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

Introduction to Turing Machines

Jing-Chao Chen
Bell Labs Research China
Lucent Technologies
Bejing

Yatsuka Nakamura
Shinshu University
Nagano

Summary. A Turing machine can be viewed as a simple kind of computer, whose operations are constrainted to reading and writing symbols on a tape, or moving along the tape to the left or right. In theory, one has proven that the computability of Turing machines is equivalent to recursive functions. This article defines and verifies the Turing machines of summation and three primitive functions which are successor, zero and project functions. It is difficult to compute sophisticated functions by simple Turing machines. Therefore, we define the combination of two Turing machines.

MML Identifier: `TURING_1`.

WWW: `http://mizar.org/JFM/Vol13/turing_1.html`

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

1. PRELIMINARIES

In this paper n, i, j, k are natural numbers.

Let A, B be non empty sets, let f be a function from A into B, and let g be a partial function from A to B. Then $f+\cdot g$ is a function from A into B.

Let X, Y be non empty sets, let a be an element of X, and let b be an element of Y. Then $a \dot{\longmapsto} b$ is a partial function from X to Y.

Let n be a natural number. The functor $\mathrm{Seg}_M n$ yielding a subset of $\mathbb{N}$ is defined as follows:

(Def. 1) $\mathrm{Seg}_M n = \{k : k \leq n\}$.

Let n be a natural number. One can verify that $\mathrm{Seg}_M n$ is finite and non empty.

One can prove the following propositions:

(1) $k \in \mathrm{Seg}_M n$ iff $k \leq n$.

(2) For every function f and for all sets x, y, z, u, v such that $u \neq x$ holds $(f+\cdot(\langle x, y\rangle \dot{\longmapsto} z))(\langle u, v\rangle) = f(\langle u, v\rangle)$.

(3) For every function f and for all sets x, y, z, u, v such that $v \neq y$ holds $(f+\cdot(\langle x, y\rangle \dot{\longmapsto} z))(\langle u, v\rangle) = f(\langle u, v\rangle)$.

In the sequel i_1, i_2, i_3, i_4 are elements of $\mathbb{Z}$.

One can prove the following propositions:

(4) $\sum\langle i_1, i_2\rangle = i_1 + i_2$.

(5) $\sum\langle i_1, i_2, i_3\rangle = i_1 + i_2 + i_3$.

(6) $\sum\langle i_1, i_2, i_3, i_4\rangle = i_1 + i_2 + i_3 + i_4$.

Let f be a finite sequence of elements of $\mathbb{N}$ and let i be a natural number. The functor $\mathrm{Prefix}(f, i)$ yielding a finite sequence of elements of $\mathbb{Z}$ is defined as follows:

(Def. 2) $\mathrm{Prefix}(f, i) = f \restriction \mathrm{Seg}\, i$.

The following propositions are true:

(7) For all natural numbers x_1, x_2 holds $\sum\mathrm{Prefix}(\langle x_1, x_2\rangle, 1) = x_1$ and $\sum\mathrm{Prefix}(\langle x_1, x_2\rangle, 2) = x_1 + x_2$.

(8) For all natural numbers x_1, x_2, x_3 holds $\sum\mathrm{Prefix}(\langle x_1, x_2, x_3\rangle, 1) = x_1$ and $\sum\mathrm{Prefix}(\langle x_1, x_2, x_3\rangle, 2) = x_1 + x_2$ and $\sum\mathrm{Prefix}(\langle x_1, x_2, x_3\rangle, 3) = x_1 + x_2 + x_3$.

2. Definitions and Terminology for Turing Machine

We consider Turing machine structures as systems

⟨ symbols, control states, a transition, an initial state, an accepting state ⟩,

where the symbols and the control states constitute finite non empty sets, the transition is a function from $[\!:$the control states, the symbols$:\!]$ into $[\!:$the control states, the symbols, $\{-1, 0, 1\}:\!]$, and the initial state and the accepting state are elements of the control states.

Let T be a Turing machine structure. A state of T is an element of the control states of T. A tape of T is an element of (the symbols of $T)^{\mathbb{Z}}$. A symbol of T is an element of the symbols of T.

Let T be a Turing machine structure, let t be a tape of T, let h be an integer, and let s be a symbol of T. The functor $\mathrm{Tape\text{-}Chg}(t, h, s)$ yields a tape of T and is defined by:

(Def. 3) $\mathrm{Tape\text{-}Chg}(t, h, s) = t {+\cdot} (h \mathbin{\dot{\longmapsto}} s)$.

Let T be a Turing machine structure. A State of T is an element of $[\!:$the control states of T, $\mathbb{Z}$, (the symbols of $T)^{\mathbb{Z}}:\!]$. A transition-source of T is an element of $[\!:$the control states of T, the symbols of $T:\!]$. A transition-target of T is an element of $[\!:$the control states of T, the symbols of T, $\{-1, 0, 1\}:\!]$.

