FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Bounded Domains and Unbounded Domains

Yatsuka Nakamura
Shinshu University
Nagano

Andrzej Trybulec
University of Białystok

Czesław Byliński
University of Białystok

Summary. First, notions of inside components and outside components are introduced for any subset of n-dimensional Euclid space. Next, notions of the bounded domain and the unbounded domain are defined using the above components. If the dimension is larger than 1, and if a subset is bounded, a unbounded domain of the subset coincides with an outside component (which is unique) of the subset. For a sphere in n-dimensional space, the similar fact is true for a bounded domain. In 2 dimensional space, any rectangle also has such property. We discussed relations between the Jordan property and the concept of boundary, which are necessary to find points in domains near a curve. In the last part, we gave the sufficient criterion for belonging to the left component of some clockwise oriented finite sequences.

MML Identifier: JORDAN2C.

The articles [44], [51], [12], [50], [53], [9], [10], [7], [22], [2], [1], [40], [54], [16], [27], [15], [24], [5], [38], [39], [20], [35], [32], [18], [42], [3], [8], [49], [46], [41], [21], [4], [26], [34], [37], [43], [6], [30], [52], [11], [25], [13], [17], [33], [14], [48], [47], [19], [23], [28], [29], [36], [45], and [31] provide the notation and terminology for this paper.

1. Definitions of Bounded Domain and Unbounded Domain

We follow the rules: m, n are natural numbers, r, s are real numbers, and x, y are sets.

The following propositions are true:

ISSN 1426–2630

(1) If $r \leqslant 0$, then $|r| = -r$.

(2) For all n, m such that $n \leqslant m$ and $m \leqslant n + 2$ holds $m = n$ or $m = n + 1$ or $m = n + 2$.

(3) For all n, m such that $n \leqslant m$ and $m \leqslant n + 3$ holds $m = n$ or $m = n + 1$ or $m = n + 2$ or $m = n + 3$.

(4) For all n, m such that $n \leqslant m$ and $m \leqslant n + 4$ holds $m = n$ or $m = n + 1$ or $m = n + 2$ or $m = n + 3$ or $m = n + 4$.

(5) For all real numbers a, b such that $a \geqslant 0$ and $b \geqslant 0$ holds $a + b \geqslant 0$.

(6) For all real numbers a, b such that $a > 0$ and $b \geqslant 0$ or $a \geqslant 0$ and $b > 0$ holds $a + b > 0$.

(7) For every finite sequence f such that $\operatorname{rng} f = \{x, y\}$ and $\operatorname{len} f = 2$ holds $f(1) = x$ and $f(2) = y$ or $f(1) = y$ and $f(2) = x$.

(8) Let f be an increasing finite sequence of elements of $\mathbb{R}$. If $\operatorname{rng} f = \{r, s\}$ and $\operatorname{len} f = 2$ and $r \leqslant s$, then $f(1) = r$ and $f(2) = s$.

In the sequel p, p_1, p_2, p_3, q, q_1, q_2 denote points of $\mathcal{E}^n_{\mathrm{T}}$.

We now state several propositions:

(9) $(p_1 + p_2) - p_3 = (p_1 - p_3) + p_2$.

(10) $||q|| = |q|$.

(11) $||q_1| - |q_2|| \leqslant |q_1 - q_2|$.

(12) $||[r]|| = |r|$.

(13) $q - 0_{\mathcal{E}^n_{\mathrm{T}}} = q$ and $0_{\mathcal{E}^n_{\mathrm{T}}} - q = -q$.

Let us consider n and let P be a subset of $\mathcal{E}^n_{\mathrm{T}}$. We say that P is n-convex if and only if:

(Def. 1) For all points w_1, w_2 of $\mathcal{E}^n_{\mathrm{T}}$ such that $w_1 \in P$ and $w_2 \in P$ holds $\mathcal{L}(w_1, w_2) \subseteq P$.

The following propositions are true:

(14) For every non empty subset P of $\mathcal{E}^n_{\mathrm{T}}$ such that P is n-convex holds P is connected.

(15) Let G be a non empty topological space, P be a subset of G, A be a subset of the carrier of G, and Q be a subset of $G \restriction A$. If $P \neq \emptyset$ and $P = Q$ and P is connected, then Q is connected.

Let us consider n and let A be a subset of $\mathcal{E}^n_{\mathrm{T}}$. We say that A is Bounded if and only if:

(Def. 2) There exists a subset C of the carrier of $\mathcal{E}^n$ such that $C = A$ and C is bounded.

One can prove the following proposition

(16) For all subsets A, B of $\mathcal{E}^n_{\mathrm{T}}$ such that B is Bounded and $A \subseteq B$ holds A is Bounded.

Let us consider n, let A be a subset of the carrier of $\mathcal{E}^n_{\rm T}$, and let B be a subset of $\mathcal{E}^n_{\rm T}$. We say that B is inside component of A if and only if:

(Def. 3) B is a component of $A^{\rm c}$ and Bounded.

Next we state the proposition

(17) Let A be a subset of the carrier of $\mathcal{E}^n_{\rm T}$ and B be a subset of $\mathcal{E}^n_{\rm T}$. Then B is inside component of A if and only if there exists a subset C of $(\mathcal{E}^n_{\rm T})\restriction A^{\rm c}$ such that $C = B$ and C is a component of $(\mathcal{E}^n_{\rm T})\restriction A^{\rm c}$ and for every subset D of the carrier of $\mathcal{E}^n$ such that $D = C$ holds D is bounded.

Let us consider n, let A be a subset of the carrier of $\mathcal{E}^n_{\rm T}$, and let B be a subset of $\mathcal{E}^n_{\rm T}$. We say that B is outside component of A if and only if:

(Def. 4) B is a component of $A^{\rm c}$ and B is not Bounded.

Next we state three propositions:

(18) Let A be a subset of the carrier of $\mathcal{E}^n_{\rm T}$ and B be a subset of $\mathcal{E}^n_{\rm T}$. Then B is outside component of A if and only if there exists a subset C of $(\mathcal{E}^n_{\rm T})\restriction A^{\rm c}$ such that $C = B$ and C is a component of $(\mathcal{E}^n_{\rm T})\restriction A^{\rm c}$ and it is not true that for every subset D of the carrier of $\mathcal{E}^n$ such that $D = C$ holds D is bounded.

(19) For all subsets A, B of $\mathcal{E}^n_{\rm T}$ such that B is inside component of A holds $B \subseteq A^{\rm c}$.

(20) For all subsets A, B of $\mathcal{E}^n_{\rm T}$ such that B is outside component of A holds $B \subseteq A^{\rm c}$.

Let us consider n and let A be a subset of the carrier of $\mathcal{E}^n_{\rm T}$. The functor $\mathrm{BDD}\, A$ yields a subset of $\mathcal{E}^n_{\rm T}$ and is defined by:

(Def. 5) $\mathrm{BDD}\, A = \bigcup\{B; B$ ranges over subsets of $\mathcal{E}^n_{\rm T}$: B is inside component of $A\}$.

Let us consider n and let A be a subset of the carrier of $\mathcal{E}^n_{\rm T}$. The functor $\mathrm{UBD}\, A$ yielding a subset of $\mathcal{E}^n_{\rm T}$ is defined by:

(Def. 6) $\mathrm{UBD}\, A = \bigcup\{B; B$ ranges over subsets of $\mathcal{E}^n_{\rm T}$: B is outside component of $A\}$.

One can prove the following propositions:

(21) $\Omega_{\mathcal{E}^n_{\rm T}}$ is n-convex.

(22) $\Omega_{\mathcal{E}^n_{\rm T}}$ is connected.

Let us consider n. One can check that $\Omega_{\mathcal{E}^n_{\rm T}}$ is connected.

We now state several propositions:

(23) $\Omega_{\mathcal{E}^n_{\rm T}}$ is a component of $\mathcal{E}^n_{\rm T}$.

(24) For every subset A of the carrier of $\mathcal{E}^n_{\rm T}$ holds $\mathrm{BDD}\, A$ is a union of components of $(\mathcal{E}^n_{\rm T})\restriction A^{\rm c}$.

(25) For every subset A of the carrier of $\mathcal{E}^n_{\rm T}$ holds $\mathrm{UBD}\, A$ is a union of components of $(\mathcal{E}^n_{\rm T})\restriction A^{\rm c}$.

(26) Let A be a subset of the carrier of $\mathcal{E}^n_{\rm T}$ and B be a subset of $\mathcal{E}^n_{\rm T}$. If B is inside component of A, then $B \subseteq {\rm BDD}\,A$.

(27) Let A be a subset of the carrier of $\mathcal{E}^n_{\rm T}$ and B be a subset of $\mathcal{E}^n_{\rm T}$. If B is outside component of A, then $B \subseteq {\rm UBD}\,A$.

(28) For every subset A of the carrier of $\mathcal{E}^n_{\rm T}$ holds ${\rm BDD}\,A \cap {\rm UBD}\,A = \emptyset$.

(29) For every subset A of the carrier of $\mathcal{E}^n_{\rm T}$ holds ${\rm BDD}\,A \subseteq A^{\rm c}$.

(30) For every subset A of the carrier of $\mathcal{E}^n_{\rm T}$ holds ${\rm UBD}\,A \subseteq A^{\rm c}$.

(31) For every subset A of the carrier of $\mathcal{E}^n_{\rm T}$ holds ${\rm BDD}\,A \cup {\rm UBD}\,A = A^{\rm c}$.

In the sequel u is a point of $\mathcal{E}^n$.

One can prove the following propositions:

(32) Let G be a non empty topological space, w_1, w_2, w_3 be points of G, h_1 be a map from $\mathbb{I}$ into G, and h_2 be a map from $\mathbb{I}$ into G. Suppose h_1 is continuous and $w_1 = h_1(0)$ and $w_2 = h_1(1)$ and h_2 is continuous and $w_2 = h_2(0)$ and $w_3 = h_2(1)$. Then there exists a map h_3 from $\mathbb{I}$ into G such that h_3 is continuous and $w_1 = h_3(0)$ and $w_3 = h_3(1)$ and ${\rm rng}\,h_3 \subseteq {\rm rng}\,h_1 \cup {\rm rng}\,h_2$.

(33) For every subset P of $\mathcal{E}^n_{\rm T}$ such that $P = \mathcal{R}^n$ holds P is connected.

Let us consider n. The functor $1 * n$ yielding a finite sequence of elements of $\mathbb{R}$ is defined by:

(Def. 7) $1 * n = n \mapsto (1$ **qua** real number$)$.

Let us consider n. Then $1 * n$ is an element of $\mathcal{R}^n$.

Let us consider n. The functor $1.{\rm REAL}\,n$ yielding a point of $\mathcal{E}^n_{\rm T}$ is defined by:

(Def. 8) $1.{\rm REAL}\,n = 1 * n$.

One can prove the following propositions:

(34) $|1 * n| = n \mapsto (1$ **qua** real number$)$.

(35) $|1 * n| = \sqrt{n}$.

(36) $1.{\rm REAL}\,1 = \langle (1$ **qua** real number$)\rangle$.

(37) $|\,1.{\rm REAL}\,n| = \sqrt{n}$.

(38) If $1 \leqslant n$, then $1 \leqslant |\,1.{\rm REAL}\,n|$.

(39) For every subset W of the carrier of $\mathcal{E}^n$ such that $n \geqslant 1$ and $W = \mathcal{R}^n$ holds W is not bounded.

(40) Let A be a subset of $\mathcal{E}^n_{\rm T}$. Then A is Bounded if and only if there exists a real number r such that for every point q of $\mathcal{E}^n_{\rm T}$ such that $q \in A$ holds $|q| < r$.

(41) If $n \geqslant 1$, then $\Omega_{\mathcal{E}^n_{\rm T}}$ is not Bounded.

(42) If $n \geqslant 1$, then ${\rm UBD}\,\emptyset_{\mathcal{E}^n_{\rm T}} = \mathcal{R}^n$.

(43) Let w_1, w_2, w_3 be points of $\mathcal{E}^n_{\rm T}$, P be a non empty subset of the carrier of $\mathcal{E}^n_{\rm T}$, and h_1, h_2 be maps from $\mathbb{I}$ into $(\mathcal{E}^n_{\rm T})\restriction P$. Suppose h_1 is continuous and $w_1 = h_1(0)$ and $w_2 = h_1(1)$ and h_2 is continuous and $w_2 = h_2(0)$ and $w_3 = h_2(1)$. Then there exists a map h_3 from $\mathbb{I}$ into $(\mathcal{E}^n_{\rm T})\restriction P$ such that h_3 is continuous and $w_1 = h_3(0)$ and $w_3 = h_3(1)$.

(44) Let P be a subset of the carrier of $\mathcal{E}^n_{\rm T}$ and w_1, w_2, w_3 be points of $\mathcal{E}^n_{\rm T}$. Suppose $w_1 \in P$ and $w_2 \in P$ and $w_3 \in P$ and $\mathcal{L}(w_1, w_2) \subseteq P$ and $\mathcal{L}(w_2, w_3) \subseteq P$. Then there exists a map h from $\mathbb{I}$ into $(\mathcal{E}^n_{\rm T})\restriction P$ such that h is continuous and $w_1 = h(0)$ and $w_3 = h(1)$.

(45) Let P be a subset of the carrier of $\mathcal{E}^n_{\rm T}$ and w_1, w_2, w_3, w_4 be points of $\mathcal{E}^n_{\rm T}$. Suppose $w_1 \in P$ and $w_2 \in P$ and $w_3 \in P$ and $w_4 \in P$ and $\mathcal{L}(w_1, w_2) \subseteq P$ and $\mathcal{L}(w_2, w_3) \subseteq P$ and $\mathcal{L}(w_3, w_4) \subseteq P$. Then there exists a map h from $\mathbb{I}$ into $(\mathcal{E}^n_{\rm T})\restriction P$ such that h is continuous and $w_1 = h(0)$ and $w_4 = h(1)$.

(46) Let P be a subset of the carrier of $\mathcal{E}^n_{\rm T}$ and w_1, w_2, w_3, w_4, w_5, w_6, w_7 be points of $\mathcal{E}^n_{\rm T}$. Suppose $w_1 \in P$ and $w_2 \in P$ and $w_3 \in P$ and $w_4 \in P$ and $w_5 \in P$ and $w_6 \in P$ and $w_7 \in P$ and $\mathcal{L}(w_1, w_2) \subseteq P$ and $\mathcal{L}(w_2, w_3) \subseteq P$ and $\mathcal{L}(w_3, w_4) \subseteq P$ and $\mathcal{L}(w_4, w_5) \subseteq P$ and $\mathcal{L}(w_5, w_6) \subseteq P$ and $\mathcal{L}(w_6, w_7) \subseteq P$. Then there exists a map h from $\mathbb{I}$ into $(\mathcal{E}^n_{\rm T})\restriction P$ such that h is continuous and $w_1 = h(0)$ and $w_7 = h(1)$.

(47) For all points w_1, w_2 of $\mathcal{E}^n_{\rm T}$ such that it is not true that there exists a real number r such that $w_1 = r \cdot w_2$ or $w_2 = r \cdot w_1$ holds $0_{\mathcal{E}^n_{\rm T}} \notin \mathcal{L}(w_1, w_2)$.

(48) Let w_1, w_2 be points of $\mathcal{E}^n_{\rm T}$ and P be a subset of $(\mathcal{E}^n)_{\rm top}$. Suppose $P = \mathcal{L}(w_1, w_2)$ and $0_{\mathcal{E}^n_{\rm T}} \notin \mathcal{L}(w_1, w_2)$. Then there exists a point w_0 of $\mathcal{E}^n_{\rm T}$ such that $w_0 \in \mathcal{L}(w_1, w_2)$ and $|w_0| > 0$ and $|w_0| = ({\rm dist}_{\rm min}(P))(0_{\mathcal{E}^n_{\rm T}})$.

(49) Let a be a real number, Q be a subset of the carrier of $\mathcal{E}^n_{\rm T}$, and w_1, w_4 be points of $\mathcal{E}^n_{\rm T}$. Suppose $Q = \{q : |q| > a\}$ and $w_1 \in Q$ and $w_4 \in Q$ and it is not true that there exists a real number r such that $w_1 = r \cdot w_4$ or $w_4 = r \cdot w_1$. Then there exist points w_2, w_3 of $\mathcal{E}^n_{\rm T}$ such that $w_2 \in Q$ and $w_3 \in Q$ and $\mathcal{L}(w_1, w_2) \subseteq Q$ and $\mathcal{L}(w_2, w_3) \subseteq Q$ and $\mathcal{L}(w_3, w_4) \subseteq Q$.

(50) Let a be a real number, Q be a subset of the carrier of $\mathcal{E}^n_{\rm T}$, and w_1, w_4 be points of $\mathcal{E}^n_{\rm T}$. Suppose $Q = \mathcal{R}^n \setminus \{q : |q| < a\}$ and $w_1 \in Q$ and $w_4 \in Q$ and it is not true that there exists a real number r such that $w_1 = r \cdot w_4$ or $w_4 = r \cdot w_1$. Then there exist points w_2, w_3 of $\mathcal{E}^n_{\rm T}$ such that $w_2 \in Q$ and $w_3 \in Q$ and $\mathcal{L}(w_1, w_2) \subseteq Q$ and $\mathcal{L}(w_2, w_3) \subseteq Q$ and $\mathcal{L}(w_3, w_4) \subseteq Q$.

(51) Let x be an element of $\mathcal{R}^n$. Then x is a finite sequence of elements of $\mathbb{R}$ and for every finite sequence f such that $f = x$ holds ${\rm len}\, f = n$.

(52) Every finite sequence f of elements of $\mathbb{R}$ is an element of $\mathcal{R}^{{\rm len}\, f}$ and a point of $\mathcal{E}^{{\rm len}\, f}_{\rm T}$.

(53) Let x be an element of $\mathcal{R}^n$, f, g be finite sequences of elements of $\mathbb{R}$, and r be a real number. Suppose $f = x$ and $g = r \cdot x$. Then ${\rm len}\, f = {\rm len}\, g$ and for

every natural number i such that $1 \leqslant i$ and $i \leqslant \operatorname{len} f$ holds $\pi_i g = r \cdot \pi_i f$.

(54) Let x be an element of $\mathcal{R}^n$ and f be a finite sequence. Suppose $x \neq \langle \underbrace{0, \ldots, 0}_{n} \rangle$ and $x = f$. Then there exists a natural number i such that $1 \leqslant i$ and $i \leqslant n$ and $f(i) \neq 0$.

(55) Let x be an element of $\mathcal{R}^n$. Suppose $n \geqslant 2$ and $x \neq \langle \underbrace{0, \ldots, 0}_{n} \rangle$. Then it is not true that there exists an element y of $\mathcal{R}^n$ and there exists a real number r such that $y = r \cdot x$ or $x = r \cdot y$.

(56) Let a be a real number, Q be a subset of the carrier of $\mathcal{E}^n_{\mathrm{T}}$, and w_1, w_7 be points of $\mathcal{E}^n_{\mathrm{T}}$. Suppose $n \geqslant 2$ and $Q = \{q : |q| > a\}$ and $w_1 \in Q$ and $w_7 \in Q$ and there exists a real number r such that $w_1 = r \cdot w_7$ or $w_7 = r \cdot w_1$. Then there exist points w_2, w_3, w_4, w_5, w_6 of $\mathcal{E}^n_{\mathrm{T}}$ such that $w_2 \in Q$ and $w_3 \in Q$ and $w_4 \in Q$ and $w_5 \in Q$ and $w_6 \in Q$ and $\mathcal{L}(w_1, w_2) \subseteq Q$ and $\mathcal{L}(w_2, w_3) \subseteq Q$ and $\mathcal{L}(w_3, w_4) \subseteq Q$ and $\mathcal{L}(w_4, w_5) \subseteq Q$ and $\mathcal{L}(w_5, w_6) \subseteq Q$ and $\mathcal{L}(w_6, w_7) \subseteq Q$.

(57) Let a be a real number, Q be a subset of the carrier of $\mathcal{E}^n_{\mathrm{T}}$, and w_1, w_7 be points of $\mathcal{E}^n_{\mathrm{T}}$. Suppose $n \geqslant 2$ and $Q = \mathcal{R}^n \setminus \{q : |q| < a\}$ and $w_1 \in Q$ and $w_7 \in Q$ and there exists a real number r such that $w_1 = r \cdot w_7$ or $w_7 = r \cdot w_1$. Then there exist points w_2, w_3, w_4, w_5, w_6 of $\mathcal{E}^n_{\mathrm{T}}$ such that $w_2 \in Q$ and $w_3 \in Q$ and $w_4 \in Q$ and $w_5 \in Q$ and $w_6 \in Q$ and $\mathcal{L}(w_1, w_2) \subseteq Q$ and $\mathcal{L}(w_2, w_3) \subseteq Q$ and $\mathcal{L}(w_3, w_4) \subseteq Q$ and $\mathcal{L}(w_4, w_5) \subseteq Q$ and $\mathcal{L}(w_5, w_6) \subseteq Q$ and $\mathcal{L}(w_6, w_7) \subseteq Q$.

(58) For every real number a such that $n \geqslant 1$ holds $\{q : |q| > a\} \neq \emptyset$.

(59) For every real number a and for every subset P of $\mathcal{E}^n_{\mathrm{T}}$ such that $n \geqslant 2$ and $P = \{q : |q| > a\}$ holds P is connected.

(60) For every real number a such that $n \geqslant 1$ holds $\mathcal{R}^n \setminus \{q : |q| < a\} \neq \emptyset$.

(61) For every real number a and for every subset P of $\mathcal{E}^n_{\mathrm{T}}$ such that $n \geqslant 2$ and $P = \mathcal{R}^n \setminus \{q : |q| < a\}$ holds P is connected.

(62) Let a be a real number, n be a natural number, and P be a subset of $\mathcal{E}^n_{\mathrm{T}}$. If $n \geqslant 1$ and $P = \mathcal{R}^n \setminus \{q; q$ ranges over points of $\mathcal{E}^n_{\mathrm{T}}$: $|q| < a\}$, then P is not Bounded.

(63) Let a be a real number and P be a subset of $\mathcal{E}^1_{\mathrm{T}}$. Suppose $P = \{q; q$ ranges over points of $\mathcal{E}^1_{\mathrm{T}}$: $\bigvee_r (q = \langle r \rangle \ \wedge \ r > a)\}$. Then P is n-convex.

(64) Let a be a real number and P be a subset of $\mathcal{E}^1_{\mathrm{T}}$. Suppose $P = \{q; q$ ranges over points of $\mathcal{E}^1_{\mathrm{T}}$: $\bigvee_r (q = \langle r \rangle \ \wedge \ r < -a)\}$. Then P is n-convex.

(65) Let a be a real number and P be a subset of $\mathcal{E}^1_{\mathrm{T}}$. Suppose $P = \{q; q$ ranges over points of $\mathcal{E}^1_{\mathrm{T}}$: $\bigvee_r (q = \langle r \rangle \ \wedge \ r > a)\}$. Then P is connected.

(66) Let a be a real number and P be a subset of $\mathcal{E}^1_{\mathrm{T}}$. Suppose $P = \{q; q$ ranges over points of $\mathcal{E}^1_{\mathrm{T}}$: $\bigvee_r (q = \langle r \rangle \ \wedge \ r < -a)\}$. Then P is connected.

(67) Let W be a subset of the carrier of $\mathcal{E}^1$, a be a real number, and P be a subset of $\mathcal{E}^1_{\rm T}$. Suppose $W = \{q; q$ ranges over points of $\mathcal{E}^1_{\rm T}$: $\bigvee_r$ $(q = \langle r \rangle \ \wedge \ r > a)\}$ and $P = W$. Then P is connected and W is not bounded.

(68) Let W be a subset of the carrier of $\mathcal{E}^1$, a be a real number, and P be a subset of $\mathcal{E}^1_{\rm T}$. Suppose $W = \{q; q$ ranges over points of $\mathcal{E}^1_{\rm T}$: $\bigvee_r$ $(q = \langle r \rangle \ \wedge \ r < -a)\}$ and $P = W$. Then P is connected and W is not bounded.

(69) Let W be a subset of the carrier of $\mathcal{E}^n$, a be a real number, and P be a subset of $\mathcal{E}^n_{\rm T}$. If $n \geqslant 2$ and $W = \{q : |q| > a\}$ and $P = W$, then P is connected and W is not bounded.

(70) Let W be a subset of the carrier of $\mathcal{E}^n$, a be a real number, and P be a subset of $\mathcal{E}^n_{\rm T}$. If $n \geqslant 2$ and $W = \mathcal{R}^n \setminus \{q : |q| < a\}$ and $P = W$, then P is connected and W is not bounded.

(71) Let P, P_1 be subsets of $\mathcal{E}^n_{\rm T}$, Q be a subset of the carrier of $\mathcal{E}^n_{\rm T}$, and W be a subset of the carrier of $\mathcal{E}^n$. Suppose $P = W$ and P is connected and W is not bounded and $P_1 = \mathrm{Component}(\mathrm{Down}(P, Q^{\rm c}))$ and $W \cap Q = \emptyset$. Then P_1 is outside component of Q.

Let S be a 1-sorted structure and let A be a subset of the carrier of S. The functor $\mathrm{RAC}\,A$ yields a subset of S and is defined as follows:

(Def. 9) $\mathrm{RAC}\,A = A$.

The following propositions are true:

(72) Let A be a subset of the carrier of $\mathcal{E}^n$, B be a non empty subset of the carrier of $\mathcal{E}^n$, and C be a subset of the carrier of $\mathcal{E}^n \restriction B$. If $A \subseteq B$ and $A = C$ and C is bounded, then A is bounded.

(73) For every subset A of $\mathcal{E}^n_{\rm T}$ such that A is compact holds A is Bounded.

(74) For every subset A of $\mathcal{E}^n_{\rm T}$ such that $1 \leqslant n$ and A is Bounded holds $A^{\rm c} \neq \emptyset$.

(75) Let r be a real number. Then

(i) there exists a subset B of the carrier of $\mathcal{E}^n$ such that $B = \{q : |q| < r\}$, and

(ii) for every subset A of the carrier of $\mathcal{E}^n$ such that $A = \{q_1 : |q_1| < r\}$ holds A is bounded.

(76) Let A be a subset of $\mathcal{E}^n_{\rm T}$. Suppose $n \geqslant 2$ and A is Bounded. Then there exists a subset B of $\mathcal{E}^n_{\rm T}$ such that B is outside component of A and $B = \mathrm{UBD}\,A$.

(77) For every real number a and for every subset P of $\mathcal{E}^n_{\rm T}$ such that $P = \{q : |q| < a\}$ holds P is n-convex.

(78) For every real number a and for every subset P of $\mathcal{E}^n_{\rm T}$ such that $P = \mathrm{Ball}(u, a)$ holds P is n-convex.

(79) For every real number a and for every subset P of $\mathcal{E}^n_{\rm T}$ such that $a > 0$ and $P = \{q : |q| < a\}$ holds P is connected.

In the sequel R denotes a subset of $\mathcal{E}^n_{\rm T}$, P denotes a subset of the carrier of $\mathcal{E}^n_{\rm T}$, and f denotes a finite sequence of elements of $\mathcal{E}^n_{\rm T}$.

Next we state a number of propositions:

(80) Suppose $p \neq q$ and $p \in \mathrm{Ball}(u, r)$ and $q \in \mathrm{Ball}(u, r)$. Then there exists a map h from $\mathbb{I}$ into $\mathcal{E}^n_{\rm T}$ such that h is continuous and $h(0) = p$ and $h(1) = q$ and $\mathrm{rng}\, h \subseteq \mathrm{Ball}(u, r)$.

(81) Let f be a map from $\mathbb{I}$ into $\mathcal{E}^n_{\rm T}$. Suppose f is continuous and $f(0) = p_1$ and $f(1) = p_2$ and $p \in \mathrm{Ball}(u, r)$ and $p_2 \in \mathrm{Ball}(u, r)$. Then there exists a map h from $\mathbb{I}$ into $\mathcal{E}^n_{\rm T}$ such that h is continuous and $h(0) = p_1$ and $h(1) = p$ and $\mathrm{rng}\, h \subseteq \mathrm{rng}\, f \cup \mathrm{Ball}(u, r)$.

(82) Let f be a map from $\mathbb{I}$ into $\mathcal{E}^n_{\rm T}$. Suppose $p \neq p_1$ and f is continuous and $\mathrm{rng}\, f \subseteq P$ and $f(0) = p_1$ and $f(1) = p_2$ and $p \in \mathrm{Ball}(u, r)$ and $p_2 \in \mathrm{Ball}(u, r)$ and $\mathrm{Ball}(u, r) \subseteq P$. Then there exists a map f_1 from $\mathbb{I}$ into $\mathcal{E}^n_{\rm T}$ such that f_1 is continuous and $\mathrm{rng}\, f_1 \subseteq P$ and $f_1(0) = p_1$ and $f_1(1) = p$.

(83) Let given p and P be a subset of $\mathcal{E}^n_{\rm T}$. Suppose that

(i) R is connected and open, and

(ii) $P = \{q : q \neq p \ \wedge \ q \in R \ \wedge \ \neg \bigvee_{f : \text{map from } \mathbb{I} \text{ into } \mathcal{E}^n_{\rm T}} (f \text{ is continuous} \wedge \mathrm{rng}\, f \subseteq R \ \wedge \ f(0) = p \ \wedge \ f(1) = q)\}$.

Then P is open.

(84) Let P be a subset of $\mathcal{E}^n_{\rm T}$. Suppose that

(i) R is connected and open,

(ii) $p \in R$, and

(iii) $P = \{q : q = p \ \vee \ \bigvee_{f : \text{map from } \mathbb{I} \text{ into } \mathcal{E}^n_{\rm T}} (f \text{ is continuous} \wedge \mathrm{rng}\, f \subseteq R \ \wedge \ f(0) = p \ \wedge \ f(1) = q)\}$.

Then P is open.

(85) Let R be a subset of the carrier of $\mathcal{E}^n_{\rm T}$. Suppose $p \in R$ and $P = \{q : q = p \ \vee \ \bigvee_{f : \text{map from } \mathbb{I} \text{ into } \mathcal{E}^n_{\rm T}} (f \text{ is continuous} \wedge \mathrm{rng}\, f \subseteq R \ \wedge \ f(0) = p \ \wedge \ f(1) = q)\}$. Then $P \subseteq R$.

(86) Let R be a subset of $\mathcal{E}^n_{\rm T}$ and p be a point of $\mathcal{E}^n_{\rm T}$. Suppose that

(i) R is connected and open,

(ii) $p \in R$, and

(iii) $P = \{q : q = p \ \vee \ \bigvee_{f : \text{map from } \mathbb{I} \text{ into } \mathcal{E}^n_{\rm T}} (f \text{ is continuous} \wedge \mathrm{rng}\, f \subseteq R \ \wedge \ f(0) = p \ \wedge \ f(1) = q)\}$.

Then $R \subseteq P$.

(87) Let R be a subset of $\mathcal{E}^n_{\rm T}$ and p, q be points of $\mathcal{E}^n_{\rm T}$. Suppose R is connected and open and $p \in R$ and $q \in R$ and $p \neq q$. Then there exists a map f from $\mathbb{I}$ into $\mathcal{E}^n_{\rm T}$ such that f is continuous and $\mathrm{rng}\, f \subseteq R$ and $f(0) = p$ and $f(1) = q$.

(88) For every subset A of $\mathcal{E}^n_{\mathrm{T}}$ and for every real number a such that $A = \{q : |q| = a\}$ holds $-A$ is open and A is closed.

(89) For every non empty subset B of $\mathcal{E}^n_{\mathrm{T}}$ such that B is open holds $(\mathcal{E}^n_{\mathrm{T}}) \restriction B$ is locally connected.

(90) Let B be a non empty subset of the carrier of $\mathcal{E}^n_{\mathrm{T}}$, A be a subset of the carrier of $\mathcal{E}^n_{\mathrm{T}}$, and a be a real number. If $A = \{q : |q| = a\}$ and $A^{\mathrm{c}} = B$, then $(\mathcal{E}^n_{\mathrm{T}}) \restriction B$ is locally connected.

(91) For every map f from $\mathcal{E}^n_{\mathrm{T}}$ into $\mathbb{R}^{\mathbf{1}}$ such that for every q holds $f(q) = |q|$ holds f is continuous.

(92) There exists a map f from $\mathcal{E}^n_{\mathrm{T}}$ into $\mathbb{R}^{\mathbf{1}}$ such that for every q holds $f(q) = |q|$ and f is continuous.

Let X, Y be non empty 1-sorted structures, let f be a map from X into Y, and let x be a set. Let us assume that x is a point of X. The functor $\pi_x f$ yielding a point of Y is defined as follows:

(Def. 10) $\pi_x f = f(x)$.

We now state four propositions:

(93) Let g be a map from $\mathbb{I}$ into $\mathcal{E}^n_{\mathrm{T}}$. Suppose g is continuous. Then there exists a map f from $\mathbb{I}$ into $\mathbb{R}^{\mathbf{1}}$ such that for every point t of $\mathbb{I}$ holds $f(t) = |g(t)|$ and f is continuous.

(94) Let g be a map from $\mathbb{I}$ into $\mathcal{E}^n_{\mathrm{T}}$ and a be a real number. Suppose g is continuous and $|\pi_0 g| \leqslant a$ and $a \leqslant |\pi_1 g|$. Then there exists a point s of $\mathbb{I}$ such that $|\pi_s g| = a$.

(95) If $q = \langle r \rangle$, then $|q| = |r|$.

(96) Let A be a subset of the carrier of $\mathcal{E}^n_{\mathrm{T}}$ and a be a real number. Suppose $n \geqslant 1$ and $a > 0$ and $A = \{q : |q| = a\}$. Then there exists a subset B of $\mathcal{E}^n_{\mathrm{T}}$ such that B is inside component of A and $B = \mathrm{BDD}\, A$.

2. Bounded and Unbounded Domains of Rectangles

In the sequel D is a non vertical non horizontal non empty compact subset of $\mathcal{E}^2_{\mathrm{T}}$.

Next we state several propositions:

(97) len the Go-board of SpStSeq D = 2 and width the Go-board of SpStSeq D = 2 and π_1 SpStSeq D = (the Go-board of SpStSeq $D)_{1,2}$ and π_2 SpStSeq D = (the Go-board of SpStSeq $D)_{2,2}$ and π_3 SpStSeq D = (the Go-board of SpStSeq $D)_{2,1}$ and π_4 SpStSeq D = (the Go-board of SpStSeq $D)_{1,1}$ and π_5 SpStSeq D = (the Go-board of SpStSeq $D)_{1,2}$.

(98) LeftComp(SpStSeq D) is not Bounded.

(99) $\mathrm{LeftComp}(\mathrm{SpStSeq}\,D) \subseteq \mathrm{UBD}\,\widetilde{\mathcal{L}}(\mathrm{SpStSeq}\,D)$.

(100) Let G be a topological space and A, B, C be subsets of G. Suppose A is a component of G and B is a component of G and C is connected and $A \cap C \neq \emptyset$ and $B \cap C \neq \emptyset$. Then $A = B$.

(101) For every subset B of $\mathcal{E}^2_{\mathrm{T}}$ such that B is a component of $(\widetilde{\mathcal{L}}(\mathrm{SpStSeq}\,D))^{\mathrm{c}}$ and B is not Bounded holds $B = \mathrm{LeftComp}(\mathrm{SpStSeq}\,D)$.

(102) $\mathrm{RightComp}(\mathrm{SpStSeq}\,D) \subseteq \mathrm{BDD}\,\widetilde{\mathcal{L}}(\mathrm{SpStSeq}\,D)$ and $\mathrm{RightComp}(\mathrm{SpStSeq}\,D)$ is Bounded.

(103) $\mathrm{LeftComp}(\mathrm{SpStSeq}\,D) = \mathrm{UBD}\,\widetilde{\mathcal{L}}(\mathrm{SpStSeq}\,D)$ and $\mathrm{RightComp}(\mathrm{SpStSeq}\,D) = \mathrm{BDD}\,\widetilde{\mathcal{L}}(\mathrm{SpStSeq}\,D)$.

(104) $\mathrm{UBD}\,\widetilde{\mathcal{L}}(\mathrm{SpStSeq}\,D) \neq \emptyset$ and $\mathrm{UBD}\,\widetilde{\mathcal{L}}(\mathrm{SpStSeq}\,D)$ is outside component of $\widetilde{\mathcal{L}}(\mathrm{SpStSeq}\,D)$ and $\mathrm{BDD}\,\widetilde{\mathcal{L}}(\mathrm{SpStSeq}\,D) \neq \emptyset$ and $\mathrm{BDD}\,\widetilde{\mathcal{L}}(\mathrm{SpStSeq}\,D)$ is inside component of $\widetilde{\mathcal{L}}(\mathrm{SpStSeq}\,D)$.

3. Jordan Property and Boundary Property

One can prove the following propositions:

(105) Let G be a non empty topological space and A be a subset of G. Suppose $A^{\mathrm{c}} \neq \emptyset$. Then A is boundary if and only if for every set x and for every subset V of G such that $x \in A$ and $x \in V$ and V is open there exists a subset B of the carrier of G such that B is a component of A^{c} and $V \cap B \neq \emptyset$.

(106) Let A be a subset of $\mathcal{E}^2_{\mathrm{T}}$. Suppose $A^{\mathrm{c}} \neq \emptyset$. Then A is boundary and Jordan if and only if there exist subsets A_1, A_2 of $\mathcal{E}^2_{\mathrm{T}}$ such that $A^{\mathrm{c}} = A_1 \cup A_2$ and $A_1 \cap A_2 = \emptyset$ and $\overline{A_1} \setminus A_1 = \overline{A_2} \setminus A_2$ and $A = \overline{A_1} \setminus A_1$ and for all subsets C_1, C_2 of $(\mathcal{E}^2_{\mathrm{T}}) \restriction A^{\mathrm{c}}$ such that $C_1 = A_1$ and $C_2 = A_2$ holds C_1 is a component of $(\mathcal{E}^2_{\mathrm{T}}) \restriction A^{\mathrm{c}}$ and C_2 is a component of $(\mathcal{E}^2_{\mathrm{T}}) \restriction A^{\mathrm{c}}$.

(107) For every point p of $\mathcal{E}^n_{\mathrm{T}}$ and for every subset P of $\mathcal{E}^n_{\mathrm{T}}$ such that $n \geqslant 1$ and $P = \{p\}$ holds P is boundary.

(108) For all points p, q of $\mathcal{E}^2_{\mathrm{T}}$ and for every r such that $p_{\mathbf{1}} = q_{\mathbf{2}}$ and $-p_{\mathbf{2}} = q_{\mathbf{1}}$ and $p = r \cdot q$ holds $p_{\mathbf{1}} = 0$ and $p_{\mathbf{2}} = 0$ and $p = 0_{\mathcal{E}^2_{\mathrm{T}}}$.

(109) For all points q_1, q_2 of $\mathcal{E}^2_{\mathrm{T}}$ holds $\mathcal{L}(q_1, q_2)$ is boundary.

Let q_1, q_2 be points of $\mathcal{E}^2_{\mathrm{T}}$. Observe that $\mathcal{L}(q_1, q_2)$ is boundary.

One can prove the following proposition

(110) For every finite sequence f of elements of $\mathcal{E}^2_{\mathrm{T}}$ holds $\widetilde{\mathcal{L}}(f)$ is boundary.

Let f be a finite sequence of elements of $\mathcal{E}^2_{\mathrm{T}}$. Note that $\widetilde{\mathcal{L}}(f)$ is boundary.

We now state several propositions:

(111) For every point e_1 of $\mathcal{E}^n$ and for all points p, q of $\mathcal{E}^n_{\rm T}$ such that $p = e_1$ and $q \in \mathrm{Ball}(e_1, r)$ holds $|p - q| < r$ and $|q - p| < r$.

(112) Let a be a real number and p be a point of $\mathcal{E}^2_{\rm T}$. Suppose $a > 0$ and $p \in \widetilde{\mathcal{L}}(\mathrm{SpStSeq}\, D)$. Then there exists a point q of $\mathcal{E}^2_{\rm T}$ such that $q \in \mathrm{UBD}\, \widetilde{\mathcal{L}}(\mathrm{SpStSeq}\, D)$ and $|p - q| < a$.

(113) $\mathcal{R}^0 = \{0_{\mathcal{E}^0_{\rm T}}\}$.

(114) For every subset A of $\mathcal{E}^n_{\rm T}$ such that A is Bounded holds $\mathrm{BDD}\, A$ is Bounded.

(115) Let G be a non empty topological space and A, B, C, D be subsets of G. Suppose A is a component of G and B is a component of G and C is a component of G and $A \cup B =$ the carrier of G and $C \cap A = \emptyset$. Then $C = B$.

(116) For every subset A of $\mathcal{E}^2_{\rm T}$ such that A is Bounded and Jordan holds $\mathrm{BDD}\, A$ is inside component of A.

(117) Let a be a real number and p be a point of $\mathcal{E}^2_{\rm T}$. Suppose $a > 0$ and $p \in \widetilde{\mathcal{L}}(\mathrm{SpStSeq}\, D)$. Then there exists a point q of $\mathcal{E}^2_{\rm T}$ such that $q \in \mathrm{BDD}\, \widetilde{\mathcal{L}}(\mathrm{SpStSeq}\, D)$ and $|p - q| < a$.

4. Points in LeftComp

In the sequel f denotes a clockwise oriented non constant standard special circular sequence.

Next we state four propositions:

(118) For every point p of $\mathcal{E}^2_{\rm T}$ such that $\pi_1 f = \text{N-min}\, \widetilde{\mathcal{L}}(f)$ and $p_{\mathbf{1}} < \text{W-bound}\, \widetilde{\mathcal{L}}(f)$ holds $p \in \mathrm{LeftComp}(f)$.

(119) For every point p of $\mathcal{E}^2_{\rm T}$ such that $\pi_1 f = \text{N-min}\, \widetilde{\mathcal{L}}(f)$ and $p_{\mathbf{1}} > \text{E-bound}\, \widetilde{\mathcal{L}}(f)$ holds $p \in \mathrm{LeftComp}(f)$.

(120) For every point p of $\mathcal{E}^2_{\rm T}$ such that $\pi_1 f = \text{N-min}\, \widetilde{\mathcal{L}}(f)$ and $p_{\mathbf{2}} < \text{S-bound}\, \widetilde{\mathcal{L}}(f)$ holds $p \in \mathrm{LeftComp}(f)$.

(121) For every point p of $\mathcal{E}^2_{\rm T}$ such that $\pi_1 f = \text{N-min}\, \widetilde{\mathcal{L}}(f)$ and $p_{\mathbf{2}} > \text{N-bound}\, \widetilde{\mathcal{L}}(f)$ holds $p \in \mathrm{LeftComp}(f)$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.

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

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

[4] Józef Białas and Yatsuka Nakamura. The theorem of Weierstrass. *Formalized Mathematics*, 5(**3**):353–359, 1996.

[5] Leszek Borys. Paracompact and metrizable spaces. *Formalized Mathematics*, 2(**4**):481–485, 1991.

[6] Czesław Byliński. Basic functions and operations on functions. *Formalized Mathematics*, 1(**1**):245–254, 1990.

[7] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(**1**):175–180, 1990.

[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] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.

[12] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.

[13] Czesław Byliński. The sum and product of finite sequences of real numbers. *Formalized Mathematics*, 1(**4**):661–668, 1990.

[14] Czesław Byliński and Piotr Rudnicki. Bounding boxes for compact sets in $\mathcal{E}^2$. *Formalized Mathematics*, 6(**3**):427–440, 1997.

[15] Agata Darmochwał. Compact spaces. *Formalized Mathematics*, 1(**2**):383–386, 1990.

[16] Agata Darmochwał. Families of subsets, subspaces and mappings in topological spaces. *Formalized Mathematics*, 1(**2**):257–261, 1990.

[17] Agata Darmochwał. The Euclidean space. *Formalized Mathematics*, 2(**4**):599–603, 1991.

