FORMALIZED MATHEMATICS
Volume 7, Number 1, 1998
University of Białystok

Another times Macro Instruction

Piotr Rudnicki[1]
University of Alberta
Edmonton

Summary. The semantics of the `times` macro is given in [2] only for the case when the body of the macro is parahalting. We remedy this by defining a new `times` macro instruction in terms of `while` (see [9, 13]). The semantics of the new `times` macro is given in a way analogous to the semantics of `while` macros. The new `times` uses an anonymous variable to control the number of its executions. We present two examples: a trivial one and a remake of the macro for the Fibonacci sequence (see [12]).

MML Identifier: `SFMASTR2`.

The terminology and notation used in this paper are introduced in the following articles: [11], [16], [21], [6], [8], [19], [5], [7], [10], [22], [3], [4], [1], [18], [17], [12], [14], [20], and [15].

1. $\mathbf{SCM}_{\mathrm{FSA}}$ Preliminaries

For simplicity, we follow the rules: s, s_1, s_2 denote states of $\mathbf{SCM}_{\mathrm{FSA}}$, a, b denote integer locations, d denotes a read-write integer location, f denotes a finite sequence location, I denotes a macro instruction, J denotes a good macro instruction, and k denotes a natural number.

One can prove the following propositions:

(1) If I is closed on Initialize(s) and halting on Initialize(s) and $b \notin$ UsedIntLoc(I), then $(\mathrm{IExec}(I,s))(b) = (\mathrm{Initialize}(s))(b)$.

(2) If I is closed on Initialize(s) and halting on Initialize(s) and $f \notin$ UsedInt* Loc(I), then $(\mathrm{IExec}(I,s))(f) = (\mathrm{Initialize}(s))(f)$.

[1]This work was partially supported by NSERC Grant OGP9207 and NATO CRG 951368.

ISSN 1426–2630

(3) Suppose I is closed on Initialize(s), halting on Initialize(s), and parahalting but $s(\mathrm{intloc}(0)) = 1$ or a is read-write but $a \notin \mathrm{UsedIntLoc}(I)$. Then $(\mathrm{IExec}(I, s))(a) = s(a)$.

(4) If $s(\mathrm{intloc}(0)) = 1$, then I is closed on s iff I is closed on Initialize(s).

(5) If $s(\mathrm{intloc}(0)) = 1$, then I is closed on s and halting on s iff I is closed on Initialize(s) and halting on Initialize(s).

(6) Let I_1 be a subset of Int-Locations and F_1 be a subset of FinSeq-Locations. Then $s_1 \restriction (I_1 \cup F_1) = s_2 \restriction (I_1 \cup F_1)$ if and only if the following conditions are satisfied:

(i) for every integer location x such that $x \in I_1$ holds $s_1(x) = s_2(x)$, and

(ii) for every finite sequence location x such that $x \in F_1$ holds $s_1(x) = s_2(x)$.

(7) Let I_1 be a subset of Int-Locations. Then $s_1 \restriction (I_1 \cup \text{FinSeq-Locations}) = s_2 \restriction (I_1 \cup \text{FinSeq-Locations})$ if and only if the following conditions are satisfied:

(i) for every integer location x such that $x \in I_1$ holds $s_1(x) = s_2(x)$, and

(ii) for every finite sequence location x holds $s_1(x) = s_2(x)$.

2. Another `times` Macro Instruction

Let a be an integer location and let I be a macro instruction. The functor $\mathrm{times}(a, I)$ yields a macro instruction and is defined by:

(Def. 1) $\mathrm{times}(a, I) = (a_1{:=}a);(\textbf{while } a_1 > 0 \textbf{ do } (I;\mathrm{SubFrom}(a_1, \mathrm{intloc}(0))))$, where $a_1 = 1^{\mathrm{st}}\text{-RWNotIn}(\{a\} \cup \mathrm{UsedIntLoc}(I))$.

We introduce $a \,\mathrm{times}\, I$ as a synonym of $\mathrm{times}(a, I)$.

Next we state two propositions:

(8) $\{b\} \cup \mathrm{UsedIntLoc}(I) \subseteq \mathrm{UsedIntLoc}(\mathrm{times}(b, I))$.

(9) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(\mathrm{times}(b, I)) = \mathrm{UsedInt}^{*}\,\mathrm{Loc}(I)$.

Let I be a good macro instruction and let a be an integer location. Observe that $\mathrm{times}(a, I)$ is good.

Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, let I be a macro instruction, and let a be an integer location. The functor $\mathrm{StepTimes}(a, I, s)$ yields a function from $\mathbb{N}$ into $\prod$(the object kind of $\mathbf{SCM}_{\mathrm{FSA}}$) and is defined by:

(Def. 2) $\mathrm{StepTimes}(a, I, s) = \mathit{StepWhile{>}0}(a_1, I;\mathrm{SubFrom}(a_1, \mathrm{intloc}(0)), \mathrm{Exec}(a_1{:=}a, \mathrm{Initialize}(s)))$, where $a_1 = 1^{\mathrm{st}}\text{-RWNotIn}(\{a\} \cup \mathrm{UsedIntLoc}(I))$.