Let T be a Turing machine structure and let g be a transition-target of T. The functor $\mathrm{offset}(g)$ yields an integer and is defined as follows:

(Def. 4) $\mathrm{offset}(g) = g_{\mathbf{3}}$.

Let T be a Turing machine structure and let s be a State of T. The functor $\mathrm{Head}(s)$ yields an integer and is defined as follows:

(Def. 5) $\mathrm{Head}(s) = s_{\mathbf{2}}$.

Let T be a Turing machine structure and let s be a State of T. The functor s-target yielding a transition-target of T is defined by:

(Def. 6) $s\text{-target} = (\text{the transition of } T)(\langle s_{\mathbf{1}}, (s_{\mathbf{3}}\ \mathbf{qua}\ \text{tape of } T)(\mathrm{Head}(s))\rangle)$.

Let T be a Turing machine structure and let s be a State of T. The functor $\mathrm{Following}(s)$ yields a State of T and is defined as follows:

(Def. 7) $\mathrm{Following}(s) = \begin{cases} \langle s\text{-target}_{\mathbf{1}}, \mathrm{Head}(s) + \mathrm{offset}(s\text{-target}), \mathrm{Tape\text{-}Chg}(s_{\mathbf{3}}, \mathrm{Head}(s), s\text{-target}_{\mathbf{2}})\rangle, & \text{if } s_{\mathbf{1}} \neq \text{the accepting state o} \\ s, & \text{otherwise.} \end{cases}$

Let T be a Turing machine structure and let s be a State of T. The functor $\mathrm{Computation}(s)$ yielding a function from $\mathbb{N}$ into $[\!:$the control states of T, $\mathbb{Z}$, (the symbols of $T)^{\mathbb{Z}}:\!]$ is defined by:

(Def. 8) $(\mathrm{Computation}(s))(0) = s$ and for every i holds $(\mathrm{Computation}(s))(i+1) = \mathrm{Following}((\mathrm{Computation}(s))(i))$.

In the sequel T is a Turing machine structure and s is a State of T.
We now state several propositions:

(9) Let T be a Turing machine structure and s be a State of T. If $s_{\mathbf{1}}$ = the accepting state of T, then $s = \mathrm{Following}(s)$.

(10) $(\mathrm{Computation}(s))(0) = s$.

(11) $(\mathrm{Computation}(s))(k+1) = \mathrm{Following}((\mathrm{Computation}(s))(k))$.

(12) $(\mathrm{Computation}(s))(1) = \mathrm{Following}(s)$.

(13) $(\mathrm{Computation}(s))(i+k) = (\mathrm{Computation}((\mathrm{Computation}(s))(i)))(k)$.

(14) If $i \leq j$ and $\mathrm{Following}((\mathrm{Computation}(s))(i)) = (\mathrm{Computation}(s))(i)$, then $(\mathrm{Computation}(s))(j) = (\mathrm{Computation}(s))(i)$.

(15) If $i \leq j$ and $(\mathrm{Computation}(s))(i)_{\mathbf{1}}$ = the accepting state of T, then $(\mathrm{Computation}(s))(j) = (\mathrm{Computation}(s))(i)$.

Let T be a Turing machine structure and let s be a State of T. We say that s is accepting if and only if:

(Def. 9) There exists k such that $(\mathrm{Computation}(s))(k)_{\mathbf{1}}$ = the accepting state of T.

Let T be a Turing machine structure and let s be a State of T. Let us assume that s is accepting. The functor $\mathrm{Result}(s)$ yields a State of T and is defined as follows:

(Def. 10) There exists k such that $\mathrm{Result}(s) = (\mathrm{Computation}(s))(k)$ and $(\mathrm{Computation}(s))(k)_{\mathbf{1}}$ = the accepting state of T.

One can prove the following proposition

(16) Let T be a Turing machine structure and s be a State of T. Suppose s is accepting. Then there exists a natural number k such that

(i) $(\mathrm{Computation}(s))(k)_{\mathbf{1}}$ = the accepting state of T,

(ii) $\mathrm{Result}(s) = (\mathrm{Computation}(s))(k)$, and

(iii) for every natural number i such that $i < k$ holds $(\mathrm{Computation}(s))(i)_{\mathbf{1}} \neq$ the accepting state of T.

Let A, B be non empty sets and let y be a set. Let us assume that $y \in B$. The functor $\mathrm{id}(A,B,y)$ yields a function from A into $[:A, B:]$ and is defined by:

(Def. 11) For every element x of A holds $(\mathrm{id}(A,B,y))(x) = \langle x, y \rangle$.