[18] Agata Darmochwał and Yatsuka Nakamura. Metric spaces as topological spaces - fundamental concepts. *Formalized Mathematics*, 2(**4**):605–608, 1991.

[19] Agata Darmochwał and Yatsuka Nakamura. The topological space $\mathcal{E}^2_{\mathrm{T}}$. Arcs, line segments and special polygonal arcs. *Formalized Mathematics*, 2(**5**):617–621, 1991.

[20] Alicia de la Cruz. Totally bounded metric spaces. *Formalized Mathematics*, 2(**4**):559–562, 1991.

[21] Adam Grabowski. Introduction to the homotopy theory. *Formalized Mathematics*, 6(**4**):449–454, 1997.

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

[23] Katarzyna Jankowska. Matrices. Abelian group of matrices. *Formalized Mathematics*, 2(**4**):475–480, 1991.

[24] Stanisława Kanas, Adam Lecko, and Mariusz Startek. Metric spaces. *Formalized Mathematics*, 1(**3**):607–610, 1990.

[25] Zbigniew Karno. Continuity of mappings over the union of subspaces. *Formalized Mathematics*, 3(**1**):1–16, 1992.

[26] Jarosław Kotowicz. Convergent real sequences. Upper and lower bound of sets of real numbers. *Formalized Mathematics*, 1(**3**):477–481, 1990.

[27] Jarosław Kotowicz. Real sequences and basic operations on them. *Formalized Mathematics*, 1(**2**):269–272, 1990.

[28] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part I. *Formalized Mathematics*, 3(**1**):107–115, 1992.

[29] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part II. *Formalized Mathematics*, 3(**1**):117–121, 1992.

[30] Eugeniusz Kusak, Wojciech Leończuk, and Michał Muzalewski. Abelian groups, fields and vector spaces. *Formalized Mathematics*, 1(**2**):335–342, 1990.

[31] Roman Matuszewski and Yatsuka Nakamura. Projections in n-dimensional Euclidean space to each coordinates. *Formalized Mathematics*, 6(**4**):505–509, 1997.

[32] Yatsuka Nakamura. Graph theoretical properties of arcs in the plane and Fashoda Meet Theorem. *Formalized Mathematics*, 7(**2**):193–201, 1998.

[33] Yatsuka Nakamura and Czesław Byliński. Extremal properties of vertices on special polygons. Part I. *Formalized Mathematics*, 5(**1**):97–102, 1996.

[34] Yatsuka Nakamura and Roman Matuszewski. Reconstructions of special sequences. *Formalized Mathematics*, 6(**2**):255–263, 1997.

[35] Yatsuka Nakamura and Andrzej Trybulec. Components and unions of components. *Formalized Mathematics*, 5(**4**):513–517, 1996.

[36] Yatsuka Nakamura and Andrzej Trybulec. Decomposing a Go-board into cells. *Formalized Mathematics*, 5(**3**):323–328, 1996.
[37] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[38] Beata Padlewska. Connected spaces. *Formalized Mathematics*, 1(**1**):239–244, 1990.
[39] Beata Padlewska. Locally connected spaces. *Formalized Mathematics*, 2(**1**):93–96, 1991.
[40] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[41] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.
[42] Konrad Raczkowski and Paweł Sadowski. Topological properties of subsets in real numbers. *Formalized Mathematics*, 1(**4**):777–780, 1990.
[43] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[44] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[45] Andrzej Trybulec. Left and right component of the complement of a special closed curve. *Formalized Mathematics*, 5(**4**):465–468, 1996.
[46] Andrzej Trybulec and Czesław Byliński. Some properties of real numbers. *Formalized Mathematics*, 1(**3**):445–449, 1990.
[47] Andrzej Trybulec and Yatsuka Nakamura. On the order on a special polygon. *Formalized Mathematics*, 6(**4**):541–548, 1997.
[48] Andrzej Trybulec and Yatsuka Nakamura. On the rectangular finite sequences of the points of the plane. *Formalized Mathematics*, 6(**4**):531–539, 1997.
[49] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[50] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[51] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[52] Toshihiko Watanabe. The Brouwer fixed point theorem for intervals. *Formalized Mathematics*, 3(**1**):85–88, 1992.
[53] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[54] Mirosław Wysocki and Agata Darmochwał. Subsets of topological spaces. *Formalized Mathematics*, 1(**1**):231–237, 1990.

Received January 7, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Rotating and Reversing

Andrzej Trybulec
University of Białystok

Summary. Quite a number of lemmas for the Jordan curve theorem, as yet in the case of the special polygonal curves, have been proved. By "special" we mean, that it is a polygonal curve with edges parallel to axes and actually the lemmas have been proved, mostly, for the triangulations i.e. for finite sequences that define the curve. Moreover some of the results deal only with a special case:

- finite sequences are clockwise oriented,
- the first member of the sequence is the member with the lowest ordinate among those with the highest abscissa (N-min f, where f is a finite sequence, in the Mizar jargon).

In the change of the orientation one has to reverse the sequence (the operation introduced in [7]) and to change the second restriction one has to rotate the sequence (the operation introduced in [26]). The goal of the paper is to prove, mostly simple, facts about the relationship between properties and attributes of the finite sequence and its rotation (similar results about reversing had been proved in [7]). Some of them deal with recounting parameters, others with properties that are invariant under the rotation. We prove also that the finite sequence is either clockwise oriented or it is such after reversing. Everything is proved for the so called standard finite sequences, which means that if a point belongs to it then every point with the same abscissa or with the same ordinate, that belongs to the polygon, belongs also to the finite sequence. It does not seem that this requirement causes serious technical obstacles.

MML Identifier: REVROT_1.

The terminology and notation used here are introduced in the following articles: [24], [29], [12], [2], [23], [20], [1], [4], [6], [3], [5], [13], [28], [14], [7], [26], [22], [30], [21], [9], [10], [11], [15], [16], [18], [25], [8], [17], [27], and [19].

ISSN 1426–2630

1. Preliminaries

For simplicity, we use the following convention: i, k, m, n are natural numbers, D is a non empty set, p is an element of D, and f is a finite sequence of elements of D.

Let S be a non trivial 1-sorted structure. Observe that the carrier of S is non trivial.

Let D be a non empty set and let f be a finite sequence of elements of D. Let us observe that f is constant if and only if:

(Def. 1) For all n, m such that $n \in \operatorname{dom} f$ and $m \in \operatorname{dom} f$ holds $\pi_n f = \pi_m f$.

One can prove the following three propositions:

(1) Let D be a non empty set and f be a finite sequence of elements of D. If f yields $\pi_{\operatorname{len} f} f$ just once, then $(\pi_{\operatorname{len} f} f) \hookleftarrow f = \operatorname{len} f$.

(2) For every non empty set D and for every finite sequence f of elements of D holds $f_{\lfloor \operatorname{len} f} = \emptyset$.

(3) For every non empty set D and for every non empty finite sequence f of elements of D holds $\pi_{\operatorname{len} f} f \in \operatorname{rng} f$.

Let D be a non empty set, let f be a finite sequence of elements of D, and let y be a set. Let us observe that f yields y just once if and only if:

(Def. 2) There exists a set x such that $x \in \operatorname{dom} f$ and $y = \pi_x f$ and for every set z such that $z \in \operatorname{dom} f$ and $z \neq x$ holds $\pi_z f \neq y$.

The following propositions are true:

(4) Let D be a non empty set and f be a finite sequence of elements of D. If f yields $\pi_{\operatorname{len} f} f$ just once, then $f -: \pi_{\operatorname{len} f} f = f$.

(5) Let D be a non empty set and f be a finite sequence of elements of D. If f yields $\pi_{\operatorname{len} f} f$ just once, then $f :- \pi_{\operatorname{len} f} f = \langle \pi_{\operatorname{len} f} f \rangle$.

(6) $1 \leqslant \operatorname{len}(f :- p)$.

(7) Let D be a non empty set, p be an element of D, and f be a finite sequence of elements of D. If $p \in \operatorname{rng} f$, then $\operatorname{len}(f :- p) \leqslant \operatorname{len} f$.

(8) For every non empty set D and for every circular non empty finite sequence f of elements of D holds $\operatorname{Rev}(f)$ is circular.

2. About the Rotation

In the sequel D denotes a non empty set, p denotes an element of D, and f denotes a finite sequence of elements of D.

We now state several propositions:

(9) If $p \in \operatorname{rng} f$ and $1 \leqslant i$ and $i \leqslant \operatorname{len}(f :- p)$, then $\pi_i f^{p}_{\circlearrowleft} = \pi_{(i -' 1) + p \leftarrow\!\!\shortmid f} f$.

(10) If $p \in \operatorname{rng} f$ and $p \leftarrow\!\!\shortmid f \leqslant i$ and $i \leqslant \operatorname{len} f$, then $\pi_i f = \pi_{(i+1) -' p \leftarrow\!\!\shortmid f} f^{p}_{\circlearrowleft}$.

(11) If $p \in \operatorname{rng} f$, then $\pi_{\operatorname{len}(f :- p)} f^{p}_{\circlearrowleft} = \pi_{\operatorname{len} f} f$.

(12) If $p \in \operatorname{rng} f$ and $\operatorname{len}(f :- p) < i$ and $i \leqslant \operatorname{len} f$, then $\pi_i f^{p}_{\circlearrowleft} = \pi_{(i + p \leftarrow\!\!\shortmid f) -' \operatorname{len} f} f$.

(13) If $p \in \operatorname{rng} f$ and $1 < i$ and $i \leqslant p \leftarrow\!\!\shortmid f$, then $\pi_i f = \pi_{(i + \operatorname{len} f) -' p \leftarrow\!\!\shortmid f} f^{p}_{\circlearrowleft}$.

(14) $\operatorname{len}(f^{p}_{\circlearrowleft}) = \operatorname{len} f$.

(15) $\operatorname{dom}(f^{p}_{\circlearrowleft}) = \operatorname{dom} f$.

(16) Let D be a non empty set, f be a circular finite sequence of elements of D, and p be an element of D. If for every i such that $1 < i$ and $i < \operatorname{len} f$ holds $\pi_i f \neq \pi_1 f$, then $(f^{p}_{\circlearrowleft})^{\pi_1 f}_{\circlearrowleft} = f$.

3. Rotating Circular Ones

In the sequel f is a circular finite sequence of elements of D.

The following propositions are true:

(17) If $p \in \operatorname{rng} f$ and $\operatorname{len}(f :- p) \leqslant i$ and $i \leqslant \operatorname{len} f$, then $\pi_i f^{p}_{\circlearrowleft} = \pi_{(i + p \leftarrow\!\!\shortmid f) -' \operatorname{len} f} f$.

(18) If $p \in \operatorname{rng} f$ and $1 \leqslant i$ and $i \leqslant p \leftarrow\!\!\shortmid f$, then $\pi_i f = \pi_{(i + \operatorname{len} f) -' p \leftarrow\!\!\shortmid f} f^{p}_{\circlearrowleft}$.

Let D be a non trivial set. Note that there exists a finite sequence of elements of D which is non constant and circular.

Let D be a non trivial set, let p be an element of D, and let f be a non constant circular finite sequence of elements of D. Note that $f^{p}_{\circlearrowleft}$ is non constant.

4. Finite Sequence on the Plane

The following proposition is true

(19) For every non empty natural number n holds $0_{\mathcal{E}^n_{\mathrm{T}}} \neq 1.\mathrm{REAL}\, n$.

Let n be a non empty natural number. Note that $\mathcal{E}^n_{\mathrm{T}}$ is non trivial.

In the sequel f, g are finite sequences of elements of $\mathcal{E}^2_{\mathrm{T}}$.

Next we state four propositions:

(20) If $\operatorname{rng} f \subseteq \operatorname{rng} g$, then $\operatorname{rng} \mathbf{X}\text{-coordinate}(f) \subseteq \operatorname{rng} \mathbf{X}\text{-coordinate}(g)$.

(21) If $\operatorname{rng} f = \operatorname{rng} g$, then $\operatorname{rng} \mathbf{X}\text{-coordinate}(f) = \operatorname{rng} \mathbf{X}\text{-coordinate}(g)$.

(22) If $\operatorname{rng} f \subseteq \operatorname{rng} g$, then $\operatorname{rng} \mathbf{Y}\text{-coordinate}(f) \subseteq \operatorname{rng} \mathbf{Y}\text{-coordinate}(g)$.

(23) If $\operatorname{rng} f = \operatorname{rng} g$, then $\operatorname{rng} \mathbf{Y}\text{-coordinate}(f) = \operatorname{rng} \mathbf{Y}\text{-coordinate}(g)$.

5. Rotating Finite Sequence on the Plane

In the sequel p denotes a point of $\mathcal{E}^2_{\rm T}$ and f denotes a finite sequence of elements of $\mathcal{E}^2_{\rm T}$.

Let p be a point of $\mathcal{E}^2_{\rm T}$ and let f be a special circular finite sequence of elements of $\mathcal{E}^2_{\rm T}$. Observe that $f^p_{\circlearrowleft}$ is special.

The following propositions are true:

(24) If $p \in \operatorname{rng} f$ and $1 \leqslant i$ and $i < \operatorname{len}(f :- p)$, then $\mathcal{L}(f^p_{\circlearrowleft}, i) = \mathcal{L}(f, (i -' 1) + p \leftarrow f)$.

(25) If $p \in \operatorname{rng} f$ and $p \leftarrow f \leqslant i$ and $i < \operatorname{len} f$, then $\mathcal{L}(f, i) = \mathcal{L}(f^p_{\circlearrowleft}, (i -' p \leftarrow f) + 1)$.

(26) For every circular finite sequence f of elements of $\mathcal{E}^2_{\rm T}$ holds $\operatorname{Inc}(\mathbf{X}\text{-coordinate}(f)) = \operatorname{Inc}(\mathbf{X}\text{-coordinate}(f^p_{\circlearrowleft}))$.

(27) For every circular finite sequence f of elements of $\mathcal{E}^2_{\rm T}$ holds $\operatorname{Inc}(\mathbf{Y}\text{-coordinate}(f)) = \operatorname{Inc}(\mathbf{Y}\text{-coordinate}(f^p_{\circlearrowleft}))$.

(28) For every non empty circular finite sequence f of elements of $\mathcal{E}^2_{\rm T}$ holds the Go-board of $f^p_{\circlearrowleft}$ = the Go-board of f.

(29) For every non constant standard special circular sequence f holds $\operatorname{Rev}(f^p_{\circlearrowleft}) = (\operatorname{Rev}(f))^p_{\circlearrowleft}$.

6. Rotating Circular Ones (on the Plane)

In the sequel f is a circular finite sequence of elements of $\mathcal{E}^2_{\rm T}$.

We now state two propositions:

(30) For every circular s.c.c. finite sequence f of elements of $\mathcal{E}^2_{\rm T}$ such that $\operatorname{len} f > 4$ holds $\mathcal{L}(f, \operatorname{len} f -' 1) \cap \mathcal{L}(f, 1) = \{\pi_1 f\}$.

(31) If $p \in \operatorname{rng} f$ and $\operatorname{len}(f:-p) \leqslant i$ and $i < \operatorname{len} f$, then $\mathcal{L}(f^p_{\circlearrowleft}, i) = \mathcal{L}(f, (i+p \leftarrow f) -' \operatorname{len} f)$.

Let p be a point of $\mathcal{E}^2_{\rm T}$ and let f be a circular s.c.c. finite sequence of elements of $\mathcal{E}^2_{\rm T}$. One can check that $f^p_{\circlearrowleft}$ is s.c.c..

Let p be a point of $\mathcal{E}^2_{\rm T}$ and let f be a non constant standard special circular sequence. Observe that $f^p_{\circlearrowleft}$ is unfolded.

Next we state three propositions:

(32) If $p \in \operatorname{rng} f$ and $1 \leqslant i$ and $i < p \leftarrow f$, then $\mathcal{L}(f, i) = \mathcal{L}(f^p_{\circlearrowleft}, (i + \operatorname{len} f) -' p \leftarrow f)$.

(33) $\widetilde{\mathcal{L}}(f^p_{\circlearrowleft}) = \widetilde{\mathcal{L}}(f)$.

(34) Let G be a Go-board. Then f is a sequence which elements belong to G if and only if $f^p_{\circlearrowleft}$ is a sequence which elements belong to G.

Let p be a point of $\mathcal{E}^2_{\rm T}$ and let f be a standard non empty circular finite sequence of elements of $\mathcal{E}^2_{\rm T}$. One can verify that $f^p_{\circlearrowleft}$ is standard.

One can prove the following three propositions:

(35) Let f be a non constant standard special circular sequence and given p, k. If $p \in \operatorname{rng} f$ and $1 \leqslant k$ and $k < p \leftarrow\!\!\!\shortmid f$, then $\operatorname{leftcell}(f, k) = \operatorname{leftcell}(f^p_{\circlearrowleft}, (k + \operatorname{len} f) -' p \leftarrow\!\!\!\shortmid f)$.

(36) For every non constant standard special circular sequence f holds $\operatorname{LeftComp}(f^p_{\circlearrowleft}) = \operatorname{LeftComp}(f)$.

(37) For every non constant standard special circular sequence f holds $\operatorname{RightComp}(f^p_{\circlearrowleft}) = \operatorname{RightComp}(f)$.

7. The Orientation

Let p be a point of $\mathcal{E}^2_{\rm T}$ and let f be a clockwise oriented non constant standard special circular sequence. One can verify that $f^p_{\circlearrowleft}$ is clockwise oriented.

One can prove the following proposition

(38) Let f be a non constant standard special circular sequence. Then f is clockwise oriented or $\operatorname{Rev}(f)$ is clockwise oriented.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.
[2] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[3] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[4] Józef Białas. Group and field definitions. *Formalized Mathematics*, 1(**3**):433–439, 1990.
[5] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.
[6] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[7] Czesław Byliński. Some properties of restrictions of finite sequences. *Formalized Mathematics*, 5(**2**):241–245, 1996.
[8] Czesław Byliński and Piotr Rudnicki. Bounding boxes for compact sets in $\mathcal{E}^2$. *Formalized Mathematics*, 6(**3**):427–440, 1997.
[9] Agata Darmochwał. Compact spaces. *Formalized Mathematics*, 1(**2**):383–386, 1990.
[10] Agata Darmochwał. The Euclidean space. *Formalized Mathematics*, 2(**4**):599–603, 1991.
[11] Agata Darmochwał and Yatsuka Nakamura. The topological space $\mathcal{E}^2_{\rm T}$. Arcs, line segments and special polygonal arcs. *Formalized Mathematics*, 2(**5**):617–621, 1991.
[12] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[13] Katarzyna Jankowska. Matrices. Abelian group of matrices. *Formalized Mathematics*, 2(**4**):475–480, 1991.
[14] Jarosław Kotowicz. Functions and finite sequences of real numbers. *Formalized Mathematics*, 3(**2**):275–278, 1992.
[15] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part I. *Formalized Mathematics*, 3(**1**):107–115, 1992.

[16] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part II. *Formalized Mathematics*, 3(**1**):117–121, 1992.
[17] Yatsuka Nakamura and Adam Grabowski. Bounding boxes for special sequences in $\mathcal{E}^2$. *Formalized Mathematics*, 7(**1**):115–121, 1998.
[18] Yatsuka Nakamura and Andrzej Trybulec. Decomposing a Go-board into cells. *Formalized Mathematics*, 5(**3**):323–328, 1996.
[19] Yatsuka Nakamura, Andrzej Trybulec, and Czesław Byliński. Bounded domains and unbounded domains. *Formalized Mathematics*, 8(**1**):1–13, 1999.
[20] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[21] Beata Padlewska. Connected spaces. *Formalized Mathematics*, 1(**1**):239–244, 1990.
[22] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[23] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.
[24] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[25] Andrzej Trybulec. Left and right component of the complement of a special closed curve. *Formalized Mathematics*, 5(**4**):465–468, 1996.
[26] Andrzej Trybulec. On the decomposition of finite sequences. *Formalized Mathematics*, 5(**3**):317–322, 1996.
[27] Andrzej Trybulec and Yatsuka Nakamura. On the order on a special polygon. *Formalized Mathematics*, 6(**4**):541–548, 1997.
[28] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[29] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[30] Mirosław Wysocki and Agata Darmochwał. Subsets of topological spaces. *Formalized Mathematics*, 1(**1**):231–237, 1990.

Received January 21, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

On the Components of the Complement of a Special Polygonal Curve

Andrzej Trybulec[1]
University of Bialystok

Yatsuka Nakamura
Shinshu University
Nagano

Summary. By the special polygonal curve we meana simple closed curve, that is a polygone and moreover has edges parallel to axes. We continue the formalization of the Takeuti-Nakamura proof [21] of the Jordan curve theorem. In the paper we prove that the complement of the special polygonal curve consists of at least two components. With the theorem which has at most two components we completed the theorem that a special polygonal curve cuts the plane into exactly two components.

MML Identifier: SPRECT_4.

The articles [22], [29], [1], [11], [3], [2], [27], [28], [19], [12], [20], [30], [7], [8], [9], [16], [4], [24], [13], [14], [15], [5], [18], [23], [17], [6], [10], [26], and [25] provide the terminology and notation for this paper.

In this paper j denotes a natural number.

One can prove the following propositions:

(1) Let f be a S-sequence in $\mathbb{R}^2$ and Q be a non empty compact subset of $\mathcal{E}^2_{\rm T}$. If $\widetilde{\mathcal{L}}(f)$ meets Q and $\pi_1 f \notin Q$, then $\widetilde{\mathcal{L}}(\downharpoonleft f, {\rm FPoint}(\widetilde{\mathcal{L}}(f), \pi_1 f, \pi_{{\rm len}\, f} f, Q)) \cap Q = \{{\rm FPoint}(\widetilde{\mathcal{L}}(f), \pi_1 f, \pi_{{\rm len}\, f} f, Q)\}$.

(2) Let f be a finite sequence of elements of $\mathcal{E}^2_{\rm T}$ and p be a point of $\mathcal{E}^2_{\rm T}$. If f is a special sequence and $p = \pi_{{\rm len}\, f} f$, then $\widetilde{\mathcal{L}}(\downharpoonleft p, f) = \{p\}$.

(3) Let f be a finite sequence of elements of $\mathcal{E}^2_{\rm T}$ and p be a point of $\mathcal{E}^2_{\rm T}$. If f is a special sequence and $p \in \widetilde{\mathcal{L}}(f)$, then $\widetilde{\mathcal{L}}(\downharpoonleft p, f) \subseteq \widetilde{\mathcal{L}}(f)$.

(4) Let f be a S-sequence in $\mathbb{R}^2$, p be a point of $\mathcal{E}^2_{\rm T}$, and given j. If $1 \leqslant j$ and $j < {\rm len}\, f$ and $p \in \widetilde{\mathcal{L}}({\rm mid}(f, j, {\rm len}\, f))$, then LE $\pi_j f$, p, $\widetilde{\mathcal{L}}(f)$, $\pi_1 f$, $\pi_{{\rm len}\, f} f$.

[1]The work had been done when the first author visited Nagano in fall of 1998.

ISSN 1426–2630

(5) Let f be a S-sequence in $\mathbb{R}^2$, p, q be points of $\mathcal{E}^2_{\rm T}$, and given j. If $1 \leqslant j$ and $j < \mathrm{len}\, f$ and $p \in \mathcal{L}(f, j)$ and $q \in \mathcal{L}(p, \pi_{j+1} f)$, then LE p, q, $\widetilde{\mathcal{L}}(f)$, $\pi_1 f$, $\pi_{\mathrm{len}\, f} f$.

(6) Let f be a S-sequence in $\mathbb{R}^2$ and Q be a non empty compact subset of $\mathcal{E}^2_{\rm T}$. If $\widetilde{\mathcal{L}}(f)$ meets Q and $\pi_{\mathrm{len}\, f} f \notin Q$, then $\widetilde{\mathcal{L}}(\downharpoonleft \mathrm{LPoint}(\widetilde{\mathcal{L}}(f), \pi_1 f, \pi_{\mathrm{len}\, f} f, Q), f) \cap Q = \{\mathrm{LPoint}(\widetilde{\mathcal{L}}(f), \pi_1 f, \pi_{\mathrm{len}\, f} f, Q)\}$.

(7) For every non constant standard special circular sequence f holds $\mathrm{LeftComp}(f) \neq \mathrm{RightComp}(f)$.

References

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

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

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

[4] Czesław Byliński. Some properties of restrictions of finite sequences. *Formalized Mathematics*, 5(**2**):241–245, 1996.

[5] Czesław Byliński and Yatsuka Nakamura. Special polygons. *Formalized Mathematics*, 5(**2**):247–252, 1996.

[6] Czesław Byliński and Piotr Rudnicki. Bounding boxes for compact sets in $\mathcal{E}^2$. *Formalized Mathematics*, 6(**3**):427–440, 1997.

[7] Agata Darmochwał. Compact spaces. *Formalized Mathematics*, 1(**2**):383–386, 1990.

[8] Agata Darmochwał. The Euclidean space. *Formalized Mathematics*, 2(**4**):599–603, 1991.

[9] Agata Darmochwał and Yatsuka Nakamura. The topological space $\mathcal{E}^2_{\rm T}$. Arcs, line segments and special polygonal arcs. *Formalized Mathematics*, 2(**5**):617–621, 1991.

[10] Adam Grabowski and Yatsuka Nakamura. The ordering of points on a curve. Part II. *Formalized Mathematics*, 6(**4**):467–473, 1997.

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

[12] Katarzyna Jankowska. Matrices. Abelian group of matrices. *Formalized Mathematics*, 2(**4**):475–480, 1991.

[13] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part I. *Formalized Mathematics*, 3(**1**):107–115, 1992.

[14] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part II. *Formalized Mathematics*, 3(**1**):117–121, 1992.

[15] Yatsuka Nakamura and Czesław Byliński. Extremal properties of vertices on special polygons. Part I. *Formalized Mathematics*, 5(**1**):97–102, 1996.

[16] Yatsuka Nakamura and Jarosław Kotowicz. Connectedness conditions using polygonal arcs. *Formalized Mathematics*, 3(**1**):101–106, 1992.

[17] Yatsuka Nakamura and Roman Matuszewski. Reconstructions of special sequences. *Formalized Mathematics*, 6(**2**):255–263, 1997.

[18] Yatsuka Nakamura and Andrzej Trybulec. Decomposing a Go-board into cells. *Formalized Mathematics*, 5(**3**):323–328, 1996.

[19] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.

[20] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.

[21] Yukio Takeuchi and Yatsuka Nakamura. On the Jordan curve theorem. Technical Report 19804, Dept. of Information Eng., Shinshu University, 500 Wakasato, Nagano city, Japan, April 1980.

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

[23] Andrzej Trybulec. Left and right component of the complement of a special closed curve. *Formalized Mathematics*, 5(**4**):465–468, 1996.

[24] Andrzej Trybulec. On the decomposition of finite sequences. *Formalized Mathematics*, 5(**3**):317–322, 1996.

[25] Andrzej Trybulec and Yatsuka Nakamura. On the order on a special polygon. *Formalized Mathematics*, 6(**4**):541–548, 1997.

[26] Andrzej Trybulec and Yatsuka Nakamura. On the rectangular finite sequences of the points of the plane. *Formalized Mathematics*, 6(**4**):531–539, 1997.

[27] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.

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

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

[30] Mirosław Wysocki and Agata Darmochwał. Subsets of topological spaces. *Formalized Mathematics*, 1(**1**):231–237, 1990.

Received January 21, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Gauges

Czesław Byliński
University of Białystok

MML Identifier: JORDAN8.

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

We follow the rules: i, i_1, i_2, j, j_1, j_2, k, m, n are natural numbers, D is a non empty set, and f is a finite sequence of elements of D.

We now state two propositions:

(1) If $\operatorname{len} f \geqslant 2$, then $f{\restriction}2 = \langle \pi_1 f, \pi_2 f \rangle$.

(2) If $k+1 \leqslant \operatorname{len} f$, then $f{\restriction}(k+1) = (f{\restriction}k) \frown \langle \pi_{k+1} f \rangle$.

In the sequel f denotes a finite sequence of elements of $\mathcal{E}^2_{\rm T}$, G denotes a Go-board, and p denotes a point of $\mathcal{E}^2_{\rm T}$.

The following propositions are true:

(3) $\varepsilon_{(\text{the carrier of } \mathcal{E}^2_{\rm T})}$ is a sequence which elements belong to G.

(4) If f is a sequence which elements belong to G, then $f{\restriction}m$ is a sequence which elements belong to G.

(5) If f is a sequence which elements belong to G, then $f_{\lfloor m}$ is a sequence which elements belong to G.

(6) Suppose $1 \leqslant k$ and $k+1 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G. Then there exist natural numbers i_1, j_1, i_2, j_2 such that
 (i) $\langle i_1, j_1 \rangle \in$ the indices of G,
 (ii) $\pi_k f = G_{i_1,j_1}$,
 (iii) $\langle i_2, j_2 \rangle \in$ the indices of G,
 (iv) $\pi_{k+1} f = G_{i_2,j_2}$, and
 (v) $i_1 = i_2$ and $j_1 + 1 = j_2$ or $i_1 + 1 = i_2$ and $j_1 = j_2$ or $i_1 = i_2 + 1$ and $j_1 = j_2$ or $i_1 = i_2$ and $j_1 = j_2 + 1$.

(7) Let f be a non empty finite sequence of elements of $\mathcal{E}^2_{\rm T}$. Suppose f is a sequence which elements belong to G. Then f is standard and special.

ISSN 1426–2630

(8) Let f be a non empty finite sequence of elements of $\mathcal{E}^2_{\mathrm{T}}$. Suppose $\mathrm{len}\, f \geqslant 2$ and f is a sequence which elements belong to G. Then f is non constant.

(9) Let f be a non empty finite sequence of elements of $\mathcal{E}^2_{\mathrm{T}}$. Suppose that

(i) f is a sequence which elements belong to G,

(ii) there exist i, j such that $\langle i,\, j\rangle \in$ the indices of G and $p = G_{i,j}$, and

(iii) for all i_1, j_1, i_2, j_2 such that $\langle i_1,\, j_1\rangle \in$ the indices of G and $\langle i_2,\, j_2\rangle \in$ the indices of G and $\pi_{\mathrm{len}\, f} f = G_{i_1,j_1}$ and $p = G_{i_2,j_2}$ holds $|i_2 - i_1| + |j_2 - j_1| = 1$.

Then $f \frown \langle p\rangle$ is a sequence which elements belong to G.

(10) If $i + k < \mathrm{len}\, G$ and $1 \leqslant j$ and $j < \mathrm{width}\, G$ and $\mathrm{cell}(G, i, j)$ meets $\mathrm{cell}(G, i + k, j)$, then $k \leqslant 1$.

(11) For every non empty compact subset C of $\mathcal{E}^2_{\mathrm{T}}$ holds C is vertical iff $\text{E-bound}\, C \leqslant \text{W-bound}\, C$.

(12) For every non empty compact subset C of $\mathcal{E}^2_{\mathrm{T}}$ holds C is horizontal iff $\text{N-bound}\, C \leqslant \text{S-bound}\, C$.

Let C be a non empty subset of $\mathcal{E}^2_{\mathrm{T}}$ and let n be a natural number. The functor $\mathrm{Gauge}(C, n)$ yielding a matrix over $\mathcal{E}^2_{\mathrm{T}}$ is defined by the conditions (Def. 1).

(Def. 1)(i) $\mathrm{len}\, \mathrm{Gauge}(C, n) = 2^n + 3$,

(ii) $\mathrm{len}\, \mathrm{Gauge}(C, n) = \mathrm{width}\, \mathrm{Gauge}(C, n)$, and

(iii) for all i, j such that $\langle i,\, j\rangle \in$ the indices of $\mathrm{Gauge}(C, n)$ holds $(\mathrm{Gauge}(C, n))_{i,j} = [\text{W-bound}\, C + \frac{\text{E-bound}\, C - \text{W-bound}\, C}{2^n} \cdot (i - 2), \text{S-bound}\, C + \frac{\text{N-bound}\, C - \text{S-bound}\, C}{2^n} \cdot (j - 2)]$.

Let C be a compact non empty subset of $\mathcal{E}^2_{\mathrm{T}}$ and let n be a natural number. Note that $\mathrm{Gauge}(C, n)$ is non trivial line **X**-constant and column **Y**-constant.

In the sequel C is a compact non vertical non horizontal non empty subset of $\mathcal{E}^2_{\mathrm{T}}$.

Let us consider C, n. Observe that $\mathrm{Gauge}(C, n)$ is line **Y**-increasing and column **X**-increasing.

The following propositions are true:

(13) $\mathrm{len}\, \mathrm{Gauge}(C, n) \geqslant 4$.

(14) If $1 \leqslant j$ and $j \leqslant \mathrm{len}\, \mathrm{Gauge}(C, n)$, then $((\mathrm{Gauge}(C, n))_{2,j})_{\mathbf{1}} = \text{W-bound}\, C$.

(15) If $1 \leqslant j$ and $j \leqslant \mathrm{len}\, \mathrm{Gauge}(C, n)$, then $((\mathrm{Gauge}(C, n))_{\mathrm{len}\, \mathrm{Gauge}(C,n) -' 1, j})_{\mathbf{1}} = \text{E-bound}\, C$.

(16) If $1 \leqslant i$ and $i \leqslant \mathrm{len}\, \mathrm{Gauge}(C, n)$, then $((\mathrm{Gauge}(C, n))_{i,2})_{\mathbf{2}} = \text{S-bound}\, C$.

(17) If $1 \leqslant i$ and $i \leqslant \mathrm{len}\, \mathrm{Gauge}(C, n)$, then $((\mathrm{Gauge}(C, n))_{i, \mathrm{len}\, \mathrm{Gauge}(C,n) -' 1})_{\mathbf{2}} = \text{N-bound}\, C$.

(18) If $i \leqslant \mathrm{len}\, \mathrm{Gauge}(C, n)$, then $\mathrm{cell}(\mathrm{Gauge}(C, n), i, \mathrm{len}\, \mathrm{Gauge}(C, n)) \cap C = \emptyset$.

(19) If $j \leqslant \mathrm{len}\, \mathrm{Gauge}(C, n)$, then $\mathrm{cell}(\mathrm{Gauge}(C, n), \mathrm{len}\, \mathrm{Gauge}(C, n), j) \cap C = \emptyset$.

(20) If $i \leqslant \mathrm{len}\, \mathrm{Gauge}(C, n)$, then $\mathrm{cell}(\mathrm{Gauge}(C, n), i, 0) \cap C = \emptyset$.

(21) If $j \leqslant \operatorname{len} \operatorname{Gauge}(C, n)$, then $\operatorname{cell}(\operatorname{Gauge}(C, n), 0, j) \cap C = \emptyset$.

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[2] Grzegorz Bancerek. Countable sets and Hessenberg's theorem. *Formalized Mathematics*, 2(**1**):65–69, 1991.
[3] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[4] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[5] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[6] Czesław Byliński and Piotr Rudnicki. Bounding boxes for compact sets in $\mathcal{E}^2$. *Formalized Mathematics*, 6(**3**):427–440, 1997.
[7] Agata Darmochwał. Compact spaces. *Formalized Mathematics*, 1(**2**):383–386, 1990.
[8] Agata Darmochwał. The Euclidean space. *Formalized Mathematics*, 2(**4**):599–603, 1991.
[9] Agata Darmochwał and Yatsuka Nakamura. The topological space $\mathcal{E}^2_{\mathrm{T}}$. Arcs, line segments and special polygonal arcs. *Formalized Mathematics*, 2(**5**):617–621, 1991.
[10] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[11] Katarzyna Jankowska. Matrices. Abelian group of matrices. *Formalized Mathematics*, 2(**4**):475–480, 1991.
[12] Jarosław Kotowicz. Functions and finite sequences of real numbers. *Formalized Mathematics*, 3(**2**):275–278, 1992.
[13] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part I. *Formalized Mathematics*, 3(**1**):107–115, 1992.
[14] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part II. *Formalized Mathematics*, 3(**1**):117–121, 1992.
[15] Yatsuka Nakamura and Czesław Byliński. Extremal properties of vertices on special polygons. Part I. *Formalized Mathematics*, 5(**1**):97–102, 1996.
[16] Yatsuka Nakamura and Andrzej Trybulec. Decomposing a Go-board into cells. *Formalized Mathematics*, 5(**3**):323–328, 1996.
[17] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[18] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[19] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.
[20] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[21] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[22] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[23] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[24] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received January 22, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

The Ring of Integers, Euclidean Rings and Modulo Integers

Christoph Schwarzweller
University of Tübingen

Summary. In this article we introduce the ring of Integers, Euclidean rings and Integers modulo p. In particular we prove that the Ring of Integers is an Euclidean ring and that the Integers modulo p constitutes a field if and only if p is a prime.

MML Identifier: `INT_3`.

The notation and terminology used here are introduced in the following papers: [16], [21], [20], [17], [22], [4], [5], [14], [10], [12], [13], [3], [8], [7], [15], [18], [2], [6], [11], [9], [1], and [19].

1. The Ring of Integers

The binary operation multint on $\mathbb{Z}$ is defined as follows:

(Def. 1) For all elements a, b of $\mathbb{Z}$ holds $(\text{multint})(a, b) = \cdot_{\mathbb{R}}(a, b)$.

The unary operation compint on $\mathbb{Z}$ is defined as follows:

(Def. 2) For every element a of $\mathbb{Z}$ holds $(\text{compint})(a) = -_{\mathbb{R}}(a)$.

The double loop structure INT.Ring is defined by:

(Def. 3) $\text{INT.Ring} = \langle \mathbb{Z}, +_{\mathbb{Z}}, \text{multint}, 1 (\in \mathbb{Z}), 0 (\in \mathbb{Z}) \rangle$.

Let us mention that INT.Ring is strict and non empty.

Let us mention that INT.Ring is Abelian add-associative right zeroed right complementable well unital distributive commutative associative integral domain-like and non degenerated.

Let a, b be elements of the carrier of INT.Ring. The predicate $a \leqslant b$ is defined by:

ISSN 1426–2630

(Def. 4) There exist integers a', b' such that $a' = a$ and $b' = b$ and $a' \leqslant b'$.

Let us notice that the predicate $a \leqslant b$ is reflexive and connected. We introduce $b \geqslant a$ as a synonym of $a \leqslant b$. We introduce $b < a$ and $a > b$ as antonyms of $a \leqslant b$.

Let a be an element of the carrier of INT.Ring. The functor $|a|$ yields an element of the carrier of INT.Ring and is defined as follows:

(Def. 5) $|a| = \begin{cases} a, & \text{if } a \geqslant 0_{\mathrm{INT.Ring}}, \\ -a, & \text{otherwise.} \end{cases}$

The function absint from the carrier of INT.Ring into $\mathbb{N}$ is defined as follows:

(Def. 6) For every element a of the carrier of INT.Ring holds $(\mathrm{absint})(a) = |\square|_{\mathbb{R}}(a)$.

One can prove the following two propositions:

(1) For every element a of the carrier of INT.Ring holds $(\mathrm{absint})(a) = |a|$.

(2) Let a, b, q_1, q_2, r_1, r_2 be elements of the carrier of INT.Ring. Suppose $b \neq 0_{\mathrm{INT.Ring}}$ and $a = q_1 \cdot b + r_1$ and $0_{\mathrm{INT.Ring}} \leqslant r_1$ and $r_1 < |b|$ and $a = q_2 \cdot b + r_2$ and $0_{\mathrm{INT.Ring}} \leqslant r_2$ and $r_2 < |b|$. Then $q_1 = q_2$ and $r_1 = r_2$.

Let a, b be elements of the carrier of INT.Ring. Let us assume that $b \neq 0_{\mathrm{INT.Ring}}$. The functor $a \div b$ yields an element of the carrier of INT.Ring and is defined by:

(Def. 7) There exists an element r of the carrier of INT.Ring such that $a = (a \div b) \cdot b + r$ and $0_{\mathrm{INT.Ring}} \leqslant r$ and $r < |b|$.

Let a, b be elements of the carrier of INT.Ring. Let us assume that $b \neq 0_{\mathrm{INT.Ring}}$. The functor $a \bmod b$ yields an element of the carrier of INT.Ring and is defined as follows:

(Def. 8) There exists an element q of the carrier of INT.Ring such that $a = q \cdot b + (a \bmod b)$ and $0_{\mathrm{INT.Ring}} \leqslant a \bmod b$ and $a \bmod b < |b|$.

Next we state the proposition

(3) For all elements a, b of the carrier of INT.Ring such that $b \neq 0_{\mathrm{INT.Ring}}$ holds $a = (a \div b) \cdot b + (a \bmod b)$.

2. Euclidean Rings

Let I be a non empty double loop structure. We say that I is Euclidian if and only if the condition (Def. 9) is satisfied.

(Def. 9) There exists a function f from the carrier of I into $\mathbb{N}$ such that for all elements a, b of the carrier of I if $b \neq 0_I$, then there exist elements q, r of the carrier of I such that $a = q \cdot b + r$ but $r = 0_I$ or $f(r) < f(b)$.

One can check that INT.Ring is Euclidian.

Let us observe that there exists a ring which is strict, Euclidian, integral domain-like, non degenerated, well unital, and distributive.

A EuclidianRing is a Euclidian integral domain-like non degenerated well unital distributive ring.

Let us mention that there exists a EuclidianRing which is strict.

Let E be a Euclidian non empty double loop structure. A function from the carrier of E into $\mathbb{N}$ is said to be a DegreeFunction of E if it satisfies the condition (Def. 10).

(Def. 10) Let a, b be elements of the carrier of E. Suppose $b \neq 0_E$. Then there exist elements q, r of the carrier of E such that $a = q \cdot b + r$ but $r = 0_E$ or $\mathrm{it}(r) < \mathrm{it}(b)$.

Next we state the proposition

(4) Every EuclidianRing is a gcdDomain.

Let us note that every integral domain-like non degenerated Abelian add-associative right zeroed right complementable associative commutative right unital right-distributive non empty double loop structure which is Euclidian is also gcd-like.

absint is a DegreeFunction of INT.Ring.

One can prove the following proposition

(5) Every commutative associative left unital field-like right zeroed non empty double loop structure is Euclidian.

Let us observe that every non empty double loop structure which is commutative, associative, left unital, field-like, right zeroed, and field-like is also Euclidian.

One can prove the following proposition

(6) Let F be a commutative associative left unital field-like right zeroed non empty double loop structure. Then every function from the carrier of F into $\mathbb{N}$ is a DegreeFunction of F.

3. Some Theorems about Div and Mod

The following propositions are true:

(7) Let n be a natural number. Suppose $n > 0$. Let a be an integer and a' be a natural number. If $a' = a$, then $a \div n = a' \div n$ and $a \bmod n = a' \bmod n$.

(8) For every natural number n such that $n > 0$ and for all integers a, k holds $(a + n \cdot k) \div n = (a \div n) + k$ and $(a + n \cdot k) \bmod n = a \bmod n$.

(9) For every natural number n such that $n > 0$ and for every integer a holds $a \bmod n \geqslant 0$ and $a \bmod n < n$.

(10) Let n be a natural number. Suppose $n > 0$. Let a be an integer. Then
 (i) if $0 \leqslant a$ and $a < n$, then $a \bmod n = a$, and
 (ii) if $0 > a$ and $a \geqslant -n$, then $a \bmod n = n + a$.

(11) For every natural number n such that $n > 0$ and for every integer a holds $a \bmod n = 0$ iff $n \mid a$.

(12) For every natural number n such that $n > 0$ and for all integers a, b holds $a \bmod n = b \bmod n$ iff $a \equiv b (\bmod n)$.

(13) For every natural number n such that $n > 0$ and for every integer a holds $a \bmod n \bmod n = a \bmod n$.

(14) For every natural number n such that $n > 0$ and for all integers a, b holds $(a + b) \bmod n = ((a \bmod n) + (b \bmod n)) \bmod n$.

(15) For every natural number n such that $n > 0$ and for all integers a, b holds $a \cdot b \bmod n = (a \bmod n) \cdot (b \bmod n) \bmod n$.

(16) For all integers a, b there exist integers s, t such that $a \gcd b = s \cdot a + t \cdot b$.

