JOURNAL OF FORMALIZED MATHEMATICS
Volume 13, Released 2001, Published 2003
Inst. of Computer Science, Univ. of Białystok

On State Machines of Calculating Type

Hisayoshi Kunimune
Shinshu University
Nagano

Grzegorz Bancerek
Białystok Technical University

Yatsuka Nakamura
Shinshu University
Nagano

Summary. In this article, we show the properties of the calculating type state machines. In the first section, we have defined calculating type state machines of which the state transition only depends on the first input. We have also proved theorems of the state machines. In the second section, we defined Moore machines with final states. We also introduced the concept of result of the Moore machines. In the last section, we proved the correctness of several calculating type of Moore machines.

MML Identifier: `FSM_2`.

WWW: `http://mizar.org/JFM/Vol13/fsm_2.html`

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

1. Calculating Type

For simplicity, we use the following convention: x, y are real numbers, i, j are non empty natural numbers, I, O are non empty sets, s, s_1, s_2, s_3 are elements of I, w, w_1, w_2 are finite sequences of elements of I, t is an element of O, S is a non empty FSM over I, and q, q_1 are states of S.

Let us consider I, S, q, w. We introduce $\mathrm{GEN}(w,q)$ as a synonym of (q,w)-admissible.

Let us consider I, S, q, w. One can verify that $\mathrm{GEN}(w,q)$ is non empty.

One can prove the following three propositions:

(1) $\mathrm{GEN}(\langle s\rangle,q) = \langle q, (\text{the transition of } S)(\langle\!\langle q, s\rangle\!\rangle)\rangle$.

(2) $\mathrm{GEN}(\langle s_1,s_2\rangle,q) = \langle q, (\text{the transition of } S)(\langle\!\langle q, s_1\rangle\!\rangle), (\text{the transition of } S)(\langle\!\langle(\text{the transition of } S)(\langle\!\langle q, s_1\rangle\!\rangle), s_2\rangle\!\rangle)\rangle$.

(3) $\mathrm{GEN}(\langle s_1,s_2,s_3\rangle,q) = \langle q, (\text{the transition of } S)(\langle\!\langle q, s_1\rangle\!\rangle), (\text{the transition of } S)(\langle\!\langle(\text{the transition of } S)(\langle\!\langle q, s_1\rangle\!\rangle), s_2\rangle\!\rangle), (\text{the transition of } S)(\langle\!\langle(\text{the transition of } S)(\langle\!\langle(\text{the transition of } S)(\langle\!\langle q, s_1\rangle\!\rangle), s_2\rangle\!\rangle), s_3\rangle\!\rangle)\rangle$.

Let us consider I, S. We say that S is calculating type if and only if the condition (Def. 1) is satisfied.

(Def. 1) Let given j and given w_1, w_2. Suppose $w_1(1) = w_2(1)$ and $j \leq \mathrm{len}\, w_1 + 1$ and $j \leq \mathrm{len}\, w_2 + 1$. Then $(\mathrm{GEN}(w_1, \text{the initial state of } S))(j) = (\mathrm{GEN}(w_2, \text{the initial state of } S))(j)$.

© Association of Mizar Users

We now state four propositions:

(4) Suppose S is calculating type. Let given w_1, w_2. Suppose $w_1(1) = w_2(1)$. Then $\mathrm{GEN}(w_1,$the initial state of $S)$ and $\mathrm{GEN}(w_2,$the initial state of $S)$ are $\subseteq$-comparable.

(5) Suppose that for all w_1, w_2 such that $w_1(1) = w_2(1)$ holds $\mathrm{GEN}(w_1,$the initial state of $S)$ and $\mathrm{GEN}(w_2,$the initial state of $S)$ are $\subseteq$-comparable. Then S is calculating type.

(6) Suppose S is calculating type. Let given w_1, w_2. Suppose $w_1(1) = w_2(1)$ and $\mathrm{len}\, w_1 = \mathrm{len}\, w_2$. Then $\mathrm{GEN}(w_1,$the initial state of $S) = \mathrm{GEN}(w_2,$the initial state of $S)$.