The function SumTran from $[:\mathrm{Seg}_M 5, \{0,1\}:]$ into $[:\mathrm{Seg}_M 5, \{0,1\}, \{-1,0,1\}:]$ is defined by:

(Def. 12) $\mathrm{SumTran} = \mathrm{id}([:\mathrm{Seg}_M 5, \{0,1\}:], \{-1,0,1\}, 0) + \cdot (\langle 0, 0\rangle \dot{\longmapsto} \langle 0, 0, 1\rangle) + \cdot (\langle 0, 1\rangle \dot{\longmapsto} \langle 1, 0, 1\rangle) + \cdot (\langle 1, 1\rangle \dot{\longmapsto} \langle 1, 1, 1\rangle) + \cdot (\langle 1, 0\rangle \dot{\longmapsto} \langle 2, 1, 1\rangle) + \cdot (\langle 2, 1\rangle \dot{\longmapsto} \langle 2, 1, 1\rangle) + \cdot (\langle 2, 0\rangle \dot{\longmapsto} \langle 3, 0, -1\rangle) + \cdot (\langle 3, 1\rangle \dot{\longmapsto} \langle 4, 0, -1\rangle) + \cdot (\langle 4, 1\rangle \dot{\longmapsto} \langle 4, 1, -1\rangle) + \cdot (\langle 4, 0\rangle \dot{\longmapsto} \langle 5, 0, 0\rangle)$.

Next we state the proposition

(17) $\mathrm{SumTran}(\langle 0, 0\rangle) = \langle 0, 0, 1\rangle$ and $\mathrm{SumTran}(\langle 0, 1\rangle) = \langle 1, 0, 1\rangle$ and $\mathrm{SumTran}(\langle 1, 1\rangle) = \langle 1, 1, 1\rangle$ and $\mathrm{SumTran}(\langle 1, 0\rangle) = \langle 2, 1, 1\rangle$ and $\mathrm{SumTran}(\langle 2, 1\rangle) = \langle 2, 1, 1\rangle$ and $\mathrm{SumTran}(\langle 2, 0\rangle) = \langle 3, 0, -1\rangle$ and $\mathrm{SumTran}(\langle 3, 1\rangle) = \langle 4, 0, -1\rangle$ and $\mathrm{SumTran}(\langle 4, 1\rangle) = \langle 4, 1, -1\rangle$ and $\mathrm{SumTran}(\langle 4, 0\rangle) = \langle 5, 0, 0\rangle$.

Let T be a Turing machine structure, let t be a tape of T, and let i, j be integers. We say that t is 1 between i, j if and only if:

(Def. 13) $t(i) = 0$ and $t(j) = 0$ and for every integer k such that $i < k$ and $k < j$ holds $t(k) = 1$.

Let f be a finite sequence of elements of $\mathbb{N}$, let T be a Turing machine structure, and let t be a tape of T. We say that t stores data f if and only if:

(Def. 14) For every natural number i such that $1 \leq i$ and $i < \operatorname{len} f$ holds t is 1 between $\sum \operatorname{Prefix}(f,i) + 2 \cdot (i-1)$, $\sum \operatorname{Prefix}(f,i+1) + 2 \cdot i$.

Next we state several propositions:

(18) Let T be a Turing machine structure, t be a tape of T, and s, n be natural numbers. If t stores data $\langle s,n \rangle$, then t is 1 between s, $s+n+2$.

(19) Let T be a Turing machine structure, t be a tape of T, and s, n be natural numbers. If t is 1 between s, $s+n+2$, then t stores data $\langle s,n \rangle$.

(20) Let T be a Turing machine structure, t be a tape of T, and s, n be natural numbers. Suppose t stores data $\langle s,n \rangle$. Then $t(s) = 0$ and $t(s+n+2) = 0$ and for every integer i such that $s < i$ and $i < s+n+2$ holds $t(i) = 1$.

(21) Let T be a Turing machine structure, t be a tape of T, and s, n_1, n_2 be natural numbers. Suppose t stores data $\langle s,n_1,n_2 \rangle$. Then t is 1 between s, $s+n_1+2$ and 1 between $s+n_1+2$, $s+n_1+n_2+4$.

(22) Let T be a Turing machine structure, t be a tape of T, and s, n_1, n_2 be natural numbers. Suppose t stores data $\langle s,n_1,n_2 \rangle$. Then

(i) $t(s) = 0$,

(ii) $t(s+n_1+2) = 0$,

(iii) $t(s+n_1+n_2+4) = 0$,

(iv) for every integer i such that $s < i$ and $i < s+n_1+2$ holds $t(i) = 1$, and

(v) for every integer i such that $s+n_1+2 < i$ and $i < s+n_1+n_2+4$ holds $t(i) = 1$.

(23) Let f be a finite sequence of elements of $\mathbb{N}$ and s be a natural number. If $\operatorname{len} f \geq 1$, then $\sum \operatorname{Prefix}(\langle s \rangle ^\frown f, 1) = s$ and $\sum \operatorname{Prefix}(\langle s \rangle ^\frown f, 2) = s + f_1$.