4. Modulo Integers

Let n be a natural number. Let us assume that $n > 0$. The functor multint n yielding a binary operation on $\mathbb{Z}_n$ is defined as follows:

(Def. 11) For all elements k, l of $\mathbb{Z}_n$ holds $(\mathrm{multint}\, n)(k, l) = k \cdot l \bmod n$.

Let n be a natural number. Let us assume that $n > 0$. The functor compint n yielding a unary operation on $\mathbb{Z}_n$ is defined by:

(Def. 12) For every element k of $\mathbb{Z}_n$ holds $(\mathrm{compint}\, n)(k) = (n - k) \bmod n$.

Next we state three propositions:

(17) Let n be a natural number. Suppose $n > 0$. Let a, b be elements of $\mathbb{Z}_n$. Then
 (i) $a + b < n$ iff $+_n(a, b) = a + b$, and
 (ii) $a + b \geqslant n$ iff $+_n(a, b) = (a + b) - n$.

(18) Let n be a natural number. Suppose $n > 0$. Let a, b be elements of $\mathbb{Z}_n$ and k be a natural number. Then $k \cdot n \leqslant a \cdot b$ and $a \cdot b < (k + 1) \cdot n$ if and only if $(\mathrm{multint}\, n)(a, b) = a \cdot b - k \cdot n$.

(19) Let n be a natural number. Suppose $n > 0$. Let a be an element of $\mathbb{Z}_n$. Then
 (i) $a = 0$ iff $(\mathrm{compint}\, n)(a) = 0$, and
 (ii) $a \neq 0$ iff $(\mathrm{compint}\, n)(a) = n - a$.

Let n be a natural number. The functor INT.Ring n yields a double loop structure and is defined by:

(Def. 13) $\mathrm{INT.Ring}\, n = \langle \mathbb{Z}_n, +_n, \mathrm{multint}\, n, 1(\in \mathbb{Z}_n), 0(\in \mathbb{Z}_n) \rangle$.

Let n be a natural number. Observe that INT.Ring n is strict and non empty.

We now state the proposition

(20) INT.Ring 1 is degenerated and INT.Ring 1 is a ring and INT.Ring 1 is field-like, well unital, and distributive.

Let us note that there exists a ring which is strict, degenerated, well unital, distributive, and field-like.

One can prove the following propositions:

(21) For every natural number n such that $n > 1$ holds INT.Ring n is non degenerated and INT.Ring n is a well unital distributive ring.

(22) Let p be a natural number. Suppose $p > 1$. Then INT.Ring p is an add-associative right zeroed right complementable Abelian commutative associative left unital distributive field-like non degenerated non empty double loop structure if and only if p is a prime number.

Let p be a prime number. Observe that INT.Ring p is add-associative right zeroed right complementable Abelian commutative associative left unital distributive field-like and non degenerated.

References

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

[2] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.

[3] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(**1**):175–180, 1990.

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

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

[6] Marek Chmur. The lattice of natural numbers and the sublattice of it. The set of prime numbers. *Formalized Mathematics*, 2(**4**):453–459, 1991.

[7] Agata Darmochwał. The Euclidean space. *Formalized Mathematics*, 2(**4**):599–603, 1991.

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

[9] Krzysztof Hryniewiecki. Recursive definitions. *Formalized Mathematics*, 1(**2**):321–328, 1990.

[10] Eugeniusz Kusak, Wojciech Leończuk, and Michał Muzalewski. Abelian groups, fields and vector spaces. *Formalized Mathematics*, 1(**2**):335–342, 1990.

[11] Rafał Kwiatek and Grzegorz Zwara. The divisibility of integers and integer relative primes. *Formalized Mathematics*, 1(**5**):829–832, 1990.

[12] Michał Muzalewski. Construction of rings and left-, right-, and bi-modules over a ring. *Formalized Mathematics*, 2(**1**):3–11, 1991.

[13] Henryk Oryszczyszyn and Krzysztof Prażmowski. Real functions spaces. *Formalized Mathematics*, 1(**3**):555–561, 1990.

[14] Christoph Schwarzweller. The correctness of the generic algorithms of Brown and Henrici concerning addition and multiplication in fraction fields. *Formalized Mathematics*, 6(**3**):381–388, 1997.

[15] Dariusz Surowik. Cyclic groups and some of their properties - part I. *Formalized Mathematics*, 2(**5**):623–627, 1991.

[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] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[19] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.
[20] Wojciech A. Trybulec. Vectors in real linear space. *Formalized Mathematics*, 1(**2**):291–296, 1990.
[21] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[22] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received February 4, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Logic Gates and Logical Equivalence of Adders

Yatsuka Nakamura
Shinshu University
Nagano

Summary. This is an experimental article which shows that logical correctness of logic circuits can be easily proven by the Mizar system. First, we define the notion of logic gates. Then we prove that an MSB carry of '4 Bit Carry Skip Adder' is equivalent to an MSB carry of a normal 4 bit adder. In the last theorem, we show that outputs of the '4 Bit Carry Look Ahead Adder' are equivalent to the corresponding outputs of the normal 4 bits adder. The policy here is as follows: when the functional (semantic) correctness of a system is already proven, and the correspondence of the system to a (normal) logic circuit is given, it is enough to prove the correctness of the new circuit if we only prove the logical equivalence between them. Although the article is very fundamental (it contains few environment files), it can be applied to real problems. The key of the method introduced here is to put the specification of the logic circuit into the Mizar propositional formulae, and to use the strong inference ability of the Mizar checker. The proof is done formally so that the automation of the proof writing is possible. Even in the 5.3.07 version of Mizar, it can handle a formulae of more than 100 lines, and a formula which contains more than 100 variables. This means that the Mizar system is enough to prove logical correctness of middle scaled logic circuits.

MML Identifier: GATE_1.

The articles [2] and [1] provide the terminology and notation for this paper.

1. Definition of Logical Values and Logic Gates

Let a be a set. We introduce NE a as an antonym of a is empty.

We now state three propositions:

ISSN 1426–2630

(1) For every set a such that $a = \{\emptyset\}$ holds NE a.

(2) There exists a set a such that NE a.

(3) NE $\emptyset$ iff *contradiction.*

let a be a set. The functor NOT1 a yielding a set is defined by:

(Def. 1) $$\mathrm{NOT1}\,a = \begin{cases} \emptyset, & \text{if NE } a, \\ \{\emptyset\}, & \text{otherwise.} \end{cases}$$

The following proposition is true

(4) For every set a holds NE NOT1 a iff not NE a.

In the sequel a, b are sets.

We now state the proposition

(5) NE NOT1 $\emptyset$.

Let a, b be sets. The functor AND2(a, b) yields a set and is defined by:

(Def. 2) $$\mathrm{AND2}(a, b) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ and NE } b, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Next we state the proposition

(6) For all sets a, b holds NE AND2(a, b) iff NE a and NE b.

Let a, b be sets. The functor OR2(a, b) yielding a set is defined as follows:

(Def. 3) $$\mathrm{OR2}(a, b) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ or NE } b, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Next we state the proposition

(7) For all sets a, b holds NE OR2(a, b) iff NE a or NE b.

Let a, b be sets. The functor XOR2(a, b) yields a set and is defined by:

(Def. 4) $$\mathrm{XOR2}(a, b) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ and not NE } b \text{ or not NE } a \text{ and NE } b, \\ \emptyset, & \text{otherwise.} \end{cases}$$

The following four propositions are true:

(8) For all sets a, b holds NE XOR2(a, b) iff NE a and not NE b or not NE a and NE b.

(9) NE XOR2(a, a) iff *contradiction.*

(10) NE XOR2$(a, \emptyset)$ iff NE a.

(11) NE XOR2(a, b) iff NE XOR2(b, a).

Let a, b be sets. The functor EQV2(a, b) yielding a set is defined by:

(Def. 5) $$\mathrm{EQV2}(a, b) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ iff NE } b, \\ \emptyset, & \text{otherwise.} \end{cases}$$

We now state two propositions:

(12) For all sets a, b holds NE EQV2(a, b) iff NE a iff NE b.

(13) NE EQV2(a, b) iff not NE XOR2(a, b).

Let a, b be sets. The functor NAND2(a, b) yielding a set is defined by:

(Def. 6) $\mathrm{NAND2}(a,b) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if not NE } a \text{ or not NE } b, \\ \emptyset, & \text{otherwise.} \end{cases}$

One can prove the following proposition

(14) For all sets a, b holds NE $\mathrm{NAND2}(a,b)$ iff not NE a or not NE b.

Let a, b be sets. The functor $\mathrm{NOR2}(a,b)$ yielding a set is defined as follows:

(Def. 7) $\mathrm{NOR2}(a,b) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if not NE } a \text{ and not NE } b, \\ \emptyset, & \text{otherwise.} \end{cases}$

We now state the proposition

(15) For all sets a, b holds NE $\mathrm{NOR2}(a,b)$ iff not NE a and not NE b.

Let a, b, c be sets. The functor $\mathrm{AND3}(a,b,c)$ yields a set and is defined by:

(Def. 8) $\mathrm{AND3}(a,b,c) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ and NE } b \text{ and NE } c, \\ \emptyset, & \text{otherwise.} \end{cases}$

One can prove the following proposition

(16) For all sets a, b, c holds NE $\mathrm{AND3}(a,b,c)$ iff NE a and NE b and NE c.

Let a, b, c be sets. The functor $\mathrm{OR3}(a,b,c)$ yielding a set is defined by:

(Def. 9) $\mathrm{OR3}(a,b,c) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ or NE } b \text{ or NE } c, \\ \emptyset, & \text{otherwise.} \end{cases}$

One can prove the following proposition

(17) For all sets a, b, c holds NE $\mathrm{OR3}(a,b,c)$ iff NE a or NE b or NE c.

Let a, b, c be sets. The functor $\mathrm{XOR3}(a,b,c)$ yielding a set is defined by:

(Def. 10) $\mathrm{XOR3}(a,b,c) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ and not NE } b \text{ or not NE } a \text{ and NE } b \text{ but not NE } c \text{ or not NE } a \text{ or not NE } b \text{ but not NE } a \text{ or not NE } b \text{ and NE } c, \\ \emptyset, & \text{otherwise.} \end{cases}$

We now state the proposition

(18) Let a, b, c be sets. Then NE $\mathrm{XOR3}(a,b,c)$ if and only if one of the following conditions is satisfied:

(i) NE a and not NE b or not NE a and NE b but not NE c, or

(ii) not NE a or not NE b but not NE a or not NE b and NE c.

Let a, b, c be sets. The functor $\mathrm{MAJ3}(a,b,c)$ yields a set and is defined as follows:

(Def. 11) $\mathrm{MAJ3}(a,b,c) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ and NE } b \text{ or NE } b \text{ and NE } c \text{ or NE } c \text{ and NE } a, \\ \emptyset, & \text{otherwise.} \end{cases}$

The following proposition is true

(19) For all sets a, b, c holds NE $\mathrm{MAJ3}(a,b,c)$ iff NE a and NE b or NE b and NE c or NE c and NE a.

Let a, b, c be sets. The functor $\mathrm{NAND3}(a,b,c)$ yielding a set is defined by:

(Def. 12) $\text{NAND3}(a,b,c) = \begin{cases} \text{NOT1}\,\emptyset, & \text{if not NE } a \text{ or not NE } b \text{ or not NE } c, \\ \emptyset, & \text{otherwise.} \end{cases}$

The following proposition is true

(20) For all sets a, b, c holds NE NAND3(a,b,c) iff not NE a or not NE b or not NE c.

Let a, b, c be sets. The functor NOR3(a,b,c) yields a set and is defined by:

(Def. 13) $\text{NOR3}(a,b,c) = \begin{cases} \text{NOT1}\,\emptyset, & \text{if not NE } a \text{ and not NE } b \text{ and not NE } c, \\ \emptyset, & \text{otherwise.} \end{cases}$

We now state the proposition

(21) For all sets a, b, c holds NE NOR3(a,b,c) iff not NE a and not NE b and not NE c.

Let a, b, c, d be sets. The functor AND4(a,b,c,d) yields a set and is defined by:

(Def. 14) $\text{AND4}(a,b,c,d) = \begin{cases} \text{NOT1}\,\emptyset, & \text{if NE } a \text{ and NE } b \text{ and NE } c \text{ and NE } d, \\ \emptyset, & \text{otherwise.} \end{cases}$

One can prove the following proposition

(22) For all sets a, b, c, d holds NE AND4(a,b,c,d) iff NE a and NE b and NE c and NE d.

Let a, b, c, d be sets. The functor OR4(a,b,c,d) yielding a set is defined as follows:

(Def. 15) $\text{OR4}(a,b,c,d) = \begin{cases} \text{NOT1}\,\emptyset, & \text{if NE } a \text{ or NE } b \text{ or NE } c \text{ or NE } d, \\ \emptyset, & \text{otherwise.} \end{cases}$

The following proposition is true

(23) For all sets a, b, c, d holds NE OR4(a,b,c,d) iff NE a or NE b or NE c or NE d.

Let a, b, c, d be sets. The functor NAND4(a,b,c,d) yielding a set is defined by:

(Def. 16) $\text{NAND4}(a,b,c,d) = \begin{cases} \text{NOT1}\,\emptyset, & \text{if not NE } a \text{ or not NE } b \text{ or not NE } c \text{ or not NE } d, \\ \emptyset, & \text{otherwise.} \end{cases}$

Next we state the proposition

(24) For all sets a, b, c, d holds NE NAND4(a,b,c,d) iff not NE a or not NE b or not NE c or not NE d.

Let a, b, c, d be sets. The functor NOR4(a,b,c,d) yielding a set is defined by:

(Def. 17) $\text{NOR4}(a,b,c,d) = \begin{cases} \text{NOT1}\,\emptyset, & \text{if not NE } a \text{ and not NE } b \text{ and not NE } c \text{ and not NE } d, \\ \emptyset, & \text{otherwise.} \end{cases}$

The following proposition is true

(25) For all sets a, b, c, d holds NE NOR4(a,b,c,d) iff not NE a and not NE b and not NE c and not NE d.

Let a, b, c, d, e be sets. The functor AND5(a,b,c,d,e) yielding a set is defined as follows:

(Def. 18) $$\mathrm{AND5}(a,b,c,d,e) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ and NE } b \text{ and NE } c \text{ and NE } d \\ & \quad \text{and NE } e, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Next we state the proposition

(26) For all sets a, b, c, d, e holds NE AND5(a,b,c,d,e) iff NE a and NE b and NE c and NE d and NE e.

Let a, b, c, d, e be sets. The functor OR5(a,b,c,d,e) yields a set and is defined by:

(Def. 19) $$\mathrm{OR5}(a,b,c,d,e) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ or NE } b \text{ or NE } c \text{ or NE } d \text{ or NE } e, \\ \emptyset, & \text{otherwise.} \end{cases}$$

The following proposition is true

(27) For all sets a, b, c, d, e holds NE OR5(a,b,c,d,e) iff NE a or NE b or NE c or NE d or NE e.

Let a, b, c, d, e be sets. The functor NAND5(a,b,c,d,e) yields a set and is defined as follows:

(Def. 20) $$\mathrm{NAND5}(a,b,c,d,e) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if not NE } a \text{ or not NE } b \text{ or not NE } c \\ & \quad \text{or not NE } d \text{ or not NE } e, \\ \emptyset, & \text{otherwise.} \end{cases}$$

The following proposition is true

(28) For all sets a, b, c, d, e holds NE NAND5(a,b,c,d,e) iff not NE a or not NE b or not NE c or not NE d or not NE e.

Let a, b, c, d, e be sets. The functor NOR5(a,b,c,d,e) yielding a set is defined as follows:

(Def. 21) $$\mathrm{NOR5}(a,b,c,d,e) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if not NE } a \text{ and not NE } b \text{ and not NE } c \\ & \quad \text{and not NE } d \text{ and not NE } e, \\ \emptyset, & \text{otherwise.} \end{cases}$$

We now state the proposition

(29) For all sets a, b, c, d, e holds NE NOR5(a,b,c,d,e) iff not NE a and not NE b and not NE c and not NE d and not NE e.

Let a, b, c, d, e, f be sets. The functor AND6(a,b,c,d,e,f) yielding a set is defined by:

(Def. 22) $$\mathrm{AND6}(a,b,c,d,e,f) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ and NE } b \text{ and NE } c \text{ and NE } d \\ & \quad \text{and NE } e \text{ and NE } f, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Next we state the proposition

(30) Let a, b, c, d, e, f be sets. Then NE AND6(a,b,c,d,e,f) if and only if the following conditions are satisfied:

(i) NE a,
(ii) NE b,
(iii) NE c,
(iv) NE d,
(v) NE e, and
(vi) NE f.

Let a, b, c, d, e, f be sets. The functor OR6(a,b,c,d,e,f) yielding a set is defined by:

(Def. 23) $\mathrm{OR6}(a,b,c,d,e,f) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ or NE } b \text{ or NE } c \text{ or NE } d \text{ or NE } e \text{ or NE } f, \\ \emptyset, & \text{otherwise.} \end{cases}$

The following proposition is true

(31) Let a, b, c, d, e, f be sets. Then NE OR6(a,b,c,d,e,f) if and only if one of the following conditions is satisfied:

(i) NE a, or
(ii) NE b, or
(iii) NE c, or
(iv) NE d, or
(v) NE e, or
(vi) NE f.

Let a, b, c, d, e, f be sets. The functor NAND6(a,b,c,d,e,f) yields a set and is defined by:

(Def. 24) $\mathrm{NAND6}(a,b,c,d,e,f) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if not NE } a \text{ or not NE } b \text{ or not NE } c \text{ or not NE } d \text{ or not NE } e \text{ or not NE } f, \\ \emptyset, & \text{otherwise.} \end{cases}$

The following proposition is true

(32) Let a, b, c, d, e, f be sets. Then NE NAND6(a,b,c,d,e,f) if and only if one of the following conditions is satisfied:

(i) not NE a, or
(ii) not NE b, or
(iii) not NE c, or
(iv) not NE d, or
(v) not NE e, or
(vi) not NE f.

Let a, b, c, d, e, f be sets. The functor NOR6(a,b,c,d,e,f) yields a set and is defined as follows:

(Def. 25) $\mathrm{NOR6}(a,b,c,d,e,f) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if not NE } a \text{ and not NE } b \text{ and not NE } c \text{ and not NE } d \text{ and not NE } e \text{ and not NE } f, \\ \emptyset, & \text{otherwise.} \end{cases}$

One can prove the following proposition

(33) Let a, b, c, d, e, f be sets. Then NE NOR6(a, b, c, d, e, f) if and only if the following conditions are satisfied:

(i) not NE a,

(ii) not NE b,

(iii) not NE c,

(iv) not NE d,

(v) not NE e, and

(vi) not NE f.

Let a, b, c, d, e, f, g be sets. The functor AND7(a, b, c, d, e, f, g) yields a set and is defined by:

(Def. 26) $$\mathrm{AND7}(a, b, c, d, e, f, g) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ and NE } b \text{ and NE } c \text{ and NE } d \text{ and NE } e \text{ and NE } f \text{ and NE } g, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Next we state the proposition

(34) Let a, b, c, d, e, f, g be sets. Then NE AND7(a, b, c, d, e, f, g) if and only if the following conditions are satisfied:
NE a and NE b and NE c and NE d and NE e and NE f and NE g.

Let a, b, c, d, e, f, g be sets. The functor OR7(a, b, c, d, e, f, g) yielding a set is defined as follows:

(Def. 27) $$\mathrm{OR7}(a, b, c, d, e, f, g) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ or NE } b \text{ or NE } c \text{ or NE } d \text{ or NE } e \text{ or NE } f \text{ or NE } g, \\ \emptyset, & \text{otherwise.} \end{cases}$$

Next we state the proposition

(35) Let a, b, c, d, e, f, g be sets. Then NE OR7(a, b, c, d, e, f, g) if and only if one of the following conditions is satisfied:
NE a or NE b or NE c or NE d or NE e or NE f or NE g.

Let a, b, c, d, e, f, g be sets. The functor NAND7(a, b, c, d, e, f, g) yielding a set is defined as follows:

(Def. 28) $$\mathrm{NAND7}(a, b, c, d, e, f, g) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if not NE } a \text{ or not NE } b \text{ or not NE } c \text{ or not NE } d \text{ or not NE } e \text{ or not NE } f \text{ or not NE } g, \\ \emptyset, & \text{otherwise.} \end{cases}$$

One can prove the following proposition

(36) Let a, b, c, d, e, f, g be sets. Then NE NAND7(a, b, c, d, e, f, g) if and only if one of the following conditions is satisfied:
not NE a or not NE b or not NE c or not NE d or not NE e or not NE f or not NE g.

Let a, b, c, d, e, f, g be sets. The functor NOR7(a, b, c, d, e, f, g) yielding a set is defined as follows:

(Def. 29) $\mathrm{NOR7}(a, b, c, d, e, f, g) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if not NE } a \text{ and not NE } b \text{ and not NE } c \text{ and not NE } d \text{ and not NE } e \text{ and not NE } f \text{ and not NE } g, \\ \emptyset, & \text{otherwise.} \end{cases}$

Next we state the proposition

(37) Let a, b, c, d, e, f, g be sets. Then NE $\mathrm{NOR7}(a, b, c, d, e, f, g)$ if and only if the following conditions are satisfied:
not NE a and not NE b and not NE c and not NE d and not NE e and not NE f and not NE g.

Let a, b, c, d, e, f, g, h be sets. The functor $\mathrm{AND8}(a, b, c, d, e, f, g, h)$ yields a set and is defined by:

(Def. 30) $\mathrm{AND8}(a, b, c, d, e, f, g, h) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ and NE } b \text{ and NE } c \text{ and NE } d \text{ and NE } e \text{ and NE } f \text{ and NE } g \text{ and NE } h, \\ \emptyset, & \text{otherwise.} \end{cases}$

The following proposition is true

(38) Let a, b, c, d, e, f, g, h be sets. Then NE $\mathrm{AND8}(a, b, c, d, e, f, g, h)$ if and only if the following conditions are satisfied:
NE a and NE b and NE c and NE d and NE e and NE f and NE g and NE h.

Let a, b, c, d, e, f, g, h be sets. The functor $\mathrm{OR8}(a, b, c, d, e, f, g, h)$ yielding a set is defined as follows:

(Def. 31) $\mathrm{OR8}(a, b, c, d, e, f, g, h) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ or NE } b \text{ or NE } c \text{ or NE } d \text{ or NE } e \text{ or NE } f \text{ or NE } g \text{ or NE } h, \\ \emptyset, & \text{otherwise.} \end{cases}$

One can prove the following proposition

(39) Let a, b, c, d, e, f, g, h be sets. Then NE $\mathrm{OR8}(a, b, c, d, e, f, g, h)$ if and only if one of the following conditions is satisfied:
NE a or NE b or NE c or NE d or NE e or NE f or NE g or NE h.

Let a, b, c, d, e, f, g, h be sets. The functor $\mathrm{NAND8}(a, b, c, d, e, f, g, h)$ yielding a set is defined as follows:

(Def. 32) $\mathrm{NAND8}(a, b, c, d, e, f, g, h) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if not NE } a \text{ or not NE } b \text{ or not NE } c \text{ or not NE } d \text{ or not NE } e \text{ or not NE } f \text{ or not NE } g \text{ or not NE } h, \\ \emptyset, & \text{otherwise.} \end{cases}$

Next we state the proposition

(40) Let a, b, c, d, e, f, g, h be sets. Then NE $\mathrm{NAND8}(a, b, c, d, e, f, g, h)$ if and only if one of the following conditions is satisfied:
not NE a or not NE b or not NE c or not NE d or not NE e or not NE f or not NE g or not NE h.

Let a, b, c, d, e, f, g, h be sets. The functor $\mathrm{NOR8}(a,b,c,d,e,f,g,h)$ yielding a set is defined as follows:

(Def. 33) $\mathrm{NOR8}(a,b,c,d,e,f,g,h) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if not NE } a \text{ and not NE } b \text{ and not NE } c \text{ and not NE } d \text{ and not NE } e \text{ and not NE } f \text{ and not NE } g \text{ and not NE } h, \\ \emptyset, & \text{otherwise.} \end{cases}$

One can prove the following proposition

(41) Let a, b, c, d, e, f, g, h be sets. Then NE $\mathrm{NOR8}(a,b,c,d,e,f,g,h)$ if and only if the following conditions are satisfied:
not NE a and not NE b and not NE c and not NE d and not NE e and not NE f and not NE g and not NE h.

2. Logical Equivalence of 4 Bits Adders

We now state the proposition

(42) Let c_1, x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4, c_2, c_3, c_4, c_5, n_1, n_2, n_3, n_4, n, c_6 be sets. Suppose that
NE c_2 iff NE $\mathrm{MAJ3}(x_1,y_1,c_1)$ and NE c_3 iff NE $\mathrm{MAJ3}(x_2,y_2,c_2)$ and NE c_4 iff NE $\mathrm{MAJ3}(x_3,y_3,c_3)$ and NE c_5 iff NE $\mathrm{MAJ3}(x_4,y_4,c_4)$ and NE n_1 iff NE $\mathrm{OR2}(x_1,y_1)$ and NE n_2 iff NE $\mathrm{OR2}(x_2,y_2)$ and NE n_3 iff NE $\mathrm{OR2}(x_3,y_3)$ and NE n_4 iff NE $\mathrm{OR2}(x_4,y_4)$ and NE n iff NE $\mathrm{AND5}(c_1,n_1,n_2,n_3,n_4)$ and NE c_6 iff NE $\mathrm{OR2}(c_5,n)$. Then NE c_5 if and only if NE c_6.

Let a, b be sets. The functor $\mathrm{MODADD2}(a,b)$ yields a set and is defined as follows:

(Def. 34) $\mathrm{MODADD2}(a,b) = \begin{cases} \mathrm{NOT1}\,\emptyset, & \text{if NE } a \text{ or NE } b \text{ but NE } a \text{ but NE } b, \\ \emptyset, & \text{otherwise.} \end{cases}$

Next we state the proposition

(43) For all sets a, b holds NE $\mathrm{MODADD2}(a,b)$ iff NE a or NE b but NE a but NE b.

Let a, b, c be sets. The functor $\mathrm{ADD1}(a,b,c)$ yields a set and is defined by:

(Def. 35) $\mathrm{ADD1}(a,b,c) = \mathrm{XOR3}(a,b,c)$.

Let a, b, c be sets. The functor $\mathrm{CARR1}(a,b,c)$ yielding a set is defined by:

(Def. 36) $\mathrm{CARR1}(a,b,c) = \mathrm{MAJ3}(a,b,c)$.

Let a_1, b_1, a_2, b_2, c be sets. The functor $\mathrm{ADD2}(a_2,b_2,a_1,b_1,c)$ yielding a set is defined as follows:

(Def. 37) $\mathrm{ADD2}(a_2,b_2,a_1,b_1,c) = \mathrm{XOR3}(a_2,b_2,\mathrm{CARR1}(a_1,b_1,c))$.

Let a_1, b_1, a_2, b_2, c be sets. The functor $\mathrm{CARR2}(a_2, b_2, a_1, b_1, c)$ yields a set and is defined as follows:

(Def. 38) $\mathrm{CARR2}(a_2, b_2, a_1, b_1, c) = \mathrm{MAJ3}(a_2, b_2, \mathrm{CARR1}(a_1, b_1, c))$.

Let a_1, b_1, a_2, b_2, a_3, b_3, c be sets. The functor $\mathrm{ADD3}(a_3, b_3, a_2, b_2, a_1, b_1, c)$ yields a set and is defined by:

(Def. 39) $\mathrm{ADD3}(a_3, b_3, a_2, b_2, a_1, b_1, c) = \mathrm{XOR3}(a_3, b_3, \mathrm{CARR2}(a_2, b_2, a_1, b_1, c))$.

Let a_1, b_1, a_2, b_2, a_3, b_3, c be sets. The functor $\mathrm{CARR3}(a_3, b_3, a_2, b_2, a_1, b_1, c)$ yields a set and is defined as follows:

(Def. 40) $\mathrm{CARR3}(a_3, b_3, a_2, b_2, a_1, b_1, c) = \mathrm{MAJ3}(a_3, b_3, \mathrm{CARR2}(a_2, b_2, a_1, b_1, c))$.

Let a_1, b_1, a_2, b_2, a_3, b_3, a_4, b_4, c be sets.

The functor $\mathrm{ADD4}(a_4, b_4, a_3, b_3, a_2, b_2, a_1, b_1, c)$ yielding a set is defined by:

(Def. 41) $\mathrm{ADD4}(a_4, b_4, a_3, b_3, a_2, b_2, a_1, b_1, c) =$
$\mathrm{XOR3}(a_4, b_4, \mathrm{CARR3}(a_3, b_3, a_2, b_2, a_1, b_1, c))$.

Let a_1, b_1, a_2, b_2, a_3, b_3, a_4, b_4, c be sets.

The functor $\mathrm{CARR4}(a_4, b_4, a_3, b_3, a_2, b_2, a_1, b_1, c)$ yields a set and is defined as follows:

(Def. 42) $\mathrm{CARR4}(a_4, b_4, a_3, b_3, a_2, b_2, a_1, b_1, c) =$
$\mathrm{MAJ3}(a_4, b_4, \mathrm{CARR3}(a_3, b_3, a_2, b_2, a_1, b_1, c))$.

One can prove the following proposition

(44) Let c_1, x_1, y_1, x_2, y_2, x_3, y_3, x_4, y_4, c_4, q_1, p_1, s_1, q_2, p_2, s_2, q_3, p_3, s_3, q_4, p_4, s_4, c_7, c_8, l_2, t_2, l_3, m_3, t_3, l_4, m_4, n_4, t_4, l_5, m_5, n_5, o_5, s_5, s_6, s_7, s_8 be sets such that NE q_1 iff NE $\mathrm{NOR2}(x_1, y_1)$ and NE p_1 iff NE $\mathrm{NAND2}(x_1, y_1)$ and NE s_1 iff NE $\mathrm{MODADD2}(x_1, y_1)$ and NE q_2 iff NE $\mathrm{NOR2}(x_2, y_2)$ and NE p_2 iff NE $\mathrm{NAND2}(x_2, y_2)$ and NE s_2 iff NE $\mathrm{MODADD2}(x_2, y_2)$ and NE q_3 iff NE $\mathrm{NOR2}(x_3, y_3)$ and NE p_3 iff NE $\mathrm{NAND2}(x_3, y_3)$ and NE s_3 iff NE $\mathrm{MODADD2}(x_3, y_3)$ and NE q_4 iff NE $\mathrm{NOR2}(x_4, y_4)$ and NE p_4 iff NE $\mathrm{NAND2}(x_4, y_4)$ and NE s_4 iff NE $\mathrm{MODADD2}(x_4, y_4)$ and NE c_7 iff NE $\mathrm{NOT1}\, c_1$ and NE c_8 iff NE $\mathrm{NOT1}\, c_7$ and NE s_5 iff NE $\mathrm{XOR2}(c_8, s_1)$ and NE l_2 iff NE $\mathrm{AND2}(c_7, p_1)$ and NE t_2 iff NE $\mathrm{NOR2}(l_2, q_1)$ and NE s_6 iff NE $\mathrm{XOR2}(t_2, s_2)$ and NE l_3 iff NE $\mathrm{AND2}(q_1, p_2)$ and NE m_3 iff NE $\mathrm{AND3}(p_2, p_1, c_7)$ and NE t_3 iff NE $\mathrm{NOR3}(l_3, m_3, q_2)$ and NE s_7 iff NE $\mathrm{XOR2}(t_3, s_3)$ and NE l_4 iff NE $\mathrm{AND2}(q_2, p_3)$ and NE m_4 iff NE $\mathrm{AND3}(q_1, p_3, p_2)$ and NE n_4 iff NE $\mathrm{AND4}(p_3, p_2, p_1, c_7)$ and NE t_4 iff NE $\mathrm{NOR4}(l_4, m_4, n_4, q_3)$ and NE s_8 iff NE $\mathrm{XOR2}(t_4, s_4)$ and NE l_5 iff NE $\mathrm{AND2}(q_3, p_4)$ and NE m_5 iff NE $\mathrm{AND3}(q_2, p_4, p_3)$ and NE n_5 iff NE $\mathrm{AND4}(q_1, p_4, p_3, p_2)$ and NE o_5 iff NE $\mathrm{AND5}(p_4, p_3, p_2, p_1, c_7)$ and NE c_4 iff NE $\mathrm{NOR5}(q_4, l_5, m_5, n_5, o_5)$. Then

(i) NE s_5 iff NE $\mathrm{ADD1}(x_1, y_1, c_1)$,

(ii) NE s_6 iff NE $\mathrm{ADD2}(x_2, y_2, x_1, y_1, c_1)$,

(iii) NE s_7 iff NE $\mathrm{ADD3}(x_3, y_3, x_2, y_2, x_1, y_1, c_1)$,

(iv) NE s_8 iff NE ADD4$(x_4, y_4, x_3, y_3, x_2, y_2, x_1, y_1, c_1)$, and
(v) NE c_4 iff NE CARR4$(x_4, y_4, x_3, y_3, x_2, y_2, x_1, y_1, c_1)$.

References

[1] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[2] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.

Received February 4, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

The Sequential Closure Operator in Sequential and Frechet Spaces

Bartłomiej Skorulski
University of Białystok

MML Identifier: `FRECHET2`.

The articles [26], [30], [2], [21], [10], [3], [11], [29], [9], [31], [6], [7], [23], [8], [4], [13], [1], [20], [19], [24], [18], [17], [14], [16], [5], [12], [22], [28], [15], [27], and [25] provide the notation and terminology for this paper.

1. The Properties of Sequences and Subsequences

Let T be a non empty 1-sorted structure, let f be a function from $\mathbb{N}$ into $\mathbb{N}$, and let S be a sequence of T. Then $S \cdot f$ is a sequence of T.

One can prove the following two propositions:

(1) Let T be a non empty 1-sorted structure, S be a sequence of T, and N_1 be an increasing sequence of naturals. Then $S \cdot N_1$ is a sequence of T.

(2) For every sequence R_1 of real numbers such that $R_1 = \mathrm{id}_{\mathbb{N}}$ holds R_1 is an increasing sequence of naturals.

Let T be a non empty 1-sorted structure and let S be a sequence of T. A sequence of T is called a subsequence of S if:

(Def. 1) There exists an increasing sequence N_1 of naturals such that it $= S \cdot N_1$.

The following two propositions are true:

(3) For every non empty 1-sorted structure T holds every sequence S of T is a subsequence of S.

(4) For every non empty 1-sorted structure T and for every sequence S of T and for every subsequence S_1 of S holds $\mathrm{rng}\, S_1 \subseteq \mathrm{rng}\, S$.

ISSN 1426–2630

Let T be a non empty 1-sorted structure, let N_1 be an increasing sequence of naturals, and let S be a sequence of T. Then $S \cdot N_1$ is a subsequence of S.

One can prove the following proposition

(5) Let T be a non empty 1-sorted structure, S_1 be a sequence of T, and S_2 be a subsequence of S_1. Then every subsequence of S_2 is a subsequence of S_1.

In this article we present several logical schemes. The scheme *SubSeqChoice* deals with a non empty 1-sorted structure $\mathcal{A}$, a sequence $\mathcal{B}$ of $\mathcal{A}$, and and states that:

There exists a subsequence S_1 of $\mathcal{B}$ such that for every natural number n holds $\mathcal{P}[S_1(n)]$

provided the following requirement is met:

- For every natural number n there exists a natural number m and there exists a point x of $\mathcal{A}$ such that $n \leqslant m$ and $x = \mathcal{B}(m)$ and $\mathcal{P}[x]$.

The scheme *SubSeqChoice1* deals with a non empty topological structure $\mathcal{A}$, a sequence $\mathcal{B}$ of $\mathcal{A}$, and and states that:

There exists a subsequence S_1 of $\mathcal{B}$ such that for every natural number n holds $\mathcal{P}[S_1(n)]$

provided the parameters have the following property:

- For every natural number n there exists a natural number m and there exists a point x of $\mathcal{A}$ such that $n \leqslant m$ and $x = \mathcal{B}(m)$ and $\mathcal{P}[x]$.

One can prove the following propositions:

(6) Let T be a non empty 1-sorted structure, S be a sequence of T, and A be a subset of the carrier of T. Suppose that for every subsequence S_1 of S holds $\operatorname{rng} S_1 \not\subseteq A$. Then there exists a natural number n such that for every natural number m such that $n \leqslant m$ holds $S(m) \notin A$.

(7) Let T be a non empty 1-sorted structure, S be a sequence of T, and A, B be subsets of the carrier of T. If $\operatorname{rng} S \subseteq A \cup B$, then there exists a subsequence S_1 of S such that $\operatorname{rng} S_1 \subseteq A$ or $\operatorname{rng} S_1 \subseteq B$.

(8) Let T be a non empty topological space. Suppose that for every sequence S of T and for all points x_1, x_2 of T such that $x_1 \in \operatorname{Lim} S$ and $x_2 \in \operatorname{Lim} S$ holds $x_1 = x_2$. Then T is a T_1 space.

(9) Let T be a non empty topological space. Suppose T is a T_2 space. Let S be a sequence of T and x_1, x_2 be points of T. If $x_1 \in \operatorname{Lim} S$ and $x_2 \in \operatorname{Lim} S$, then $x_1 = x_2$.

(10) Let T be a non empty topological space. Suppose T is first-countable. Then T is a T_2 space if and only if for every sequence S of T and for all points x_1, x_2 of T such that $x_1 \in \operatorname{Lim} S$ and $x_2 \in \operatorname{Lim} S$ holds $x_1 = x_2$.

(11) For every non empty topological structure T and for every sequence S of T such that S is not convergent holds $\operatorname{Lim} S = \emptyset$.

(12) Let T be a non empty topological space and A be a subset of T. If A is closed, then for every sequence S of T such that $\operatorname{rng} S \subseteq A$ holds $\operatorname{Lim} S \subseteq A$.

(13) Let T be a non empty topological structure, S be a sequence of T, and x be a point of T. Suppose S is not convergent to x. Then there exists a subsequence S_1 of S such that every subsequence of S_1 is not convergent to x.

2. The Continuous Maps

One can prove the following two propositions:

(14) Let T_1, T_2 be non empty topological spaces and f be a map from T_1 into T_2. Suppose f is continuous. Let S_1 be a sequence of T_1 and S_2 be a sequence of T_2. If $S_2 = f \cdot S_1$, then $f^\circ \operatorname{Lim} S_1 \subseteq \operatorname{Lim} S_2$.

(15) Let T_1, T_2 be non empty topological spaces and f be a map from T_1 into T_2. Suppose T_1 is sequential. Then f is continuous if and only if for every sequence S_1 of T_1 and for every sequence S_2 of T_2 such that $S_2 = f \cdot S_1$ holds $f^\circ \operatorname{Lim} S_1 \subseteq \operatorname{Lim} S_2$.

3. The Sequential Closure Operator

Let T be a non empty topological structure and let A be a subset of the carrier of T. The functor $\mathrm{Cl}_{\mathrm{Seq}} A$ yielding a subset of T is defined by:

(Def. 2) For every point x of T holds $x \in \mathrm{Cl}_{\mathrm{Seq}} A$ iff there exists a sequence S of T such that $\operatorname{rng} S \subseteq A$ and $x \in \operatorname{Lim} S$.

The following propositions are true:

(16) Let T be a non empty topological structure, A be a subset of T, S be a sequence of T, and x be a point of T. If $\operatorname{rng} S \subseteq A$ and $x \in \operatorname{Lim} S$, then $x \in \overline{A}$.

(17) For every non empty topological structure T and for every subset A of T holds $\mathrm{Cl}_{\mathrm{Seq}} A \subseteq \overline{A}$.

(18) Let T be a non empty topological structure, S be a sequence of T, S_1 be a subsequence of S, and x be a point of T. If S is convergent to x, then S_1 is convergent to x.

(19) Let T be a non empty topological structure, S be a sequence of T, and S_1 be a subsequence of S. Then $\operatorname{Lim} S \subseteq \operatorname{Lim} S_1$.

(20) For every non empty topological structure T holds $\mathrm{Cl}_{\mathrm{Seq}}(\emptyset_T) = \emptyset$.

(21) For every non empty topological structure T and for every subset A of T holds $A \subseteq \mathrm{Cl}_{\mathrm{Seq}} A$.

(22) For every non empty topological structure T and for all subsets A, B of T holds $\mathrm{Cl}_{\mathrm{Seq}} A \cup \mathrm{Cl}_{\mathrm{Seq}} B = \mathrm{Cl}_{\mathrm{Seq}}(A \cup B)$.

(23) Let T be a non empty topological structure. Then T is Frechet if and only if for every subset A of the carrier of T holds $\overline{A} = \mathrm{Cl}_{\mathrm{Seq}} A$.

(24) Let T be a non empty topological space. Suppose T is Frechet. Let A, B be subsets of T. Then $\mathrm{Cl}_{\mathrm{Seq}}(\emptyset_T) = \emptyset$ and $A \subseteq \mathrm{Cl}_{\mathrm{Seq}} A$ and $\mathrm{Cl}_{\mathrm{Seq}}(A \cup B) = \mathrm{Cl}_{\mathrm{Seq}} A \cup \mathrm{Cl}_{\mathrm{Seq}} B$ and $\mathrm{Cl}_{\mathrm{Seq}} \mathrm{Cl}_{\mathrm{Seq}} A = \mathrm{Cl}_{\mathrm{Seq}} A$.

(25) Let T be a non empty topological space. Suppose T is sequential. If for every subset A of T holds $\mathrm{Cl}_{\mathrm{Seq}} \mathrm{Cl}_{\mathrm{Seq}} A = \mathrm{Cl}_{\mathrm{Seq}} A$, then T is Frechet.

(26) Let T be a non empty topological space. Suppose T is sequential. Then T is Frechet if and only if for all subsets A, B of T holds $\mathrm{Cl}_{\mathrm{Seq}}(\emptyset_T) = \emptyset$ and $A \subseteq \mathrm{Cl}_{\mathrm{Seq}} A$ and $\mathrm{Cl}_{\mathrm{Seq}}(A \cup B) = \mathrm{Cl}_{\mathrm{Seq}} A \cup \mathrm{Cl}_{\mathrm{Seq}} B$ and $\mathrm{Cl}_{\mathrm{Seq}} \mathrm{Cl}_{\mathrm{Seq}} A = \mathrm{Cl}_{\mathrm{Seq}} A$.

4. The Limit

Let T be a non empty topological space and let S be a sequence of T. Let us assume that there exists a point x of T such that $\mathrm{Lim}\, S = \{x\}$. The functor $\lim S$ yields a point of T and is defined as follows:

(Def. 3) S is convergent to $\lim S$.

The following propositions are true:

(27) Let T be a non empty topological space. Suppose T is a T_2 space. Let S be a sequence of T. If S is convergent, then there exists a point x of T such that $\mathrm{Lim}\, S = \{x\}$.

(28) Let T be a non empty topological space. Suppose T is a T_2 space. Let S be a sequence of T and x be a point of T. Then S is convergent to x if and only if S is convergent and $x = \lim S$.

(29) For every metric structure M holds every sequence of M is a sequence of M_{top}.

(30) For every non empty metric structure M holds every sequence of M_{top} is a sequence of M.

(31) Let M be a non empty metric space, S be a sequence of M, x be a point of M, S' be a sequence of M_{top}, and x' be a point of M_{top}. Suppose $S = S'$ and $x = x'$. Then S is convergent to x if and only if S' is convergent to x'.

(32) Let M be a non empty metric space, S_3 be a sequence of M, and S_4 be a sequence of M_{top}. If $S_3 = S_4$, then S_3 is convergent iff S_4 is convergent.

(33) Let M be a non empty metric space, S_3 be a sequence of M, and S_4 be a sequence of M_{top}. If $S_3 = S_4$ and S_3 is convergent, then $\lim S_3 = \lim S_4$.

5. The Cluster Points

Let T be a topological structure, let S be a sequence of T, and let x be a point of T. We say that x is a cluster point of S if and only if the condition (Def. 4) is satisfied.