Next we state several propositions:

(10) $(\mathrm{StepTimes}(a, J, s))(0)(\mathrm{intloc}(0)) = 1$.

(11) If $s(\mathrm{intloc}(0)) = 1$ or a is read-write, then $(\mathrm{StepTimes}(a, J, s))(0)(1^{\mathrm{st}}\text{-RWNotIn}(\{a\} \cup \mathrm{UsedIntLoc}(J))) = s(a)$.

(12) Suppose $(\mathrm{StepTimes}(a, J, s))(k)(\mathrm{intloc}(0)) = 1$ and J is closed on $(\mathrm{StepTimes}(a, J, s))(k)$ and halting on $(\mathrm{StepTimes}(a, J, s))(k)$. Then $(\mathrm{StepTimes}(a, J, s))(k + 1)(\mathrm{intloc}(0)) = 1$ and if $(\mathrm{StepTimes}(a, J, s))(k)(1^{\mathrm{st}}\text{-RWNotIn}(\{a\} \cup \mathrm{UsedIntLoc}(J))) > 0$, then $(\mathrm{StepTimes}(a, J, s))(k+1)(1^{\mathrm{st}}\text{-RWNotIn}(\{a\} \cup \mathrm{UsedIntLoc}(J))) = (\mathrm{StepTimes}(a, J, s))(k)(1^{\mathrm{st}}\text{-RWNotIn}(\{a\} \cup \mathrm{UsedIntLoc}(J))) - 1$.

(13) If $s(\mathrm{intloc}(0)) = 1$ or a is read-write, then $(\mathrm{StepTimes}(a, I, s))(0)(a) = s(a)$.

(14) $(\mathrm{StepTimes}(a, I, s))(0)(f) = s(f)$.

Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, let a be an integer location, and let I be a macro instruction. We say that ProperTimesBody a, I, s if and only if:

(Def. 3) For every natural number k such that $k < s(a)$ holds I is closed on $(\mathrm{StepTimes}(a, I, s))(k)$ and halting on $(\mathrm{StepTimes}(a, I, s))(k)$.

One can prove the following propositions:

(15) If I is parahalting, then ProperTimesBody a, I, s.

(16) If ProperTimesBody a, J, s, then for every k such that $k \leqslant s(a)$ holds $(\mathrm{StepTimes}(a, J, s))(k)(\mathrm{intloc}(0)) = 1$.

(17) Suppose $s(\mathrm{intloc}(0)) = 1$ or a is read-write but ProperTimesBody a, J, s. Let given k. If $k \leqslant s(a)$, then $(\mathrm{StepTimes}(a, J, s))(k)(1^{\mathrm{st}}\text{-RWNotIn}(\{a\} \cup \mathrm{UsedIntLoc}(J))) + k = s(a)$.

(18) Suppose ProperTimesBody a, J, s but $0 \leqslant s(a)$ but $s(\mathrm{intloc}(0)) = 1$ or a is read-write. Let given k. If $k \geqslant s(a)$, then $(\mathrm{StepTimes}(a, J, s))(k)(1^{\mathrm{st}}\text{-RWNotIn}(\{a\} \cup \mathrm{UsedIntLoc}(J))) = 0$ and $(\mathrm{StepTimes}(a, J, s))(k)(\mathrm{intloc}(0)) = 1$.

(19) If $s(\mathrm{intloc}(0)) = 1$, then $(\mathrm{StepTimes}(a, I, s))(0){\restriction}(\mathrm{UsedIntLoc}(I) \cup \text{FinSeq-Locations}) = s{\restriction}(\mathrm{UsedIntLoc}(I) \cup \text{FinSeq-Locations})$.

(20) Suppose $(\mathrm{StepTimes}(a, J, s))(k)(\mathrm{intloc}(0)) = 1$ and J is halting on $\mathrm{Initialize}((\mathrm{StepTimes}(a, J, s))(k))$ and closed on $\mathrm{Initialize}((\mathrm{StepTimes}(a, J, s))(k))$ and $(\mathrm{StepTimes}(a, J, s))(k)(1^{\mathrm{st}}\text{-RWNotIn}(\{a\} \cup \mathrm{UsedIntLoc}(J))) > 0$. Then $(\mathrm{StepTimes}(a, J, s))(k + 1){\restriction}(\mathrm{UsedIntLoc}(J) \cup \text{FinSeq-Locations}) = \mathrm{IExec}(J, (\mathrm{StepTimes}(a, J, s))(k)){\restriction}(\mathrm{UsedIntLoc}(J) \cup \text{FinSeq-Locations})$.