(24) Let f be a finite sequence of elements of $\mathbb{N}$ and s be a natural number. Suppose $\operatorname{len} f \geq 3$. Then $\sum \operatorname{Prefix}(\langle s \rangle ^\frown f, 1) = s$ and $\sum \operatorname{Prefix}(\langle s \rangle ^\frown f, 2) = s + f_1$ and $\sum \operatorname{Prefix}(\langle s \rangle ^\frown f, 3) = s + f_1 + f_2$ and $\sum \operatorname{Prefix}(\langle s \rangle ^\frown f, 4) = s + f_1 + f_2 + f_3$.

(25) Let T be a Turing machine structure, t be a tape of T, s be a natural number, and f be a finite sequence of elements of $\mathbb{N}$. If $\operatorname{len} f \geq 1$ and t stores data $\langle s \rangle ^\frown f$, then t is 1 between s, $s + f_1 + 2$.

(26) Let T be a Turing machine structure, t be a tape of T, s be a natural number, and f be a finite sequence of elements of $\mathbb{N}$. Suppose $\operatorname{len} f \geq 3$ and t stores data $\langle s \rangle ^\frown f$. Then t is 1 between s, $s+f_1+2$, 1 between $s+f_1+2$, $s+f_1+f_2+4$, and 1 between $s+f_1+f_2+4$, $s+f_1+f_2+f_3+6$.

3. Summation of Two Natural Numbers

The strict Turing machine structure SumTuring is defined by the conditions (Def. 15).

(Def. 15)(i) The symbols of SumTuring $= \{0,1\}$,

(ii) the control states of SumTuring $= \operatorname{Seg}_M 5$,

(iii) the transition of SumTuring $=$ SumTran,

(iv) the initial state of SumTuring $= 0$, and

(v) the accepting state of SumTuring $= 5$.

One can prove the following propositions:

(27) Let T be a Turing machine structure, s be a State of T, and p, h, t be sets. If $s = \langle p, h, t \rangle$, then $\mathrm{Head}(s) = h$.

(28) Let T be a Turing machine structure, t be a tape of T, h be an integer, and s be a symbol of T. If $t(h) = s$, then $\text{Tape-Chg}(t,h,s) = t$.

(29) Let T be a Turing machine structure, s be a State of T, and p, h, t be sets. Suppose $s = \langle p, h, t \rangle$ and $p \neq$ the accepting state of T. Then $\mathrm{Following}(s) = \langle s\text{-target}_{\mathbf{1}}, \mathrm{Head}(s) + \mathrm{offset}(s\text{-target}), \text{Tape-Chg}(s_{\mathbf{3}}, \mathrm{Head}(s), s\text{-target}_{\mathbf{2}}) \rangle$.

(30) Let T be a Turing machine structure, t be a tape of T, h be an integer, s be a symbol of T, and i be a set. Then $(\text{Tape-Chg}(t,h,s))(h) = s$ and if $i \neq h$, then $(\text{Tape-Chg}(t,h,s))(i) = t(i)$.

(31) Let s be a State of SumTuring, t be a tape of SumTuring, and h_1, n_1, n_2 be natural numbers. Suppose $s = \langle 0, h_1, t \rangle$ and t stores data $\langle h_1, n_1, n_2 \rangle$. Then s is accepting and $(\mathrm{Result}(s))_{\mathbf{2}} = 1 + h_1$ and $(\mathrm{Result}(s))_{\mathbf{3}}$ stores data $\langle 1 + h_1, n_1 + n_2 \rangle$.

Let T be a Turing machine structure and let F be a function. We say that T computes F if and only if the condition (Def. 16) is satisfied.

(Def. 16) Let s be a State of T, t be a tape of T, a be a natural number, and x be a finite sequence of elements of $\mathbb{N}$. Suppose $x \in \mathrm{dom}\, F$ and $s = \langle$the initial state of T, $a, t\rangle$ and t stores data $\langle a \rangle ^\frown x$. Then s is accepting and there exist natural numbers b, y such that $(\mathrm{Result}(s))_{\mathbf{2}} = b$ and $y = F(x)$ and $(\mathrm{Result}(s))_{\mathbf{3}}$ stores data $\langle b \rangle ^\frown \langle y \rangle$.

We now state two propositions:

(32) $\mathrm{dom}[+] \subseteq \mathbb{N}^2$.

(33) SumTuring computes $[+]$.

4. Computing Successor Function

The function SuccTran from $[: \mathrm{Seg}_M 4, \{0,1\} :]$ into $[: \mathrm{Seg}_M 4, \{0,1\}, \{-1,0,1\} :]$ is defined by:

(Def. 17) $\mathrm{SuccTran} = \mathrm{id}([: \mathrm{Seg}_M 4, \{0,1\} :], \{-1,0,1\}, 0) +\cdot (\langle 0, 0 \rangle \dot\longmapsto \langle 1, 0, 1 \rangle) +\cdot (\langle 1, 1 \rangle \dot\longmapsto \langle 1, 1, 1 \rangle) +\cdot (\langle 1, 0 \rangle \dot\longmapsto \langle 2, 1, 1 \rangle) +\cdot (\langle 2, 0 \rangle \dot\longmapsto \langle 3, 0, -1 \rangle) +\cdot (\langle 2, 1 \rangle \dot\longmapsto \langle 3, 0, -1 \rangle) +\cdot (\langle 3, 1 \rangle \dot\longmapsto \langle 3, 1, -1 \rangle) +\cdot (\langle 3, 0 \rangle \dot\longmapsto \langle 4, 0, 0 \rangle)$.