(Def. 4) Let O be a subset of T and n be a natural number. Suppose O is open and $x \in O$. Then there exists a natural number m such that $n \leqslant m$ and $S(m) \in O$.

Next we state several propositions:

(34) Let T be a non empty topological structure, S be a sequence of T, and x be a point of T. If there exists a subsequence of S which is convergent to x, then x is a cluster point of S.

(35) Let T be a non empty topological structure, S be a sequence of T, and x be a point of T. If S is convergent to x, then x is a cluster point of S.

(36) Let T be a non empty topological structure, S be a sequence of T, x be a point of T, and Y be a subset of the carrier of T. If $Y = \{y; y$ ranges over points of T: $x \in \overline{\{y\}}\}$ and $\mathrm{rng}\, S \subseteq Y$, then S is convergent to x.

(37) Let T be a non empty topological structure, S be a sequence of T, and x, y be points of T. Suppose that for every natural number n holds $S(n) = y$ and S is convergent to x. Then $x \in \overline{\{y\}}$.

(38) Let T be a non empty topological structure, x be a point of T, Y be a subset of the carrier of T, and S be a sequence of T. Suppose $Y = \{y; y$ ranges over points of T: $x \in \overline{\{y\}}\}$ and $\mathrm{rng}\, S \cap Y = \emptyset$ and S is convergent to x. Then there exists a subsequence of S which is one-to-one.

(39) Let T be a non empty topological structure and S_1, S_2 be sequences of T. Suppose $\mathrm{rng}\, S_2 \subseteq \mathrm{rng}\, S_1$ and S_2 is one-to-one. Then there exists a permutation P of $\mathbb{N}$ such that $S_2 \cdot P$ is a subsequence of S_1.

Now we present two schemes. The scheme *PermSeq* deals with a non empty 1-sorted structure $\mathcal{A}$, a sequence $\mathcal{B}$ of $\mathcal{A}$, a permutation $\mathcal{C}$ of $\mathbb{N}$, and and states that:

> There exists a natural number n such that for every natural number m such that $n \leqslant m$ holds $\mathcal{P}[(\mathcal{B} \cdot \mathcal{C})(m)]$

provided the following condition is satisfied:

- There exists a natural number n such that for every natural number m and for every point x of $\mathcal{A}$ if $n \leqslant m$ and $x = \mathcal{B}(m)$, then $\mathcal{P}[x]$.

The scheme *PermSeq2* deals with a non empty topological structure $\mathcal{A}$, a sequence $\mathcal{B}$ of $\mathcal{A}$, a permutation $\mathcal{C}$ of $\mathbb{N}$, and and states that:

> There exists a natural number n such that for every natural number m such that $n \leqslant m$ holds $\mathcal{P}[(\mathcal{B} \cdot \mathcal{C})(m)]$

provided the parameters meet the following condition:

- There exists a natural number n such that for every natural number m and for every point x of $\mathcal{A}$ if $n \leqslant m$ and $x = \mathcal{B}(m)$, then $\mathcal{P}[x]$.

We now state several propositions:

(40) Let T be a non empty topological structure, S be a sequence of T, P be a permutation of $\mathbb{N}$, and x be a point of T. If S is convergent to x, then $S \cdot P$ is convergent to x.

(41) Let n_0 be a natural number. Then there exists an increasing sequence N_1 of naturals such that for every natural number n holds $N_1(n) = n + n_0$.

(42) Let T be a non empty 1-sorted structure, S be a sequence of T, and n_0 be a natural number. Then there exists a subsequence S_1 of S such that for every natural number n holds $S_1(n) = S(n + n_0)$.

(43) Let T be a non empty topological structure, S be a sequence of T, x be a point of T, and S_1 be a subsequence of S. Suppose x is a cluster point of S and there exists a natural number n_0 such that for every natural number n holds $S_1(n) = S(n + n_0)$. Then x is a cluster point of S_1.

(44) Let T be a non empty topological structure, S be a sequence of T, and x be a point of T. If x is a cluster point of S, then $x \in \overline{\operatorname{rng} S}$.

(45) Let T be a non empty topological structure. Suppose T is Frechet. Let S be a sequence of T and x be a point of T. If x is a cluster point of S, then there exists a subsequence of S which is convergent to x.

6. Auxiliary Theorems

We now state several propositions:

(46) Let T be a non empty topological space. Suppose T is first-countable. Let x be a point of T. Then there exists a basis B of x and there exists a function S such that $\operatorname{dom} S = \mathbb{N}$ and $\operatorname{rng} S = B$ and for all natural numbers n, m such that $m \geqslant n$ holds $S(m) \subseteq S(n)$.

(47) For every non empty topological space T holds T is a T_1 space iff for every point p of T holds $\overline{\{p\}} = \{p\}$.

(48) For every non empty topological space T such that T is a T_2 space holds T is a T_1 space.

(49) Let T be a non empty topological space. Suppose T is not a T_1 space. Then there exist points x_1, x_2 of T and there exists a sequence S of T such that $S = \mathbb{N} \longmapsto x_1$ and $x_1 \neq x_2$ and S is convergent to x_2.

(50) For every function f such that $\operatorname{dom} f$ is infinite and f is one-to-one holds $\operatorname{rng} f$ is infinite.

(51) For every non empty finite subset X of $\mathbb{N}$ and for every natural number x such that $x \in X$ holds $x \leqslant \max X$.

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[2] Grzegorz Bancerek. The ordinal numbers. *Formalized Mathematics*, 1(**1**):91–96, 1990.
[3] Grzegorz Bancerek. Countable sets and Hessenberg's theorem. *Formalized Mathematics*, 2(**1**):65–69, 1991.
[4] Józef Białas and Yatsuka Nakamura. Dyadic numbers and T_4 topological spaces. *Formalized Mathematics*, 5(**3**):361–366, 1996.
[5] Leszek Borys. Paracompact and metrizable spaces. *Formalized Mathematics*, 2(**4**):481–485, 1991.
[6] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[7] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[8] Agata Darmochwał. Compact spaces. *Formalized Mathematics*, 1(**2**):383–386, 1990.
[9] Agata Darmochwał. Families of subsets, subspaces and mappings in topological spaces. *Formalized Mathematics*, 1(**2**):257–261, 1990.
[10] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[11] Agata Darmochwał and Andrzej Trybulec. Similarity of formulae. *Formalized Mathematics*, 2(**5**):635–642, 1991.
[12] Alicia de la Cruz. Totally bounded metric spaces. *Formalized Mathematics*, 2(**4**):559–562, 1991.
[13] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[14] Krzysztof Hryniewiecki. Recursive definitions. *Formalized Mathematics*, 1(**2**):321–328, 1990.
[15] Stanisława Kanas and Adam Lecko. Sequences in metric spaces. *Formalized Mathematics*, 2(**5**):657–661, 1991.
[16] Stanisława Kanas, Adam Lecko, and Mariusz Startek. Metric spaces. *Formalized Mathematics*, 1(**3**):607–610, 1990.
[17] Jarosław Kotowicz. Monotone real sequences. Subsequences. *Formalized Mathematics*, 1(**3**):471–475, 1990.
[18] Jarosław Kotowicz. Real sequences and basic operations on them. *Formalized Mathematics*, 1(**2**):269–272, 1990.
[19] Jarosław Kotowicz. The limit of a real function at infinity. *Formalized Mathematics*, 2(**1**):17–28, 1991.
[20] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, I. *Formalized Mathematics*, 5(**2**):167–172, 1996.
[21] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[22] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[23] Jan Popiołek. Real normed space. *Formalized Mathematics*, 2(**1**):111–115, 1991.
[24] Alexander Yu. Shibakov and Andrzej Trybulec. The Cantor set. *Formalized Mathematics*, 5(**2**):233–236, 1996.
[25] Bartłomiej Skorulski. First-countable, sequential, and Frechet spaces. *Formalized Mathematics*, 7(**1**):81–86, 1998.

[26] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[27] Andrzej Trybulec. Baire spaces, Sober spaces. *Formalized Mathematics*, 6(**2**):289–294, 1997.
[28] Wojciech A. Trybulec. Binary operations on finite sequences. *Formalized Mathematics*, 1(**5**):979–981, 1990.
[29] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[30] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[31] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received February 13, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Properties of the Product of Compact Topological Spaces

Adam Grabowski
University of Białystok

MML Identifier: BORSUK_3.

The notation and terminology used in this paper are introduced in the following articles: [12], [16], [15], [4], [17], [9], [2], [11], [6], [18], [5], [13], [19], [14], [7], [1], [3], [10], and [8].

1. Preliminaries

One can prove the following proposition

(1) For all topological spaces S, T holds $\Omega_{[: S, T :]} = [: \Omega_S, \Omega_T :]$.

Let X be a set and let Y be an empty set. Note that $[: X, Y :]$ is empty.

Let X be an empty set and let Y be a set. Observe that $[: X, Y :]$ is empty.

We now state the proposition

(2) Let X, Y be non empty topological spaces and x be a point of X. Then $Y \longmapsto x$ is a continuous map from Y into $X \restriction \{x\}$.

Let T be a non empty topological structure. One can verify that id_T is homeomorphism.

Let S, T be non empty topological structures. Let us notice that the predicate S and T are homeomorphic is reflexive and symmetric.

The following proposition is true

(3) Let S, T, V be non empty topological spaces. Suppose S and T are homeomorphic and T and V are homeomorphic. Then S and V are homeomorphic.

ISSN 1426–2630

2. On the Projections and Empty Topological Spaces

Let T be a topological structure and let P be an empty subset of the carrier of T. One can verify that $T \restriction P$ is empty.

One can check that there exists a topological space which is strict and empty.

One can prove the following propositions:

(4) For every topological space T_1 and for every empty topological space T_2 holds $[: T_1, T_2 :]$ is empty and $[: T_2, T_1 :]$ is empty.

(5) Every empty topological space is compact.

Let us note that every topological space which is empty is also compact.

Let T_1 be a topological space and let T_2 be an empty topological space. Observe that $[: T_1, T_2 :]$ is empty.

One can prove the following propositions:

(6) Let X, Y be non empty topological spaces, x be a point of X, and f be a map from $[: Y, X \restriction \{x\} :]$ into Y. If $f = \pi_1((\text{the carrier of } Y) \times \{x\})$, then f is one-to-one.

(7) Let X, Y be non empty topological spaces, x be a point of X, and f be a map from $[: X \restriction \{x\}, Y :]$ into Y. If $f = \pi_2(\{x\} \times \text{the carrier of } Y)$, then f is one-to-one.

(8) Let X, Y be non empty topological spaces, x be a point of X, and f be a map from $[: Y, X \restriction \{x\} :]$ into Y. If $f = \pi_1((\text{the carrier of } Y) \times \{x\})$, then $f^{-1} = \langle \mathrm{id}_Y, Y \longmapsto x \rangle$.

(9) Let X, Y be non empty topological spaces, x be a point of X, and f be a map from $[: X \restriction \{x\}, Y :]$ into Y. If $f = \pi_2(\{x\} \times \text{the carrier of } Y)$, then $f^{-1} = \langle Y \longmapsto x, \mathrm{id}_Y \rangle$.

(10) Let X, Y be non empty topological spaces, x be a point of X, and f be a map from $[: Y, X \restriction \{x\} :]$ into Y. If $f = \pi_1((\text{the carrier of } Y) \times \{x\})$, then f is a homeomorphism.

(11) Let X, Y be non empty topological spaces, x be a point of X, and f be a map from $[: X \restriction \{x\}, Y :]$ into Y. If $f = \pi_2(\{x\} \times \text{the carrier of } Y)$, then f is a homeomorphism.

3. On the Product of Compact Spaces

One can prove the following propositions:

(12) Let X be a non empty topological space, Y be a compact non empty topological space, G be an open subset of $[: X, Y :]$, and x be a set. Suppose $x \in \{x'; x' \text{ ranges over points of } X\colon [: \{x'\}, \text{the carrier of } Y :] \subseteq G\}$. Then

there exists a many sorted set f indexed by the carrier of Y such that for every set i if $i \in$ the carrier of Y, then there exists a subset G_1 of X and there exists a subset H_1 of Y such that $f(i) = \langle G_1, H_1 \rangle$ and $\langle x, i \rangle \in [\!: G_1, H_1 :\!]$ and G_1 is open and H_1 is open and $[\!: G_1, H_1 :\!] \subseteq G$.

(13) Let X be a non empty topological space, Y be a compact non empty topological space, G be an open subset of $[\!: Y, X :\!]$, and x be a set. Suppose $x \in \{y; y$ ranges over points of X: $[\!: \Omega_Y, \{y\} :\!] \subseteq G\}$. Then there exists an open subset R of X such that $x \in R$ and $R \subseteq \{y; y$ ranges over points of X: $[\!: \Omega_Y, \{y\} :\!] \subseteq G\}$.

(14) Let X be a non empty topological space, Y be a compact non empty topological space, and G be an open subset of $[\!: Y, X :\!]$. Then $\{x; x$ ranges over points of X: $[\!: \Omega_Y, \{x\} :\!] \subseteq G\} \in$ the topology of X.

(15) For all non empty topological spaces X, Y and for every point x of X holds $[\!: X \restriction \{x\}, Y :\!]$ and Y are homeomorphic.

(16) For all non empty topological spaces S, T such that S and T are homeomorphic and S is compact holds T is compact.

(17) For all topological spaces X, Y and for every subspace X_1 of X holds $[\!: Y, X_1 :\!]$ is a subspace of $[\!: Y, X :\!]$.

(18) Let X be a non empty topological space, Y be a compact non empty topological space, x be a point of X, and Z be a subset of $[\!: Y, X :\!]$. If $Z = [\!: \Omega_Y, \{x\} :\!]$, then Z is compact.

(19) Let X be a non empty topological space, Y be a compact non empty topological space, and x be a point of X. Then $[\!: Y, X \restriction \{x\} :\!]$ is compact.

(20) Let X, Y be compact non empty topological spaces and R be a family of subsets of X. Suppose $R = \{Q; Q$ ranges over open subsets of X: $[\!: \Omega_Y, Q :\!] \subseteq \bigcup \mathrm{BaseAppr}(\Omega_{[\!: Y, X :\!]})\}$. Then R is open and a cover of Ω_X.

(21) Let X, Y be compact non empty topological spaces, R be a family of subsets of X, and F be a family of subsets of $[\!: Y, X :\!]$. Suppose that

(i) F is a cover of $[\!: Y, X :\!]$ and open, and

(ii) $R = \{Q; Q$ ranges over open subsets of X: $\bigvee_{F_1 : \text{family of subsets of } [\!: Y, X :\!]}$ $(F_1 \subseteq F \ \wedge \ F_1$ is finite $\wedge$ $[\!: \Omega_Y, Q :\!] \subseteq \bigcup F_1)\}$.

Then R is open and a cover of X.

(22) Let X, Y be compact non empty topological spaces, R be a family of subsets of X, and F be a family of subsets of $[\!: Y, X :\!]$. Suppose that

(i) F is a cover of $[\!: Y, X :\!]$ and open, and

(ii) $R = \{Q; Q$ ranges over open subsets of X: $\bigvee_{F_1 : \text{family of subsets of } [\!: Y, X :\!]}$ $(F_1 \subseteq F \ \wedge \ F_1$ is finite $\wedge$ $[\!: \Omega_Y, Q :\!] \subseteq \bigcup F_1)\}$.

Then there exists a family C of subsets of X such that $C \subseteq R$ and C is finite and a cover of X.

(23) Let X, Y be compact non empty topological spaces and F be a family of

subsets of $[: Y, X :]$. Suppose F is a cover of $[: Y, X :]$ and open. Then there exists a family G of subsets of $[: Y, X :]$ such that $G \subseteq F$ and G is a cover of $[: Y, X :]$ and finite.

(24) For all topological spaces T_1, T_2 such that T_1 is compact and T_2 is compact holds $[: T_1, T_2 :]$ is compact.

Let T_1, T_2 be compact topological spaces. Observe that $[: T_1, T_2 :]$ is compact. Next we state two propositions:

(25) Let X, Y be non empty topological spaces, X_1 be a non empty subspace of X, and Y_1 be a non empty subspace of Y. Then $[: X_1, Y_1 :]$ is a subspace of $[: X, Y :]$.

(26) Let X, Y be non empty topological spaces, Z be a non empty subset of $[: Y, X :]$, V be a non empty subset of X, and W be a non empty subset of Y. Suppose $Z = [: W, V :]$. Then the topological structure of $[: Y \restriction W, X \restriction V :]$ = the topological structure of $[: Y, X :] \restriction Z$.

Let T be a topological space. Observe that there exists a subset of T which is compact.

Let T be a topological space and let P be a compact subset of T. Note that $T \restriction P$ is compact.

We now state the proposition

(27) Let T_1, T_2 be topological spaces, S_1 be a subset of T_1, and S_2 be a subset of T_2. If S_1 is compact and S_2 is compact, then $[: S_1, S_2 :]$ is a compact subset of $[: T_1, T_2 :]$.

References

[1] Czesław Byliński. Basic functions and operations on functions. *Formalized Mathematics*, 1(**1**):245–254, 1990.
[2] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[3] Czesław Byliński. Introduction to categories and functors. *Formalized Mathematics*, 1(**2**):409–420, 1990.
[4] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[5] Agata Darmochwał. Compact spaces. *Formalized Mathematics*, 1(**2**):383–386, 1990.
[6] Agata Darmochwał. Families of subsets, subspaces and mappings in topological spaces. *Formalized Mathematics*, 1(**2**):257–261, 1990.
[7] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[8] Michał Muzalewski. Categories of groups. *Formalized Mathematics*, 2(**4**):563–571, 1991.
[9] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[10] Beata Padlewska. Locally connected spaces. *Formalized Mathematics*, 2(**1**):93–96, 1991.
[11] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[12] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[13] Andrzej Trybulec. A Borsuk theorem on homotopy types. *Formalized Mathematics*, 2(**4**):535–545, 1991.
[14] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[15] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.

[16] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[17] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[18] Mirosław Wysocki and Agata Darmochwał. Subsets of topological spaces. *Formalized Mathematics*, 1(**1**):231–237, 1990.
[19] Mariusz Żynel and Adam Guzowski. T_0 topological spaces. *Formalized Mathematics*, 5(**1**):75–77, 1996.

Received February 13, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Compactness of the Bounded Closed Subsets of $\mathcal{E}^2_{\rm T}$

Artur Korniłowicz[1]
University of Bialystok

Summary. This paper contains theorems which describe the correspondence between topological properties of real numbers subsets introduced in [40] and introduced in [38], [16]. We also show the homeomorphism between the cartesian product of two R^1 and $\mathcal{E}^2_{\rm T}$. The compactness of the bounded closed subset of $\mathcal{E}^2_{\rm T}$ is proven.

MML Identifier: `TOPREAL6`.

The articles [41], [48], [12], [49], [10], [11], [6], [47], [7], [18], [24], [43], [1], [39], [35], [8], [14], [28], [27], [26], [45], [25], [23], [3], [9], [13], [29], [2], [46], [40], [38], [50], [17], [36], [37], [16], [42], [5], [19], [4], [20], [21], [22], [51], [33], [32], [15], [31], [30], [44], and [34] provide the notation and terminology for this paper.

1. Real Numbers

For simplicity, we use the following convention: a, b are real numbers, r is a real number, i, j, n are natural numbers, M is a non empty metric space, p, q, s are points of $\mathcal{E}^2_{\rm T}$, e is a point of $\mathcal{E}^2$, w is a point of $\mathcal{E}^n$, z is a point of M, A, B are subsets of $\mathcal{E}^n_{\rm T}$, P is a subset of $\mathcal{E}^2_{\rm T}$, and D is a non empty subset of $\mathcal{E}^2_{\rm T}$.

One can prove the following propositions:

(2)[2] $a - 2 \cdot a = -a$.

(3) $-a + 2 \cdot a = a$.

[1]This paper was written while the author visited Shinshu University, winter 1999.
[2]The proposition (1) has been removed.

ISSN 1426–2630

(4) $a - \frac{a}{2} = \frac{a}{2}$.

(5) If $a \neq 0$ and $b \neq 0$, then $\frac{a}{\frac{a}{b}} = b$.

(6) For all real numbers a, b such that $0 \leqslant a$ and $0 \leqslant b$ holds $\sqrt{a+b} \leqslant \sqrt{a} + \sqrt{b}$.

(7) If $0 \leqslant a$ and $a \leqslant b$, then $|a| \leqslant |b|$.

(8) If $b \leqslant a$ and $a \leqslant 0$, then $|a| \leqslant |b|$.

(9) $\prod(0 \mapsto r) = 1$.

(10) $\prod(1 \mapsto r) = r$.

(11) $\prod(2 \mapsto r) = r \cdot r$.

(12) $\prod((n+1) \mapsto r) = \prod(n \mapsto r) \cdot r$.

(13) $j \neq 0$ and $r = 0$ iff $\prod(j \mapsto r) = 0$.

(14) If $r \neq 0$ and $j \leqslant i$, then $\prod((i -' j) \mapsto r) = \frac{\prod(i \mapsto r)}{\prod(j \mapsto r)}$.

(15) If $r \neq 0$ and $j \leqslant i$, then $r^{i -' j} = \frac{r^i}{r^j}$.

In the sequel a, b denote real numbers.

The following propositions are true:

(16) ${}^2\langle a, b \rangle = \langle a^{\mathbf{2}}, b^{\mathbf{2}} \rangle$.

(17) For every finite sequence F of elements of $\mathbb{R}$ such that $i \in \operatorname{dom}|F|$ and $a = F(i)$ holds $|F|(i) = |a|$.

(18) $|\langle a, b \rangle| = \langle |a|, |b| \rangle$.

(19) For all real numbers a, b, c, d such that $a \leqslant b$ and $c \leqslant d$ holds $|b - a| + |d - c| = (b - a) + (d - c)$.

(20) If $r > 0$, then $a \in]a - r, a + r[$.

(21) If $r \geqslant 0$, then $a \in [a - r, a + r]$.

(22) If $a < b$, then $\inf]a, b[= a$ and $\sup]a, b[= b$.

(23) $]a, b[\subseteq [a, b]$.

(24) For every bounded subset A of $\mathbb{R}$ holds $A \subseteq [\inf A, \sup A]$.

2. Topological Preliminaries

Let T be a topological structure and let A be a finite subset of the carrier of T. One can verify that $T \restriction A$ is finite.

Let us observe that there exists a topological space which is finite, non empty, and strict.

Let T be a topological structure. Note that every subset of T which is empty is also connected.

Let T be a topological space. Observe that every subset of T which is finite is also compact.

Let T be T_2 non empty topological space. Observe that every subset of T which is compact is also closed.

The following two propositions are true:

(25) For all topological spaces S, T such that S and T are homeomorphic and S is connected holds T is connected.

(26) Let T be a topological space and F be a finite family of subsets of T. Suppose that for every subset X of T such that $X \in F$ holds X is compact. Then $\bigcup F$ is compact.

3. Points and Subsets in $\mathcal{E}^2_{\rm T}$

The following propositions are true:

(27) For every non empty set X and for every set Y such that $X \subseteq Y$ holds X meets Y.

(28) For all sets A, B, C, D, X such that $A \cup B = X$ and $C \cup D = X$ and $A \cap B = \emptyset$ and $C \cap D = \emptyset$ and $B = D$ holds $A = C$.

(29) For all sets A, B, C, D, a, b such that $A \subseteq B$ and $C \subseteq D$ holds $\prod[a \longmapsto A, b \longmapsto C] \subseteq \prod[a \longmapsto B, b \longmapsto D]$.

(30) For all subsets A, B of $\mathbb{R}$ holds $\prod[1 \longmapsto A, 2 \longmapsto B]$ is a subset of $\mathcal{E}^2_{\rm T}$.

(31) $|[0, a]| = |a|$ and $|[a, 0]| = |a|$.

(32) For every point p of $\mathcal{E}^2$ and for every point q of $\mathcal{E}^2_{\rm T}$ such that $p = 0_{\mathcal{E}^2_{\rm T}}$ and $p = q$ holds $q = \langle 0, 0 \rangle$ and $q_{\bf 1} = 0$ and $q_{\bf 2} = 0$.

(33) For all points p, q of $\mathcal{E}^2$ and for every point z of $\mathcal{E}^2_{\rm T}$ such that $p = 0_{\mathcal{E}^2_{\rm T}}$ and $q = z$ holds $\rho(p, q) = |z|$.

(34) $r \cdot p = [r \cdot p_{\bf 1}, r \cdot p_{\bf 2}]$.

(35) If $s = (1 - r) \cdot p + r \cdot q$ and $s \neq p$ and $0 \leqslant r$, then $0 < r$.

(36) If $s = (1 - r) \cdot p + r \cdot q$ and $s \neq q$ and $r \leqslant 1$, then $r < 1$.

(37) If $s \in \mathcal{L}(p, q)$ and $s \neq p$ and $s \neq q$ and $p_{\bf 1} < q_{\bf 1}$, then $p_{\bf 1} < s_{\bf 1}$ and $s_{\bf 1} < q_{\bf 1}$.

(38) If $s \in \mathcal{L}(p, q)$ and $s \neq p$ and $s \neq q$ and $p_{\bf 2} < q_{\bf 2}$, then $p_{\bf 2} < s_{\bf 2}$ and $s_{\bf 2} < q_{\bf 2}$.

(39) For every point p of $\mathcal{E}^2_{\rm T}$ there exists a point q of $\mathcal{E}^2_{\rm T}$ such that $q_{\bf 1} <$ W-bound D and $p \neq q$.

(40) For every point p of $\mathcal{E}^2_{\rm T}$ there exists a point q of $\mathcal{E}^2_{\rm T}$ such that $q_{\bf 1} >$ E-bound D and $p \neq q$.

(41) For every point p of $\mathcal{E}^2_{\rm T}$ there exists a point q of $\mathcal{E}^2_{\rm T}$ such that $q_{\bf 2} >$ N-bound D and $p \neq q$.

(42) For every point p of $\mathcal{E}^2_{\rm T}$ there exists a point q of $\mathcal{E}^2_{\rm T}$ such that $q_{\bf 2} <$ S-bound D and $p \neq q$.

One can verify the following observations:

* every subset of $\mathcal{E}^2_{\rm T}$ which is convex and non empty is also connected,
* every subset of $\mathcal{E}^2_{\rm T}$ which is non horizontal is also non empty,
* every subset of $\mathcal{E}^2_{\rm T}$ which is non vertical is also non empty,
* every subset of $\mathcal{E}^2_{\rm T}$ which is region is also open and connected, and
* every subset of $\mathcal{E}^2_{\rm T}$ which is open and connected is also region.

Let us observe that every subset of $\mathcal{E}^2_{\rm T}$ which is empty is also horizontal and every subset of $\mathcal{E}^2_{\rm T}$ which is empty is also vertical.

Let us mention that there exists a subset of $\mathcal{E}^2_{\rm T}$ which is non empty and convex.

Let a, b be points of $\mathcal{E}^2_{\rm T}$. Observe that $\mathcal{L}(a, b)$ is convex and connected.

Let us mention that $\square_{\mathcal{E}^2}$ is connected.

Let us observe that every subset of $\mathcal{E}^2_{\rm T}$ which is simple closed curve is also connected and compact.

One can prove the following propositions:

(43) $\mathcal{L}(\text{NE-corner}\, P, \text{SE-corner}\, P) \subseteq \widetilde{\mathcal{L}}(\text{SpStSeq}\, P)$.

(44) $\mathcal{L}(\text{SW-corner}\, P, \text{SE-corner}\, P) \subseteq \widetilde{\mathcal{L}}(\text{SpStSeq}\, P)$.

(45) $\mathcal{L}(\text{SW-corner}\, P, \text{NW-corner}\, P) \subseteq \widetilde{\mathcal{L}}(\text{SpStSeq}\, P)$.

(46) For every subset C of $\mathcal{E}^2_{\rm T}$ holds $\{p; p$ ranges over points of $\mathcal{E}^2_{\rm T}$: $p_{\bf 1} <$ W-bound $C\}$ is a non empty convex connected subset of $\mathcal{E}^2_{\rm T}$.

4. Balls as subsets of $\mathcal{E}^n_{\rm T}$

We now state a number of propositions:

(47) If $e = q$ and $p \in \text{Ball}(e, r)$, then $q_{\bf 1} - r < p_{\bf 1}$ and $p_{\bf 1} < q_{\bf 1} + r$.

(48) If $e = q$ and $p \in \text{Ball}(e, r)$, then $q_{\bf 2} - r < p_{\bf 2}$ and $p_{\bf 2} < q_{\bf 2} + r$.

(49) If $p = e$, then $\prod[1 \longmapsto]p_{\bf 1} - \frac{r}{\sqrt{2}}, p_{\bf 1} + \frac{r}{\sqrt{2}}[, 2 \longmapsto]p_{\bf 2} - \frac{r}{\sqrt{2}}, p_{\bf 2} + \frac{r}{\sqrt{2}}[] \subseteq$ $\text{Ball}(e, r)$.

(50) If $p = e$, then $\text{Ball}(e, r) \subseteq \prod[1 \longmapsto]p_{\bf 1} - r, p_{\bf 1} + r[, 2 \longmapsto]p_{\bf 2} - r, p_{\bf 2} + r[]$.

(51) If $P = \text{Ball}(e, r)$ and $p = e$, then $(\text{proj1})^\circ P =]p_{\bf 1} - r, p_{\bf 1} + r[$.

(52) If $P = \text{Ball}(e, r)$ and $p = e$, then $(\text{proj2})^\circ P =]p_{\bf 2} - r, p_{\bf 2} + r[$.

(53) If $D = \text{Ball}(e, r)$ and $p = e$, then W-bound $D = p_{\bf 1} - r$.

(54) If $D = \text{Ball}(e, r)$ and $p = e$, then E-bound $D = p_{\bf 1} + r$.

(55) If $D = \text{Ball}(e, r)$ and $p = e$, then S-bound $D = p_{\bf 2} - r$.

(56) If $D = \text{Ball}(e, r)$ and $p = e$, then N-bound $D = p_{\bf 2} + r$.

(57) If $D = \mathrm{Ball}(e, r)$, then D is non horizontal.

(58) If $D = \mathrm{Ball}(e, r)$, then D is non vertical.

(59) For every point f of $\mathcal{E}^2$ and for every point x of $\mathcal{E}^2_\mathrm{T}$ such that $x \in \mathrm{Ball}(f, a)$ holds $[x_\mathbf{1} - 2 \cdot a, x_\mathbf{2}] \notin \mathrm{Ball}(f, a)$.

(60) Let X be a non empty compact subset of $\mathcal{E}^2_\mathrm{T}$ and p be a point of $\mathcal{E}^2$. If $p = 0_{\mathcal{E}^2_\mathrm{T}}$ and $a > 0$, then $X \subseteq \mathrm{Ball}(p, |\,\text{E-bound}\,X| + |\,\text{N-bound}\,X| + |\,\text{W-bound}\,X| + |\,\text{S-bound}\,X| + a)$.

(61) Let M be a Reflexive symmetric triangle non empty metric structure and z be a point of M. If $r < 0$, then $\mathrm{Sphere}(z, r) = \emptyset$.

(62) For every Reflexive discernible non empty metric structure M and for every point z of M holds $\mathrm{Sphere}(z, 0) = \{z\}$.

(63) Let M be a Reflexive symmetric triangle non empty metric structure and z be a point of M. If $r < 0$, then $\overline{\mathrm{Ball}}(z, r) = \emptyset$.

(64) $\overline{\mathrm{Ball}}(z, 0) = \{z\}$.

(65) For every subset A of M_top such that $A = \overline{\mathrm{Ball}}(z, r)$ holds A is closed.

(66) If $A = \overline{\mathrm{Ball}}(w, r)$, then A is closed.

(67) $\overline{\mathrm{Ball}}(z, r)$ is bounded.

(68) For every subset A of M_top such that $A = \mathrm{Sphere}(z, r)$ holds A is closed.

(69) If $A = \mathrm{Sphere}(w, r)$, then A is closed.

(70) $\mathrm{Sphere}(z, r)$ is bounded.

(71) If A is Bounded, then $\overline{A}$ is Bounded.

(72) For every non empty metric structure M holds M is bounded iff every subset of the carrier of M is bounded.

(73) Let M be a Reflexive symmetric triangle non empty metric structure and X, Y be subsets of the carrier of M. Suppose the carrier of $M = X \cup Y$ and M is non bounded and X is bounded. Then Y is non bounded.

(74) For all subsets X, Y of $\mathcal{E}^n_\mathrm{T}$ such that $n \geqslant 1$ and the carrier of $\mathcal{E}^n_\mathrm{T} = X \cup Y$ and X is Bounded holds Y is non Bounded.

(76)[3] If A is Bounded and B is Bounded, then $A \cup B$ is Bounded.

5. Topological Properties of Real Numbers Subsets

Let X be a non empty subset of $\mathbb{R}$. Observe that $\overline{X}$ is non empty.

Let D be a lower bounded subset of $\mathbb{R}$. One can verify that $\overline{D}$ is lower bounded.

[3]The proposition (75) has been removed.

Let D be an upper bounded subset of $\mathbb{R}$. One can verify that $\overline{D}$ is upper bounded.

We now state two propositions:

(77) For every non empty lower bounded subset D of $\mathbb{R}$ holds $\inf D = \inf \overline{D}$.

(78) For every non empty upper bounded subset D of $\mathbb{R}$ holds $\sup D = \sup \overline{D}$.

Let us observe that $\mathbb{R}^{\mathbf{1}}$ is T_2.

The following three propositions are true:

(79) For every subset A of $\mathbb{R}$ and for every subset B of $\mathbb{R}^{\mathbf{1}}$ such that $A = B$ holds A is closed iff B is closed.

(80) For every subset A of $\mathbb{R}$ and for every subset B of $\mathbb{R}^{\mathbf{1}}$ such that $A = B$ holds $\overline{A} = \overline{B}$.

(81) For every subset A of $\mathbb{R}$ and for every subset B of $\mathbb{R}^{\mathbf{1}}$ such that $A = B$ holds A is compact iff B is compact.

One can check that every subset of $\mathbb{R}$ which is finite is also compact.

Let a, b be real numbers. Note that $[a, b]$ is compact.

Next we state the proposition

(82) $a \neq b$ iff $\overline{]a, b[} = [a, b]$.

Let us observe that there exists a subset of $\mathbb{R}$ which is non empty, finite, and bounded.

The following propositions are true:

(83) Let T be a topological structure, f be a real map of T, and g be a map from T into $\mathbb{R}^{\mathbf{1}}$. If $f = g$, then f is continuous iff g is continuous.

(84) Let A, B be subsets of $\mathbb{R}$ and f be a map from $[\!: \mathbb{R}^{\mathbf{1}}, \mathbb{R}^{\mathbf{1}} :\!]$ into $\mathcal{E}^2_{\rm T}$. If for all real numbers x, y holds $f(\langle x, y \rangle) = \langle x, y \rangle$, then $f^\circ[\!: A, B :\!] = \prod[1 \longmapsto A, 2 \longmapsto B]$.

(85) For every map f from $[\!: \mathbb{R}^{\mathbf{1}}, \mathbb{R}^{\mathbf{1}} :\!]$ into $\mathcal{E}^2_{\rm T}$ such that for all real numbers x, y holds $f(\langle x, y \rangle) = \langle x, y \rangle$ holds f is a homeomorphism.

(86) $[\!: \mathbb{R}^{\mathbf{1}}, \mathbb{R}^{\mathbf{1}} :\!]$ and $\mathcal{E}^2_{\rm T}$ are homeomorphic.

6. Bounded Subsets

One can prove the following propositions:

(87) For all compact subsets A, B of $\mathbb{R}$ holds $\prod[1 \longmapsto A, 2 \longmapsto B]$ is a compact subset of $\mathcal{E}^2_{\rm T}$.

(88) If P is Bounded and closed, then P is compact.

(89) If P is Bounded, then for every continuous real map g of $\mathcal{E}^2_{\rm T}$ holds $\overline{g^\circ P} \subseteq g^\circ \overline{P}$.

(90) $(\mathrm{proj1})^\circ \overline{P} \subseteq \overline{(\mathrm{proj1})^\circ P}$.

(91) $(\mathrm{proj2})^\circ\overline{P} \subseteq \overline{(\mathrm{proj2})^\circ P}$.

(92) If P is Bounded, then $\overline{(\mathrm{proj1})^\circ P} = (\mathrm{proj1})^\circ\overline{P}$.

(93) If P is Bounded, then $\overline{(\mathrm{proj2})^\circ P} = (\mathrm{proj2})^\circ\overline{P}$.

(94) If D is Bounded, then W-bound D = W-bound $\overline{D}$.

(95) If D is Bounded, then E-bound D = E-bound $\overline{D}$.

(96) If D is Bounded, then N-bound D = N-bound $\overline{D}$.

(97) If D is Bounded, then S-bound D = S-bound $\overline{D}$.

Acknowledgments

I would like to thank Professor Yatsuka Nakamura for his help in the preparation of the article.

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[2] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[3] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[4] Józef Białas and Yatsuka Nakamura. The theorem of Weierstrass. *Formalized Mathematics*, 5(**3**):353–359, 1996.
[5] Leszek Borys. Paracompact and metrizable spaces. *Formalized Mathematics*, 2(**4**):481–485, 1991.
[6] Czesław Byliński. Basic functions and operations on functions. *Formalized Mathematics*, 1(**1**):245–254, 1990.
[7] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(**1**):175–180, 1990.
[8] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.
[9] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.
[10] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[11] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[12] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[13] Czesław Byliński. The sum and product of finite sequences of real numbers. *Formalized Mathematics*, 1(**4**):661–668, 1990.
[14] Czesław Byliński. Products and coproducts in categories. *Formalized Mathematics*, 2(**5**):701–709, 1991.
[15] Czesław Byliński and Piotr Rudnicki. Bounding boxes for compact sets in $\mathcal{E}^2$. *Formalized Mathematics*, 6(**3**):427–440, 1997.
[16] Agata Darmochwał. Compact spaces. *Formalized Mathematics*, 1(**2**):383–386, 1990.
[17] Agata Darmochwał. Families of subsets, subspaces and mappings in topological spaces. *Formalized Mathematics*, 1(**2**):257–261, 1990.
[18] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[19] Agata Darmochwał. The Euclidean space. *Formalized Mathematics*, 2(**4**):599–603, 1991.
[20] Agata Darmochwał and Yatsuka Nakamura. Metric spaces as topological spaces - fundamental concepts. *Formalized Mathematics*, 2(**4**):605–608, 1991.
[21] Agata Darmochwał and Yatsuka Nakamura. The topological space $\mathcal{E}^2_\mathrm{T}$. Arcs, line segments and special polygonal arcs. *Formalized Mathematics*, 2(**5**):617–621, 1991.
[22] Agata Darmochwał and Yatsuka Nakamura. The topological space $\mathcal{E}^2_\mathrm{T}$. Simple closed curves. *Formalized Mathematics*, 2(**5**):663–664, 1991.

[23] Alicia de la Cruz. Totally bounded metric spaces. *Formalized Mathematics*, 2(**4**):559–562, 1991.
[24] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[25] Stanisława Kanas, Adam Lecko, and Mariusz Startek. Metric spaces. *Formalized Mathematics*, 1(**3**):607–610, 1990.
[26] Jarosław Kotowicz. Convergent real sequences. Upper and lower bound of sets of real numbers. *Formalized Mathematics*, 1(**3**):477–481, 1990.
[27] Jarosław Kotowicz. Convergent sequences and the limit of sequences. *Formalized Mathematics*, 1(**2**):273–275, 1990.
[28] Jarosław Kotowicz. Real sequences and basic operations on them. *Formalized Mathematics*, 1(**2**):269–272, 1990.
[29] Rafał Kwiatek. Factorial and Newton coefficients. *Formalized Mathematics*, 1(**5**):887–890, 1990.
[30] Yatsuka Nakamura. Graph theoretical properties of arcs in the plane and Fashoda Meet Theorem. *Formalized Mathematics*, 7(**2**):193–201, 1998.
[31] Yatsuka Nakamura and Czesław Byliński. Extremal properties of vertices on special polygons. Part I. *Formalized Mathematics*, 5(**1**):97–102, 1996.
[32] Yatsuka Nakamura and Jarosław Kotowicz. Connectedness conditions using polygonal arcs. *Formalized Mathematics*, 3(**1**):101–106, 1992.
[33] Yatsuka Nakamura and Jarosław Kotowicz. The Jordan's property for certain subsets of the plane. *Formalized Mathematics*, 3(**2**):137–142, 1992.
[34] Yatsuka Nakamura, Andrzej Trybulec, and Czesław Byliński. Bounded domains and unbounded domains. *Formalized Mathematics*, 8(**1**):1–13, 1999.
[35] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[36] Beata Padlewska. Connected spaces. *Formalized Mathematics*, 1(**1**):239–244, 1990.
[37] Beata Padlewska. Locally connected spaces. *Formalized Mathematics*, 2(**1**):93–96, 1991.
[38] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[39] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.
[40] Konrad Raczkowski and Paweł Sadowski. Topological properties of subsets in real numbers. *Formalized Mathematics*, 1(**4**):777–780, 1990.
[41] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[42] Andrzej Trybulec. A Borsuk theorem on homotopy types. *Formalized Mathematics*, 2(**4**):535–545, 1991.
[43] Andrzej Trybulec and Czesław Byliński. Some properties of real numbers. *Formalized Mathematics*, 1(**3**):445–449, 1990.
[44] Andrzej Trybulec and Yatsuka Nakamura. On the rectangular finite sequences of the points of the plane. *Formalized Mathematics*, 6(**4**):531–539, 1997.
[45] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.
[46] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[47] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[48] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[49] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[50] Mirosław Wysocki and Agata Darmochwał. Subsets of topological spaces. *Formalized Mathematics*, 1(**1**):231–237, 1990.
[51] Mariusz Żynel and Adam Guzowski. T_0 topological spaces. *Formalized Mathematics*, 5(**1**):75–77, 1996.

Received February 19, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Hilbert Positive Propositional Calculus

Adam Grabowski
University of Białystok

MML Identifier: HILBERT1.

The papers [4], [5], [3], [1], and [2] provide the notation and terminology for this paper.

1. Definition of the Language

Let D be a set. We say that D has VERUM if and only if:

(Def. 1) $\langle 0 \rangle \in D$.

Let D be a set. We say that D has implication if and only if:

(Def. 2) For all finite sequences p, q such that $p \in D$ and $q \in D$ holds $\langle 1 \rangle ^\frown p ^\frown q \in D$.

Let D be a set. We say that D has conjunction if and only if:

(Def. 3) For all finite sequences p, q such that $p \in D$ and $q \in D$ holds $\langle 2 \rangle ^\frown p ^\frown q \in D$.

Let D be a set. We say that D has propositional variables if and only if:

(Def. 4) For every natural number n holds $\langle 3 + n \rangle \in D$.

Let D be a set. We say that D is HP-closed if and only if:

(Def. 5) $D \subseteq \mathbb{N}^*$ and D has VERUM, implication, conjunction, and propositional variables.

Let us note that every set which is HP-closed is also non empty and has VERUM, implication, conjunction, and propositional variables and every subset of $\mathbb{N}^*$ which has VERUM, implication, conjunction, and propositional variables is HP-closed.

The set HP-WFF is defined as follows:

ISSN 1426–2630

(Def. 6) HP-WFF is HP-closed and for every set D such that D is HP-closed holds HP-WFF $\subseteq D$.

Let us note that HP-WFF is HP-closed.

Let us mention that there exists a set which is HP-closed and non empty.

One can verify that every element of HP-WFF is relation-like and function-like.

Let us mention that every element of HP-WFF is finite sequence-like.

A HP-formula is an element of HP-WFF.

The HP-formula VERUM is defined by:

(Def. 7) $\mathrm{VERUM} = \langle 0 \rangle$.

Let p, q be elements of HP-WFF. The functor $p \Rightarrow q$ yielding a HP-formula is defined by:

(Def. 8) $p \Rightarrow q = \langle 1 \rangle \frown p \frown q$.