(7) Suppose that for all w_1, w_2 such that $w_1(1) = w_2(1)$ and $\mathrm{len}\, w_1 = \mathrm{len}\, w_2$ holds $\mathrm{GEN}(w_1,$the initial state of $S) = \mathrm{GEN}(w_2,$the initial state of $S)$. Then S is calculating type.

Let us consider I, S, q, s. We say that q is accessible via s if and only if:

(Def. 2) There exists a finite sequence w of elements of I such that the initial state of $S \xrightarrow{\langle s\rangle \frown w} q$.

Let us consider I, S, q. We say that q is accessible if and only if:

(Def. 3) There exists a finite sequence w of elements of I such that the initial state of $S \xrightarrow{w} q$.

One can prove the following propositions:

(8) If q is accessible via s, then q is accessible.

(9) If q is accessible and $q \neq$ the initial state of S, then there exists s such that q is accessible via s.

(10) The initial state of S is accessible.

(11) Suppose S is calculating type and q is accessible via s. Then there exists a non empty natural number m such that for every w if $\mathrm{len}\, w + 1 \geq m$ and $w(1) = s$, then $q = (\mathrm{GEN}(w,$the initial state of $S))(m)$ and for every i such that $i < m$ holds $(\mathrm{GEN}(w,$the initial state of $S))(i) \neq q$.

Let us consider I, S. We say that S is regular if and only if:

(Def. 4) Every state of S is accessible.

Next we state several propositions:

(12) If for all s_1, s_2, q holds (the transition of S)$(\langle q, s_1\rangle) =$ (the transition of S)$(\langle q, s_2\rangle)$, then S is calculating type.

(13) Let given S. Suppose that

(i) for all s_1, s_2, q such that $q \neq$ the initial state of S holds (the transition of S)$(\langle q, s_1\rangle) =$ (the transition of S)$(\langle q, s_2\rangle)$, and

(ii) for all s, q_1 holds (the transition of S)$(\langle q_1, s\rangle) \neq$ the initial state of S.

Then S is calculating type.

(14) Suppose S is regular and calculating type. Let given s_1, s_2, q. If $q \neq$ the initial state of S, then $(\mathrm{GEN}(\langle s_1\rangle, q))(2) = (\mathrm{GEN}(\langle s_2\rangle, q))(2)$.

(15) Suppose S is regular and calculating type. Let given s_1, s_2, q. Suppose $q \neq$ the initial state of S. Then (the transition of S)$(\langle q, s_1\rangle) =$ (the transition of S)$(\langle q, s_2\rangle)$.

(16) Suppose S is regular and calculating type. Let given s, s_1, s_2, q. Suppose (the transition of S)$(\langle$the initial state of S, $s_1\rangle) \neq$ (the transition of S)$(\langle$the initial state of S, $s_2\rangle)$. Then (the transition of S)$(\langle q, s\rangle) \neq$ the initial state of S.

2. STATE MACHINE WITH FINAL STATES

Let I be a set. We introduce state machines over I with final states which are extensions of FSM over I and are systems

$\langle$ a carrier, a transition, an initial state, final states $\rangle$,

where the carrier is a set, the transition is a function from $[\!:$ the carrier, $I:\!]$ into the carrier, the initial state is an element of the carrier, and the final states constitute a subset of the carrier.

Let I be a set. Observe that there exists a state machine over I with final states which is non empty.

Let us consider I, s and let S be a non empty state machine over I with final states. We say that s leads to final state of S if and only if:

(Def. 5) There exists a state q of S such that q is accessible via s and $q \in$ the final states of S.

Let us consider I and let S be a non empty state machine over I with final states. We say that S is halting if and only if:

(Def. 6) s leads to final state of S.

Let I be a set and let O be a non empty set. We consider Moore state machines over I and O with final states as extensions of state machine over I with final states and Moore-FSM over I, O as systems

$\langle$ a carrier, a transition, an output function, an initial state, final states $\rangle$,

where the carrier is a set, the transition is a function from $[\!:$ the carrier, $I:\!]$ into the carrier, the output function is a function from the carrier into O, the initial state is an element of the carrier, and the final states constitute a subset of the carrier.