We now state the proposition

(34) $\mathrm{SuccTran}(\langle 0, 0 \rangle) = \langle 1, 0, 1 \rangle$ and $\mathrm{SuccTran}(\langle 1, 1 \rangle) = \langle 1, 1, 1 \rangle$ and $\mathrm{SuccTran}(\langle 1, 0 \rangle) = \langle 2, 1, 1 \rangle$ and $\mathrm{SuccTran}(\langle 2, 0 \rangle) = \langle 3, 0, -1 \rangle$ and $\mathrm{SuccTran}(\langle 2, 1 \rangle) = \langle 3, 0, -1 \rangle$ and $\mathrm{SuccTran}(\langle 3, 1 \rangle) = \langle 3, 1, -1 \rangle$ and $\mathrm{SuccTran}(\langle 3, 0 \rangle) = \langle 4, 0, 0 \rangle$.

The strict Turing machine structure SuccTuring is defined by the conditions (Def. 18).

(Def. 18)(i) The symbols of SuccTuring $= \{0, 1\}$,

(ii) the control states of SuccTuring $= \mathrm{Seg}_M 4$,

(iii) the transition of SuccTuring $=$ SuccTran,

(iv) the initial state of SuccTuring $= 0$, and

(v) the accepting state of SuccTuring $= 4$.

The following two propositions are true:

(36)[1] Let s be a State of SuccTuring, t be a tape of SuccTuring, and h_1, n be natural numbers. Suppose $s = \langle 0, h_1, t \rangle$ and t stores data $\langle h_1, n \rangle$. Then s is accepting and $(\mathrm{Result}(s))_{\mathbf{2}} = h_1$ and $(\mathrm{Result}(s))_{\mathbf{3}}$ stores data $\langle h_1, n + 1 \rangle$.

(37) SuccTuring computes $\mathrm{succ}_1(1)$.

[1] The proposition (35) has been removed.

5. Computing Zero Function

The function ZeroTran from $[:\mathrm{Seg}_M 4, \{0,1\}:]$ into $[:\mathrm{Seg}_M 4, \{0,1\}, \{-1,0,1\}:]$ is defined by:

(Def. 19) ZeroTran $=$ $\mathrm{id}([:\mathrm{Seg}_M 4, \{0,1\}:], \{-1,0,1\}, 1) +\cdot (\langle 0, 0\rangle \mathbin{\dot{\longmapsto}} \langle 1, 0, 1\rangle) +\cdot (\langle 1, 1\rangle \mathbin{\dot{\longmapsto}} \langle 2, 1, 1\rangle) +\cdot (\langle 2, 0\rangle \mathbin{\dot{\longmapsto}} \langle 3, 0, -1\rangle) +\cdot (\langle 2, 1\rangle \mathbin{\dot{\longmapsto}} \langle 3, 0, -1\rangle) +\cdot (\langle 3, 1\rangle \mathbin{\dot{\longmapsto}} \langle 4, 1, -1\rangle)$.

The following proposition is true

(38) $\mathrm{ZeroTran}(\langle 0, 0\rangle) = \langle 1, 0, 1\rangle$ and $\mathrm{ZeroTran}(\langle 1, 1\rangle) = \langle 2, 1, 1\rangle$ and $\mathrm{ZeroTran}(\langle 2, 0\rangle) = \langle 3, 0, -1\rangle$ and $\mathrm{ZeroTran}(\langle 2, 1\rangle) = \langle 3, 0, -1\rangle$ and $\mathrm{ZeroTran}(\langle 3, 1\rangle) = \langle 4, 1, -1\rangle$.

The strict Turing machine structure ZeroTuring is defined by the conditions (Def. 20).

(Def. 20)(i) The symbols of ZeroTuring $= \{0,1\}$,

(ii) the control states of ZeroTuring $= \mathrm{Seg}_M 4$,

(iii) the transition of ZeroTuring $=$ ZeroTran,

(iv) the initial state of ZeroTuring $= 0$, and

(v) the accepting state of ZeroTuring $= 4$.

Next we state two propositions:

(39) Let s be a State of ZeroTuring, t be a tape of ZeroTuring, h_1 be a natural number, and f be a finite sequence of elements of $\mathbb{N}$. Suppose $\mathrm{len}\, f \geq 1$ and $s = \langle 0, h_1, t\rangle$ and t stores data $\langle h_1\rangle \frown f$. Then s is accepting and $(\mathrm{Result}(s))_{\mathbf{2}} = h_1$ and $(\mathrm{Result}(s))_{\mathbf{3}}$ stores data $\langle h_1, 0\rangle$.