The functor $p \wedge q$ yielding a HP-formula is defined as follows:

(Def. 9) $p \wedge q = \langle 2 \rangle \frown p \frown q$.

We follow the rules: T, X, Y denote subsets of HP-WFF and p, q, r, s denote elements of HP-WFF.

Let T be a subset of HP-WFF. We say that T is Hilbert theory if and only if the conditions (Def. 10) are satisfied.

(Def. 10)(i) $\mathrm{VERUM} \in T$, and

(ii) for all elements p, q, r of HP-WFF holds $p \Rightarrow (q \Rightarrow p) \in T$ and $(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r)) \in T$ and $p \wedge q \Rightarrow p \in T$ and $p \wedge q \Rightarrow q \in T$ and $p \Rightarrow (q \Rightarrow p \wedge q) \in T$ and if $p \in T$ and $p \Rightarrow q \in T$, then $q \in T$.

Let us consider X. The functor $\mathrm{CnPos}\, X$ yields a subset of HP-WFF and is defined by:

(Def. 11) $r \in \mathrm{CnPos}\, X$ iff for every T such that T is Hilbert theory and $X \subseteq T$ holds $r \in T$.

The subset HP_TAUT of HP-WFF is defined by:

(Def. 12) $\mathrm{HP\_TAUT} = \mathrm{CnPos}\, \emptyset_{\mathrm{HP\text{-}WFF}}$.

The following propositions are true:

(1) $\mathrm{VERUM} \in \mathrm{CnPos}\, X$.

(2) $p \Rightarrow (q \Rightarrow p \wedge q) \in \mathrm{CnPos}\, X$.

(3) $(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r)) \in \mathrm{CnPos}\, X$.

(4) $p \Rightarrow (q \Rightarrow p) \in \mathrm{CnPos}\, X$.

(5) $p \wedge q \Rightarrow p \in \mathrm{CnPos}\, X$.

(6) $p \wedge q \Rightarrow q \in \mathrm{CnPos}\, X$.

(7) If $p \in \mathrm{CnPos}\, X$ and $p \Rightarrow q \in \mathrm{CnPos}\, X$, then $q \in \mathrm{CnPos}\, X$.

(8) If T is Hilbert theory and $X \subseteq T$, then $\mathrm{CnPos}\, X \subseteq T$.

(9) $X \subseteq \mathrm{CnPos}\, X$.

(10) If $X \subseteq Y$, then $\mathrm{CnPos}\, X \subseteq \mathrm{CnPos}\, Y$.

(11) $\mathrm{CnPos}\, \mathrm{CnPos}\, X = \mathrm{CnPos}\, X$.

Let X be a subset of HP-WFF. One can verify that $\mathrm{CnPos}\, X$ is Hilbert theory.

We now state two propositions:

(12) T is Hilbert theory iff $\mathrm{CnPos}\, T = T$.

(13) If T is Hilbert theory, then $\mathrm{HP\_TAUT} \subseteq T$.

Let us mention that HP_TAUT is Hilbert theory.

2. The Tautologies of the Hilbert Calculus - Implicational Part

We now state a number of propositions:

(14) $p \Rightarrow p \in \mathrm{HP\_TAUT}$.

(15) If $q \in \mathrm{HP\_TAUT}$, then $p \Rightarrow q \in \mathrm{HP\_TAUT}$.

(16) $p \Rightarrow \mathrm{VERUM} \in \mathrm{HP\_TAUT}$.

(17) $(p \Rightarrow q) \Rightarrow (p \Rightarrow p) \in \mathrm{HP\_TAUT}$.

(18) $(q \Rightarrow p) \Rightarrow (p \Rightarrow p) \in \mathrm{HP\_TAUT}$.

(19) $(q \Rightarrow r) \Rightarrow ((p \Rightarrow q) \Rightarrow (p \Rightarrow r)) \in \mathrm{HP\_TAUT}$.

(20) If $p \Rightarrow (q \Rightarrow r) \in \mathrm{HP\_TAUT}$, then $q \Rightarrow (p \Rightarrow r) \in \mathrm{HP\_TAUT}$.

(21) $(p \Rightarrow q) \Rightarrow ((q \Rightarrow r) \Rightarrow (p \Rightarrow r)) \in \mathrm{HP\_TAUT}$.

(22) If $p \Rightarrow q \in \mathrm{HP\_TAUT}$, then $(q \Rightarrow r) \Rightarrow (p \Rightarrow r) \in \mathrm{HP\_TAUT}$.

(23) If $p \Rightarrow q \in \mathrm{HP\_TAUT}$ and $q \Rightarrow r \in \mathrm{HP\_TAUT}$, then $p \Rightarrow r \in \mathrm{HP\_TAUT}$.

(24) $(p \Rightarrow (q \Rightarrow r)) \Rightarrow ((s \Rightarrow q) \Rightarrow (p \Rightarrow (s \Rightarrow r))) \in \mathrm{HP\_TAUT}$.

(25) $((p \Rightarrow q) \Rightarrow r) \Rightarrow (q \Rightarrow r) \in \mathrm{HP\_TAUT}$.

(26) $(p \Rightarrow (q \Rightarrow r)) \Rightarrow (q \Rightarrow (p \Rightarrow r)) \in \mathrm{HP\_TAUT}$.

(27) $(p \Rightarrow (p \Rightarrow q)) \Rightarrow (p \Rightarrow q) \in \mathrm{HP\_TAUT}$.

(28) $q \Rightarrow ((q \Rightarrow p) \Rightarrow p) \in \mathrm{HP\_TAUT}$.

(29) If $s \Rightarrow (q \Rightarrow p) \in \mathrm{HP\_TAUT}$ and $q \in \mathrm{HP\_TAUT}$, then $s \Rightarrow p \in \mathrm{HP\_TAUT}$.

3. Conjunctional Part of the Calculus

The following propositions are true:

(30) $p \Rightarrow p \wedge p \in \mathrm{HP\_TAUT}$.

(31) $p \wedge q \in \mathrm{HP\_TAUT}$ iff $p \in \mathrm{HP\_TAUT}$ and $q \in \mathrm{HP\_TAUT}$.

(32) $p \wedge q \in \mathrm{HP\_TAUT}$ iff $q \wedge p \in \mathrm{HP\_TAUT}$.

(33) $(p \wedge q \Rightarrow r) \Rightarrow (p \Rightarrow (q \Rightarrow r)) \in \mathrm{HP\_TAUT}$.

(34) $(p \Rightarrow (q \Rightarrow r)) \Rightarrow (p \wedge q \Rightarrow r) \in \mathrm{HP\_TAUT}$.

(35) $(r \Rightarrow p) \Rightarrow ((r \Rightarrow q) \Rightarrow (r \Rightarrow p \wedge q)) \in \mathrm{HP\_TAUT}$.

(36) $(p \Rightarrow q) \wedge p \Rightarrow q \in \mathrm{HP\_TAUT}$.

(37) $(p \Rightarrow q) \wedge p \wedge s \Rightarrow q \in \mathrm{HP\_TAUT}$.

(38) $(q \Rightarrow s) \Rightarrow (p \wedge q \Rightarrow s) \in \mathrm{HP\_TAUT}$.

(39) $(q \Rightarrow s) \Rightarrow (q \wedge p \Rightarrow s) \in \mathrm{HP\_TAUT}$.

(40) $(p \wedge s \Rightarrow q) \Rightarrow (p \wedge s \Rightarrow q \wedge s) \in \mathrm{HP\_TAUT}$.

(41) $(p \Rightarrow q) \Rightarrow (p \wedge s \Rightarrow q \wedge s) \in \mathrm{HP\_TAUT}$.

(42) $(p \Rightarrow q) \wedge (p \wedge s) \Rightarrow q \wedge s \in \mathrm{HP\_TAUT}$.

(43) $p \wedge q \Rightarrow q \wedge p \in \mathrm{HP\_TAUT}$.

(44) $(p \Rightarrow q) \wedge (p \wedge s) \Rightarrow s \wedge q \in \mathrm{HP\_TAUT}$.

(45) $(p \Rightarrow q) \Rightarrow (p \wedge s \Rightarrow s \wedge q) \in \mathrm{HP\_TAUT}$.

(46) $(p \Rightarrow q) \Rightarrow (s \wedge p \Rightarrow s \wedge q) \in \mathrm{HP\_TAUT}$.

(47) $p \wedge (s \wedge q) \Rightarrow p \wedge (q \wedge s) \in \mathrm{HP\_TAUT}$.

(48) $(p \Rightarrow q) \wedge (p \Rightarrow s) \Rightarrow (p \Rightarrow q \wedge s) \in \mathrm{HP\_TAUT}$.

(49) $p \wedge q \wedge s \Rightarrow p \wedge (q \wedge s) \in \mathrm{HP\_TAUT}$.

(50) $p \wedge (q \wedge s) \Rightarrow p \wedge q \wedge s \in \mathrm{HP\_TAUT}$.

References

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

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

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

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

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

Received February 20, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Homeomorphism between $[:\mathcal{E}^i_{\mathrm{T}}, \mathcal{E}^j_{\mathrm{T}}:]$ and $\mathcal{E}^{i+j}_{\mathrm{T}}$

Artur Korniłowicz[1]
University of Bialystok

Summary. In this paper we introduce the cartesian product of two metric spaces. As the distance between two points in the product we take maximal distance between coordinates of these points. In the main theorem we show the homeomorphism between $[:\mathcal{E}^i_{\mathrm{T}}, \mathcal{E}^j_{\mathrm{T}}:]$ and $\mathcal{E}^{i+j}_{\mathrm{T}}$.

MML Identifier: `TOPREAL7`.

The notation and terminology used in this paper have been introduced in the following articles: [20], [9], [25], [7], [8], [4], [16], [24], [21], [19], [13], [18], [23], [1], [2], [10], [5], [17], [11], [3], [22], [14], [12], [6], [26], and [15].

We use the following convention: i, j, n denote natural numbers, f, g, h, k denote finite sequences of elements of $\mathbb{R}$, and M, N denote non empty metric spaces.

We now state a number of propositions:

(1) For all real numbers a, b such that $\max(a,b) \leqslant a$ holds $\max(a,b) = a$.

(2) For all real numbers a, b, c, d holds $\max(a+c, b+d) \leqslant \max(a,b) + \max(c,d)$.

(3) For all real numbers a, b, c, d, e, f such that $a \leqslant b+c$ and $d \leqslant e+f$ holds $\max(a,d) \leqslant \max(b,e) + \max(c,f)$.

(4) For all finite sequences f, g holds $\mathrm{dom}\, g \subseteq \mathrm{dom}(f \frown g)$.

(5) For all finite sequences f, g such that $\mathrm{len}\, f < i$ and $i \leqslant \mathrm{len}\, f + \mathrm{len}\, g$ holds $i - \mathrm{len}\, f \in \mathrm{dom}\, g$.

(6) For all finite sequences f, g, h, k such that $f \frown g = h \frown k$ and $\mathrm{len}\, f = \mathrm{len}\, h$ and $\mathrm{len}\, g = \mathrm{len}\, k$ holds $f = h$ and $g = k$.

(7) If $\mathrm{len}\, f = \mathrm{len}\, g$ or $\mathrm{dom}\, f = \mathrm{dom}\, g$, then $\mathrm{len}(f+g) = \mathrm{len}\, f$ and $\mathrm{dom}(f+g) = \mathrm{dom}\, f$.

[1]This paper was written while the author visited Shinshu University, winter 1999.

ISSN 1426–2630

(8) If $\operatorname{len} f = \operatorname{len} g$ or $\operatorname{dom} f = \operatorname{dom} g$, then $\operatorname{len}(f - g) = \operatorname{len} f$ and $\operatorname{dom}(f - g) = \operatorname{dom} f$.

(9) $\operatorname{len} f = \operatorname{len}^2 f$ and $\operatorname{dom} f = \operatorname{dom}^2 f$.

(10) $\operatorname{len} f = \operatorname{len}|f|$ and $\operatorname{dom} f = \operatorname{dom}|f|$.

(11) ${}^2(f \frown g) = ({}^2 f) \frown ({}^2 g)$.

(12) $|f \frown g| = |f| \frown |g|$.

(13) If $\operatorname{len} f = \operatorname{len} h$ and $\operatorname{len} g = \operatorname{len} k$, then ${}^2(f \frown g + h \frown k) = ({}^2(f + h)) \frown ({}^2(g + k))$.

(14) If $\operatorname{len} f = \operatorname{len} h$ and $\operatorname{len} g = \operatorname{len} k$, then $|f \frown g + h \frown k| = |f + h| \frown |g + k|$.

(15) If $\operatorname{len} f = \operatorname{len} h$ and $\operatorname{len} g = \operatorname{len} k$, then ${}^2(f \frown g - h \frown k) = ({}^2(f - h)) \frown ({}^2(g - k))$.

(16) If $\operatorname{len} f = \operatorname{len} h$ and $\operatorname{len} g = \operatorname{len} k$, then $|f \frown g - h \frown k| = |f - h| \frown |g - k|$.

(17) If $\operatorname{len} f = n$, then $f \in$ the carrier of $\mathcal{E}^n$.

(18) If $\operatorname{len} f = n$, then $f \in$ the carrier of $\mathcal{E}^n_{\mathrm{T}}$.

(19) For every finite sequence f such that $f \in$ the carrier of $\mathcal{E}^n$ holds $\operatorname{len} f = n$.

Let M, N be non empty metric structures. The functor max-Prod2(M, N) yielding a strict metric structure is defined by the conditions (Def. 1).

(Def. 1)(i) The carrier of max-Prod2$(M, N) = [\!:$ the carrier of M, the carrier of $N :\!]$, and

(ii) for all points x, y of max-Prod2(M, N) there exist points x_1, y_1 of M and there exist points x_2, y_2 of N such that $x = \langle x_1, x_2 \rangle$ and $y = \langle y_1, y_2 \rangle$ and (the distance of max-Prod2$(M, N))(x, y) = \max(($the distance of $M)(x_1, y_1), ($the distance of $N)(x_2, y_2))$.

Let M, N be non empty metric structures. One can verify that max-Prod2(M, N) is non empty.

Let M, N be non empty metric structures, let x be a point of M, and let y be a point of N. Then $\langle x, y \rangle$ is an element of max-Prod2(M, N).

Let M, N be non empty metric structures and let x be a point of max-Prod2(M, N). Then $x_{\mathbf{1}}$ is an element of M. Then $x_{\mathbf{2}}$ is an element of N.

The following three propositions are true:

(20) Let M, N be non empty metric structures, m_1, m_2 be points of M, and n_1, n_2 be points of N. Then $\rho(\langle m_1, n_1 \rangle, \langle m_2, n_2 \rangle) = \max(\rho(m_1, m_2), \rho(n_1, n_2))$.

(21) For all non empty metric structures M, N and for all points m, n of max-Prod2(M, N) holds $\rho(m, n) = \max(\rho(m_{\mathbf{1}}, n_{\mathbf{1}}), \rho(m_{\mathbf{2}}, n_{\mathbf{2}}))$.

(22) For all Reflexive non empty metric structures M, N holds max-Prod2(M, N) is Reflexive.

Let M, N be Reflexive non empty metric structures. Observe that max-Prod2(M,N) is Reflexive.

Next we state the proposition

(23) For all symmetric non empty metric structures M, N holds max-Prod2(M,N) is symmetric.

Let M, N be symmetric non empty metric structures. Note that max-Prod2(M,N) is symmetric.

Next we state the proposition

(24) For all triangle non empty metric structures M, N holds max-Prod2(M,N) is triangle.

Let M, N be triangle non empty metric structures. One can check that max-Prod2(M,N) is triangle.

Let M, N be non empty metric spaces. Note that max-Prod2(M,N) is discernible.

The following three propositions are true:

(25) $[: M_{\rm top}, N_{\rm top} :] = (\text{max-Prod2}(M,N))_{\rm top}$.

(26) Suppose that

(i) the carrier of $M =$ the carrier of N,

(ii) for every point m of M and for every point n of N and for every real number r such that $r > 0$ and $m = n$ there exists a real number r_1 such that $r_1 > 0$ and $\text{Ball}(n, r_1) \subseteq \text{Ball}(m, r)$, and

(iii) for every point m of M and for every point n of N and for every real number r such that $r > 0$ and $m = n$ there exists a real number r_1 such that $r_1 > 0$ and $\text{Ball}(m, r_1) \subseteq \text{Ball}(n, r)$.

Then $M_{\rm top} = N_{\rm top}$.

(27) $[: \mathcal{E}^i_{\rm T}, \mathcal{E}^j_{\rm T} :]$ and $\mathcal{E}^{i+j}_{\rm T}$ are homeomorphic.

Acknowledgments

I would like to thank Professor Yatsuka Nakamura for his help in the preparation of the article.

References

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

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

[3] Leszek Borys. Paracompact and metrizable spaces. *Formalized Mathematics*, 2(**4**):481–485, 1991.

[4] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(**1**):175–180, 1990.

[5] Czesław Byliński. Binary operations applied to finite sequences. *Formalized Mathematics*, 1(**4**):643–649, 1990.

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

[7] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[8] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[9] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[10] Czesław Byliński. The sum and product of finite sequences of real numbers. *Formalized Mathematics*, 1(**4**):661–668, 1990.
[11] Agata Darmochwał. Families of subsets, subspaces and mappings in topological spaces. *Formalized Mathematics*, 1(**2**):257–261, 1990.
[12] Agata Darmochwał. The Euclidean space. *Formalized Mathematics*, 2(**4**):599–603, 1991.
[13] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[14] Stanisława Kanas, Adam Lecko, and Mariusz Startek. Metric spaces. *Formalized Mathematics*, 1(**3**):607–610, 1990.
[15] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part I. *Formalized Mathematics*, 3(**1**):107–115, 1992.
[16] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[17] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[18] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.
[19] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.
[20] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[21] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[22] Andrzej Trybulec. A Borsuk theorem on homotopy types. *Formalized Mathematics*, 2(**4**):535–545, 1991.
[23] Andrzej Trybulec and Czesław Byliński. Some properties of real numbers. *Formalized Mathematics*, 1(**3**):445–449, 1990.
[24] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[25] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[26] Mariusz Żynel and Adam Guzowski. T_0 topological spaces. *Formalized Mathematics*, 5(**1**):75–77, 1996.

Received February 21, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Full Subtracter Circuit. Part I

Katsumi Wasaki
Shinshu University
Nagano

Noboru Endou
Shinshu University
Nagano

Summary. We formalize the concept of the full subtracter circuit, define the structures of bit subtract/borrow units for binary operations, and prove the stability of the circuit.

MML Identifier: FSCIRC_1.

The terminology and notation used in this paper are introduced in the following papers: [11], [14], [13], [10], [17], [3], [4], [1], [16], [9], [12], [8], [6], [7], [5], [15], and [2].

1. Bit Subtract and Borrow Circuit

In this paper x, y, c are sets.

Let x, y, c be sets. The functor BitSubtracterOutput(x, y, c) yields an element of InnerVertices(2GatesCircStr(x, y, c, xor)) and is defined as follows:

(Def. 1) BitSubtracterOutput$(x, y, c) =$ 2GatesCircOutput(x, y, c, xor).

Let x, y, c be sets. The functor BitSubtracterCirc(x, y, c) yields a strict Boolean circuit of 2GatesCircStr(x, y, c, xor) with denotation held in gates and is defined as follows:

(Def. 2) BitSubtracterCirc$(x, y, c) =$ 2GatesCircuit(x, y, c, xor).

Let x, y, c be sets. The functor BorrowIStr(x, y, c) yields an unsplit non void strict non empty many sorted signature with arity held in gates and Boolean denotation held in gates and is defined by:

(Def. 3) BorrowIStr(x, y, c) = 1GateCircStr$(\langle x, y\rangle, \mathrm{and}_{2a})$+· 1GateCircStr$(\langle y, c\rangle, \mathrm{and}_2)$+· 1GateCircStr$(\langle x, c\rangle, \mathrm{and}_{2a})$.

ISSN 1426–2630

Let x, y, c be sets. The functor BorrowStr(x, y, c) yielding an unsplit non void strict non empty many sorted signature with arity held in gates and Boolean denotation held in gates is defined by:

(Def. 4) $\text{BorrowStr}(x, y, c) = \text{BorrowIStr}(x, y, c) +\cdot \text{1GateCircStr}(\langle \langle \langle x, y \rangle, \text{and}_{2a} \rangle, \langle \langle y, c \rangle, \text{and}_2 \rangle, \langle \langle x, c \rangle, \text{and}_{2a} \rangle \rangle, \text{or}_3)$.

Let x, y, c be sets. The functor BorrowICirc(x, y, c) yielding a strict Boolean circuit of BorrowIStr(x, y, c) with denotation held in gates is defined by:

(Def. 5) $\text{BorrowICirc}(x, y, c) = \text{1GateCircuit}(x, y, \text{and}_{2a}) +\cdot \text{1GateCircuit}(y, c, \text{and}_2) +\cdot \text{1GateCircuit}(x, c, \text{and}_{2a})$.

The following propositions are true:

(1) InnerVertices(BorrowStr(x, y, c)) is a binary relation.

(2) For all non pair sets x, y, c holds InputVertices(BorrowStr(x, y, c)) has no pairs.

(3) For every state s of BorrowICirc(x, y, c) and for all elements a, b of *Boolean* such that $a = s(x)$ and $b = s(y)$ holds $(\text{Following}(s))(\langle \langle x, y \rangle, \text{and}_{2a} \rangle) = \neg a \wedge b$.

(4) For every state s of BorrowICirc(x, y, c) and for all elements a, b of *Boolean* such that $a = s(y)$ and $b = s(c)$ holds $(\text{Following}(s))(\langle \langle y, c \rangle, \text{and}_2 \rangle) = a \wedge b$.

(5) For every state s of BorrowICirc(x, y, c) and for all elements a, b of *Boolean* such that $a = s(x)$ and $b = s(c)$ holds $(\text{Following}(s))(\langle \langle x, c \rangle, \text{and}_{2a} \rangle) = \neg a \wedge b$.

Let x, y, c be sets. The functor BorrowOutput(x, y, c) yields an element of InnerVertices(BorrowStr(x, y, c)) and is defined by:

(Def. 6) $\text{BorrowOutput}(x, y, c) = \langle \langle \langle \langle x, y \rangle, \text{and}_{2a} \rangle, \langle \langle y, c \rangle, \text{and}_2 \rangle, \langle \langle x, c \rangle, \text{and}_{2a} \rangle \rangle, \text{or}_3 \rangle$.

Let x, y, c be sets. The functor BorrowCirc(x, y, c) yielding a strict Boolean circuit of BorrowStr(x, y, c) with denotation held in gates is defined by:

(Def. 7) $\text{BorrowCirc}(x, y, c) = \text{BorrowICirc}(x, y, c) +\cdot \text{1GateCircuit}(\langle \langle x, y \rangle, \text{and}_{2a} \rangle, \langle \langle y, c \rangle, \text{and}_2 \rangle, \langle \langle x, c \rangle, \text{and}_{2a} \rangle, \text{or}_3)$.

Next we state a number of propositions:

(6) $x \in$ the carrier of BorrowStr(x, y, c) and $y \in$ the carrier of BorrowStr(x, y, c) and $c \in$ the carrier of BorrowStr(x, y, c).

(7) $\langle \langle x, y \rangle, \text{and}_{2a} \rangle \in$ InnerVertices(BorrowStr(x, y, c)) and $\langle \langle y, c \rangle, \text{and}_2 \rangle \in$ InnerVertices(BorrowStr(x, y, c)) and $\langle \langle x, c \rangle, \text{and}_{2a} \rangle \in$ InnerVertices(BorrowStr(x, y, c)).

(8) For all non pair sets x, y, c holds $x \in$ InputVertices(BorrowStr(x, y, c)) and $y \in$ InputVertices(BorrowStr(x, y, c)) and $c \in$ InputVertices(BorrowStr(x, y, c)).

(9) For all non pair sets x, y, c holds InputVertices(BorrowStr(x, y, c)) $= \{x, y, c\}$ and InnerVertices(BorrowStr(x, y, c)) $= \{\langle\langle x, y\rangle, \mathrm{and}_{2a}\rangle, \langle\langle y, c\rangle, \mathrm{and}_{2}\rangle, \langle\langle x, c\rangle, \mathrm{and}_{2a}\rangle\} \cup \{$BorrowOutput$(x, y, c)\}$.

(10) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(y)$, then (Following(s))$(\langle\langle x, y\rangle, \mathrm{and}_{2a}\rangle) = \neg a_1 \wedge a_2$.

(11) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_2, a_3 be elements of *Boolean*. If $a_2 = s(y)$ and $a_3 = s(c)$, then (Following(s))$(\langle\langle y, c\rangle, \mathrm{and}_{2}\rangle) = a_2 \wedge a_3$.

(12) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_1, a_3 be elements of *Boolean*. If $a_1 = s(x)$ and $a_3 = s(c)$, then (Following(s))$(\langle\langle x, c\rangle, \mathrm{and}_{2a}\rangle) = \neg a_1 \wedge a_3$.

(13) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_1, a_2, a_3 be elements of *Boolean*. If $a_1 = s(\langle\langle x, y\rangle, \mathrm{and}_{2a}\rangle)$ and $a_2 = s(\langle\langle y, c\rangle, \mathrm{and}_{2}\rangle)$ and $a_3 = s(\langle\langle x, c\rangle, \mathrm{and}_{2a}\rangle)$, then (Following$(s)$)(BorrowOutput$(x, y, c)$) $= a_1 \vee a_2 \vee a_3$.

(14) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_1, a_2 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(y)$, then (Following$(s, 2)$)$(\langle\langle x, y\rangle, \mathrm{and}_{2a}\rangle) = \neg a_1 \wedge a_2$.

(15) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_2, a_3 be elements of *Boolean*. If $a_2 = s(y)$ and $a_3 = s(c)$, then (Following$(s, 2)$)$(\langle\langle y, c\rangle, \mathrm{and}_{2}\rangle) = a_2 \wedge a_3$.

(16) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_1, a_3 be elements of *Boolean*. If $a_1 = s(x)$ and $a_3 = s(c)$, then (Following$(s, 2)$)$(\langle\langle x, c\rangle, \mathrm{and}_{2a}\rangle) = \neg a_1 \wedge a_3$.

(17) Let x, y, c be non pair sets, s be a state of BorrowCirc(x, y, c), and a_1, a_2, a_3 be elements of *Boolean*. If $a_1 = s(x)$ and $a_2 = s(y)$ and $a_3 = s(c)$, then (Following$(s, 2)$)(BorrowOutput(x, y, c)) $= \neg a_1 \wedge a_2 \vee a_2 \wedge a_3 \vee \neg a_1 \wedge a_3$.

(18) For all non pair sets x, y, c and for every state s of BorrowCirc(x, y, c) holds Following$(s, 2)$ is stable.

2. Bit Subtracter with Borrow Circuit

Let x, y, c be sets. The functor BitSubtracterWithBorrowStr(x, y, c) yields an unsplit non void strict non empty many sorted signature with arity held in gates and Boolean denotation held in gates and is defined by:

(Def. 8) BitSubtracterWithBorrowStr(x, y, c) = 2GatesCircStr(x, y, c, xor) $+\cdot$ BorrowStr(x, y, c).

The following propositions are true:

(19) For all non pair sets x, y, c holds InputVertices(BitSubtracterWithBorrowStr(x, y, c)) $= \{x, y, c\}$.

(20) For all non pair sets x, y, c holds InnerVertices(BitSubtracterWithBorrowStr(x, y, c)) $= \{\langle\langle x, y\rangle, \text{xor}\rangle$, 2GatesCircOutput$(x, y, c, \text{xor})\} \cup \{\langle\langle x, y\rangle, \text{and}_{2a}\rangle, \langle\langle y, c\rangle, \text{and}_2\rangle, \langle\langle x, c\rangle, \text{and}_{2a}\rangle\} \cup \{$BorrowOutput$(x, y, c)\}$.

(21) Let S be a non empty many sorted signature. Suppose $S =$ BitSubtracterWithBorrowStr(x, y, c). Then $x \in$ the carrier of S and $y \in$ the carrier of S and $c \in$ the carrier of S.

Let x, y, c be sets. The functor BitSubtracterWithBorrowCirc(x, y, c) yields a strict Boolean circuit of BitSubtracterWithBorrowStr(x, y, c) with denotation held in gates and is defined as follows:

(Def. 9) BitSubtracterWithBorrowCirc(x, y, c) = BitSubtracterCirc(x, y, c) $+\cdot$ BorrowCirc(x, y, c).

We now state several propositions:

(22) InnerVertices(BitSubtracterWithBorrowStr(x, y, c)) is a binary relation.

(23) For all non pair sets x, y, c holds InputVertices(BitSubtracterWithBorrowStr(x, y, c)) has no pairs.

(24) BitSubtracterOutput(x, y, c) $\in$ InnerVertices(BitSubtracterWithBorrowStr(x, y, c)) and BorrowOutput (x, y, c) $\in$ InnerVertices(BitSubtracterWithBorrowStr(x, y, c)).

(25) Let x, y, c be non pair sets, s be a state of BitSubtracterWithBorrowCirc (x, y, c), and a_1, a_2, a_3 be elements of *Boolean*. Suppose $a_1 = s(x)$ and $a_2 = s(y)$ and $a_3 = s(c)$. Then (Following($s, 2$))(BitSubtracterOutput(x, y, c)) $= a_1 \oplus a_2 \oplus a_3$ and (Following($s, 2$))(BorrowOutput(x, y, c)) $= \neg a_1 \wedge a_2 \vee a_2 \wedge a_3 \vee \neg a_1 \wedge a_3$.

(26) For all non pair sets x, y, c and for every state s of BitSubtracterWithBorrowCirc(x, y, c) holds Following($s, 2$) is stable.

References

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

[2] Grzegorz Bancerek and Yatsuka Nakamura. Full adder circuit. Part I. *Formalized Mathematics*, 5(**3**):367–380, 1996.

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

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

[5] Yatsuka Nakamura and Grzegorz Bancerek. Combining of circuits. *Formalized Mathematics*, 5(**2**):283–295, 1996.

[6] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Introduction to circuits, I. *Formalized Mathematics*, 5(**2**):227–232, 1996.

[7] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Introduction to circuits, II. *Formalized Mathematics*, 5(**2**):273–278, 1996.

[8] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, II. *Formalized Mathematics*, 5(**2**):215–220, 1996.
[9] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[10] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[11] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[12] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[13] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[14] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[15] Katsumi Wasaki and Pauline N. Kawamoto. 2's complement circuit. *Formalized Mathematics*, 6(**2**):189–197, 1997.
[16] Edmund Woronowicz. Many–argument relations. *Formalized Mathematics*, 1(**4**):733–737, 1990.
[17] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received March 13, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Correctness of Binary Counter Circuits

Yuguang Yang
Shinshu University
Nagano

Katsumi Wasaki
Shinshu University
Nagano

Yasushi Fuwa
Shinshu University
Nagano

Yatsuka Nakamura
Shinshu University
Nagano

Summary. This article introduces the verification of the correctness for the operations and the specification of the 3-bit counter. Both cases: without reset input and with reset input are considered. The proof was proposed by Y. Nakamura in [1].

MML Identifier: GATE_2.

The paper [1] provides the terminology and notation for this paper.

In this paper a, b, c, d denote sets.

Next we state four propositions:

(1) Let s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, n_0, n_1, n_2, n_3, n_4, n_5, n_6, n_7, q_1, q_2, q_3, n_8, n_9, n_{10} be sets such that NE s_0 iff NE AND3(NOT1 q_3, NOT1 q_2, NOT1 q_1) and NE s_1 iff NE AND3(NOT1 q_3, NOT1 q_2, q_1) and NE s_2 iff NE AND3(NOT1 q_3, q_2, NOT1 q_1) and NE s_3 iff NE AND3(NOT1 q_3, q_2, q_1) and NE s_4 iff NE AND3(q_3, NOT1 q_2, NOT1 q_1) and NE s_5 iff NE AND3(q_3, NOT1 q_2, q_1) and NE s_6 iff NE AND3(q_3, q_2, NOT1 q_1) and NE s_7 iff NE AND3(q_3, q_2, q_1) and NE n_0 iff NE AND3(NOT1 n_{10}, NOT1 n_9, NOT1 n_8) and NE n_1 iff NE AND3(NOT1 n_{10}, NOT1 n_9, n_8) and NE n_2 iff NE AND3(NOT1 n_{10}, n_9, NOT1 n_8) and NE n_3 iff NE AND3(NOT1 n_{10}, n_9, n_8) and NE n_4 iff NE AND3(n_{10}, NOT1 n_9, NOT1 n_8) and NE n_5 iff NE AND3(n_{10}, NOT1 n_9, n_8) and NE n_6 iff NE AND3(n_{10}, n_9, NOT1 n_8) and NE n_7 iff NE AND3(n_{10}, n_9, n_8) and NE n_8 iff NE NOT1 q_1 and NE n_9 iff NE XOR2(q_1, q_2) and NE n_{10} iff NE OR2(AND2(q_3, NOT1 q_1), AND2(q_1, XOR2(q_2, q_3))). Then

ISSN 1426–2630

(i) NE n_1 iff NE s_0,
(ii) NE n_2 iff NE s_1,
(iii) NE n_3 iff NE s_2,
(iv) NE n_4 iff NE s_3,
(v) NE n_5 iff NE s_4,
(vi) NE n_6 iff NE s_5,
(vii) NE n_7 iff NE s_6, and
(viii) NE n_0 iff NE s_7.

(2) NE $\text{AND3}(\text{AND2}(a,d),\text{AND2}(b,d),\text{AND2}(c,d))$ iff NE $\text{AND2}(\text{AND3}(a,b,c),d)$.

(3)(i) Not NE $\text{AND2}(a,b)$ iff NE $\text{OR2}(\text{NOT1}\,a,\text{NOT1}\,b)$,
(ii) NE $\text{OR2}(a,b)$ and NE $\text{OR2}(c,b)$ iff NE $\text{OR2}(\text{AND2}(a,c),b)$,
(iii) NE $\text{OR2}(a,b)$ and NE $\text{OR2}(c,b)$ and NE $\text{OR2}(d,b)$ iff NE $\text{OR2}(\text{AND3}(a,c,d),b)$, and
(iv) if NE $\text{OR2}(a,b)$ and NE a iff NE c, then NE $\text{OR2}(c,b)$.

(4) Let s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, n_0, n_1, n_2, n_3, n_4, n_5, n_6, n_7, q_1, q_2, q_3, n_8, n_9, n_{10}, R be sets such that NE s_0 iff NE $\text{AND3}(\text{NOT1}\,q_3,\text{NOT1}\,q_2,\text{NOT1}\,q_1)$ and NE s_1 iff NE $\text{AND3}(\text{NOT1}\,q_3,\text{NOT1}\,q_2,q_1)$ and NE s_2 iff NE $\text{AND3}(\text{NOT1}\,q_3,q_2,\text{NOT1}\,q_1)$ and NE s_3 iff NE $\text{AND3}(\text{NOT1}\,q_3,q_2,q_1)$ and NE s_4 iff NE $\text{AND3}(q_3,\text{NOT1}\,q_2,\text{NOT1}\,q_1)$ and NE s_5 iff NE $\text{AND3}(q_3,\text{NOT1}\,q_2,q_1)$ and NE s_6 iff NE $\text{AND3}(q_3,q_2,\text{NOT1}\,q_1)$ and NE s_7 iff NE $\text{AND3}(q_3,q_2,q_1)$ and NE n_0 iff NE $\text{AND3}(\text{NOT1}\,n_{10},\text{NOT1}\,n_9,\text{NOT1}\,n_8)$ and NE n_1 iff NE $\text{AND3}(\text{NOT1}\,n_{10},\text{NOT1}\,n_9,n_8)$ and NE n_2 iff NE $\text{AND3}(\text{NOT1}\,n_{10},n_9,\text{NOT1}\,n_8)$ and NE n_3 iff NE $\text{AND3}(\text{NOT1}\,n_{10},n_9,n_8)$ and NE n_4 iff NE $\text{AND3}(n_{10},\text{NOT1}\,n_9,\text{NOT1}\,n_8)$ and NE n_5 iff NE $\text{AND3}(n_{10},\text{NOT1}\,n_9,n_8)$ and NE n_6 iff NE $\text{AND3}(n_{10},n_9,\text{NOT1}\,n_8)$ and NE n_7 iff NE $\text{AND3}(n_{10},n_9,n_8)$ and NE n_8 iff NE $\text{AND2}(\text{NOT1}\,q_1,R)$ and NE n_9 iff NE $\text{AND2}(\text{XOR2}(q_1,q_2),R)$ and NE n_{10} iff NE $\text{AND2}(\text{OR2}(\text{AND2}(q_3,\text{NOT1}\,q_1),\text{AND2}(q_1,\text{XOR2}(q_2,q_3))),R)$. Then

(i) NE n_1 iff NE $\text{AND2}(s_0,R)$,
(ii) NE n_2 iff NE $\text{AND2}(s_1,R)$,
(iii) NE n_3 iff NE $\text{AND2}(s_2,R)$,
(iv) NE n_4 iff NE $\text{AND2}(s_3,R)$,
(v) NE n_5 iff NE $\text{AND2}(s_4,R)$,
(vi) NE n_6 iff NE $\text{AND2}(s_5,R)$,
(vii) NE n_7 iff NE $\text{AND2}(s_6,R)$, and
(viii) NE n_0 iff NE $\text{OR2}(s_7,\text{NOT1}\,R)$.

References

[1] Yatsuka Nakamura. Logic gates and logical equivalence of adders. *Formalized Mathematics*, 8(**1**):35–45, 1999.

Received March 13, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Correctness of Johnson Counter Circuits

Yuguang Yang
Shinshu University
Nagano

Katsumi Wasaki
Shinshu University
Nagano

Yasushi Fuwa
Shinshu University
Nagano

Yatsuka Nakamura
Shinshu University
Nagano

Summary. This article introduces the verification of the correctness for the operations and the specification of the Johnson counter. We formalize the concepts of 2-bit, 3-bit and 4-bit Johnson counter circuits with a reset input, and define the specification of the state transitions without the minor loop.

MML Identifier: GATE_3.

The notation and terminology used here are introduced in the paper [1].

The following propositions are true:

(1) Let s_0, s_1, s_2, s_3, n_0, n_1, n_2, n_3, q_1, q_2, n_4, n_5 be sets such that NE s_0 iff NE AND2(NOT1 q_2, NOT1 q_1) and NE s_1 iff NE AND2(NOT1 q_2, q_1) and NE s_2 iff NE AND2(q_2, NOT1 q_1) and NE s_3 iff NE AND2(q_2, q_1) and NE n_0 iff NE AND2(NOT1 n_5, NOT1 n_4) and NE n_1 iff NE AND2(NOT1 n_5, n_4) and NE n_2 iff NE AND2(n_5, NOT1 n_4) and NE n_3 iff NE AND2(n_5, n_4) and NE n_4 iff NE NOT1 q_2 and NE n_5 iff NE q_1. Then

(i) NE n_1 iff NE s_0,

(ii) NE n_3 iff NE s_1,

(iii) NE n_2 iff NE s_3, and

(iv) NE n_0 iff NE s_2.

(2) Let s_0, s_1, s_2, s_3, n_0, n_1, n_2, n_3, q_1, q_2, n_4, n_5, R be sets such that NE s_0 iff NE AND2(NOT1 q_2, NOT1 q_1) and NE s_1 iff NE AND2(NOT1 q_2, q_1) and NE s_2 iff NE AND2(q_2, NOT1 q_1) and NE s_3 iff NE AND2(q_2, q_1) and NE n_0 iff NE AND2(NOT1 n_5, NOT1 n_4) and NE n_1 iff NE AND2(NOT1 n_5, n_4) and NE n_2 iff NE AND2(n_5, NOT1 n_4) and NE n_3

ISSN 1426–2630

iff NE AND2(n_5, n_4) and NE n_4 iff NE AND2(NOT1 q_2, R) and NE n_5 iff NE AND2(q_1, R). Then

(i) NE n_1 iff NE AND2(s_0, R),

(ii) NE n_3 iff NE AND2(s_1, R),

(iii) NE n_2 iff NE AND2(s_3, R), and

(iv) NE n_0 iff NE OR2(AND2(s_2, R), NOT1 R).

(3) Let s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, n_0, n_1, n_2, n_3, n_6, n_7, n_8, n_9, q_1, q_2, q_3, n_4, n_5, n_{10} be sets such that NE s_0 iff NE AND3(NOT1 q_3, NOT1 q_2, NOT1 q_1) and NE s_1 iff NE AND3(NOT1 q_3, NOT1 q_2, q_1) and NE s_2 iff NE AND3(NOT1 q_3, q_2, NOT1 q_1) and NE s_3 iff NE AND3(NOT1 q_3, q_2, q_1) and NE s_4 iff NE AND3(q_3, NOT1 q_2, NOT1 q_1) and NE s_5 iff NE AND3(q_3, NOT1 q_2, q_1) and NE s_6 iff NE AND3(q_3, q_2, NOT1 q_1) and NE s_7 iff NE AND3(q_3, q_2, q_1) and NE n_0 iff NE AND3(NOT1 n_{10}, NOT1 n_5, NOT1 n_4) and NE n_1 iff NE AND3(NOT1 n_{10}, NOT1 n_5, n_4) and NE n_2 iff NE AND3(NOT1 n_{10}, n_5, NOT1 n_4) and NE n_3 iff NE AND3(NOT1 n_{10}, n_5, n_4) and NE n_6 iff NE AND3(n_{10}, NOT1 n_5, NOT1 n_4) and NE n_7 iff NE AND3(n_{10}, NOT1 n_5, n_4) and NE n_8 iff NE AND3(n_{10}, n_5, NOT1 n_4) and NE n_9 iff NE AND3(n_{10}, n_5, n_4) and NE n_4 iff NE NOT1 q_3 and NE n_5 iff NE q_1 and NE n_{10} iff NE q_2. Then

(i) NE n_1 iff NE s_0,

(ii) NE n_3 iff NE s_1,

(iii) NE n_9 iff NE s_3,

(iv) NE n_8 iff NE s_7,

(v) NE n_6 iff NE s_6,

(vi) NE n_0 iff NE s_4,

(vii) NE n_2 iff NE s_5, and

(viii) NE n_7 iff NE s_2.

(4) Let s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, n_0, n_1, n_2, n_3, n_6, n_7, n_8, n_9, q_1, q_2, q_3, n_4, n_5, n_{10}, R be sets such that NE s_0 iff NE AND3(NOT1 q_3, NOT1 q_2, NOT1 q_1) and NE s_1 iff NE AND3(NOT1 q_3, NOT1 q_2, q_1) and NE s_2 iff NE AND3(NOT1 q_3, q_2, NOT1 q_1) and NE s_3 iff NE AND3(NOT1 q_3, q_2, q_1) and NE s_4 iff NE AND3(q_3, NOT1 q_2, NOT1 q_1) and NE s_5 iff NE AND3(q_3, NOT1 q_2, q_1) and NE s_6 iff NE AND3(q_3, q_2, NOT1 q_1) and NE s_7 iff NE AND3(q_3, q_2, q_1) and NE n_0 iff NE AND3(NOT1 n_{10}, NOT1 n_5, NOT1 n_4) and NE n_1 iff NE AND3(NOT1 n_{10}, NOT1 n_5, n_4) and NE n_2 iff NE AND3(NOT1 n_{10}, n_5, NOT1 n_4) and NE n_3 iff NE AND3(NOT1 n_{10}, n_5, n_4) and NE n_6 iff NE AND3(n_{10}, NOT1 n_5, NOT1 n_4) and NE n_7 iff NE AND3(n_{10}, NOT1 n_5, n_4) and NE n_8 iff NE AND3(n_{10}, n_5, NOT1 n_4) and NE n_9 iff NE AND3(n_{10}, n_5, n_4) and NE n_4 iff NE AND2(NOT1 q_3, R) and NE n_5 iff NE AND2(q_1, R) and NE n_{10} iff NE AND2(q_2, R). Then

(i) NE n_1 iff NE AND2(s_0, R),
(ii) NE n_3 iff NE AND2(s_1, R),
(iii) NE n_9 iff NE AND2(s_3, R),
(iv) NE n_8 iff NE AND2(s_7, R),
(v) NE n_6 iff NE AND2(s_6, R),
(vi) NE n_0 iff NE OR2(AND2(s_4, R), NOT1 R),
(vii) NE n_2 iff NE AND2(s_5, R), and
(viii) NE n_7 iff NE AND2(s_2, R).