Let I be a set and let O be a non empty set. One can verify that there exists a Moore state machine over I and O with final states which is non empty and strict.

Let us consider I, O, let i, f be sets, and let o be a function from $\{i, f\}$ into O. The functor I-TwoStatesMooreSM(i, f, o) yields a non empty strict Moore state machine over I and O with final states and is defined by the conditions (Def. 7).

(Def. 7)(i) The carrier of I-TwoStatesMooreSM$(i, f, o) = \{i, f\}$,

(ii) the transition of I-TwoStatesMooreSM$(i, f, o) = [\!: \{i, f\}, I :\!] \longmapsto f$,

(iii) the output function of I-TwoStatesMooreSM$(i, f, o) = o$,

(iv) the initial state of I-TwoStatesMooreSM$(i, f, o) = i$, and

(v) the final states of I-TwoStatesMooreSM$(i, f, o) = \{f\}$.

Next we state the proposition

(17) Let i, f be sets, o be a function from $\{i, f\}$ into O, and given j. If $1 < j$ and $j \leq \operatorname{len} w + 1$, then (GEN($w$, the initial state of I-TwoStatesMooreSM(i, f, o)))$(j) = f$.

Let us consider I, O, let i, f be sets, and let o be a function from $\{i, f\}$ into O. Observe that I-TwoStatesMooreSM(i, f, o) is calculating type.

Let us consider I, O, let i, f be sets, and let o be a function from $\{i, f\}$ into O. Note that I-TwoStatesMooreSM(i, f, o) is halting.

In the sequel n, m are non empty natural numbers and M is a non empty Moore state machine over I and O with final states.

One can prove the following proposition

(18) Suppose that

(i) M is calculating type,

(ii) s leads to final state of M, and

(iii) the initial state of $M \notin$ the final states of M.

Then there exists a non empty natural number m such that

(iv) for every w such that $\operatorname{len} w + 1 \geq m$ and $w(1) = s$ holds (GEN(w, the initial state of M))(m) $\in$ the final states of M, and

(v) for all w, j such that $j \leq \operatorname{len} w + 1$ and $w(1) = s$ and $j < m$ holds (GEN(w, the initial state of M))(j) $\notin$ the final states of M.

3. Correctness of a Result of State Machine

Let us consider I, O, M, s and let t be a set. We say that t is a result of s in M if and only if the condition (Def. 8) is satisfied.

(Def. 8) There exists m such that for every w if $w(1) = s$, then if $m \leq \operatorname{len} w + 1$, then $t =$ (the output function of M)((GEN(w, the initial state of M))(m)) and (GEN(w, the initial state of M))(m) $\in$ the final states of M and for every n such that $n < m$ and $n \leq \operatorname{len} w + 1$ holds (GEN(w, the initial state of M))(n) $\notin$ the final states of M.

One can prove the following propositions:

(19) Suppose the initial state of $M \in$ the final states of M. Then (the output function of M)(the initial state of M) is a result of s in M.

(20) Suppose that

(i) M is calculating type,

(ii) s leads to final state of M, and

(iii) the initial state of $M \notin$ the final states of M.

Then there exists t which is a result of s in M.

(21) Suppose M is calculating type and s leads to final state of M. Let t_1, t_2 be elements of O. If t_1 is a result of s in M and t_2 is a result of s in M, then $t_1 = t_2$.

(22) Let X be a non empty set, f be a binary operation on X, and M be a non empty Moore state machine over $[:X, X:]$ and $X \cup \{X\}$ with final states. Suppose that

(i) M is calculating type,

(ii) the carrier of $M = X \cup \{X\}$,

(iii) the final states of $M = X$,

(iv) the initial state of $M = X$,

(v) the output function of $M = \mathrm{id}_{\text{the carrier of } M}$, and

(vi) for all elements x, y of X holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle$) $= f(x, y)$.

Then M is halting and for all elements x, y of X holds $f(x, y)$ is a result of $\langle x, y\rangle$ in M.