(40) If $n \geq 1$, then ZeroTuring computes $\mathrm{const}_n(0)$.

6. Computing n-ary Project Function

The function n-proj3Tran from $[:\mathrm{Seg}_M 3, \{0,1\}:]$ into $[:\mathrm{Seg}_M 3, \{0,1\}, \{-1,0,1\}:]$ is defined as follows:

(Def. 21) n-proj3Tran $=$ $\mathrm{id}([:\mathrm{Seg}_M 3, \{0,1\}:], \{-1,0,1\}, 0) +\cdot (\langle 0, 0\rangle \mathbin{\dot{\longmapsto}} \langle 1, 0, 1\rangle) +\cdot (\langle 1, 1\rangle \mathbin{\dot{\longmapsto}} \langle 1, 0, 1\rangle) +\cdot (\langle 1, 0\rangle \mathbin{\dot{\longmapsto}} \langle 2, 0, 1\rangle) +\cdot (\langle 2, 1\rangle \mathbin{\dot{\longmapsto}} \langle 2, 0, 1\rangle) +\cdot (\langle 2, 0\rangle \mathbin{\dot{\longmapsto}} \langle 3, 0, 0\rangle)$.

The following proposition is true

(41) n-proj3Tran$(\langle 0, 0\rangle) = \langle 1, 0, 1\rangle$ and n-proj3Tran$(\langle 1, 1\rangle) = \langle 1, 0, 1\rangle$ and n-proj3Tran$(\langle 1, 0\rangle) = \langle 2, 0, 1\rangle$ and n-proj3Tran$(\langle 2, 1\rangle) = \langle 2, 0, 1\rangle$ and n-proj3Tran$(\langle 2, 0\rangle) = \langle 3, 0, 0\rangle$.

The strict Turing machine structure n-proj3Turing is defined by the conditions (Def. 22).

(Def. 22)(i) The symbols of n-proj3Turing $= \{0,1\}$,

(ii) the control states of n-proj3Turing $= \mathrm{Seg}_M 3$,

(iii) the transition of n-proj3Turing $=$ n-proj3Tran,

(iv) the initial state of n-proj3Turing $= 0$, and

(v) the accepting state of n-proj3Turing $= 3$.

One can prove the following propositions:

(42) Let s be a State of n-proj3Turing, t be a tape of n-proj3Turing, h_1 be a natural number, and f be a finite sequence of elements of $\mathbb{N}$. Suppose $\mathrm{len}\, f \geq 3$ and $s = \langle 0, h_1, t\rangle$ and t stores data $\langle h_1\rangle \frown f$. Then s is accepting and $(\mathrm{Result}(s))_{\mathbf{2}} = h_1 + f_1 + f_2 + 4$ and $(\mathrm{Result}(s))_{\mathbf{3}}$ stores data $\langle h_1 + f_1 + f_2 + 4, f_3\rangle$.

(43) If $n \geq 3$, then n-proj3Turing computes $\mathrm{proj}_n(3)$.

7. Combining Two Turing Machines into One

Let t_1, t_2 be Turing machine structures. The functor $\mathrm{SeqStates}(t_1,t_2)$ yields a finite non empty set and is defined by the condition (Def. 23).

(Def. 23) $\mathrm{SeqStates}(t_1,t_2) = [:$the control states of t_1, $\{$the initial state of $t_2\}:] \cup [:\{$the accepting state of $t_1\}$, the control states of $t_2:]$.

We now state four propositions:

(44) Let t_1, t_2 be Turing machine structures. Then

(i) $\langle$the initial state of t_1, the initial state of $t_2\rangle \in \mathrm{SeqStates}(t_1,t_2)$, and

(ii) $\langle$the accepting state of t_1, the accepting state of $t_2\rangle \in \mathrm{SeqStates}(t_1,t_2)$.

(45) For all Turing machine structures s, t and for every state x of s holds $\langle x$, the initial state of $t\rangle \in \mathrm{SeqStates}(s,t)$.

(46) For all Turing machine structures s, t and for every state x of t holds $\langle$the accepting state of s, $x\rangle \in \mathrm{SeqStates}(s,t)$.

(47) Let s, t be Turing machine structures and x be an element of $\mathrm{SeqStates}(s,t)$. Then there exists a state x_1 of s and there exists a state x_2 of t such that $x = \langle x_1, x_2\rangle$.

Let s, t be Turing machine structures and let x be a transition-target of s. The functor $1^{\mathrm{st}}\mathrm{SeqTran}(s,t,x)$ yields an element of $[:\mathrm{SeqStates}(s,t)$, (the symbols of s) $\cup$ (the symbols of t), $\{-1,0,1\}:]$ and is defined by:

(Def. 24) $1^{\mathrm{st}}\mathrm{SeqTran}(s,t,x) = \langle\langle x_{\mathbf{1}}$, the initial state of $t\rangle, x_{\mathbf{2}}, x_{\mathbf{3}}\rangle$.