(5) Let s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, n_0, n_1, n_2, n_3, n_6, n_7, n_8, n_9, n_{11}, n_{12}, n_{13}, n_{14}, n_{15}, n_{16}, n_{17}, n_{18}, q_1, q_2, q_3, q_4, n_4, n_5, n_{10}, n_{19} be sets such that NE s_0 iff NE AND4(NOT1 q_4, NOT1 q_3, NOT1 q_2, NOT1 q_1) and NE s_1 iff NE AND4(NOT1 q_4, NOT1 q_3, NOT1 q_2, q_1) and NE s_2 iff NE AND4(NOT1 q_4, NOT1 q_3, q_2, NOT1 q_1) and NE s_3 iff NE AND4(NOT1 q_4, NOT1 q_3, q_2, q_1) and NE s_4 iff NE AND4(NOT1 q_4, q_3, NOT1 q_2, NOT1 q_1) and NE s_5 iff NE AND4(NOT1 q_4, q_3, NOT1 q_2, q_1) and NE s_6 iff NE AND4(NOT1 q_4, q_3, q_2, NOT1 q_1) and NE s_7 iff NE AND4(NOT1 q_4, q_3, q_2, q_1) and NE s_8 iff NE AND4(q_4, NOT1 q_3, NOT1 q_2, NOT1 q_1) and NE s_9 iff NE AND4(q_4, NOT1 q_3, NOT1 q_2, q_1) and NE s_{10} iff NE AND4(q_4, NOT1 q_3, q_2, NOT1 q_1) and NE s_{11} iff NE AND4(q_4, NOT1 q_3, q_2, q_1) and NE s_{12} iff NE AND4(q_4, q_3, NOT1 q_2, NOT1 q_1) and NE s_{13} iff NE AND4(q_4, q_3, NOT1 q_2, q_1) and NE s_{14} iff NE AND4(q_4, q_3, q_2, NOT1 q_1) and NE s_{15} iff NE AND4(q_4, q_3, q_2, q_1) and NE n_0 iff NE AND4(NOT1 n_{19}, NOT1 n_{10}, NOT1 n_5, NOT1 n_4) and NE n_1 iff NE AND4(NOT1 n_{19}, NOT1 n_{10}, NOT1 n_5, n_4) and NE n_2 iff NE AND4(NOT1 n_{19}, NOT1 n_{10}, n_5, NOT1 n_4) and NE n_3 iff NE AND4(NOT1 n_{19}, NOT1 n_{10}, n_5, n_4) and NE n_6 iff NE AND4(NOT1 n_{19}, n_{10}, NOT1 n_5, NOT1 n_4) and NE n_7 iff NE AND4(NOT1 n_{19}, n_{10}, NOT1 n_5, n_4) and NE n_8 iff NE AND4(NOT1 n_{19}, n_{10}, n_5, NOT1 n_4) and NE n_9 iff NE AND4(NOT1 n_{19}, n_{10}, n_5, n_4) and NE n_{11} iff NE AND4(n_{19}, NOT1 n_{10}, NOT1 n_5, NOT1 n_4) and NE n_{12} iff NE AND4(n_{19}, NOT1 n_{10}, NOT1 n_5, n_4) and NE n_{13} iff NE AND4(n_{19}, NOT1 n_{10}, n_5, NOT1 n_4) and NE n_{14} iff NE AND4(n_{19}, NOT1 n_{10}, n_5, n_4) and NE n_{15} iff NE AND4(n_{19}, n_{10}, NOT1 n_5, NOT1 n_4) and NE n_{16} iff NE AND4(n_{19}, n_{10}, NOT1 n_5, n_4) and NE n_{17} iff NE AND4(n_{19}, n_{10}, n_5, NOT1 n_4) and NE n_{18} iff NE AND4(n_{19}, n_{10}, n_5, n_4) and NE n_4 iff NE NOT1 q_4 and NE n_5 iff NE q_1 and NE n_{10} iff NE q_2 and NE n_{19} iff NE q_3. Then

(i) NE n_1 iff NE s_0,
(ii) NE n_3 iff NE s_1,
(iii) NE n_9 iff NE s_3,
(iv) NE n_{18} iff NE s_7,
(v) NE n_{17} iff NE s_{15},

(vi) NE n_{15} iff NE s_{14},
(vii) NE n_{11} iff NE s_{12},
(viii) NE n_0 iff NE s_8,
(ix) NE n_7 iff NE s_2,
(x) NE n_{14} iff NE s_5,
(xi) NE n_8 iff NE s_{11},
(xii) NE n_{16} iff NE s_6,
(xiii) NE n_{13} iff NE s_{13},
(xiv) NE n_6 iff NE s_{10},
(xv) NE n_{12} iff NE s_4, and
(xvi) NE n_2 iff NE s_9.

(6) Let s_0, s_1, s_2, s_3, s_4, s_5, s_6, s_7, s_8, s_9, s_{10}, s_{11}, s_{12}, s_{13}, s_{14}, s_{15}, n_0, n_1, n_2, n_3, n_6, n_7, n_8, n_9, n_{11}, n_{12}, n_{13}, n_{14}, n_{15}, n_{16}, n_{17}, n_{18}, q_1, q_2, q_3, q_4, n_4, n_5, n_{10}, n_{19}, R be sets such that NE s_0 iff NE AND4(NOT1 q_4, NOT1 q_3, NOT1 q_2, NOT1 q_1) and NE s_1 iff NE AND4(NOT1 q_4, NOT1 q_3, NOT1 q_2, q_1) and NE s_2 iff NE AND4(NOT1 q_4, NOT1 q_3, q_2, NOT1 q_1) and NE s_3 iff NE AND4(NOT1 q_4, NOT1 q_3, q_2, q_1) and NE s_4 iff NE AND4(NOT1 q_4, q_3, NOT1 q_2, NOT1 q_1) and NE s_5 iff NE AND4(NOT1 q_4, q_3, NOT1 q_2, q_1) and NE s_6 iff NE AND4(NOT1 q_4, q_3, q_2, NOT1 q_1) and NE s_7 iff NE AND4(NOT1 q_4, q_3, q_2, q_1) and NE s_8 iff NE AND4(q_4, NOT1 q_3, NOT1 q_2, NOT1 q_1) and NE s_9 iff NE AND4(q_4, NOT1 q_3, NOT1 q_2, q_1) and NE s_{10} iff NE AND4(q_4, NOT1 q_3, q_2, NOT1 q_1) and NE s_{11} iff NE AND4(q_4, NOT1 q_3, q_2, q_1) and NE s_{12} iff NE AND4(q_4, q_3, NOT1 q_2, NOT1 q_1) and NE s_{13} iff NE AND4(q_4, q_3, NOT1 q_2, q_1) and NE s_{14} iff NE AND4(q_4, q_3, q_2, NOT1 q_1) and NE s_{15} iff NE AND4(q_4, q_3, q_2, q_1) and NE n_0 iff NE AND4(NOT1 n_{19}, NOT1 n_{10}, NOT1 n_5, NOT1 n_4) and NE n_1 iff NE AND4(NOT1 n_{19}, NOT1 n_{10}, NOT1 n_5, n_4) and NE n_2 iff NE AND4(NOT1 n_{19}, NOT1 n_{10}, n_5, NOT1 n_4) and NE n_3 iff NE AND4(NOT1 n_{19}, NOT1 n_{10}, n_5, n_4) and NE n_6 iff NE AND4(NOT1 n_{19}, n_{10}, NOT1 n_5, NOT1 n_4) and NE n_7 iff NE AND4(NOT1 n_{19}, n_{10}, NOT1 n_5, n_4) and NE n_8 iff NE AND4(NOT1 n_{19}, n_{10}, n_5, NOT1 n_4) and NE n_9 iff NE AND4(NOT1 n_{19}, n_{10}, n_5, n_4) and NE n_{11} iff NE AND4(n_{19}, NOT1 n_{10}, NOT1 n_5, NOT1 n_4) and NE n_{12} iff NE AND4(n_{19}, NOT1 n_{10}, NOT1 n_5, n_4) and NE n_{13} iff NE AND4(n_{19}, NOT1 n_{10}, n_5, NOT1 n_4) and NE n_{14} iff NE AND4(n_{19}, NOT1 n_{10}, n_5, n_4) and NE n_{15} iff NE AND4(n_{19}, n_{10}, NOT1 n_5, NOT1 n_4) and NE n_{16} iff NE AND4(n_{19}, n_{10}, NOT1 n_5, n_4) and NE n_{17} iff NE AND4(n_{19}, n_{10}, n_5, NOT1 n_4) and NE n_{18} iff NE AND4(n_{19}, n_{10}, n_5, n_4) and NE n_4 iff NE AND2(NOT1 q_4, R) and NE n_5 iff NE AND2(q_1, R) and NE n_{10} iff NE AND2(q_2, R) and NE n_{19} iff NE AND2(q_3, R). Then

(i) NE n_1 iff NE AND2(s_0, R),

(ii) NE n_3 iff NE AND2(s_1, R),
(iii) NE n_9 iff NE AND2(s_3, R),
(iv) NE n_{18} iff NE AND2(s_7, R),
(v) NE n_{17} iff NE AND2(s_{15}, R),
(vi) NE n_{15} iff NE AND2(s_{14}, R),
(vii) NE n_{11} iff NE AND2(s_{12}, R),
(viii) NE n_0 iff NE OR2(AND2(s_8, R), NOT1 R),
(ix) NE n_7 iff NE AND2(s_2, R),
(x) NE n_{14} iff NE AND2(s_5, R),
(xi) NE n_8 iff NE AND2(s_{11}, R),
(xii) NE n_{16} iff NE AND2(s_6, R),
(xiii) NE n_{13} iff NE AND2(s_{13}, R),
(xiv) NE n_6 iff NE AND2(s_{10}, R),
(xv) NE n_{12} iff NE AND2(s_4, R), and
(xvi) NE n_2 iff NE AND2(s_9, R).

References

[1] Yatsuka Nakamura. Logic gates and logical equivalence of adders. *Formalized Mathematics*, 8(**1**):35–45, 1999.

Received March 13, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

The Definition of the Riemann Definite Integral and some Related Lemmas

Noboru Endou
Shinshu University
Nagano

Artur Korniłowicz[1]
University of Bialystok

Summary. This article introduces the Riemann definite integral on the closed interval of real. We present the definitions and related lemmas of the closed interval. We formalize the concept of the Riemann definite integral and the division of the closed interval of real, and prove the additivity of the integral.

MML Identifier: `INTEGRA1`.

The notation and terminology used in this paper are introduced in the following papers: [28], [31], [8], [14], [2], [5], [6], [30], [22], [32], [18], [15], [7], [20], [26], [10], [12], [3], [27], [21], [4], [29], [16], [17], [24], [9], [11], [19], [25], [13], [23], and [1].

1. Definition of Closed Interval and its Properties

For simplicity, we adopt the following rules: a, a_1, a_2, b, b_1, b_2 are real numbers, p is a finite sequence, F, G, H are finite sequences of elements of $\mathbb{R}$, i, j, k are natural numbers, f is a function from $\mathbb{R}$ into $\mathbb{R}$, and x_1 is a set.

Let I_1 be a subset of $\mathbb{R}$. We say that I_1 is closed-interval if and only if:

(Def. 1) There exist real numbers a, b such that $a \leqslant b$ and $I_1 = [a, b]$.

Let us mention that there exists a subset of $\mathbb{R}$ which is closed-interval.

In the sequel A, A_1, A_2 are closed-interval subsets of $\mathbb{R}$.

The following propositions are true:

(1) Every closed-interval subset of $\mathbb{R}$ is compact.

[1]This paper was written while the second author visited Shinshu University, winter 1999.

ISSN 1426–2630

(2) If A is a closed-interval subset of $\mathbb{R}$, then A is non empty.

Let us observe that every subset of $\mathbb{R}$ which is closed-interval is also non empty and compact.

The following proposition is true

(3) If A is a closed-interval subset of $\mathbb{R}$, then A is lower bounded and upper bounded.

Let us observe that every subset of $\mathbb{R}$ which is closed-interval is also bounded.

One can verify that there exists a subset of $\mathbb{R}$ which is closed-interval.

Next we state three propositions:

(4) If A is a closed-interval subset of $\mathbb{R}$, then there exist a, b such that $a \leqslant b$ and $a = \inf A$ and $b = \sup A$.

(5) If A is a closed-interval subset of $\mathbb{R}$, then $A = [\inf A, \sup A]$.

(6) If $A = [a_1, b_1]$ and $A = [a_2, b_2]$, then $a_1 = a_2$ and $b_1 = b_2$.

2. Definition of Division of Closed Interval and its Properties

Let A be a closed-interval subset of $\mathbb{R}$. A non empty increasing finite sequence of elements of $\mathbb{R}$ is said to be a DivisionPoint of A if:

(Def. 2) $\operatorname{rng} \mathrm{it} \subseteq A$ and $\mathrm{it}(\operatorname{len} \mathrm{it}) = \sup A$.

Let A be a closed-interval subset of $\mathbb{R}$. The functor $\operatorname{divs} A$ yielding a set is defined by:

(Def. 3) $x_1 \in \operatorname{divs} A$ iff x_1 is a DivisionPoint of A.

Let A be a closed-interval subset of $\mathbb{R}$. One can check that $\operatorname{divs} A$ is non empty.

Let A be a closed-interval subset of $\mathbb{R}$. A non empty set is called a Division of A if:

(Def. 4) $x_1 \in \mathrm{it}$ iff x_1 is a DivisionPoint of A.

Let A be a closed-interval subset of $\mathbb{R}$. Observe that there exists a Division of A which is non empty.

The following proposition is true

(7) For every closed-interval subset A of $\mathbb{R}$ and for every non empty Division S of A holds every element of S is a DivisionPoint of A.

Let A be a closed-interval subset of $\mathbb{R}$ and let S be a non empty Division of A. We see that the element of S is a DivisionPoint of A.

In the sequel S denotes a non empty Division of A and D, D_1, D_2 denote elements of S.

Next we state two propositions:

(8) If $i \in \operatorname{dom} D$, then $D(i) \in A$.

(9) If $i \in \operatorname{dom} D$ and $i \neq 1$, then $i-1 \in \operatorname{dom} D$ and $D(i-1) \in A$ and $i-1 \in \mathbb{N}$.

Let A be a closed-interval subset of $\mathbb{R}$, let S be a non empty Division of A, let D be an element of S, and let i be a natural number. Let us assume that $i \in \operatorname{dom} D$. The functor $\operatorname{divset}(D,i)$ yielding a closed-interval subset of $\mathbb{R}$ is defined as follows:

(Def. 5)(i) $\inf \operatorname{divset}(D,i) = \inf A$ and $\sup \operatorname{divset}(D,i) = D(i)$ if $i = 1$,

(ii) $\inf \operatorname{divset}(D,i) = D(i-1)$ and $\sup \operatorname{divset}(D,i) = D(i)$, otherwise.

Next we state the proposition

(10) If $i \in \operatorname{dom} D$, then $\operatorname{divset}(D,i) \subseteq A$.

Let A be a subset of $\mathbb{R}$. The functor $\operatorname{vol}(A)$ yielding a real number is defined by:

(Def. 6) $\operatorname{vol}(A) = \sup A - \inf A$.

One can prove the following proposition

(11) For every closed-interval subset A of $\mathbb{R}$ holds $0 \leqslant \operatorname{vol}(A)$.

3. Definitions of Integrability and Related Topics

Let A be a closed-interval subset of $\mathbb{R}$, let f be a partial function from A to $\mathbb{R}$, let S be a non empty Division of A, and let D be an element of S. The functor upper_volume(f,D) yielding a finite sequence of elements of $\mathbb{R}$ is defined as follows:

(Def. 7) len upper_volume(f,D) = len D and for every i such that $i \in \operatorname{Seg} \operatorname{len} D$ holds $(\text{upper\_volume}(f,D))(i) = \sup \operatorname{rng}(f \restriction \operatorname{divset}(D,i)) \cdot \operatorname{vol}(\operatorname{divset}(D,i))$.

The functor lower_volume(f,D) yielding a finite sequence of elements of $\mathbb{R}$ is defined by:

(Def. 8) len lower_volume(f,D) = len D and for every i such that $i \in \operatorname{Seg} \operatorname{len} D$ holds $(\text{lower\_volume}(f,D))(i) = \inf \operatorname{rng}(f \restriction \operatorname{divset}(D,i)) \cdot \operatorname{vol}(\operatorname{divset}(D,i))$.

Let A be a closed-interval subset of $\mathbb{R}$, let f be a partial function from A to $\mathbb{R}$, let S be a non empty Division of A, and let D be an element of S. The functor upper_sum(f,D) yields a real number and is defined by:

(Def. 9) $\text{upper\_sum}(f,D) = \sum \text{upper\_volume}(f,D)$.

The functor lower_sum(f,D) yields a real number and is defined by:

(Def. 10) $\text{lower\_sum}(f,D) = \sum \text{lower\_volume}(f,D)$.

Let A be a closed-interval subset of $\mathbb{R}$. Then divs A is a Division of A.

Let A be a closed-interval subset of $\mathbb{R}$ and let f be a partial function from A to $\mathbb{R}$. The functor upper_sum_set f yielding a partial function from divs A to $\mathbb{R}$ is defined as follows:

(Def. 11) dom upper_sum_set f = divs A and for every element D of divs A such that $D \in$ dom upper_sum_set f holds (upper_sum_set f)(D) = upper_sum(f, D).

The functor lower_sum_set f yields a partial function from divs A to $\mathbb{R}$ and is defined as follows:

(Def. 12) dom lower_sum_set f = divs A and for every element D of divs A such that $D \in$ dom lower_sum_set f holds (lower_sum_set f)(D) = lower_sum(f, D).

Let A be a closed-interval subset of $\mathbb{R}$ and let f be a partial function from A to $\mathbb{R}$. We say that f is upper integrable on A if and only if:

(Def. 13) rng upper_sum_set f is lower bounded.

We say that f is lower integrable on A if and only if:

(Def. 14) rng lower_sum_set f is upper bounded.

Let A be a closed-interval subset of $\mathbb{R}$ and let f be a partial function from A to $\mathbb{R}$. The functor upper_integral f yielding a real number is defined by:

(Def. 15) upper_integral f = inf rng upper_sum_set f.

Let A be a closed-interval subset of $\mathbb{R}$ and let f be a partial function from A to $\mathbb{R}$. The functor lower_integral f yields a real number and is defined as follows:

(Def. 16) lower_integral f = sup rng lower_sum_set f.

Let A be a closed-interval subset of $\mathbb{R}$ and let f be a partial function from A to $\mathbb{R}$. We say that f is integrable on A if and only if:

(Def. 17) f is upper integrable on A and f is lower integrable on A and upper_integral f = lower_integral f.

Let A be a closed-interval subset of $\mathbb{R}$ and let f be a partial function from A to $\mathbb{R}$. The functor integral f yields a real number and is defined by:

(Def. 18) integral f = upper_integral f.

4. Real Function's Properties

Next we state several propositions:

(12) For every non empty set X and for all partial functions f, g from X to $\mathbb{R}$ holds $\operatorname{rng}(f + g) \subseteq \operatorname{rng} f + \operatorname{rng} g$.

(13) Let A be a closed-interval subset of $\mathbb{R}$ and f be a partial function from A to $\mathbb{R}$. If f is lower bounded on A, then rng f is lower bounded.

(14) Let A be a closed-interval subset of $\mathbb{R}$ and f be a partial function from A to $\mathbb{R}$. If $\operatorname{rng} f$ is lower bounded, then f is lower bounded on A.

(15) Let A be a closed-interval subset of $\mathbb{R}$ and f be a partial function from A to $\mathbb{R}$. If f is upper bounded on A, then $\operatorname{rng} f$ is upper bounded.

(16) Let A be a closed-interval subset of $\mathbb{R}$ and f be a partial function from A to $\mathbb{R}$. If $\operatorname{rng} f$ is upper bounded, then f is upper bounded on A.

(17) Let A be a closed-interval subset of $\mathbb{R}$ and f be a partial function from A to $\mathbb{R}$. If f is bounded on A, then $\operatorname{rng} f$ is bounded.

5. Characteristic Function's Properties

The following propositions are true:

(18) For every closed-interval subset A of $\mathbb{R}$ holds $\chi_{A,A}$ is a constant on A.

(19) For every closed-interval subset A of $\mathbb{R}$ holds $\operatorname{rng}(\chi_{A,A}) = \{1\}$.

(20) For every closed-interval subset A of $\mathbb{R}$ and for every set B such that $B \cap \operatorname{dom}(\chi_{A,A}) \neq \emptyset$ holds $\operatorname{rng}(\chi_{A,A} \restriction B) = \{1\}$.

(21) If $i \in \operatorname{Seg} \operatorname{len} D$, then $\operatorname{vol}(\operatorname{divset}(D, i)) = (\operatorname{lower\_volume}(\chi_{A,A}, D))(i)$.

(22) If $i \in \operatorname{Seg} \operatorname{len} D$, then $\operatorname{vol}(\operatorname{divset}(D, i)) = (\operatorname{upper\_volume}(\chi_{A,A}, D))(i)$.

(23) If $\operatorname{len} F = \operatorname{len} G$ and $\operatorname{len} F = \operatorname{len} H$ and for every k such that $k \in \operatorname{dom} F$ holds $H(k) = F_k + G_k$, then $\sum H = \sum F + \sum G$.

(24) If $\operatorname{len} F = \operatorname{len} G$ and $\operatorname{len} F = \operatorname{len} H$ and for every k such that $k \in \operatorname{dom} F$ holds $H(k) = F_k - G_k$, then $\sum H = \sum F - \sum G$.

(25) Let A be a closed-interval subset of $\mathbb{R}$, S be a non empty Division of A, and D be an element of S. Then $\sum \operatorname{lower\_volume}(\chi_{A,A}, D) = \operatorname{vol}(A)$.

(26) Let A be a closed-interval subset of $\mathbb{R}$, S be a non empty Division of A, and D be an element of S. Then $\sum \operatorname{upper\_volume}(\chi_{A,A}, D) = \operatorname{vol}(A)$.

6. Some Properties of Darboux Sum

Let A be a closed-interval subset of $\mathbb{R}$, let f be a partial function from A to $\mathbb{R}$, let S be a non empty Division of A, and let D be an element of S. Then $\operatorname{upper\_volume}(f, D)$ is a non empty finite sequence of elements of $\mathbb{R}$.

Let A be a closed-interval subset of $\mathbb{R}$, let f be a partial function from A to $\mathbb{R}$, let S be a non empty Division of A, and let D be an element of S. Then $\operatorname{lower\_volume}(f, D)$ is a non empty finite sequence of elements of $\mathbb{R}$.

One can prove the following propositions:

(27) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, and D be an element of S. If f is total and lower bounded on A, then $\inf \operatorname{rng} f \cdot \operatorname{vol}(A) \leqslant \text{lower\_sum}(f, D)$.

(28) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, D be an element of S, and i be a natural number. Suppose f is total and upper bounded on A and $i \in \operatorname{Seg} \operatorname{len} D$. Then $\sup \operatorname{rng} f \cdot \operatorname{vol}(\operatorname{divset}(D, i)) \geqslant \sup \operatorname{rng}(f \restriction \operatorname{divset}(D, i)) \cdot \operatorname{vol}(\operatorname{divset}(D, i))$.

(29) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, and D be an element of S. If f is total and upper bounded on A, then $\text{upper\_sum}(f, D) \leqslant \sup \operatorname{rng} f \cdot \operatorname{vol}(A)$.

(30) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, and D be an element of S. If f is total and bounded on A, then $\text{lower\_sum}(f, D) \leqslant \text{upper\_sum}(f, D)$.

Let x be a non empty finite sequence of elements of $\mathbb{R}$. Then $\operatorname{rng} x$ is a finite non empty subset of $\mathbb{R}$.

Let A be a closed-interval subset of $\mathbb{R}$ and let D be an element of $\operatorname{divs} A$. The functor δ_D yielding a real number is defined by:

(Def. 19) $\delta_D = \max \operatorname{rng} \text{upper\_volume}(\chi_{A,A}, D)$.

Let A be a closed-interval subset of $\mathbb{R}$, let S be a non empty Division of A, and let D_1, D_2 be elements of S. The predicate $D_1 \leqslant D_2$ is defined as follows:

(Def. 20) $\operatorname{len} D_1 \leqslant \operatorname{len} D_2$ and $\operatorname{rng} D_1 \subseteq \operatorname{rng} D_2$.

We introduce $D_2 \geqslant D_1$ as a synonym of $D_1 \leqslant D_2$.

One can prove the following propositions:

(31) Let A be a closed-interval subset of $\mathbb{R}$, S be a non empty Division of A, and D_1, D_2 be elements of S. If $\operatorname{len} D_1 = 1$, then $D_1 \leqslant D_2$.

(32) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, and D_1, D_2 be elements of S. If f is total and upper bounded on A and $\operatorname{len} D_1 = 1$, then $\text{upper\_sum}(f, D_1) \geqslant \text{upper\_sum}(f, D_2)$.

(33) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, and D_1, D_2 be elements of S. If f is total and lower bounded on A and $\operatorname{len} D_1 = 1$, then $\text{lower\_sum}(f, D_1) \leqslant \text{lower\_sum}(f, D_2)$.

(34) Let A be a closed-interval subset of $\mathbb{R}$, S be a non empty Division of A, and D be an element of S. If $i \in \operatorname{dom} D$, then there exist A_1, A_2 such that $A_1 = [\inf A, D(i)]$ and $A_2 = [D(i), \sup A]$ and $A = A_1 \cup A_2$.

(35) Let A be a closed-interval subset of $\mathbb{R}$, S be a non empty Division of A, and D_1, D_2 be elements of S. If $i \in \operatorname{dom} D_1$, then if $D_1 \leqslant D_2$, then there exists j such that $j \in \operatorname{dom} D_2$ and $D_1(i) = D_2(j)$.

Let A be a closed-interval subset of $\mathbb{R}$, let S be a non empty Division of A, let D_1, D_2 be elements of S, and let i be a natural number. Let us assume that $D_1 \leqslant D_2$. The functor $\mathrm{indx}(D_2, D_1, i)$ yields a natural number and is defined as follows:

(Def. 21)(i) $\mathrm{indx}(D_2, D_1, i) \in \mathrm{dom}\, D_2$ and $D_1(i) = D_2(\mathrm{indx}(D_2, D_1, i))$ if $i \in \mathrm{dom}\, D_1$,

(ii) $\mathrm{indx}(D_2, D_1, i) = 0$, otherwise.

Next we state four propositions:

(36) Let p be an increasing finite sequence of elements of $\mathbb{R}$ and n be a natural number. Suppose $n \leqslant \mathrm{len}\, p$. Then $p_{\restriction n}$ is an increasing finite sequence of elements of $\mathbb{R}$.

(37) Let p be an increasing finite sequence of elements of $\mathbb{R}$ and i, j be natural numbers. Suppose $j \in \mathrm{dom}\, p$ and $i \leqslant j$. Then $\mathrm{mid}(p, i, j)$ is an increasing finite sequence of elements of $\mathbb{R}$.

(38) Let A be a closed-interval subset of $\mathbb{R}$, S be a non empty Division of A, D be an element of S, and i, j be natural numbers. Suppose $i \in \mathrm{dom}\, D$ and $j \in \mathrm{dom}\, D$ and $i \leqslant j$. Then there exists a closed-interval subset B of $\mathbb{R}$ such that $\inf B = (\mathrm{mid}(D, i, j))(1)$ and $\sup B = (\mathrm{mid}(D, i, j))(\mathrm{len}\,\mathrm{mid}(D, i, j))$ and $\mathrm{len}\,\mathrm{mid}(D, i, j) = (j - i) + 1$ and $\mathrm{mid}(D, i, j)$ is a DivisionPoint of B.

(39) Let A, B be closed-interval subsets of $\mathbb{R}$, S be a non empty Division of A, S_1 be a non empty Division of B, D be an element of S, and i, j be natural numbers. Suppose $i \in \mathrm{dom}\, D$ and $j \in \mathrm{dom}\, D$ and $i \leqslant j$ and $D(i) \geqslant \inf B$ and $D(j) = \sup B$. Then $\mathrm{mid}(D, i, j)$ is an element of S_1.

Let p be a finite sequence of elements of $\mathbb{R}$. The functor $\mathrm{PartSums}\, p$ yielding a finite sequence of elements of $\mathbb{R}$ is defined by:

(Def. 22) $\mathrm{len}\,\mathrm{PartSums}\, p = \mathrm{len}\, p$ and for every i such that $i \in \mathrm{Seg}\,\mathrm{len}\, p$ holds $(\mathrm{PartSums}\, p)(i) = \sum(p{\restriction}i)$.

We now state a number of propositions:

(40) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, and D_1, D_2 be elements of S. Suppose $D_1 \leqslant D_2$ and f is total and upper bounded on A. Let i be a non empty natural number. If $i \in \mathrm{dom}\, D_1$, then $\sum(\mathrm{upper\_volume}(f, D_1){\restriction}i) \geqslant \sum(\mathrm{upper\_volume}(f, D_2){\restriction}\,\mathrm{indx}(D_2, D_1, i))$.

(41) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, and D_1, D_2 be elements of S. Suppose $D_1 \leqslant D_2$ and f is total and lower bounded on A. Let i be a non empty natural number. If $i \in \mathrm{dom}\, D_1$, then $\sum(\mathrm{lower\_volume}(f, D_1){\restriction}i) \leqslant \sum(\mathrm{lower\_volume}(f, D_2){\restriction}\,\mathrm{indx}(D_2, D_1, i))$.

(42) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, D_1, D_2 be elements of S, and i

be a natural number. Suppose $D_1 \leqslant D_2$ and $i \in \operatorname{dom} D_1$ and f is total and upper bounded on A. Then $(\text{PartSums upper\_volume}(f, D_1))(i) \geqslant (\text{PartSums upper\_volume}(f, D_2))(\text{indx}(D_2, D_1, i))$.

(43) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, D_1, D_2 be elements of S, and i be a natural number. Suppose $D_1 \leqslant D_2$ and $i \in \operatorname{dom} D_1$ and f is total and lower bounded on A. Then $(\text{PartSums lower\_volume}(f, D_1))(i) \leqslant (\text{PartSums lower\_volume}(f, D_2))(\text{indx}(D_2, D_1, i))$.

(44) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, and D be an element of S. Then $(\text{PartSums upper\_volume}(f, D))(\operatorname{len} D) = \text{upper\_sum}(f, D)$.

(45) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, and D be an element of S. Then $(\text{PartSums lower\_volume}(f, D))(\operatorname{len} D) = \text{lower\_sum}(f, D)$.

(46) Let A be a closed-interval subset of $\mathbb{R}$, S be a non empty Division of A, and D_1, D_2 be elements of S. If $D_1 \leqslant D_2$, then $\text{indx}(D_2, D_1, \operatorname{len} D_1) = \operatorname{len} D_2$.

(47) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, and D_1, D_2 be elements of S. If $D_1 \leqslant D_2$ and f is total and upper bounded on A, then $\text{upper\_sum}(f, D_2) \leqslant \text{upper\_sum}(f, D_1)$.

(48) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, and D_1, D_2 be elements of S. If $D_1 \leqslant D_2$ and f is total and lower bounded on A, then $\text{lower\_sum}(f, D_2) \geqslant \text{lower\_sum}(f, D_1)$.

(49) Let A be a closed-interval subset of $\mathbb{R}$, S be a non empty Division of A, and D_1, D_2 be elements of S. Then there exists an element D of S such that $D_1 \leqslant D$ and $D_2 \leqslant D$.

(50) Let A be a closed-interval subset of $\mathbb{R}$, f be a partial function from A to $\mathbb{R}$, S be a non empty Division of A, and D_1, D_2 be elements of S. If f is total and bounded on A, then $\text{lower\_sum}(f, D_1) \leqslant \text{upper\_sum}(f, D_2)$.

7. Additivity of Integral

One can prove the following propositions:

(51) Let A be a closed-interval subset of $\mathbb{R}$ and f be a partial function from A to $\mathbb{R}$. Suppose f is upper integrable on A and f is lower integrable on A and f is total and bounded on A. Then $\text{upper\_integral}\, f \geqslant \text{lower\_integral}\, f$.

(52) For all subsets X, Y of $\mathbb{R}$ holds $-X + -Y = -(X + Y)$.

(53) For all subsets X, Y of $\mathbb{R}$ such that X is upper bounded and $Y \neq \emptyset$ and Y is upper bounded holds $X + Y$ is upper bounded.

(54) For all non empty subsets X, Y of $\mathbb{R}$ such that X is upper bounded and Y is upper bounded holds $\sup(X + Y) = \sup X + \sup Y$.

(55) Let A be a closed-interval subset of $\mathbb{R}$, f, g be partial functions from A to $\mathbb{R}$, S be a non empty Division of A, and D be an element of S. Suppose $i \in \operatorname{Seg} \operatorname{len} D$ and f is upper bounded on A and total and g is upper bounded on A and total. Then $(\text{upper\_volume}(f + g, D))(i) \leqslant (\text{upper\_volume}(f, D))(i) + (\text{upper\_volume}(g, D))(i)$.

(56) Let A be a closed-interval subset of $\mathbb{R}$, f, g be partial functions from A to $\mathbb{R}$, S be a non empty Division of A, and D be an element of S. Suppose $i \in \operatorname{Seg} \operatorname{len} D$ and f is lower bounded on A and total and g is lower bounded on A and total. Then $(\text{lower\_volume}(f, D))(i) + (\text{lower\_volume}(g, D))(i) \leqslant (\text{lower\_volume}(f + g, D))(i)$.

(57) Let A be a closed-interval subset of $\mathbb{R}$, f, g be partial functions from A to $\mathbb{R}$, S be a non empty Division of A, and D be an element of S. Suppose f is upper bounded on A and total and g is upper bounded on A and total. Then $\text{upper\_sum}(f + g, D) \leqslant \text{upper\_sum}(f, D) + \text{upper\_sum}(g, D)$.

(58) Let A be a closed-interval subset of $\mathbb{R}$, f, g be partial functions from A to $\mathbb{R}$, S be a non empty Division of A, and D be an element of S. Suppose f is lower bounded on A and total and g is lower bounded on A and total. Then $\text{lower\_sum}(f, D) + \text{lower\_sum}(g, D) \leqslant \text{lower\_sum}(f + g, D)$.

(59) Let X be a non empty set and f be a partial function from X to $\mathbb{R}$. If f is upper bounded on X and total, then $\operatorname{rng} f$ is upper bounded.

(60) Let X be a non empty set and f be a partial function from X to $\mathbb{R}$. If $\operatorname{rng} f$ is upper bounded and f is total, then f is upper bounded on X.

(61) Let X be a non empty set and f be a partial function from X to $\mathbb{R}$. If f is lower bounded on X and total, then $\operatorname{rng} f$ is lower bounded.

(62) Let X be a non empty set and f be a partial function from X to $\mathbb{R}$. If $\operatorname{rng} f$ is lower bounded and f is total, then f is lower bounded on X.

(63) Let A be a closed-interval subset of $\mathbb{R}$ and f, g be partial functions from A to $\mathbb{R}$. Suppose that

- (i) f is total and bounded on A,
- (ii) g is total and bounded on A,
- (iii) f is integrable on A, and
- (iv) g is integrable on A.

Then $f + g$ is integrable on A and $\operatorname{integral} f + g = \operatorname{integral} f + \operatorname{integral} g$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.

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

[3] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[4] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.
[5] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[6] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[7] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.
[8] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[9] Czesław Byliński. The sum and product of finite sequences of real numbers. *Formalized Mathematics*, 1(**4**):661–668, 1990.
[10] Czesław Byliński and Piotr Rudnicki. Bounding boxes for compact sets in $\mathcal{E}^2$. *Formalized Mathematics*, 6(**3**):427–440, 1997.
[11] Czesław Byliński and Andrzej Trybulec. Complex spaces. *Formalized Mathematics*, 2(**1**):151–158, 1991.
[12] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[13] Agata Darmochwał and Yatsuka Nakamura. The topological space $\mathcal{E}^2_{\mathrm{T}}$. Arcs, line segments and special polygonal arcs. *Formalized Mathematics*, 2(**5**):617–621, 1991.
[14] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[15] Jarosław Kotowicz. Convergent real sequences. Upper and lower bound of sets of real numbers. *Formalized Mathematics*, 1(**3**):477–481, 1990.
[16] Jarosław Kotowicz. Partial functions from a domain to a domain. *Formalized Mathematics*, 1(**4**):697–702, 1990.
[17] Jarosław Kotowicz. Partial functions from a domain to the set of real numbers. *Formalized Mathematics*, 1(**4**):703–709, 1990.
[18] Jarosław Kotowicz. Real sequences and basic operations on them. *Formalized Mathematics*, 1(**2**):269–272, 1990.
[19] Jarosław Kotowicz. Functions and finite sequences of real numbers. *Formalized Mathematics*, 3(**2**):275–278, 1992.
[20] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part I. *Formalized Mathematics*, 3(**1**):107–115, 1992.
[21] Eugeniusz Kusak, Wojciech Leończuk, and Michał Muzalewski. Abelian groups, fields and vector spaces. *Formalized Mathematics*, 1(**2**):335–342, 1990.
[22] Robert Milewski. Natural numbers. *Formalized Mathematics*, 7(**1**):19–22, 1998.
[23] Yatsuka Nakamura and Roman Matuszewski. Reconstructions of special sequences. *Formalized Mathematics*, 6(**2**):255–263, 1997.
[24] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, I. *Formalized Mathematics*, 5(**2**):167–172, 1996.
[25] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[26] Konrad Raczkowski and Paweł Sadowski. Topological properties of subsets in real numbers. *Formalized Mathematics*, 1(**4**):777–780, 1990.
[27] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[28] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[29] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[30] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[31] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[32] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received March 13, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Properties of the Trigonometric Function

Takashi Mitsuishi
Shinshu University
Nagano

Yuguang Yang
Shinshu University
Nagano

Summary. This article introduces the monotone increasing and the monotone decreasing of *sinus* and *cosine*, and definitions of hyperbolic *sinus*, hyperbolic *cosine* and hyperbolic *tangent*, and some related formulas about them.

MML Identifier: `SIN_COS2`.

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

1. Monotone Increasing and Monotone Decreasing of Sinus and Cosine

We adopt the following rules: p, q, r, t_1 are elements of $\mathbb{R}$ and n is a natural number.

Next we state a number of propositions:

(1) If $p \geqslant 0$ and $r \geqslant 0$, then $p + r \geqslant 2 \cdot \sqrt{p \cdot r}$.

(2) sin is increasing on $]0, \frac{\mathrm{Pai}}{2}[$.

(3) sin is decreasing on $]\frac{\mathrm{Pai}}{2}, \mathrm{Pai}[$.

(4) cos is decreasing on $]0, \frac{\mathrm{Pai}}{2}[$.

(5) cos is decreasing on $]\frac{\mathrm{Pai}}{2}, \mathrm{Pai}[$.

(6) sin is decreasing on $]\mathrm{Pai}, \frac{3}{2} \cdot \mathrm{Pai}[$.

(7) sin is increasing on $]\frac{3}{2} \cdot \mathrm{Pai}, 2 \cdot \mathrm{Pai}[$.

(8) cos is increasing on $]\mathrm{Pai}, \frac{3}{2} \cdot \mathrm{Pai}[$.

ISSN 1426–2630

(9) cos is increasing on $]\frac{3}{2} \cdot \mathrm{Pai}, 2 \cdot \mathrm{Pai}[$.

(10) $(\sin)(t_1) = (\sin)(2 \cdot \mathrm{Pai} \cdot n + t_1)$.

(11) $(\cos)(t_1) = (\cos)(2 \cdot \mathrm{Pai} \cdot n + t_1)$.

2. Hyperbolic Sinus, Hyperbolic Cosine and Hyperbolic Tangent

The partial function sinh from $\mathbb{R}$ to $\mathbb{R}$ is defined as follows:

(Def. 1) $\mathrm{dom}\,\mathrm{sinh} = \mathbb{R}$ and for every real number d holds $(\sinh)(d) = \frac{(\exp)(d) - (\exp)(-d)}{2}$.

Let d be a real number. The functor $\sinh d$ yielding an element of $\mathbb{R}$ is defined by:

(Def. 2) $\sinh d = (\sinh)(d)$.

The partial function cosh from $\mathbb{R}$ to $\mathbb{R}$ is defined as follows:

(Def. 3) $\mathrm{dom}\,\mathrm{cosh} = \mathbb{R}$ and for every real number d holds $(\cosh)(d) = \frac{(\exp)(d) + (\exp)(-d)}{2}$.

Let d be a real number. The functor $\cosh d$ yields an element of $\mathbb{R}$ and is defined as follows:

(Def. 4) $\cosh d = (\cosh)(d)$.

The partial function tanh from $\mathbb{R}$ to $\mathbb{R}$ is defined as follows:

(Def. 5) $\mathrm{dom}\,\mathrm{tanh} = \mathbb{R}$ and for every real number d holds $(\tanh)(d) = \frac{(\exp)(d) - (\exp)(-d)}{(\exp)(d) + (\exp)(-d)}$.

Let d be a real number. The functor $\tanh d$ yields an element of $\mathbb{R}$ and is defined as follows:

(Def. 6) $\tanh d = (\tanh)(d)$.

We now state a number of propositions:

(12) $(\exp)(p + q) = (\exp)(p) \cdot (\exp)(q)$.

(13) $(\exp)(0) = 1$.

(14) $(\cosh)(p)^{\mathbf{2}} - (\sinh)(p)^{\mathbf{2}} = 1$ and $(\cosh)(p) \cdot (\cosh)(p) - (\sinh)(p) \cdot (\sinh)(p) = 1$.

(15) $(\cosh)(p) \neq 0$ and $(\cosh)(p) > 0$ and $(\cosh)(0) = 1$.

(16) $(\sinh)(0) = 0$.

(17) $(\tanh)(p) = \frac{(\sinh)(p)}{(\cosh)(p)}$.

(18) $(\sinh)(p)^{\mathbf{2}} = \frac{1}{2} \cdot ((\cosh)(2 \cdot p) - 1)$ and $(\cosh)(p)^{\mathbf{2}} = \frac{1}{2} \cdot ((\cosh)(2 \cdot p) + 1)$.

(19) $(\cosh)(-p) = (\cosh)(p)$ and $(\sinh)(-p) = -(\sinh)(p)$ and $(\tanh)(-p) = -(\tanh)(p)$.

(20) $(\cosh)(p + r) = (\cosh)(p) \cdot (\cosh)(r) + (\sinh)(p) \cdot (\sinh)(r)$ and $(\cosh)(p - r) = (\cosh)(p) \cdot (\cosh)(r) - (\sinh)(p) \cdot (\sinh)(r)$.

(21) $(\sinh)(p+r) = (\sinh)(p) \cdot (\cosh)(r) + (\cosh)(p) \cdot (\sinh)(r)$ and $(\sinh)(p - r) = (\sinh)(p) \cdot (\cosh)(r) - (\cosh)(p) \cdot (\sinh)(r)$.

(22) $(\tanh)(p+r) = \frac{(\tanh)(p)+(\tanh)(r)}{1+(\tanh)(p)\cdot(\tanh)(r)}$ and $(\tanh)(p-r) = \frac{(\tanh)(p)-(\tanh)(r)}{1-(\tanh)(p)\cdot(\tanh)(r)}$.