(23) Let M be a non empty Moore state machine over $[:\mathbb{R}, \mathbb{R}:]$ and $\mathbb{R} \cup \{\mathbb{R}\}$ with final states. Suppose that M is calculating type and the carrier of $M = \mathbb{R} \cup \{\mathbb{R}\}$ and the final states of $M = \mathbb{R}$ and the initial state of $M = \mathbb{R}$ and the output function of $M = \mathrm{id}_{\text{the carrier of } M}$ and for all x, y such that $x \geq y$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle$) $= x$ and for all x, y such that $x < y$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle$) $= y$. Let x, y be elements of $\mathbb{R}$. Then $\max(x, y)$ is a result of $\langle x, y\rangle$ in M.

(24) Let M be a non empty Moore state machine over $[:\mathbb{R}, \mathbb{R}:]$ and $\mathbb{R} \cup \{\mathbb{R}\}$ with final states. Suppose that M is calculating type and the carrier of $M = \mathbb{R} \cup \{\mathbb{R}\}$ and the final states of $M = \mathbb{R}$ and the initial state of $M = \mathbb{R}$ and the output function of $M = \mathrm{id}_{\text{the carrier of } M}$ and for all x, y such that $x < y$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle$) $= x$ and for all x, y such that $x \geq y$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle$) $= y$. Let x, y be elements of $\mathbb{R}$. Then $\min(x, y)$ is a result of $\langle x, y\rangle$ in M.

(25) Let M be a non empty Moore state machine over $[:\mathbb{R}, \mathbb{R}:]$ and $\mathbb{R} \cup \{\mathbb{R}\}$ with final states. Suppose that

(i) M is calculating type,

(ii) the carrier of $M = \mathbb{R} \cup \{\mathbb{R}\}$,

(iii) the final states of $M = \mathbb{R}$,

(iv) the initial state of $M = \mathbb{R}$,

(v) the output function of $M = \mathrm{id}_{\text{the carrier of } M}$, and

(vi) for all x, y holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle) = x + y$.

Let x, y be elements of $\mathbb{R}$. Then $x + y$ is a result of $\langle x, y\rangle$ in M.

(26) Let M be a non empty Moore state machine over $[:\mathbb{R}, \mathbb{R}:]$ and $\mathbb{R} \cup \{\mathbb{R}\}$ with final states. Suppose that M is calculating type and the carrier of $M = \mathbb{R} \cup \{\mathbb{R}\}$ and the final states of $M = \mathbb{R}$ and the initial state of $M = \mathbb{R}$ and the output function of $M = \mathrm{id}_{\text{the carrier of } M}$ and for all x, y such that $x > 0$ or $y > 0$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle) = 1$ and for all x, y such that $x = 0$ or $y = 0$ but $x \leq 0$ but $y \leq 0$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle) = 0$ and for all x, y such that $x < 0$ and $y < 0$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle) = -1$. Let x, y be elements of $\mathbb{R}$. Then $\max(\mathrm{sgn}\, x, \mathrm{sgn}\, y)$ is a result of $\langle x, y\rangle$ in M.

Let us consider I, O. One can check that there exists a non empty Moore state machine over I and O with final states which is calculating type and halting.

Let us consider I. Observe that there exists a non empty state machine over I with final states which is calculating type and halting.

Let us consider I, O, let M be a calculating type halting non empty Moore state machine over I and O with final states, and let us consider s. The functor $\mathrm{Result}(s, M)$ yielding an element of O is defined by:

(Def. 9) $\mathrm{Result}(s, M)$ is a result of s in M.

We now state several propositions:

(27) For every function f from $\{0, 1\}$ into O holds $\mathrm{Result}(s, I\text{-TwoStatesMooreSM}(0, 1, f)) = f(1)$.

(28) Let M be a calculating type halting non empty Moore state machine over $[:\mathbb{R}, \mathbb{R}:]$ and $\mathbb{R} \cup \{\mathbb{R}\}$ with final states. Suppose that

(i) the carrier of $M = \mathbb{R} \cup \{\mathbb{R}\}$,

(ii) the final states of $M = \mathbb{R}$,

(iii) the initial state of $M = \mathbb{R}$,

(iv) the output function of $M = \mathrm{id}_{\text{the carrier of } M}$,

(v) for all x, y such that $x \geq y$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle) = x$, and

(vi) for all x, y such that $x < y$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle) = y$.