Let s, t be Turing machine structures and let x be a transition-target of t. The functor $2^{\mathrm{nd}}\mathrm{SeqTran}(s,t,x)$ yields an element of $[:\mathrm{SeqStates}(s,t)$, (the symbols of s) $\cup$ (the symbols of t), $\{-1,0,1\}:]$ and is defined as follows:

(Def. 25) $2^{\mathrm{nd}}\mathrm{SeqTran}(s,t,x) = \langle\langle$the accepting state of s, $x_{\mathbf{1}}\rangle, x_{\mathbf{2}}, x_{\mathbf{3}}\rangle$.

Let s, t be Turing machine structures and let x be an element of $\mathrm{SeqStates}(s,t)$. Then $x_{\mathbf{1}}$ is a state of s. Then $x_{\mathbf{2}}$ is a state of t.

Let s, t be Turing machine structures and let x be an element of $[:\mathrm{SeqStates}(s,t)$, (the symbols of s) $\cup$ (the symbols of t)$:]$. The functor $1^{\mathrm{st}}\mathrm{SeqState}\,x$ yields a state of s and is defined as follows:

(Def. 26) $1^{\mathrm{st}}\mathrm{SeqState}\,x = (x_{\mathbf{1}})_{\mathbf{1}}$.

The functor $2^{\mathrm{nd}}\mathrm{SeqState}\,x$ yields a state of t and is defined as follows:

(Def. 27) $2^{\mathrm{nd}}\mathrm{SeqState}\,x = (x_{\mathbf{1}})_{\mathbf{2}}$.

Let X, Y, Z be non empty sets and let x be an element of $[:X, Y \cup Z:]$. Let us assume that there exist a set u and an element y of Y such that $x = \langle u, y\rangle$. The functor $1^{\mathrm{st}}\mathrm{SeqSymbol}\,x$ yielding an element of Y is defined by:

(Def. 28) $1^{\mathrm{st}}\mathrm{SeqSymbol}\,x = x_{\mathbf{2}}$.

Let X, Y, Z be non empty sets and let x be an element of $[:X, Y \cup Z:]$. Let us assume that there exist a set u and an element z of Z such that $x = \langle u, z\rangle$. The functor $2^{\mathrm{nd}}\mathrm{SeqSymbol}\,x$ yielding an element of Z is defined as follows:

(Def. 29) $2^{\mathrm{nd}}\mathrm{SeqSymbol}\,x = x_{\mathbf{2}}$.

Let s, t be Turing machine structures and let x be an element of $[:\mathrm{SeqStates}(s,t)$, (the symbols of s) $\cup$ (the symbols of t)$:]$. The functor $\mathrm{SeqTran}(s,t,x)$ yielding an element of $[:\mathrm{SeqStates}(s,t)$, (the symbols of s) $\cup$ (the symbols of t), $\{-1,0,1\}:]$ is defined as follows:

(Def. 30) $\mathrm{SeqTran}(s,t,x) = \begin{cases} 1^{\mathrm{st}}\mathrm{SeqTran}(s,t,(\text{the transition of } s)(\langle 1^{\mathrm{st}}\mathrm{SeqState}\, x,\ 1^{\mathrm{st}}\mathrm{SeqSymbol}\, x\rangle)), & \text{if thereexists a state } p \text{ of } s \\ 2^{\mathrm{nd}}\mathrm{SeqTran}(s,t,(\text{the transition of } t)(\langle 2^{\mathrm{nd}}\mathrm{SeqState}\, x,\ 2^{\mathrm{nd}}\mathrm{SeqSymbol}\, x\rangle)), & \text{if thereexists a state } q \text{ of } t \\ \langle x_{\mathbf{1}}, x_{\mathbf{2}}, -1\rangle, & \text{otherwise.} \end{cases}$

Let s, t be Turing machine structures. The functor $\mathrm{SeqTran}(s,t)$ yielding a function from $[:\mathrm{SeqStates}(s,t), (\text{the symbols of } s) \cup (\text{the symbols of } t):]$ into $[:\mathrm{SeqStates}(s,t), (\text{the symbols of } s) \cup (\text{the symbols of } t), \{-1,0,1\}:]$ is defined by:

(Def. 31) For every element x of $[:\mathrm{SeqStates}(s,t), (\text{the symbols of } s) \cup (\text{the symbols of } t):]$ holds $(\mathrm{SeqTran}(s,t))(x) = \mathrm{SeqTran}(s,t,x)$.

Let T_1, T_2 be Turing machine structures. The functor $T_1; T_2$ yielding a strict Turing machine structure is defined by the conditions (Def. 32).