(23) $(\sinh)(2 \cdot p) = 2 \cdot (\sinh)(p) \cdot (\cosh)(p)$ and $(\cosh)(2 \cdot p) = 2 \cdot (\cosh)(p)^{\mathbf{2}} - 1$ and $(\tanh)(2 \cdot p) = \frac{2\cdot(\tanh)(p)}{1+(\tanh)(p)^{\mathbf{2}}}$.

(24) $(\sinh)(p)^{\mathbf{2}} - (\sinh)(q)^{\mathbf{2}} = (\sinh)(p+q) \cdot (\sinh)(p-q)$ and $(\sinh)(p+q) \cdot (\sinh)(p-q) = (\cosh)(p)^{\mathbf{2}} - (\cosh)(q)^{\mathbf{2}}$ and $(\sinh)(p)^{\mathbf{2}} - (\sinh)(q)^{\mathbf{2}} = (\cosh)(p)^{\mathbf{2}} - (\cosh)(q)^{\mathbf{2}}$.

(25) $(\sinh)(p)^{\mathbf{2}} + (\cosh)(q)^{\mathbf{2}} = (\cosh)(p+q) \cdot (\cosh)(p-q)$ and $(\cosh)(p+q) \cdot (\cosh)(p-q) = (\cosh)(p)^{\mathbf{2}} + (\sinh)(q)^{\mathbf{2}}$ and $(\sinh)(p)^{\mathbf{2}} + (\cosh)(q)^{\mathbf{2}} = (\cosh)(p)^{\mathbf{2}} + (\sinh)(q)^{\mathbf{2}}$.

(26) $(\sinh)(p) + (\sinh)(r) = 2 \cdot (\sinh)(\frac{p}{2} + \frac{r}{2}) \cdot (\cosh)(\frac{p}{2} - \frac{r}{2})$ and $(\sinh)(p) - (\sinh)(r) = 2 \cdot (\sinh)(\frac{p}{2} - \frac{r}{2}) \cdot (\cosh)(\frac{p}{2} + \frac{r}{2})$.

(27) $(\cosh)(p) + (\cosh)(r) = 2 \cdot (\cosh)(\frac{p}{2} + \frac{r}{2}) \cdot (\cosh)(\frac{p}{2} - \frac{r}{2})$ and $(\cosh)(p) - (\cosh)(r) = 2 \cdot (\sinh)(\frac{p}{2} - \frac{r}{2}) \cdot (\sinh)(\frac{p}{2} + \frac{r}{2})$.

(28) $(\tanh)(p) + (\tanh)(r) = \frac{(\sinh)(p+r)}{(\cosh)(p)\cdot(\cosh)(r)}$ and $(\tanh)(p) - (\tanh)(r) = \frac{(\sinh)(p-r)}{(\cosh)(p)\cdot(\cosh)(r)}$.

(29) $((\cosh)(p) + (\sinh)(p))^{n}_{\mathbb{N}} = (\cosh)(n \cdot p) + (\sinh)(n \cdot p)$.

One can check the following observations:

* sinh is total,
* cosh is total, and
* tanh is total.

One can prove the following propositions:

(30) $\operatorname{dom}\sinh = \mathbb{R}$ and $\operatorname{dom}\cosh = \mathbb{R}$ and $\operatorname{dom}\tanh = \mathbb{R}$.

(31) sinh is differentiable in p and $(\sinh)'(p) = (\cosh)(p)$.

(32) cosh is differentiable in p and $(\cosh)'(p) = (\sinh)(p)$.

(33) tanh is differentiable in p and $(\tanh)'(p) = \frac{1}{(\cosh)(p)^{\mathbf{2}}}$.

(34) sinh is differentiable on $\mathbb{R}$ and $(\sinh)'(p) = (\cosh)(p)$.

(35) cosh is differentiable on $\mathbb{R}$ and $(\cosh)'(p) = (\sinh)(p)$.

(36) tanh is differentiable on $\mathbb{R}$ and $(\tanh)'(p) = \frac{1}{(\cosh)(p)^{\mathbf{2}}}$.

(37) $(\cosh)(p) \geqslant 1$.

(38) sinh is continuous in p.

(39) cosh is continuous in p.

(40) tanh is continuous in p.

(41) sinh is continuous on $\mathbb{R}$.

(42) cosh is continuous on $\mathbb{R}$.

(43) tanh is continuous on $\mathbb{R}$.

(44) $(\tanh)(p) < 1$ and $(\tanh)(p) > -1$.

References

[1] Agnieszka Banachowicz and Anna Winnicka. Complex sequences. *Formalized Mathematics*, 4(**1**):121–124, 1993.
[2] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[3] Czesław Byliński. The complex numbers. *Formalized Mathematics*, 1(**3**):507–513, 1990.
[4] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[5] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.
[6] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[7] Jarosław Kotowicz. Convergent sequences and the limit of sequences. *Formalized Mathematics*, 1(**2**):273–275, 1990.
[8] Jarosław Kotowicz. Monotone real sequences. Subsequences. *Formalized Mathematics*, 1(**3**):471–475, 1990.
[9] Jarosław Kotowicz. Partial functions from a domain to the set of real numbers. *Formalized Mathematics*, 1(**4**):703–709, 1990.
[10] Jarosław Kotowicz. Properties of real functions. *Formalized Mathematics*, 1(**4**):781–786, 1990.
[11] Jarosław Kotowicz. Real sequences and basic operations on them. *Formalized Mathematics*, 1(**2**):269–272, 1990.
[12] Eugeniusz Kusak, Wojciech Leończuk, and Michał Muzalewski. Abelian groups, fields and vector spaces. *Formalized Mathematics*, 1(**2**):335–342, 1990.
[13] Konrad Raczkowski. Integer and rational exponents. *Formalized Mathematics*, 2(**1**):125–130, 1991.
[14] Konrad Raczkowski and Paweł Sadowski. Real function continuity. *Formalized Mathematics*, 1(**4**):787–791, 1990.
[15] Konrad Raczkowski and Paweł Sadowski. Real function differentiability. *Formalized Mathematics*, 1(**4**):797–801, 1990.
[16] Konrad Raczkowski and Paweł Sadowski. Topological properties of subsets in real numbers. *Formalized Mathematics*, 1(**4**):777–780, 1990.
[17] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[18] Andrzej Trybulec and Czesław Byliński. Some properties of real numbers. *Formalized Mathematics*, 1(**3**):445–449, 1990.
[19] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 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.
[23] Yuguang Yang and Yasunari Shidama. Trigonometric functions and existence of circle ratio. *Formalized Mathematics*, 7(**2**):255–263, 1998.

Received March 13, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Predicate Calculus for Boolean Valued Functions. Part II

Shunichi Kobayashi
Shinshu University
Nagano

Yatsuka Nakamura
Shinshu University
Nagano

Summary. In this paper, we have proved some elementary predicate calculus formulae containing the quantifiers of Boolean valued functions with respect to partitions. Such a theory is an analogy of usual predicate logic.

MML Identifier: `BVFUNC_4`.

The terminology and notation used in this paper are introduced in the following articles: [8], [10], [11], [2], [3], [7], [6], [9], [1], [4], and [5].

1. Preliminaries

In this paper Y denotes a non empty set.

Next we state a number of propositions:

(1) For all elements a, b, c of $\mathrm{BVF}(Y)$ such that $a \Subset b \Rightarrow c$ holds $a \wedge b \Subset c$.

(2) For all elements a, b, c of $\mathrm{BVF}(Y)$ such that $a \wedge b \Subset c$ holds $a \Subset b \Rightarrow c$.

(3) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \vee a \wedge b = a$.

(4) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \wedge (a \vee b) = a$.

(5) For every element a of $\mathrm{BVF}(Y)$ holds $a \wedge \neg a = \mathit{false}(Y)$.

(6) For every element a of $\mathrm{BVF}(Y)$ holds $a \vee \neg a = \mathit{true}(Y)$.

(7) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \Leftrightarrow b = (a \Rightarrow b) \wedge (b \Rightarrow a)$.

(8) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b = \neg a \vee b$.

(9) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \oplus b = \neg a \wedge b \vee a \wedge \neg b$.

ISSN 1426–2630

(10) For all elements a, b of BVF(Y) holds $a \Leftrightarrow b = \mathit{true}(Y)$ iff $a \Rightarrow b = \mathit{true}(Y)$ and $b \Rightarrow a = \mathit{true}(Y)$.

(11) For all elements a, b, c of BVF(Y) such that $a \Leftrightarrow b = \mathit{true}(Y)$ and $b \Leftrightarrow c = \mathit{true}(Y)$ holds $a \Leftrightarrow c = \mathit{true}(Y)$.

(12) For all elements a, b of BVF(Y) such that $a \Leftrightarrow b = \mathit{true}(Y)$ holds $\neg a \Leftrightarrow \neg b = \mathit{true}(Y)$.

(13) For all elements a, b, c, d of BVF(Y) such that $a \Leftrightarrow b = \mathit{true}(Y)$ and $c \Leftrightarrow d = \mathit{true}(Y)$ holds $a \wedge c \Leftrightarrow b \wedge d = \mathit{true}(Y)$.

(14) For all elements a, b, c, d of BVF(Y) such that $a \Leftrightarrow b = \mathit{true}(Y)$ and $c \Leftrightarrow d = \mathit{true}(Y)$ holds $a \Rightarrow c \Leftrightarrow b \Rightarrow d = \mathit{true}(Y)$.

(15) For all elements a, b, c, d of BVF(Y) such that $a \Leftrightarrow b = \mathit{true}(Y)$ and $c \Leftrightarrow d = \mathit{true}(Y)$ holds $a \vee c \Leftrightarrow b \vee d = \mathit{true}(Y)$.

(16) For all elements a, b, c, d of BVF(Y) such that $a \Leftrightarrow b = \mathit{true}(Y)$ and $c \Leftrightarrow d = \mathit{true}(Y)$ holds $a \Leftrightarrow c \Leftrightarrow b \Leftrightarrow d = \mathit{true}(Y)$.

2. Predicate Calculus

Next we state a number of propositions:

(17) Let a, b be elements of BVF(Y), G be a subset of PARTITIONS(Y), and P_1 be a partition of Y. If G is a coordinate and $P_1 \in G$, then $\forall_{a \Leftrightarrow b, P_1} G = \forall_{a \Rightarrow b, P_1} G \wedge \forall_{b \Rightarrow a, P_1} G$.

(18) Let a be an element of BVF(Y), G be a subset of PARTITIONS(Y), and P_1, P_2 be partitions of Y. Suppose G is a coordinate and $P_1 \in G$ and $P_2 \in G$. Then $\forall_{a,P_1} G \Subset \exists_{a,P_1} G$ and $\forall_{a,P_1} G \Subset \exists_{a,P_2} G$.

(19) Let a, u be elements of BVF(Y), G be a subset of PARTITIONS(Y), and P_1 be a partition of Y. Suppose G is a coordinate and $P_1 \in G$ and u is independent of P_1, G. If $a \Rightarrow u = \mathit{true}(Y)$, then $\forall_{a,P_1} G \Rightarrow u = \mathit{true}(Y)$.

(20) Let u be an element of BVF(Y), G be a subset of PARTITIONS(Y), and P_1 be a partition of Y. Suppose G is a coordinate and $P_1 \in G$ and u is independent of P_1, G. Then $\exists_{u,P_1} G \Subset u$.

(21) Let u be an element of BVF(Y), G be a subset of PARTITIONS(Y), and P_1 be a partition of Y. Suppose G is a coordinate and $P_1 \in G$ and u is independent of P_1, G. Then $u \Subset \forall_{u,P_1} G$.

(22) Let u be an element of BVF(Y), G be a subset of PARTITIONS(Y), and P_1, P_2 be partitions of Y. Suppose G is a coordinate and $P_1 \in G$ and $P_2 \in G$ and u is independent of P_2, G. Then $\forall_{u,P_1} G \Subset \forall_{u,P_2} G$.

(23) Let u be an element of BVF(Y), G be a subset of PARTITIONS(Y), and P_1, P_2 be partitions of Y. Suppose G is a coordinate and $P_1 \in G$ and $P_2 \in G$ and u is independent of P_1, G. Then $\exists_{u,P_1} G \Subset \exists_{u,P_2} G$.

(24) Let a, b be elements of BVF(Y), G be a subset of PARTITIONS(Y), and P_1 be a partition of Y. If G is a coordinate and $P_1 \in G$, then $\forall_{a \Leftrightarrow b, P_1} G \Subset \forall_{a,P_1} G \Leftrightarrow \forall_{b,P_1} G$.

(25) Let a, b be elements of BVF(Y), G be a subset of PARTITIONS(Y), and P_1 be a partition of Y. If G is a coordinate and $P_1 \in G$, then $\forall_{a \wedge b, P_1} G \Subset a \wedge \forall_{b,P_1} G$.

(26) Let a, u be elements of BVF(Y), G be a subset of PARTITIONS(Y), and P_1 be a partition of Y. Suppose G is a coordinate and $P_1 \in G$ and u is independent of P_1, G. Then $\forall_{a,P_1} G \Rightarrow u \Subset \exists_{a \Rightarrow u, P_1} G$.

(27) Let a, b be elements of BVF(Y), G be a subset of PARTITIONS(Y), and P_1 be a partition of Y. Suppose G is a coordinate and $P_1 \in G$. If $a \Leftrightarrow b = \mathit{true}(Y)$, then $\forall_{a,P_1} G \Leftrightarrow \forall_{b,P_1} G = \mathit{true}(Y)$.

(28) Let a, b be elements of BVF(Y), G be a subset of PARTITIONS(Y), and P_1 be a partition of Y. Suppose G is a coordinate and $P_1 \in G$. If $a \Leftrightarrow b = \mathit{true}(Y)$, then $\exists_{a,P_1} G \Leftrightarrow \exists_{b,P_1} G = \mathit{true}(Y)$.

References

[1] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.
[2] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[3] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[4] Shunichi Kobayashi and Kui Jia. A theory of Boolean valued functions and partitions. *Formalized Mathematics*, 7(**2**):249–254, 1998.
[5] Shunichi Kobayashi and Yatsuka Nakamura. A theory of Boolean valued functions and quantifiers with respect to partitions. *Formalized Mathematics*, 7(**2**):307–312, 1998.
[6] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[7] Konrad Raczkowski and Paweł Sadowski. Equivalence relations and classes of abstraction. *Formalized Mathematics*, 1(**3**):441–444, 1990.
[8] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[9] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[10] Edmund Woronowicz. Many–argument relations. *Formalized Mathematics*, 1(**4**):733–737, 1990.
[11] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received March 13, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Propositional Calculus for Boolean Valued Functions. Part I

Shunichi Kobayashi
Shinshu University
Nagano

Yatsuka Nakamura
Shinshu University
Nagano

Summary. In this paper, we have proved some elementary propositional calculus formulae for Boolean valued functions.

MML Identifier: `BVFUNC_5`.

The terminology and notation used in this paper have been introduced in the following articles: [6], [8], [9], [2], [3], [5], [1], [7], and [4].

In this paper Y is a non empty set.

Next we state a number of propositions:

(1) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a = \mathit{true}(Y)$ and $b = \mathit{true}(Y)$ iff $a \wedge b = \mathit{true}(Y)$.

(2) For all elements a, b of $\mathrm{BVF}(Y)$ such that $a = \mathit{true}(Y)$ and $a \Rightarrow b = \mathit{true}(Y)$ holds $b = \mathit{true}(Y)$.

(3) For all elements a, b of $\mathrm{BVF}(Y)$ such that $a = \mathit{true}(Y)$ holds $a \vee b = \mathit{true}(Y)$.

(5)[1] For all elements a, b of $\mathrm{BVF}(Y)$ such that $b = \mathit{true}(Y)$ holds $a \Rightarrow b = \mathit{true}(Y)$.

(6) For all elements a, b of $\mathrm{BVF}(Y)$ such that $\neg a = \mathit{true}(Y)$ holds $a \Rightarrow b = \mathit{true}(Y)$.

(7) For every element a of $\mathrm{BVF}(Y)$ holds $\neg(a \wedge \neg a) = \mathit{true}(Y)$.

(8) For every element a of $\mathrm{BVF}(Y)$ holds $a \Rightarrow a = \mathit{true}(Y)$.

(9) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b = \mathit{true}(Y)$ iff $\neg b \Rightarrow \neg a = \mathit{true}(Y)$.

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

ISSN 1426–2630

(10) For all elements a, b, c of BVF(Y) such that $a \Rightarrow b = \mathit{true}(Y)$ and $b \Rightarrow c = \mathit{true}(Y)$ holds $a \Rightarrow c = \mathit{true}(Y)$.

(11) For all elements a, b of BVF(Y) such that $a \Rightarrow b = \mathit{true}(Y)$ and $a \Rightarrow \neg b = \mathit{true}(Y)$ holds $\neg a = \mathit{true}(Y)$.

(12) For every element a of BVF(Y) holds $\neg a \Rightarrow a \Rightarrow a = \mathit{true}(Y)$.

(13) For all elements a, b, c of BVF(Y) holds $a \Rightarrow b \Rightarrow \neg(b \wedge c) \Rightarrow \neg(a \wedge c) = \mathit{true}(Y)$.

(14) For all elements a, b, c of BVF(Y) holds $a \Rightarrow b \Rightarrow b \Rightarrow c \Rightarrow a \Rightarrow c = \mathit{true}(Y)$.

(15) For all elements a, b, c of BVF(Y) such that $a \Rightarrow b = \mathit{true}(Y)$ holds $b \Rightarrow c \Rightarrow a \Rightarrow c = \mathit{true}(Y)$.

(16) For all elements a, b of BVF(Y) holds $b \Rightarrow a \Rightarrow b = \mathit{true}(Y)$.

(17) For all elements a, b, c of BVF(Y) holds $a \Rightarrow b \Rightarrow c \Rightarrow b \Rightarrow c = \mathit{true}(Y)$.

(18) For all elements a, b of BVF(Y) holds $b \Rightarrow b \Rightarrow a \Rightarrow a = \mathit{true}(Y)$.

(19) For all elements a, b, c of BVF(Y) holds $c \Rightarrow b \Rightarrow a \Rightarrow b \Rightarrow c \Rightarrow a = \mathit{true}(Y)$.

(20) For all elements a, b, c of BVF(Y) holds $b \Rightarrow c \Rightarrow a \Rightarrow b \Rightarrow a \Rightarrow c = \mathit{true}(Y)$.

(21) For all elements a, b, c of BVF(Y) holds $b \Rightarrow b \Rightarrow c \Rightarrow b \Rightarrow c = \mathit{true}(Y)$.

(22) For all elements a, b, c of BVF(Y) holds $a \Rightarrow b \Rightarrow c \Rightarrow a \Rightarrow b \Rightarrow a \Rightarrow c = \mathit{true}(Y)$.

(23) For all elements a, b of BVF(Y) such that $a = \mathit{true}(Y)$ holds $a \Rightarrow b \Rightarrow b = \mathit{true}(Y)$.

(24) For all elements a, b, c of BVF(Y) such that $c \Rightarrow b \Rightarrow a = \mathit{true}(Y)$ holds $b \Rightarrow c \Rightarrow a = \mathit{true}(Y)$.

(25) For all elements a, b, c of BVF(Y) such that $c \Rightarrow b \Rightarrow a = \mathit{true}(Y)$ and $b = \mathit{true}(Y)$ holds $c \Rightarrow a = \mathit{true}(Y)$.

(26) For all elements a, b, c of BVF(Y) such that $c \Rightarrow b \Rightarrow a = \mathit{true}(Y)$ and $b = \mathit{true}(Y)$ and $c = \mathit{true}(Y)$ holds $a = \mathit{true}(Y)$.

(27) For all elements b, c of BVF(Y) such that $b \Rightarrow b \Rightarrow c = \mathit{true}(Y)$ holds $b \Rightarrow c = \mathit{true}(Y)$.

(28) For all elements a, b, c of BVF(Y) such that $a \Rightarrow b \Rightarrow c = \mathit{true}(Y)$ holds $a \Rightarrow b \Rightarrow a \Rightarrow c = \mathit{true}(Y)$.

(29) For all elements a, b, c of BVF(Y) such that $a \Rightarrow b \Rightarrow c = \mathit{true}(Y)$ and $a \Rightarrow b = \mathit{true}(Y)$ holds $a \Rightarrow c = \mathit{true}(Y)$.

(30) For all elements a, b, c of BVF(Y) such that $a \Rightarrow b \Rightarrow c = \mathit{true}(Y)$ and $a \Rightarrow b = \mathit{true}(Y)$ and $a = \mathit{true}(Y)$ holds $c = \mathit{true}(Y)$.

(31) For all elements a, b, c, d of BVF(Y) such that $a \Rightarrow b \Rightarrow c = \mathit{true}(Y)$ and $a \Rightarrow c \Rightarrow d = \mathit{true}(Y)$ holds $a \Rightarrow b \Rightarrow d = \mathit{true}(Y)$.

(32) For all elements a, b of $\mathrm{BVF}(Y)$ holds $\neg a \Rightarrow \neg b \Rightarrow b \Rightarrow a = \mathit{true}(Y)$.

(33) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \Rightarrow \neg b \Rightarrow \neg a = \mathit{true}(Y)$.

(34) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \Rightarrow \neg b \Rightarrow b \Rightarrow \neg a = \mathit{true}(Y)$.

(35) For all elements a, b of $\mathrm{BVF}(Y)$ holds $\neg a \Rightarrow b \Rightarrow \neg b \Rightarrow a = \mathit{true}(Y)$.

(36) For every element a of $\mathrm{BVF}(Y)$ holds $a \Rightarrow \neg a \Rightarrow \neg a = \mathit{true}(Y)$.

(37) For all elements a, b of $\mathrm{BVF}(Y)$ holds $\neg a \Rightarrow a \Rightarrow b = \mathit{true}(Y)$.

(38) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $\neg(a \wedge b \wedge c) = \neg a \vee \neg b \vee \neg c$.

(39) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $\neg(a \vee b \vee c) = \neg a \wedge \neg b \wedge \neg c$.

(40) For all elements a, b, c, d of $\mathrm{BVF}(Y)$ holds $a \vee b \wedge c \wedge d = (a \vee b) \wedge (a \vee c) \wedge (a \vee d)$.

(41) For all elements a, b, c, d of $\mathrm{BVF}(Y)$ holds $a \wedge (b \vee c \vee d) = a \wedge b \vee a \wedge c \vee a \wedge d$.

References

[1] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.
[2] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[3] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[4] Shunichi Kobayashi and Kui Jia. A theory of Boolean valued functions and partitions. *Formalized Mathematics*, 7(**2**):249–254, 1998.
[5] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[6] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[7] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[8] Edmund Woronowicz. Many–argument relations. *Formalized Mathematics*, 1(**4**):733–737, 1990.
[9] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received March 13, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Propositional Calculus for Boolean Valued Functions. Part II

Shunichi Kobayashi
Shinshu University
Nagano

Yatsuka Nakamura
Shinshu University
Nagano

Summary. In this paper, we have proved some elementary propositional calculus formulae for Boolean valued functions.

MML Identifier: BVFUNC_6.

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

In this paper Y denotes a non empty set.

The following propositions are true:

(1) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \Rightarrow a \wedge b = \mathit{true}(Y)$.

(2) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \Rightarrow b \Rightarrow a \Rightarrow a \Leftrightarrow b = \mathit{true}(Y)$.

(3) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \vee b \Leftrightarrow b \vee a = \mathit{true}(Y)$.

(4) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \wedge b \Rightarrow c \Rightarrow a \Rightarrow b \Rightarrow c = \mathit{true}(Y)$.

(5) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \Rightarrow c \Rightarrow a \wedge b \Rightarrow c = \mathit{true}(Y)$.

(6) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $c \Rightarrow a \Rightarrow c \Rightarrow b \Rightarrow c \Rightarrow a \wedge b = \mathit{true}(Y)$.

(7) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \vee b \Rightarrow c \Rightarrow (a \Rightarrow c) \vee (b \Rightarrow c) = \mathit{true}(Y)$.

(8) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \Rightarrow c \Rightarrow b \Rightarrow c \Rightarrow a \vee b \Rightarrow c = \mathit{true}(Y)$.

(9) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $(a \Rightarrow c) \wedge (b \Rightarrow c) \Rightarrow a \vee b \Rightarrow c = \mathit{true}(Y)$.

ISSN 1426–2630

(10) For all elements a, b of BVF(Y) holds $a \Rightarrow b \wedge \neg b \Rightarrow \neg a = true(Y)$.

(11) For all elements a, b, c of BVF(Y) holds $(a \vee b) \wedge (a \vee c) \Rightarrow a \vee b \wedge c = true(Y)$.

(12) For all elements a, b, c of BVF(Y) holds $a \wedge (b \vee c) \Rightarrow a \wedge b \vee a \wedge c = true(Y)$.

(13) For all elements a, b, c of BVF(Y) holds $(a \vee c) \wedge (b \vee c) \Rightarrow a \wedge b \vee c = true(Y)$.

(14) For all elements a, b, c of BVF(Y) holds $(a \vee b) \wedge c \Rightarrow a \wedge c \vee b \wedge c = true(Y)$.

(15) For all elements a, b of BVF(Y) such that $a \wedge b = true(Y)$ holds $a \vee b = true(Y)$.

(16) For all elements a, b, c of BVF(Y) such that $a \Rightarrow b = true(Y)$ holds $a \vee c \Rightarrow b \vee c = true(Y)$.

(17) For all elements a, b, c of BVF(Y) such that $a \Rightarrow b = true(Y)$ holds $a \wedge c \Rightarrow b \wedge c = true(Y)$.

(18) For all elements a, b, c of BVF(Y) such that $c \Rightarrow a = true(Y)$ and $c \Rightarrow b = true(Y)$ holds $c \Rightarrow a \wedge b = true(Y)$.

(19) For all elements a, b, c of BVF(Y) such that $a \Rightarrow c = true(Y)$ and $b \Rightarrow c = true(Y)$ holds $a \vee b \Rightarrow c = true(Y)$.

(20) For all elements a, b of BVF(Y) such that $a \vee b = true(Y)$ and $\neg a = true(Y)$ holds $b = true(Y)$.

(21) For all elements a, b, c, d of BVF(Y) such that $a \Rightarrow b = true(Y)$ and $c \Rightarrow d = true(Y)$ holds $a \wedge c \Rightarrow b \wedge d = true(Y)$.

(22) For all elements a, b, c, d of BVF(Y) such that $a \Rightarrow b = true(Y)$ and $c \Rightarrow d = true(Y)$ holds $a \vee c \Rightarrow b \vee d = true(Y)$.

(23) For all elements a, b of BVF(Y) such that $a \wedge \neg b \Rightarrow \neg a = true(Y)$ holds $a \Rightarrow b = true(Y)$.

(24) For all elements a, b of BVF(Y) such that $\neg a \Rightarrow \neg b = true(Y)$ holds $b \Rightarrow a = true(Y)$.

(25) For all elements a, b of BVF(Y) such that $a \Rightarrow \neg b = true(Y)$ holds $b \Rightarrow \neg a = true(Y)$.

(26) For all elements a, b of BVF(Y) such that $\neg a \Rightarrow b = true(Y)$ holds $\neg b \Rightarrow a = true(Y)$.

(27) For all elements a, b of BVF(Y) holds $a \Rightarrow a \vee b = true(Y)$.

(28) For all elements a, b of BVF(Y) holds $a \vee b \Rightarrow \neg a \Rightarrow b = true(Y)$.

(29) For all elements a, b of BVF(Y) holds $\neg(a \vee b) \Rightarrow \neg a \wedge \neg b = true(Y)$.

(30) For all elements a, b of BVF(Y) holds $\neg a \wedge \neg b \Rightarrow \neg(a \vee b) = true(Y)$.

(31) For all elements a, b of BVF(Y) holds $\neg(a \vee b) \Rightarrow \neg a = true(Y)$.

(32) For every element a of BVF(Y) holds $a \vee a \Rightarrow a = true(Y)$.

(33) For all elements a, b of BVF(Y) holds $a \wedge \neg a \Rightarrow b = true(Y)$.

(34) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \Rightarrow \neg a \vee b = \mathit{true}(Y)$.

(35) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \wedge b \Rightarrow \neg(a \Rightarrow \neg b) = \mathit{true}(Y)$.

(36) For all elements a, b of $\mathrm{BVF}(Y)$ holds $\neg(a \Rightarrow \neg b) \Rightarrow a \wedge b = \mathit{true}(Y)$.

(37) For all elements a, b of $\mathrm{BVF}(Y)$ holds $\neg(a \wedge b) \Rightarrow \neg a \vee \neg b = \mathit{true}(Y)$.

(38) For all elements a, b of $\mathrm{BVF}(Y)$ holds $\neg a \vee \neg b \Rightarrow \neg(a \wedge b) = \mathit{true}(Y)$.

(39) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \wedge b \Rightarrow a = \mathit{true}(Y)$.

(40) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \wedge b \Rightarrow a \vee b = \mathit{true}(Y)$.

(41) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \wedge b \Rightarrow b = \mathit{true}(Y)$.

(42) For every element a of $\mathrm{BVF}(Y)$ holds $a \Rightarrow a \wedge a = \mathit{true}(Y)$.

(43) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \Leftrightarrow b \Rightarrow a \Rightarrow b = \mathit{true}(Y)$.

(44) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \Leftrightarrow b \Rightarrow b \Rightarrow a = \mathit{true}(Y)$.

(45) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \vee b \vee c \Rightarrow a \vee (b \vee c) = \mathit{true}(Y)$.

(46) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \wedge b \wedge c \Rightarrow a \wedge (b \wedge c) = \mathit{true}(Y)$.

(47) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \vee (b \vee c) \Rightarrow a \vee b \vee c = \mathit{true}(Y)$.

References

[1] Shunichi Kobayashi and Kui Jia. A theory of Boolean valued functions and partitions. *Formalized Mathematics*, 7(**2**):249–254, 1998.

[2] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.

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

[4] Edmund Woronowicz. Many–argument relations. *Formalized Mathematics*, 1(**4**):733–737, 1990.

Received March 13, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Insert Sort on $\mathbf{SCM}_{\mathrm{FSA}}$[1]

Jing-Chao Chen
Shanghai Jiaotong University

Summary. This article describes the insert sorting algorithm using macro instructions such as if-Macro (conditional branch macro instructions), for-loop macro instructions and While-Macro instructions etc. From the viewpoint of initialization, we generalize the halting and computing problem of the While-Macro. Generally speaking, it is difficult to judge whether the While-Macro is halting or not by way of loop inspection. For this reason, we introduce a practical and simple method, called body-inspection. That is, in many cases, we can prove the halting problem of the While-Macro by only verifying the nature of the body of the While-Macro, rather than the While-Macro itself. In fact, we have used this method in justifying the halting of the insert sorting algorithm. Finally, we prove that the insert sorting algorithm given in the article is autonomic and its computing result is correct.

MML Identifier: `SCMISORT`.

The articles [28], [39], [20], [8], [13], [40], [14], [38], [15], [16], [12], [7], [10], [9], [23], [30], [11], [26], [34], [31], [32], [33], [25], [5], [6], [3], [1], [17], [2], [35], [37], [18], [27], [29], [24], [4], [22], [19], [21], and [36] provide the terminology and notation for this paper.

1. Preliminaries

Let i be a good instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Observe that Macro($i$) is good.

Let a be a read-write integer location and let b be an integer location. Note that AddTo(a, b) is good.

We now state several propositions:

[1]This research is supported by the National Natural Science Foundation of China Grant No. 69873033.

ISSN 1426–2630

(1) For every function f and for all sets d, r such that $d \in \operatorname{dom} f$ holds $\operatorname{dom} f = \operatorname{dom}(f +\cdot (d \dot{\longmapsto} r))$.

(2) Let p be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$, and i_1 be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $l \in \operatorname{dom} p$ and there exists an instruction p_1 of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $p_1 = p(l)$ and $\mathrm{UsedIntLoc}(p_1) = \mathrm{UsedIntLoc}(i_1)$. Then $\mathrm{UsedIntLoc}(p) = \mathrm{UsedIntLoc}(p +\cdot (l \dot{\longmapsto} i_1))$.

(3) For every integer location a and for every macro instruction I holds $(\mathbf{if}\ a > 0\ \mathbf{then}\ I; \mathrm{Goto}(\mathrm{insloc}(0))\ \mathbf{else}\ (\mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}}))(\mathrm{insloc}(\operatorname{card} I + 4)) = \mathrm{goto}\ \ \mathrm{insloc}(\operatorname{card} I + 4)$.

(4) Let p be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$, and i_1 be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $l \in \operatorname{dom} p$ and there exists an instruction p_1 of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $p_1 = p(l)$ and $\mathrm{UsedInt}^* \mathrm{Loc}(p_1) = \mathrm{UsedInt}^* \mathrm{Loc}(i_1)$. Then $\mathrm{UsedInt}^* \mathrm{Loc}(p) = \mathrm{UsedInt}^* \mathrm{Loc}(p +\cdot (l \dot{\longmapsto} i_1))$.

(5) For every natural number k holds $k + 1 > 0$.

For simplicity, we adopt the following convention: s is a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I is a macro instruction, a is a read-write integer location, and j, k, n are natural numbers.

Next we state a number of propositions:

(6) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every macro instruction I such that $s(\mathrm{intloc}(0)) = 1$ and $\mathbf{IC}_s = \mathrm{insloc}(0)$ holds $s +\cdot I = s +\cdot \mathrm{Initialized}(I)$.

(7) Let I be a macro instruction and a, b be integer locations. If I does not destroy b, then $\mathbf{while}\ a > 0\ \mathbf{do}\ I$ does not destroy b.

(8) If $n \leqslant 11$, then $n = 0$ or $n = 1$ or $n = 2$ or $n = 3$ or $n = 4$ or $n = 5$ or $n = 6$ or $n = 7$ or $n = 8$ or $n = 9$ or $n = 10$ or $n = 11$.

(9) Let f, g be finite sequences of elements of $\mathbb{Z}$ and m, n be natural numbers. Suppose $1 \leqslant n$ and $n \leqslant \operatorname{len} f$ and $1 \leqslant m$ and $m \leqslant \operatorname{len} f$ and $g = f +\cdot (m, \pi_n f) +\cdot (n, \pi_m f)$. Then

(i) $f(m) = g(n)$,

(ii) $f(n) = g(m)$,

(iii) for every set k such that $k \neq m$ and $k \neq n$ and $k \in \operatorname{dom} f$ holds $f(k) = g(k)$, and

(iv) f and g are fiberwise equipotent.

(10) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. Suppose I is halting on $\mathrm{Initialize}(s)$. Let a be an integer location. Then $(\mathrm{IExec}(I, s))(a) = (\mathrm{Computation}(\mathrm{Initialize}(s) +\cdot (I +\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))$ $(\mathrm{LifeSpan}(\mathrm{Initialize}(s) +\cdot (I +\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(a)$.

(11) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a InitHalting macro instruction. Suppose $\mathrm{Initialized}(I) \subseteq s_1$ and $\mathrm{Initialized}(I) \subseteq$

s_2 and s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$. Let k be a natural number. Then (Computation(s_1))(k) and (Computation(s_2))(k) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$ and CurInstr((Computation(s_1))(k)) = CurInstr((Computation (s_2))(k)).

(12) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a InitHalting macro instruction. Suppose Initialized(I) $\subseteq s_1$ and Initialized(I) $\subseteq s_2$ and s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$. Then LifeSpan($s_1$) = LifeSpan($s_2$) and Result($s_1$) and Result($s_2$) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(13) For every macro instruction I and for every finite sequence location f holds $f \notin \mathrm{dom}\, I$.

(14) For every macro instruction I and for every integer location a holds $a \notin \mathrm{dom}\, I$.

(15) Let N be a non empty set with non empty elements, S be a halting von Neumann definite AMI over N, and s be a state of S. If LifeSpan(s) $\leqslant j$ and s is halting, then (Computation(s))(j) = (Computation(s))(LifeSpan(s)).

2. Basic Property of **while** Macro

We now state several propositions:

(16) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I be a macro instruction, and a be a read-write integer location. Suppose $s(a) \leqslant 0$. Then **while** $a > 0$ **do** I is halting onInit s and **while** $a > 0$ **do** I is closed onInit s.

(17) Let a be an integer location, I be a macro instruction, s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and k be a natural number. Suppose that

(i) I is closed onInit s,

(ii) I is halting onInit s,

(iii) $k <$ LifeSpan(s+· Initialized(I)),

(iv) $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathbf{while}\ a>0\ \mathbf{do}\ I)))(1+k)} = \mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(I)))(k)} + 4$, and

(v) (Computation(s+· Initialized(**while** $a > 0$ **do** I)))($1 + k$)$\upharpoonright D$ = (Computation(s+· Initialized(I)))(k)$\upharpoonright D$.

Then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathbf{while}\ a>0\ \mathbf{do}\ I)))(1+k+1)} = \mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(I)))(k+1)} + 4$ and (Computation(s+· Initialized (**while** $a > 0$ **do** I)))($1+k+1$)$\upharpoonright D$ = (Computation(s+· Initialized(I)))($k+1$)$\upharpoonright D$, where D = Int-Locations $\cup$ FinSeq-Locations.

(18) Let a be an integer location, I be a macro instruction, and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose I is closed onInit s and I is halting onInit s and $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathbf{while}\ a>0\ \mathbf{do}\ I)))(1+\mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(I)))} = \mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(I)))(\mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(I)))} + 4$.
Then $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathbf{while}\ a > 0\ \mathbf{do}\ I)))(1 + \mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(I)))) = \mathrm{goto}\ \mathrm{insloc}(\mathrm{card}\, I + 4)$.

(19) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I be a macro instruction, and a be a read-write integer location. Suppose I is closed onInit s and I is halting onInit s and $s(a) > 0$. Then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathbf{while}\ a>0\ \mathbf{do}\ I)))(\mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(I))+3)} = \mathrm{insloc}(0)$ and for every natural number k such that $k \leqslant \mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(I)) + 3$ holds $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathbf{while}\ a>0\ \mathbf{do}\ I)))(k)} \in \mathrm{dom}(\mathbf{while}\ a > 0\ \mathbf{do}\ I)$.

(20) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I be a macro instruction, and a be a read-write integer location. Suppose I is closed onInit s and I is halting onInit s and $s(a) > 0$. Let k be a natural number. If $k \leqslant \mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(I)) + 3$, then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathbf{while}\ a>0\ \mathbf{do}\ I)))(k)} \in \mathrm{dom}(\mathbf{while}\ a > 0\ \mathbf{do}\ I)$.

(21) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I be a macro instruction, and a be a read-write integer location. Suppose I is closed onInit s and I is halting onInit s and $s(a) > 0$. Then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathbf{while}\ a>0\ \mathbf{do}\ I)))(\mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(I))+3)} = \mathrm{insloc}(0)$ and $(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(\mathbf{while}\ a > 0\ \mathbf{do}\ I)))(\mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(I)) + 3)\restriction D = (\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(I)))(\mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(I)))\restriction D$, where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

(22) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I be a InitHalting macro instruction, and a be a read-write integer location. Suppose $s(a) > 0$. Then there exists a state s_2 of $\mathbf{SCM}_{\mathrm{FSA}}$ and there exists a natural number k such that
- (i) $s_2 = s+\cdot\,\mathrm{Initialized}(\mathbf{while}\ a > 0\ \mathbf{do}\ I)$,
- (ii) $k = \mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(I)) + 3$,
- (iii) $\mathbf{IC}_{(\mathrm{Computation}(s_2))(k)} = \mathrm{insloc}(0)$,
- (iv) for every integer location b holds $(\mathrm{Computation}(s_2))(k)(b) = (\mathrm{IExec}(I, s))(b)$, and
- (v) for every finite sequence location f holds $(\mathrm{Computation}(s_2))(k)(f) = (\mathrm{IExec}(I, s))(f)$.

Let us consider s, I, a. The functor *StepWhile>0*(a, s, I) yields a function from $\mathbb{N}$ into $\prod$(the object kind of $\mathbf{SCM}_{\mathrm{FSA}}$) and is defined by the conditions (Def. 1).

(Def. 1)(i) (*StepWhile>0*$(a, s, I))(0) = s$ **qua** element of $\prod$(the object kind of $\mathbf{SCM}_{\mathrm{FSA}}$) **qua** non empty set, and

(ii) for every natural number i and for every element x of $\prod$(the object kind of $\mathbf{SCM}_{\mathrm{FSA}}$) **qua** non empty set such that $x = (\mathit{StepWhile}{>}0(a, s, I))(i)$ holds $(\mathit{StepWhile}{>}0(a, s, I))(i + 1) = (\mathrm{Computation}(x{+\cdot}\,\mathrm{Initialized}(\mathbf{while}\ a > 0\ \mathbf{do}\ I)))(\mathrm{LifeSpan}(x{+\cdot}\,\mathrm{Initialized}(I)) + 3)$.

We now state several propositions:

(23) $(\mathit{StepWhile}{>}0(a, s, I))(0) = s$.

(24) $(\mathit{StepWhile}{>}0(a, s, I))(k{+}1) = (\mathrm{Computation}((\mathit{StepWhile}{>}0(a, s, I))(k) {+\cdot}\,\mathrm{Initialized}(\mathbf{while}\ a > 0\ \mathbf{do}\ I)))(\mathrm{LifeSpan}((\mathit{StepWhile}{>}0(a, s, I))(k){+\cdot}\,\mathrm{Initialized}(I)) + 3)$.

(25) $(\mathit{StepWhile}{>}0(a, s, I))(k{+}1) = (\mathit{StepWhile}{>}0(a, (\mathit{StepWhile}{>}0(a, s, I))(k), I))(1)$.

(26) Let I be a macro instruction, a be a read-write integer location, and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $(\mathit{StepWhile}{>}0(a, s, I))(0 + 1) = (\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(\mathbf{while}\ a > 0\ \mathbf{do}\ I)))(\mathrm{LifeSpan}(s{+\cdot}\,\mathrm{Initialized}(I)) + 3)$.

(27) Let I be a macro instruction, a be a read-write integer location, s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and k, n be natural numbers. Suppose $\mathbf{IC}_{(\mathit{StepWhile}>0(a,s,I))(k)} = \mathrm{insloc}(0)$ and $(\mathit{StepWhile}{>}0(a, s, I))(k) = (\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(\mathbf{while}\ a > 0\ \mathbf{do}\ I)))(n)$ and $(\mathit{StepWhile}{>}0(a, s, I))(k)(\mathrm{intloc}(0)) = 1$.
Then $(\mathit{StepWhile}{>}0(a, s, I))(k) = (\mathit{StepWhile}{>}0(a, s, I))(k){+\cdot}\,\mathrm{Initialized}(\mathbf{while}\ a > 0\ \mathbf{do}\ I)$ and $(\mathit{StepWhile}{>}0(a, s, I))(k{+}1) = (\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(\mathbf{while}\ a > 0\ \mathbf{do}\ I)))(n{+}(\mathrm{LifeSpan}((\mathit{StepWhile}{>}0(a, s, I))(k){+\cdot}\,\mathrm{Initialized}(I)) + 3))$.

(28) Let I be a macro instruction, a be a read-write integer location, and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Given a function f from $\prod$(the object kind of $\mathbf{SCM}_{\mathrm{FSA}}$) into $\mathbb{N}$ such that let k be a natural number. Then

(i) if $f((\mathit{StepWhile}{>}0(a, s, I))(k)) \neq 0$, then $f((\mathit{StepWhile}{>}0(a, s, I))(k + 1)) < f((\mathit{StepWhile}{>}0(a, s, I))(k))$ and I is closed onInit $(\mathit{StepWhile}{>}0(a, s, I))(k)$ and I is halting onInit $(\mathit{StepWhile}{>}0(a, s, I))(k)$,

(ii) $(\mathit{StepWhile}{>}0(a, s, I))(k + 1)(\mathrm{intloc}(0)) = 1$, and

(iii) $f((\mathit{StepWhile}{>}0(a, s, I))(k)) = 0$ iff $(\mathit{StepWhile}{>}0(a, s, I))(k)(a) \leqslant 0$.

Then **while** $a > 0$ **do** I is halting onInit s and **while** $a > 0$ **do** I is closed onInit s.

