$ and $x_2 = f(t_2)$ holds $f(n_1\text{-tree}(t_1, t_2)) = \max(x_1, x_2)$.

One can prove the following propositions:

(11) For every terminal t of $\mathrm{AE_{SCM}}$ holds $\max_{\mathrm{DL}}$(the root tree of t) $= \mathbf{d}^{-1}(t)$.

(12) For every nonterminal n_1 of $\mathrm{AE_{SCM}}$ and for all binary terms t_1, t_2 holds $\max_{\mathrm{DL}}(n_1\text{-tree}(t_1, t_2)) = \max(\max_{\mathrm{DL}}(t_1), \max_{\mathrm{DL}}(t_2))$.

(13) Let t_5 be a binary term and let s_1, s_2 be states of **SCM**. Suppose that for every natural number d_2 such that $d_2 \leq \max_{\mathrm{DL}}(t_5)$ holds $s_1(\mathbf{d}_{(d_2)}) = s_2(\mathbf{d}_{(d_2)})$. Then $t_5 \ {}^{@}\ s_1 = t_5 \ {}^{@}\ s_2$.

(14) Let t_5 be a binary term, and let a_1, n, k be natural numbers, and let s be a state with instruction counter on n, with $\mathrm{Compile}(t_5, a_1)$ located

from n, and $\varepsilon_{\mathbb{Z}}$ from k. Suppose $a_1 > \max_{\rm DL}(t_5)$. Then there exists a natural number i and there exists a state u of **SCM** such that

(i) $u = ({\rm Computation}(s))(i+1)$,

(ii) $i+1 = {\rm len\,Compile}(t_5, a_1)$,

(iii) $\mathbf{IC}_{({\rm Computation}(s))(i)} = \mathbf{i}_{n+i}$,

(iv) $\mathbf{IC}_u = \mathbf{i}_{n+(i+1)}$,

(v) $u(\mathbf{d}_{(a_1)}) = t_5\,^{@}\, s$, and

(vi) for every natural number d_2 such that $d_2 < a_1$ holds $s(\mathbf{d}_{(d_2)}) = u(\mathbf{d}_{(d_2)})$.

(15) Let t_5 be a binary term, and let a_1, n, k be natural numbers, and let s be a state with instruction counter on n, with $({\rm Compile}(t_5, a_1)) \frown \langle \mathbf{halt_{SCM}} \rangle$ located from n, and $\varepsilon_{\mathbb{Z}}$ from k. Suppose $a_1 > \max_{\rm DL}(t_5)$. Then s is halting and $({\rm Result}(s))(\mathbf{d}_{(a_1)}) = t_5\,^{@}\, s$ and the complexity of $s = {\rm len\,Compile}(t_5, a_1)$.

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.

[2] Grzegorz Bancerek. Joining of decorated trees. *Formalized Mathematics*, 4(**1**):77–82, 1993.

[3] Grzegorz Bancerek. König's lemma. *Formalized Mathematics*, 2(**3**):397–402, 1991.

[4] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.

[5] Grzegorz Bancerek and Piotr Rudnicki. Development of terminology for **SCM**. *Formalized Mathematics*, 4(**1**):61–67, 1993.

[6] Grzegorz Bancerek and Piotr Rudnicki. On defining functions on binary trees. *Formalized Mathematics*, 5(**1**):9–13, 1996.

[7] Grzegorz Bancerek and Piotr Rudnicki. On defining functions on trees. *Formalized Mathematics*, 4(**1**):91–101, 1993.

[8] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.

[9] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.

[10] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.

[11] Patricia L. Carlson and Grzegorz Bancerek. Context-free grammar - part 1. *Formalized Mathematics*, 2(**5**):683–687, 1991.

[12] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.

[13] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.

[14] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.

[15] Andrzej Trybulec. Function domains and Frænkel operator. *Formalized Mathematics*, 1(**3**):495–500, 1990.

[16] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.

[17] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.

[18] Andrzej Trybulec and Czesław Byliński. Some properties of real numbers. *Formalized Mathematics*, 1(**3**):445–449, 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] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received December 30, 1993