(Def. 32)(i) The symbols of $T_1; T_2 = (\text{the symbols of } T_1) \cup (\text{the symbols of } T_2)$,

(ii) the control states of $T_1; T_2 = \mathrm{SeqStates}(T_1, T_2)$,

(iii) the transition of $T_1; T_2 = \mathrm{SeqTran}(T_1, T_2)$,

(iv) the initial state of $T_1; T_2 = \langle$the initial state of T_1, the initial state of $T_2\rangle$, and

(v) the accepting state of $T_1; T_2 = \langle$the accepting state of T_1, the accepting state of $T_2\rangle$.

The following propositions are true:

(48) Let T_1, T_2 be Turing machine structures, g be a transition-target of T_1, p be a state of T_1, and y be a symbol of T_1. Suppose $p \neq$ the accepting state of T_1 and $g = (\text{the transition of } T_1)(\langle p, y\rangle)$. Then $(\text{the transition of } T_1; T_2)(\langle\langle p, \text{the initial state of } T_2\rangle, y\rangle) = \langle\langle g_{\mathbf{1}}, \text{the initial state of } T_2\rangle, g_{\mathbf{2}}, g_{\mathbf{3}}\rangle$.

(49) Let T_1, T_2 be Turing machine structures, g be a transition-target of T_2, q be a state of T_2, and y be a symbol of T_2. Suppose $g = (\text{the transition of } T_2)(\langle q, y\rangle)$. Then $(\text{the transition of } T_1; T_2)(\langle\langle \text{the accepting state of } T_1,\ q\rangle, y\rangle) = \langle\langle \text{the accepting state of } T_1,\ g_{\mathbf{1}}\rangle, g_{\mathbf{2}}, g_{\mathbf{3}}\rangle$.

(50) Let T_1, T_2 be Turing machine structures, s_1 be a State of T_1, h be a natural number, t be a tape of T_1, s_2 be a State of T_2, and s_3 be a State of $T_1; T_2$. Suppose that

(i) s_1 is accepting,

(ii) $s_1 = \langle$the initial state of T_1, $h, t\rangle$,

(iii) s_2 is accepting,

(iv) $s_2 = \langle$the initial state of T_2, $(\mathrm{Result}(s_1))_{\mathbf{2}}$, $(\mathrm{Result}(s_1))_{\mathbf{3}}\rangle$, and

(v) $s_3 = \langle$the initial state of $T_1; T_2, h, t\rangle$.

Then s_3 is accepting and $(\mathrm{Result}(s_3))_{\mathbf{2}} = (\mathrm{Result}(s_2))_{\mathbf{2}}$ and $(\mathrm{Result}(s_3))_{\mathbf{3}} = (\mathrm{Result}(s_2))_{\mathbf{3}}$.

(51) Let t_3, t_4 be Turing machine structures and t be a tape of t_3. If the symbols of $t_3 =$ the symbols of t_4, then t is a tape of $t_3; t_4$.

(52) Let t_3, t_4 be Turing machine structures and t be a tape of $t_3; t_4$. Suppose the symbols of $t_3 =$ the symbols of t_4. Then t is a tape of t_3 and a tape of t_4.

(53) Let f be a finite sequence of elements of $\mathbb{N}$, t_3, t_4 be Turing machine structures, t_1 be a tape of t_3, and t_2 be a tape of t_4. If $t_1 = t_2$ and t_1 stores data f, then t_2 stores data f.

(54) Let s be a State of ZeroTuring; SuccTuring, t be a tape of ZeroTuring, and h_1, n be natural numbers. Suppose $s = \langle\langle 0, 0\rangle, h_1, t\rangle$ and t stores data $\langle h_1, n\rangle$. Then s is accepting and $(\mathrm{Result}(s))_{\mathbf{2}} = h_1$ and $(\mathrm{Result}(s))_{\mathbf{3}}$ stores data $\langle h_1, 1\rangle$.

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. The ordinal numbers. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/ordinal1.html`.

[3] Grzegorz Bancerek. Sequences of ordinal numbers. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/ordinal2.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. The set of primitive recursive functions. *Journal of Formalized Mathematics*, 13, 2001. `http://mizar.org/JFM/Vol13/comput_1.html`.

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

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

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

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

[10] Czesław Byliński. A classical first order language. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/cqc_lang.html`.

[11] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/finseq_2.html`.

[12] 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`.

[13] 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`.

[14] Agata Darmochwał. Finite sets. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/finset_1.html`.

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

[16] Dariusz Surowik. Cyclic groups and some of their properties — part I. *Journal of Formalized Mathematics*, 3, 1991. `http://mizar.org/JFM/Vol3/gr_cy_1.html`.

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

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

[19] Andrzej Trybulec. Tuples, projections and Cartesian products. *Journal of Formalized Mathematics*, 1, 1989. `http://mizar.org/JFM/Vol1/mcart_1.html`.

[20] Andrzej Trybulec. Function domains and Frænkel operator. *Journal of Formalized Mathematics*, 2, 1990. `http://mizar.org/JFM/Vol2/fraenkel.html`.

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

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

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

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

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

Received July 27, 2001

Published January 2, 2004