(21) Suppose ProperTimesBody a, J, s or J is parahalting but $k < s(a)$ but $s(\mathrm{intloc}(0)) = 1$ or a is read-write. Then $(\mathrm{StepTimes}(a, J, s))(k + 1){\restriction}(\mathrm{UsedIntLoc}(J) \cup \text{FinSeq-Locations}) = \mathrm{IExec}(J, (\mathrm{StepTimes}(a, J, s))(k)){\restriction}(\mathrm{UsedIntLoc}(J) \cup \text{FinSeq-Locations})$.

(22) If $s(a) \leqslant 0$ and $s(\mathrm{intloc}(0)) = 1$, then $\mathrm{IExec}(\mathrm{times}(a, I), s){\restriction}(\mathrm{UsedIntLoc}(I) \cup \text{FinSeq-Locations}) = s{\restriction}(\mathrm{UsedIntLoc}(I) \cup \text{FinSeq-Locations})$.

(23) Suppose $s(a) = k$ but ProperTimesBody a, J, s or J is parahalting but $s(\mathrm{intloc}(0)) = 1$ or a is read-write. Then $\mathrm{IExec}(\mathrm{times}(a, J), s)\restriction D = (\mathrm{StepTimes}(a, J, s))(k)\restriction D$, where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

(24) If $s(\mathrm{intloc}(0)) = 1$ and if ProperTimesBody a, J, s or J is parahalting, then $\mathrm{times}(a, J)$ is closed on s and $\mathrm{times}(a, J)$ is halting on s.

3. A Trivial Example

Let d be a read-write integer location. The functor triv-times(d) yields a macro instruction and is defined as follows:

(Def. 4) triv-times$(d) =$
  $\mathrm{times}(d, (\textbf{while } d = 0 \textbf{ do } \mathrm{Macro}(d{:=}d));$
  $\mathrm{SubFrom}(d, \mathrm{intloc}(0)))$.

One can prove the following propositions:

(25) If $s(d) \leqslant 0$, then $(\mathrm{IExec}(\text{triv-times}(d), s))(d) = s(d)$.

(26) If $0 \leqslant s(d)$, then $(\mathrm{IExec}(\text{triv-times}(d), s))(d) = 0$.

4. A Macro for the Fibonacci Sequence

Let N, r_1 be integer locations. The functor Fib-macro(N, r_1) yields a macro instruction and is defined by:

(Def. 5) Fib-macro$(N, r_1) =$
  $(N_1{:=}N);$
  $\mathrm{SubFrom}(r_1, r_1);$
  $(n_1{:=}\,\mathrm{intloc}(0));$
  $\mathrm{times}(N, \mathrm{AddTo}(r_1, n_1); \mathrm{swap}(r_1, n_1));$
  $(N{:=}N_1)$,
  where $N_1 = 1^{\mathrm{st}}$-NotUsed$(\mathrm{times}(N, \mathrm{AddTo}(r_1, n_1); \mathrm{swap}(r_1, n_1)))$ and $n_1 = 1^{\mathrm{st}}$-RWNotIn$(\{N, r_1\})$.

One can prove the following proposition

(27) Let N, r_1 be read-write integer locations. Suppose $N \neq r_1$. Let n be a natural number. If $n = s(N)$, then $(\mathrm{IExec}(\text{Fib-macro}(N, r_1), s))(r_1) = \mathrm{Fib}(n)$ and $(\mathrm{IExec}(\text{Fib-macro}(N, r_1), s))(N) = s(N)$.

References

[1] Noriko Asamoto. Constant assignment macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Part II. *Formalized Mathematics*, 6(**1**):59–63, 1997.

[2] Noriko Asamoto. The `loop` and `Times` macroinstruction for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 6(**4**):483–497, 1997.

[3] Noriko Asamoto, Yatsuka Nakamura, Piotr Rudnicki, and Andrzej Trybulec. On the composition of macro instructions. Part II. *Formalized Mathematics*, 6(**1**):41–47, 1997.

[4] Noriko Asamoto, Yatsuka Nakamura, Piotr Rudnicki, and Andrzej Trybulec. On the composition of macro instructions. Part III. *Formalized Mathematics*, 6(**1**):53–57, 1997.

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

[6] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.

[7] Grzegorz Bancerek and Piotr Rudnicki. Two programs for **scm**. Part I - preliminaries. *Formalized Mathematics*, 4(**1**):69–72, 1993.

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

[9] Jing-Chao Chen. While macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 6(**4**):553–561, 1997.

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

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

[12] Piotr Rudnicki. On the composition of non-parahalting macro instructions. *Formalized Mathematics*, 7(**1**):87–92, 1998.

[13] Piotr Rudnicki. The `while` macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Part II. *Formalized Mathematics*, 7(**1**):93–100, 1998.

[14] Piotr Rudnicki and Andrzej Trybulec. Memory handling for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 6(**1**):29–36, 1997.

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

[16] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.

[17] Andrzej Trybulec and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):571–576, 1996.

[18] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\mathrm{FSA}}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.

[19] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.

[20] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.

[21] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.

[22] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received June 4, 1998