Let x, y be elements of $\mathbb{R}$. Then $\mathrm{Result}(\langle x, y\rangle, M) = \max(x, y)$.

(29) Let M be a calculating type halting non empty Moore state machine over $[:\mathbb{R}, \mathbb{R}:]$ and $\mathbb{R} \cup \{\mathbb{R}\}$ with final states. Suppose that

(i) the carrier of $M = \mathbb{R} \cup \{\mathbb{R}\}$,

(ii) the final states of $M = \mathbb{R}$,

(iii) the initial state of $M = \mathbb{R}$,

(iv) the output function of $M = \mathrm{id}_{\text{the carrier of } M}$,

(v) for all x, y such that $x < y$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle) = x$, and

(vi) for all x, y such that $x \geq y$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle) = y$.

Let x, y be elements of $\mathbb{R}$. Then Result$(\langle x, y\rangle, M) = \min(x, y)$.

(30) Let M be a calculating type halting non empty Moore state machine over $[:\mathbb{R}, \mathbb{R}:]$ and $\mathbb{R} \cup \{\mathbb{R}\}$ with final states. Suppose that

(i) the carrier of $M = \mathbb{R} \cup \{\mathbb{R}\}$,

(ii) the final states of $M = \mathbb{R}$,

(iii) the initial state of $M = \mathbb{R}$,

(iv) the output function of $M = \mathrm{id}_{\text{the carrier of } M}$, and

(v) for all x, y holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle) = x + y$.

Let x, y be elements of $\mathbb{R}$. Then Result$(\langle x, y\rangle, M) = x + y$.

(31) Let M be a calculating type halting non empty Moore state machine over $[:\mathbb{R}, \mathbb{R}:]$ and $\mathbb{R} \cup \{\mathbb{R}\}$ with final states. Suppose that the carrier of $M = \mathbb{R} \cup \{\mathbb{R}\}$ and the final states of $M = \mathbb{R}$ and the initial state of $M = \mathbb{R}$ and the output function of $M = \mathrm{id}_{\text{the carrier of } M}$ and for all x, y such that $x > 0$ or $y > 0$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle) = 1$ and for all x, y such that $x = 0$ or $y = 0$ but $x \leq 0$ but $y \leq 0$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle) = 0$ and for all x, y such that $x < 0$ and $y < 0$ holds (the transition of M)($\langle$the initial state of M, $\langle x, y\rangle\rangle) = -1$. Let x, y be elements of $\mathbb{R}$. Then Result$(\langle x, y\rangle, M) = \max(\mathrm{sgn}\, x, \mathrm{sgn}\, y)$.

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] Czesław Byliński. Binary operations. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/binop_1.html`.

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

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

[6] Czesław Byliński. Partial functions. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/partfun1.html`.

[7] Czesław Byliński. Some basic properties of sets. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/zfmisc_1.html`.

[8] Jing-Chao Chen. A small computer model with push-down stack. *Journal of Formalized Mathematics*, 11, 1999. `http://mizar.org/JFM/Vol11/scmpds_1.html`.

[9] Krzysztof Hryniewiecki. Basic properties of real numbers. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/real_1.html`.

[10] Miroslava Kaloper and Piotr Rudnicki. Minimization of finite state machines. *Journal of Formalized Mathematics*, 6, 1994. `http://mizar.org/JFM/Vol6/fsm_1.html`.

[11] Jan Popiołek. Some properties of functions modul and signum. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/absvalue.html`.

[12] Andrzej Trybulec. Binary operations applied to functions. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/funcop_1.html`.

[13] Andrzej Trybulec. Domains and their Cartesian products. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/domain_1.html`.

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

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

[16] Andrzej Trybulec and Czesław Byliński. Some properties of real numbers operations: min, max, square, and square root. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/square_1.html`.

[17] Wojciech A. Trybulec. Pigeon hole principle. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/finseq_4.html`.

[18] Zinaida Trybulec. Properties of subsets. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/subset_1.html`.

Received December 3, 2001

Published January 2, 2004