(29) Let I be a good InitHalting macro instruction and a be a read-write integer location. Suppose that for every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $s(a) > 0$ holds $(\mathrm{IExec}(I, s))(a) < s(a)$. Then **while** $a > 0$ **do** I is InitHalting.

(30) Let I be a good InitHalting macro instruction and a be a read-write integer location. Suppose that for every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ holds

(IExec$(I, s))(a) < s(a)$ or (IExec$(I, s))(a) \leqslant 0$. Then **while** $a > 0$ **do** I is InitHalting.

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

One can prove the following propositions:

(31) Let I be a good InitHalting macro instruction and a be a read-write integer location. Given a function f from $\prod$ (the object kind of $\mathbf{SCM}_{\mathrm{FSA}}$) into $\mathbb{Z}$ such that let s, t be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Then
  (i) if $f(s) > 0$, then $f(\mathrm{IExec}(I, s)) < f(s)$,
  (ii) if $s \restriction D = t \restriction D$, then $f(s) = f(t)$, and
  (iii) $f(s) \leqslant 0$ iff $s(a) \leqslant 0$.
  Then **while** $a > 0$ **do** I is InitHalting, where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

(32) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I be a macro instruction, and a be a read-write integer location. If $s(a) \leqslant 0$, then IExec(**while** $a > 0$ **do** $I, s) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = \text{Initialize}(s) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(33) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I be a good InitHalting macro instruction, and a be a read-write integer location. If $s(a) > 0$ and **while** $a > 0$ **do** I is InitHalting, then IExec(**while** $a > 0$ **do** $I, s) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) =$ IExec(**while** $a > 0$ **do** $I, \mathrm{IExec}(I, s)) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(34) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I be a macro instruction, f be a finite sequence location, and a be a read-write integer location. If $s(a) \leqslant 0$, then (IExec(**while** $a > 0$ **do** $I, s))(f) = s(f)$.

(35) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I be a macro instruction, b be an integer location, and a be a read-write integer location. If $s(a) \leqslant 0$, then (IExec(**while** $a > 0$ **do** $I, s))(b) = (\mathrm{Initialize}(s))(b)$.

(36) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I be a good InitHalting macro instruction, f be a finite sequence location, and a be a read-write integer location. If $s(a) > 0$ and **while** $a > 0$ **do** I is InitHalting, then (IExec(**while** $a > 0$ **do** $I, s))(f) =$ (IExec(**while** $a > 0$ **do** $I, \mathrm{IExec}(I, s)))(f)$.

(37) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I be a good InitHalting macro instruction, b be an integer location, and a be a read-write integer location. If $s(a) > 0$ and **while** $a > 0$ **do** I is InitHalting, then (IExec(**while** $a > 0$ **do** $I, s))(b) =$ (IExec(**while** $a > 0$ **do** $I, \mathrm{IExec}(I, s)))(b)$.

3. Insert Sort Algorithm

Let f be a finite sequence location. The functor insert − sort f yields a macro instruction and is defined as follows:

(Def. 2) insert − sort $f = i_2$;(a_1:=len$f$);SubFrom(a_1, a_0); Times(a_1, (a_2:=len$f$); SubFrom(a_2, a_1);(a_3:=a_2);AddTo(a_3, a_0);(a_6:=f_{a_3});SubFrom(a_4, a_4); (**while** $a_2 > 0$ **do** ((a_5:=f_{a_2});SubFrom(a_5, a_6);(**if** $a_5 > 0$ **then** Macro (SubFrom(a_2, a_2)) **else** (AddTo(a_4, a_0);SubFrom(a_2, a_0))))); Times(a_4, (a_2:=a_3);SubFrom(a_3, a_0);(a_5:=f_{a_2});(a_6:=f_{a_3});(f_{a_2}:=a_6);(f_{a_3}:=a_5))), where i_2 = (a_2:=a_0);(a_3:=a_0);(a_4:=a_0);(a_5:=a_0);(a_6:=a_0), a_2 = intloc(2), a_0 = intloc(0), a_3 = intloc(3), a_4 = intloc(4), a_5 = intloc(5), a_6 = intloc(6), and a_1 = intloc(1).

The macro instruction Insert − Sort − Algorithm is defined by:

(Def. 3) Insert − Sort − Algorithm = insert − sort fsloc(0).

We now state a number of propositions:

(38) For every finite sequence location f holds UsedIntLoc(insert − sort f) = $\{a_0, a_1, a_2, a_3, a_4, a_5, a_6\}$, where a_0 = intloc(0), a_1 = intloc(1), a_2 = intloc(2), a_3 = intloc(3), a_4 = intloc(4), a_5 = intloc(5), and a_6 = intloc(6).

(39) For every finite sequence location f holds UsedInt* Loc(insert − sort f) = $\{f\}$.

(40) For all instructions k_1, k_2, k_3, k_4 of $\mathbf{SCM}_{\mathrm{FSA}}$ holds card(k_1;k_2;k_3;k_4) = 8.

(41) For all instructions k_1, k_2, k_3, k_4, k_5 of $\mathbf{SCM}_{\mathrm{FSA}}$ holds card(k_1;k_2;k_3;k_4;k_5) = 10.

(42) For every finite sequence location f holds card insert − sort f = 82.

(43) For every finite sequence location f and for every natural number k such that $k < 82$ holds insloc(k) ∈ dom insert − sort f.

(44) insert − sort fsloc(0) is keepInt0 1 and InitHalting.

(45) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Then

(i) $s(f_0)$ and (IExec(insert − sort f_0, s))(f_0) are fiberwise equipotent, and

(ii) for all natural numbers i, j such that $i \geqslant 1$ and $j \leqslant$ len $s(f_0)$ and $i < j$ and for all integers x_1, x_2 such that x_1 = (IExec(insert − sort f_0, s))(f_0)(i) and x_2 = (IExec(insert − sort f_0, s))(f_0)(j) holds $x_1 \geqslant x_2$,

where f_0 = fsloc(0).

(46) Let i be a natural number, s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and w be a finite sequence of elements of $\mathbb{Z}$. If Initialized(Insert − Sort − Algorithm)+·(fsloc(0) $\dot{\longmapsto} w$) ⊆ s, then $\mathbf{IC}_{(\mathrm{Computation}(s))(i)}$ ∈ dom Insert − Sort − Algorithm.

(47) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and t be a finite sequence of elements of $\mathbb{Z}$. Suppose Initialized(Insert − Sort − Algorithm)+·(fsloc(0)$\dot{\longmapsto} t$) ⊆ s. Then there exists a finite sequence u of elements of $\mathbb{R}$ such that

(i) t and u are fiberwise equipotent,
(ii) u is non-increasing and a finite sequence of elements of $\mathbb{Z}$, and
(iii) $(\text{Result}(s))(\text{fsloc}(0)) = u$.

(48) For every finite sequence w of elements of $\mathbb{Z}$ holds Initialized(Insert − Sort − Algorithm)$+\cdot$(fsloc(0)$\longmapsto w$) is autonomic.

(49) Initialized(Insert − Sort − Algorithm) computes Sorting-Function.

References

[1] Noriko Asamoto. Conditional branch macro instructions of $\mathbf{SCM}_{\rm FSA}$. Part I. *Formalized Mathematics*, 6(**1**):65–72, 1997.
[2] Noriko Asamoto. Conditional branch macro instructions of $\mathbf{SCM}_{\rm FSA}$. Part II. *Formalized Mathematics*, 6(**1**):73–80, 1997.
[3] Noriko Asamoto. Constant assignment macro instructions of $\mathbf{SCM}_{\rm FSA}$. Part II. *Formalized Mathematics*, 6(**1**):59–63, 1997.
[4] Noriko Asamoto. The `loop` and `Times` macroinstruction for $\mathbf{SCM}_{\rm FSA}$. *Formalized Mathematics*, 6(**4**):483–497, 1997.
[5] Noriko Asamoto, Yatsuka Nakamura, Piotr Rudnicki, and Andrzej Trybulec. On the composition of macro instructions. Part II. *Formalized Mathematics*, 6(**1**):41–47, 1997.
[6] Noriko Asamoto, Yatsuka Nakamura, Piotr Rudnicki, and Andrzej Trybulec. On the composition of macro instructions. Part III. *Formalized Mathematics*, 6(**1**):53–57, 1997.
[7] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.
[8] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[9] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[10] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[11] Grzegorz Bancerek and Piotr Rudnicki. Development of terminology for **scm**. *Formalized Mathematics*, 4(**1**):61–67, 1993.
[12] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.
[13] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.
[14] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[15] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[16] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[17] Jing-Chao Chen. While macro instructions of $\mathbf{SCM}_{\rm FSA}$. *Formalized Mathematics*, 6(**4**):553–561, 1997.
[18] Jing-Chao Chen and Yatsuka Nakamura. Bubble sort on $\mathbf{SCM}_{\rm FSA}$. *Formalized Mathematics*, 7(**1**):153–161, 1998.
[19] Jing-Chao Chen and Yatsuka Nakamura. Initialization halting concepts and their basic properties of $\mathbf{SCM}_{\rm FSA}$. *Formalized Mathematics*, 7(**1**):139–151, 1998.
[20] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[21] Krzysztof Hryniewiecki. Recursive definitions. *Formalized Mathematics*, 1(**2**):321–328, 1990.
[22] Jarosław Kotowicz. Functions and finite sequences of real numbers. *Formalized Mathematics*, 3(**2**):275–278, 1992.
[23] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[24] Andrzej Nędzusiak. σ-fields and probability. *Formalized Mathematics*, 1(**2**):401–407, 1990.
[25] Piotr Rudnicki and Andrzej Trybulec. Memory handling for $\mathbf{SCM}_{\rm FSA}$. *Formalized Mathematics*, 6(**1**):29–36, 1997.

[26] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[27] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[28] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[29] Andrzej Trybulec and Agata Darmochwał. Boolean domains. *Formalized Mathematics*, 1(**1**):187–190, 1990.
[30] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[31] Andrzej Trybulec and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):571–576, 1996.
[32] Andrzej Trybulec and Yatsuka Nakamura. Relocability for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):583–586, 1996.
[33] Andrzej Trybulec, Yatsuka Nakamura, and Noriko Asamoto. On the compositions of macro instructions. Part I. *Formalized Mathematics*, 6(**1**):21–27, 1997.
[34] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\mathrm{FSA}}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[35] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[36] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.
[37] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[38] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[39] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[40] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received March 13, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Correctness of a Cyclic Redundancy Check Code Generator

Yuguang Yang
Shinshu University
Nagano

Katsumi Wasaki
Shinshu University
Nagano

Yasushi Fuwa
Shinshu University
Nagano

Yatsuka Nakamura
Shinshu University
Nagano

Summary. We prove the correctness of the division circuit and the CRC (cyclic redundancy checks) circuit by verifying the contents of the register after one shift. Circuits with 12-bit register and 16-bit register are taken as examples. All the proofs are done formally.

MML Identifier: GATE_4.

The terminology and notation used here are introduced in the article [1].

1. Correctness of Division Circuits with 12-bit Register and 16-bit Register

One can prove the following propositions:

(1) Let g_0, g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_{10}, g_{11}, g_{12}, a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, p be sets such that NE g_0 and NE g_{12} and NE b_0 iff NE $\mathrm{XOR2}(p, \mathrm{AND2}(g_0, a_{11}))$ and NE b_1 iff NE $\mathrm{XOR2}(a_0, \mathrm{AND2}(g_1, a_{11}))$ and NE b_2 iff NE $\mathrm{XOR2}(a_1, \mathrm{AND2}(g_2, a_{11}))$ and NE b_3 iff NE $\mathrm{XOR2}(a_2, \mathrm{AND2}(g_3, a_{11}))$ and NE b_4 iff NE $\mathrm{XOR2}(a_3, \mathrm{AND2}(g_4, a_{11}))$ and NE b_5 iff NE $\mathrm{XOR2}(a_4, \mathrm{AND2}(g_5, a_{11}))$ and NE b_6 iff NE $\mathrm{XOR2}(a_5, \mathrm{AND2}(g_6, a_{11}))$ and NE b_7 iff NE $\mathrm{XOR2}(a_6, \mathrm{AND2}(g_7, a_{11}))$

ISSN 1426–2630

and NE b_8 iff NE XOR2$(a_7, \text{AND2}(g_8, a_{11}))$ and NE b_9 iff NE XOR2$(a_8, \text{AND2}(g_9, a_{11}))$ and NE b_{10} iff NE XOR2$(a_9, \text{AND2}(g_{10}, a_{11}))$ and NE b_{11} iff NE XOR2$(a_{10}, \text{AND2}(g_{11}, a_{11}))$. Then

(i) NE a_{11} iff NE AND2(g_{12}, a_{11}),
(ii) NE a_{10} iff NE XOR2$(b_{11}, \text{AND2}(g_{11}, a_{11}))$,
(iii) NE a_9 iff NE XOR2$(b_{10}, \text{AND2}(g_{10}, a_{11}))$,
(iv) NE a_8 iff NE XOR2$(b_9, \text{AND2}(g_9, a_{11}))$,
(v) NE a_7 iff NE XOR2$(b_8, \text{AND2}(g_8, a_{11}))$,
(vi) NE a_6 iff NE XOR2$(b_7, \text{AND2}(g_7, a_{11}))$,
(vii) NE a_5 iff NE XOR2$(b_6, \text{AND2}(g_6, a_{11}))$,
(viii) NE a_4 iff NE XOR2$(b_5, \text{AND2}(g_5, a_{11}))$,
(ix) NE a_3 iff NE XOR2$(b_4, \text{AND2}(g_4, a_{11}))$,
(x) NE a_2 iff NE XOR2$(b_3, \text{AND2}(g_3, a_{11}))$,
(xi) NE a_1 iff NE XOR2$(b_2, \text{AND2}(g_2, a_{11}))$,
(xii) NE a_0 iff NE XOR2$(b_1, \text{AND2}(g_1, a_{11}))$, and
(xiii) NE p iff NE XOR2$(b_0, \text{AND2}(g_0, a_{11}))$.

(2) Let g_0, g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_{10}, g_{11}, g_{12}, g_{13}, g_{14}, g_{15}, g_{16}, a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{15}, p be sets such that NE g_0 and NE g_{16} and NE b_0 iff NE XOR2$(p, \text{AND2}(g_0, a_{15}))$ and NE b_1 iff NE XOR2$(a_0, \text{AND2}(g_1, a_{15}))$ and NE b_2 iff NE XOR2$(a_1, \text{AND2}(g_2, a_{15}))$ and NE b_3 iff NE XOR2$(a_2, \text{AND2}(g_3, a_{15}))$ and NE b_4 iff NE XOR2$(a_3, \text{AND2}(g_4, a_{15}))$ and NE b_5 iff NE XOR2$(a_4, \text{AND2}(g_5, a_{15}))$ and NE b_6 iff NE XOR2$(a_5, \text{AND2}(g_6, a_{15}))$ and NE b_7 iff NE XOR2$(a_6, \text{AND2}(g_7, a_{15}))$ and NE b_8 iff NE XOR2$(a_7, \text{AND2}(g_8, a_{15}))$ and NE b_9 iff NE XOR2$(a_8, \text{AND2}(g_9, a_{15}))$ and NE b_{10} iff NE XOR2$(a_9, \text{AND2}(g_{10}, a_{15}))$ and NE b_{11} iff NE XOR2$(a_{10}, \text{AND2}(g_{11}, a_{15}))$ and NE b_{12} iff NE XOR2$(a_{11}, \text{AND2}(g_{12}, a_{15}))$ and NE b_{13} iff NE XOR2$(a_{12}, \text{AND2}(g_{13}, a_{15}))$ and NE b_{14} iff NE XOR2$(a_{13}, \text{AND2}(g_{14}, a_{15}))$ and NE b_{15} iff NE XOR2$(a_{14}, \text{AND2}(g_{15}, a_{15}))$. Then

(i) NE a_{15} iff NE AND2(g_{16}, a_{15}),
(ii) NE a_{14} iff NE XOR2$(b_{15}, \text{AND2}(g_{15}, a_{15}))$,
(iii) NE a_{13} iff NE XOR2$(b_{14}, \text{AND2}(g_{14}, a_{15}))$,
(iv) NE a_{12} iff NE XOR2$(b_{13}, \text{AND2}(g_{13}, a_{15}))$,
(v) NE a_{11} iff NE XOR2$(b_{12}, \text{AND2}(g_{12}, a_{15}))$,
(vi) NE a_{10} iff NE XOR2$(b_{11}, \text{AND2}(g_{11}, a_{15}))$,
(vii) NE a_9 iff NE XOR2$(b_{10}, \text{AND2}(g_{10}, a_{15}))$,
(viii) NE a_8 iff NE XOR2$(b_9, \text{AND2}(g_9, a_{15}))$,
(ix) NE a_7 iff NE XOR2$(b_8, \text{AND2}(g_8, a_{15}))$,
(x) NE a_6 iff NE XOR2$(b_7, \text{AND2}(g_7, a_{15}))$,
(xi) NE a_5 iff NE XOR2$(b_6, \text{AND2}(g_6, a_{15}))$,
(xii) NE a_4 iff NE XOR2$(b_5, \text{AND2}(g_5, a_{15}))$,

(xiii) NE a_3 iff NE XOR2(b_4, AND2(g_4, a_{15})),
(xiv) NE a_2 iff NE XOR2(b_3, AND2(g_3, a_{15})),
(xv) NE a_1 iff NE XOR2(b_2, AND2(g_2, a_{15})),
(xvi) NE a_0 iff NE XOR2(b_1, AND2(g_1, a_{15})), and
(xvii) NE p iff NE XOR2(b_0, AND2(g_0, a_{15})).

2. Correctness of CRC Circuits with Generator Polynomial of Degree 12 and 16

Next we state two propositions:

(3) Let g_0, g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_{10}, g_{11}, g_{12}, a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, z, p be sets such that NE g_0 and NE g_{12} and not NE z and NE b_0 iff NE XOR2(p, a_{11}) and NE b_1 iff NE XOR2(a_0, AND2(g_1, b_0)) and NE b_2 iff NE XOR2(a_1, AND2(g_2, b_0)) and NE b_3 iff NE XOR2(a_2, AND2(g_3, b_0)) and NE b_4 iff NE XOR2(a_3, AND2(g_4, b_0)) and NE b_5 iff NE XOR2(a_4, AND2(g_5, b_0)) and NE b_6 iff NE XOR2(a_5, AND2(g_6, b_0)) and NE b_7 iff NE XOR2(a_6, AND2(g_7, b_0)) and NE b_8 iff NE XOR2(a_7, AND2(g_8, b_0)) and NE b_9 iff NE XOR2(a_8, AND2(g_9, b_0)) and NE b_{10} iff NE XOR2(a_9, AND2(g_{10}, b_0)) and NE b_{11} iff NE XOR2(a_{10}, AND2(g_{11}, b_0)). Then

(i) NE b_{11} iff NE XOR2(XOR2(a_{10}, AND2(g_{11}, a_{11})), XOR2(z, AND2(g_{11}, p))),
(ii) NE b_{10} iff NE XOR2(XOR2(a_9, AND2(g_{10}, a_{11})), XOR2(z, AND2(g_{10}, p))),
(iii) NE b_9 iff NE XOR2(XOR2(a_8, AND2(g_9, a_{11})), XOR2(z, AND2(g_9, p))),
(iv) NE b_8 iff NE XOR2(XOR2(a_7, AND2(g_8, a_{11})), XOR2(z, AND2(g_8, p))),
(v) NE b_7 iff NE XOR2(XOR2(a_6, AND2(g_7, a_{11})), XOR2(z, AND2(g_7, p))),
(vi) NE b_6 iff NE XOR2(XOR2(a_5, AND2(g_6, a_{11})), XOR2(z, AND2(g_6, p))),
(vii) NE b_5 iff NE XOR2(XOR2(a_4, AND2(g_5, a_{11})), XOR2(z, AND2(g_5, p))),
(viii) NE b_4 iff NE XOR2(XOR2(a_3, AND2(g_4, a_{11})), XOR2(z, AND2(g_4, p))),
(ix) NE b_3 iff NE XOR2(XOR2(a_2, AND2(g_3, a_{11})), XOR2(z, AND2(g_3, p))),
(x) NE b_2 iff NE XOR2(XOR2(a_1, AND2(g_2, a_{11})), XOR2(z, AND2(g_2, p))),
(xi) NE b_1 iff NE XOR2(XOR2(a_0, AND2(g_1, a_{11})), XOR2(z, AND2(g_1, p))), and
(xii) NE b_0 iff NE XOR2(XOR2(z, AND2(g_0, a_{11})), XOR2(z, AND2(g_0, p))).

(4) Let g_0, g_1, g_2, g_3, g_4, g_5, g_6, g_7, g_8, g_9, g_{10}, g_{11}, g_{12}, g_{13}, g_{14}, g_{15}, g_{16}, a_0, a_1, a_2, a_3, a_4, a_5, a_6, a_7, a_8, a_9, a_{10}, a_{11}, a_{12}, a_{13}, a_{14}, a_{15}, b_0, b_1, b_2, b_3, b_4, b_5, b_6, b_7, b_8, b_9, b_{10}, b_{11}, b_{12}, b_{13}, b_{14}, b_{15}, z, p be sets such that NE g_0 and NE g_{16} and not NE z and NE b_0 iff NE XOR2(p, a_{15}) and NE b_1 iff NE XOR2(a_0, AND2(g_1, b_0)) and NE b_2 iff NE XOR2(a_1, AND2(g_2, b_0)) and NE b_3 iff NE XOR2(a_2, AND2(g_3, b_0))

and NE b_4 iff NE XOR2(a_3, AND2(g_4, b_0)) and NE b_5 iff NE XOR2(a_4, AND2(g_5, b_0)) and NE b_6 iff NE XOR2(a_5, AND2(g_6, b_0)) and NE b_7 iff NE XOR2(a_6, AND2(g_7, b_0)) and NE b_8 iff NE XOR2(a_7, AND2(g_8, b_0)) and NE b_9 iff NE XOR2(a_8, AND2(g_9, b_0)) and NE b_{10} iff NE XOR2(a_9, AND2(g_{10}, b_0)) and NE b_{11} iff NE XOR2(a_{10}, AND2(g_{11}, b_0)) and NE b_{12} iff NE XOR2(a_{11}, AND2(g_{12}, b_0)) and NE b_{13} iff NE XOR2(a_{12}, AND2(g_{13}, b_0)) and NE b_{14} iff NE XOR2(a_{13}, AND2(g_{14}, b_0)) and NE b_{15} iff NE XOR2(a_{14}, AND2(g_{15}, b_0)). Then

(i) NE b_{15} iff NE XOR2(XOR2(a_{14}, AND2(g_{15}, a_{15})), XOR2(z, AND2(g_{15}, p))),
(ii) NE b_{14} iff NE XOR2(XOR2(a_{13}, AND2(g_{14}, a_{15})), XOR2(z, AND2(g_{14}, p))),
(iii) NE b_{13} iff NE XOR2(XOR2(a_{12}, AND2(g_{13}, a_{15})), XOR2(z, AND2(g_{13}, p))),
(iv) NE b_{12} iff NE XOR2(XOR2(a_{11}, AND2(g_{12}, a_{15})), XOR2(z, AND2(g_{12}, p))),
(v) NE b_{11} iff NE XOR2(XOR2(a_{10}, AND2(g_{11}, a_{15})), XOR2(z, AND2(g_{11}, p))),
(vi) NE b_{10} iff NE XOR2(XOR2(a_9, AND2(g_{10}, a_{15})), XOR2(z, AND2(g_{10}, p))),
(vii) NE b_9 iff NE XOR2(XOR2(a_8, AND2(g_9, a_{15})), XOR2(z, AND2(g_9, p))),
(viii) NE b_8 iff NE XOR2(XOR2(a_7, AND2(g_8, a_{15})), XOR2(z, AND2(g_8, p))),
(ix) NE b_7 iff NE XOR2(XOR2(a_6, AND2(g_7, a_{15})), XOR2(z, AND2(g_7, p))),
(x) NE b_6 iff NE XOR2(XOR2(a_5, AND2(g_6, a_{15})), XOR2(z, AND2(g_6, p))),
(xi) NE b_5 iff NE XOR2(XOR2(a_4, AND2(g_5, a_{15})), XOR2(z, AND2(g_5, p))),
(xii) NE b_4 iff NE XOR2(XOR2(a_3, AND2(g_4, a_{15})), XOR2(z, AND2(g_4, p))),
(xiii) NE b_3 iff NE XOR2(XOR2(a_2, AND2(g_3, a_{15})), XOR2(z, AND2(g_3, p))),
(xiv) NE b_2 iff NE XOR2(XOR2(a_1, AND2(g_2, a_{15})), XOR2(z, AND2(g_2, p))),
(xv) NE b_1 iff NE XOR2(XOR2(a_0, AND2(g_1, a_{15})), XOR2(z, AND2(g_1, p))),

and

(xvi) NE b_0 iff NE XOR2(XOR2(z, AND2(g_0, a_{15})), XOR2(z, AND2(g_0, p))).

References

[1] Yatsuka Nakamura. Logic gates and logical equivalence of adders. *Formalized Mathematics*, 8(**1**):35–45, 1999.

Received April 16, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Defining by Structural Induction in the Positive Propositional Language

Andrzej Trybulec
University of Białystok

Summary. The main goal of the paper consists in proving schemes for defining by structural induction in the language defined by Adam Grabowski [13]. The article consists of four parts. Besides the preliminaries where we prove some simple facts still missing in the library, they are:

- "About the language" in which the consequences of the fact that the algebra of formulae is free are formulated,

- "Defining by structural induction" in which two schemes are proved,

- "The tree of the subformulae" in which a scheme proved in the previous section is used to define the tree of subformulae; also some simple facts about the tree are proved.

MML Identifier: HILBERT2.

The terminology and notation used in this paper are introduced in the following papers: [16], [19], [1], [14], [20], [10], [12], [18], [8], [15], [9], [11], [3], [17], [2], [4], [5], [6], [7], and [13].

1. Preliminaries

In this paper X, x denote sets.

We now state four propositions:

(1) Let Z be a set and M be a many sorted set indexed by Z. Suppose that for every set x such that $x \in Z$ holds $M(x)$ is a many sorted set indexed by x. Let f be a function. If $f = \mathrm{Union}\, M$, then $\mathrm{dom}\, f = \bigcup Z$.

(2) For all sets x, y and for all finite sequences f, g such that $\langle x\rangle ^\frown f = \langle y\rangle ^\frown g$ holds $f = g$.

ISSN 1426–2630

(3) If $\langle x \rangle$ is a finite sequence of elements of X, then $x \in X$.

(4) Let given X and f be a finite sequence of elements of X. Suppose $f \neq \varepsilon$. Then there exists a finite sequence g of elements of X and there exists an element d of X such that $f = g \mathbin{^\frown} \langle d \rangle$.

We adopt the following rules: m, n are natural numbers, p, q, r, s are elements of HP-WFF, and T_1, T_2 are trees.

Next we state the proposition

(5) $\langle x \rangle \in \overbrace{T_1, T_2}$ iff $x = 0$ or $x = 1$.

Let us mention that ε is tree yielding.

The scheme *InTreeInd* deals with a tree $\mathcal{A}$ and and states that:

For every element f of $\mathcal{A}$ holds $\mathcal{P}[f]$

provided the following conditions are satisfied:

- $\mathcal{P}[\varepsilon_{\mathbb{N}}]$, and
- For every element f of $\mathcal{A}$ such that $\mathcal{P}[f]$ and for every n such that $f \mathbin{^\frown} \langle n \rangle \in \mathcal{A}$ holds $\mathcal{P}[f \mathbin{^\frown} \langle n \rangle]$.

In the sequel D is a non empty set and T_1, T_2 are decorated trees.

Next we state three propositions:

(6) For every set x and for all T_1, T_2 holds $(x\text{-tree}(T_1, T_2))(\varepsilon) = x$.

(7) $(x\text{-tree}(T_1, T_2))(\langle 0 \rangle) = T_1(\varepsilon)$ and $(x\text{-tree}(T_1, T_2))(\langle 1 \rangle) = T_2(\varepsilon)$.

(8) $(x\text{-tree}(T_1, T_2)) \restriction \langle 0 \rangle = T_1$ and $(x\text{-tree}(T_1, T_2)) \restriction \langle 1 \rangle = T_2$.

Let us consider x and let p be a decorated tree yielding non empty finite sequence. Observe that x-tree(p) is non root.

Let us consider x and let T_1 be a decorated tree. Observe that x-tree(T_1) is non root. Let T_2 be a decorated tree. Observe that x-tree(T_1, T_2) is non root.

2. About the Language

Let us consider n. The functor prop n yielding an element of HP-WFF is defined as follows:

(Def. 1) $\text{prop}\, n = \langle 3 + n \rangle$.

Let D be a set. Let us observe that D has VERUM if and only if:

(Def. 2) VERUM $\in D$.

Let us observe that D has propositional variables if and only if:

(Def. 3) For every n holds $\text{prop}\, n \in D$.

Let D be a subset of HP-WFF. Let us observe that D has implication if and only if:

(Def. 4) For all p, q such that $p \in D$ and $q \in D$ holds $p \Rightarrow q \in D$.

Let us observe that D has conjunction if and only if:

(Def. 5) For all p, q such that $p \in D$ and $q \in D$ holds $p \wedge q \in D$.

In the sequel t denotes a finite sequence.

Let us consider p. We say that p is conjunctive if and only if:

(Def. 6) There exist r, s such that $p = r \wedge s$.

We say that p is conditional if and only if:

(Def. 7) There exist r, s such that $p = r \Rightarrow s$.

We say that p is simple if and only if:

(Def. 8) There exists n such that $p = \mathrm{prop}\, n$.

The scheme *HP Ind* concerns and states that:

For every r holds $\mathcal{P}[r]$

provided the following requirements are met:

- $\mathcal{P}[\mathrm{VERUM}]$,
- For every n holds $\mathcal{P}[\mathrm{prop}\, n]$, and
- For all r, s such that $\mathcal{P}[r]$ and $\mathcal{P}[s]$ holds $\mathcal{P}[r \wedge s]$ and $\mathcal{P}[r \Rightarrow s]$.

Next we state a number of propositions:

(9) p is conjunctive, or conditional, or simple or $p = \mathrm{VERUM}$.

(10) $\mathrm{len}\, p \geqslant 1$.

(11) If $p(1) = 1$, then p is conditional.

(12) If $p(1) = 2$, then p is conjunctive.

(13) If $p(1) = 3 + n$, then p is simple.

(14) If $p(1) = 0$, then $p = \mathrm{VERUM}$.

(15) $\mathrm{len}\, p < \mathrm{len}(p \wedge q)$ and $\mathrm{len}\, q < \mathrm{len}(p \wedge q)$.

(16) $\mathrm{len}\, p < \mathrm{len}(p \Rightarrow q)$ and $\mathrm{len}\, q < \mathrm{len}(p \Rightarrow q)$.

(17) If $p = q \frown t$, then $p = q$.

(18) If $p \frown q = r \frown s$, then $p = r$ and $q = s$.

(19) If $p \wedge q = r \wedge s$, then $p = r$ and $s = q$.

(20) If $p \Rightarrow q = r \Rightarrow s$, then $p = r$ and $s = q$.

(21) If $\mathrm{prop}\, n = \mathrm{prop}\, m$, then $n = m$.

(22) $p \wedge q \neq r \Rightarrow s$.

(23) $p \wedge q \neq \mathrm{VERUM}$.

(24) $p \wedge q \neq \mathrm{prop}\, n$.

(25) $p \Rightarrow q \neq \mathrm{VERUM}$.

(26) $p \Rightarrow q \neq \mathrm{prop}\, n$.

(27) $p \wedge q \neq p$ and $p \wedge q \neq q$.

(28) $p \Rightarrow q \neq p$ and $p \Rightarrow q \neq q$.

(29) $\mathrm{VERUM} \neq \mathrm{prop}\, n$.

3. Defining by Structural Induction

Now we present two schemes. The scheme *HP MSSExL* deals with a set $\mathcal{A}$, a unary functor $\mathcal{F}$ yielding a set, and a 5-ary predicate $\mathcal{Q}$, and states that:

There exists a many sorted set M indexed by HP-WFF such that

(i) $M(\text{VERUM}) = \mathcal{A}$,

(ii) for every n holds $M(\text{prop}\, n) = \mathcal{F}(n)$, and

(iii) for all p, q and for all sets a, b, c, d such that $a = M(p)$ and $b = M(q)$ and $c = M(p \wedge q)$ and $d = M(p \Rightarrow q)$ holds $\mathcal{P}[p, q, a, b, c]$ and $\mathcal{Q}[p, q, a, b, d]$

provided the following conditions are met:

- For all p, q and for all sets a, b there exists a set c such that $\mathcal{P}[p, q, a, b, c]$,
- For all p, q and for all sets a, b there exists a set d such that $\mathcal{Q}[p, q, a, b, d]$,
- For all p, q and for all sets a, b, c, d such that $\mathcal{P}[p, q, a, b, c]$ and $\mathcal{P}[p, q, a, b, d]$ holds $c = d$, and
- For all p, q and for all sets a, b, c, d such that $\mathcal{Q}[p, q, a, b, c]$ and $\mathcal{Q}[p, q, a, b, d]$ holds $c = d$.

The scheme *HP MSSLambda* deals with a set $\mathcal{A}$, a unary functor $\mathcal{F}$ yielding a set, and two binary functors $\mathcal{G}$ and $\mathcal{H}$ yielding sets, and states that:

There exists a many sorted set M indexed by HP-WFF such that

(i) $M(\text{VERUM}) = \mathcal{A}$,

(ii) for every n holds $M(\text{prop}\, n) = \mathcal{F}(n)$, and

(iii) for all p, q and for all sets x, y such that $x = M(p)$ and $y = M(q)$ holds $M(p \wedge q) = \mathcal{G}(x, y)$ and $M(p \Rightarrow q) = \mathcal{H}(x, y)$

for all values of the parameters.

4. The Tree of the Subformulae

The many sorted set HP-Subformulae indexed by HP-WFF is defined by the conditions (Def. 9).

(Def. 9)(i) (HP-Subformulae)(VERUM) = the root tree of VERUM,

(ii) for every n holds (HP-Subformulae)(prop n) = the root tree of prop n, and

(iii) for all p, q there exist trees p', q' decorated with elements of HP-WFF such that p' = (HP-Subformulae)(p) and q' = (HP-Subformulae)(q) and (HP-Subformulae)($p \wedge q$) = $p \wedge q$-tree(p', q') and (HP-Subformulae)($p \Rightarrow q$) = ($p \Rightarrow q$)-tree(p', q').

Let us consider p. The functor Subformulae p yielding a tree decorated with elements of HP-WFF is defined by:

(Def. 10) Subformulae p = (HP-Subformulae)(p).

The following propositions are true:

(30) Subformulae VERUM = the root tree of VERUM.

(31) Subformulae prop n = the root tree of prop n.

(32) Subformulae$(p \wedge q) = p \wedge q$-tree(Subformulae p, Subformulae q).

(33) Subformulae$(p \Rightarrow q) = (p \Rightarrow q)$-tree(Subformulae p, Subformulae q).

(34) (Subformulae p)$(\varepsilon) = p$.

(35) For every element f of dom Subformulae p holds Subformulae $p \restriction f$ = Subformulae(Subformulae p)(f).

(36) If $p \in$ Leaves(Subformulae q), then p = VERUM or p is simple.

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[2] Grzegorz Bancerek. Introduction to trees. *Formalized Mathematics*, 1(**2**):421–427, 1990.
[3] Grzegorz Bancerek. Cartesian product of functions. *Formalized Mathematics*, 2(**4**):547–552, 1991.
[4] Grzegorz Bancerek. König's lemma. *Formalized Mathematics*, 2(**3**):397–402, 1991.
[5] Grzegorz Bancerek. Sets and functions of trees and joining operations of trees. *Formalized Mathematics*, 3(**2**):195–204, 1992.
[6] Grzegorz Bancerek. Joining of decorated trees. *Formalized Mathematics*, 4(**1**):77–82, 1993.
[7] Grzegorz Bancerek. Subtrees. *Formalized Mathematics*, 5(**2**):185–190, 1996.
[8] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[9] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.
[10] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[11] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[12] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.
[13] Adam Grabowski. Hilbert positive propositional calculus. *Formalized Mathematics*, 8(**1**):69–72, 1999.
[14] Andrzej Nędzusiak. σ-fields and probability. *Formalized Mathematics*, 1(**2**):401–407, 1990.
[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. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[18] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[19] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[20] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received April 23, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Some Properties of Cells on Go-Board

Czesław Byliński
University of Białystok

MML Identifier: GOBRD13.

The terminology and notation used in this paper have been introduced in the following articles: [23], [9], [13], [3], [20], [22], [25], [26], [7], [8], [2], [1], [5], [6], [24], [10], [19], [4], [15], [14], [21], [11], [12], [16], [17], and [18].

We use the following convention: i, i_1, i_2, j, j_1, j_2, k, n are natural numbers, D is a non empty set, and f is a finite sequence of elements of D.

Let E be a non empty set, let S be a non empty set of finite sequences of the carrier of $\mathcal{E}^2_{\rm T}$, let F be a function from E into S, and let e be an element of E. Then $F(e)$ is a finite sequence of elements of $\mathcal{E}^2_{\rm T}$.

Let F be a function. The functor $\mathrm{Values}\, F$ yielding a set is defined by:

(Def. 1) $\mathrm{Values}\, F = \mathrm{Union}(\mathrm{rng}_\kappa\, F(\kappa))$.

We now state three propositions:

(1) Let M be a finite sequence of elements of D^*. If $i \in \mathrm{dom}\, M$, then $M(i)$ is a finite sequence of elements of D.

(2) For every finite sequence M of elements of D^* holds $\mathrm{dom}(\mathrm{rng}_\kappa\, M(\kappa)) = \mathrm{dom}\, M$.

(3) For every finite sequence M of elements of D^* holds $\mathrm{Values}\, M = \bigcup\{\mathrm{rng}\, f;\ f$ ranges over elements of D^*: $f \in \mathrm{rng}\, M\}$.

Let D be a non empty set and let M be a finite sequence of elements of D^*. Note that $\mathrm{Values}\, M$ is finite.

The following propositions are true:

(4) For every matrix M over D such that $i \in \mathrm{dom}\, M$ and $M(i) = f$ holds $\mathrm{len}\, f = \mathrm{width}\, M$.

(5) For every matrix M over D such that $i \in \mathrm{dom}\, M$ and $M(i) = f$ and $j \in \mathrm{dom}\, f$ holds $\langle i, j\rangle \in$ the indices of M.

(6) For every matrix M over D such that $\langle i, j\rangle \in$ the indices of M and $M(i) = f$ holds $\mathrm{len}\, f = \mathrm{width}\, M$ and $j \in \mathrm{dom}\, f$.

ISSN 1426–2630

(7) For every matrix M over D holds $\mathrm{Values}\, M = \{M_{i,j} : \langle i, j\rangle \in$ the indices of $M\}$.

(8) For every non empty set D and for every matrix M over D holds $\mathrm{card}\, \mathrm{Values}\, M \leqslant \mathrm{len}\, M \cdot \mathrm{width}\, M$.

In the sequel f, f_1, f_2 are finite sequences of elements of $\mathcal{E}^2_{\mathrm{T}}$ and G is a Go-board.

Next we state a number of propositions:

(9) If f is a sequence which elements belong to G, then $\mathrm{rng}\, f \subseteq \mathrm{Values}\, G$.

(10) For all Go-boards G_1, G_2 such that $\mathrm{Values}\, G_1 \subseteq \mathrm{Values}\, G_2$ and $\langle i_1, j_1\rangle \in$ the indices of G_1 and $1 \leqslant j_2$ and $j_2 \leqslant \mathrm{width}\, G_2$ and $(G_1)_{i_1,j_1} = (G_2)_{1,j_2}$ holds $i_1 = 1$.

(11) For all Go-boards G_1, G_2 such that $\mathrm{Values}\, G_1 \subseteq \mathrm{Values}\, G_2$ and $\langle i_1, j_1\rangle \in$ the indices of G_1 and $1 \leqslant j_2$ and $j_2 \leqslant \mathrm{width}\, G_2$ and $(G_1)_{i_1,j_1} = (G_2)_{\mathrm{len}\, G_2,j_2}$ holds $i_1 = \mathrm{len}\, G_1$.

(12) For all Go-boards G_1, G_2 such that $\mathrm{Values}\, G_1 \subseteq \mathrm{Values}\, G_2$ and $\langle i_1, j_1\rangle \in$ the indices of G_1 and $1 \leqslant i_2$ and $i_2 \leqslant \mathrm{len}\, G_2$ and $(G_1)_{i_1,j_1} = (G_2)_{i_2,1}$ holds $j_1 = 1$.

(13) For all Go-boards G_1, G_2 such that $\mathrm{Values}\, G_1 \subseteq \mathrm{Values}\, G_2$ and $\langle i_1, j_1\rangle \in$ the indices of G_1 and $1 \leqslant i_2$ and $i_2 \leqslant \mathrm{len}\, G_2$ and $(G_1)_{i_1,j_1} = (G_2)_{i_2,\mathrm{width}\, G_2}$ holds $j_1 = \mathrm{width}\, G_1$.

(14) Let G_1, G_2 be Go-boards. Suppose $\mathrm{Values}\, G_1 \subseteq \mathrm{Values}\, G_2$ and $1 \leqslant i_1$ and $i_1 < \mathrm{len}\, G_1$ and $1 \leqslant j_1$ and $j_1 \leqslant \mathrm{width}\, G_1$ and $1 \leqslant i_2$ and $i_2 < \mathrm{len}\, G_2$ and $1 \leqslant j_2$ and $j_2 \leqslant \mathrm{width}\, G_2$ and $(G_1)_{i_1,j_1} = (G_2)_{i_2,j_2}$. Then $((G_2)_{i_2+1,j_2})_{\mathbf{1}} \leqslant ((G_1)_{i_1+1,j_1})_{\mathbf{1}}$.

(15) Let G_1, G_2 be Go-boards. Suppose $\mathrm{Values}\, G_1 \subseteq \mathrm{Values}\, G_2$ and $1 < i_1$ and $i_1 \leqslant \mathrm{len}\, G_1$ and $1 \leqslant j_1$ and $j_1 \leqslant \mathrm{width}\, G_1$ and $1 < i_2$ and $i_2 \leqslant \mathrm{len}\, G_2$ and $1 \leqslant j_2$ and $j_2 \leqslant \mathrm{width}\, G_2$ and $(G_1)_{i_1,j_1} = (G_2)_{i_2,j_2}$. Then $((G_1)_{i_1-'1,j_1})_{\mathbf{1}} \leqslant ((G_2)_{i_2-'1,j_2})_{\mathbf{1}}$.

(16) Let G_1, G_2 be Go-boards. Suppose $\mathrm{Values}\, G_1 \subseteq \mathrm{Values}\, G_2$ and $1 \leqslant i_1$ and $i_1 \leqslant \mathrm{len}\, G_1$ and $1 \leqslant j_1$ and $j_1 < \mathrm{width}\, G_1$ and $1 \leqslant i_2$ and $i_2 \leqslant \mathrm{len}\, G_2$ and $1 \leqslant j_2$ and $j_2 < \mathrm{width}\, G_2$ and $(G_1)_{i_1,j_1} = (G_2)_{i_2,j_2}$. Then $((G_2)_{i_2,j_2+1})_{\mathbf{2}} \leqslant ((G_1)_{i_1,j_1+1})_{\mathbf{2}}$.

(17) Let G_1, G_2 be Go-boards. Suppose $\mathrm{Values}\, G_1 \subseteq \mathrm{Values}\, G_2$ and $1 \leqslant i_1$ and $i_1 \leqslant \mathrm{len}\, G_1$ and $1 < j_1$ and $j_1 \leqslant \mathrm{width}\, G_1$ and $1 \leqslant i_2$ and $i_2 \leqslant \mathrm{len}\, G_2$ and $1 < j_2$ and $j_2 \leqslant \mathrm{width}\, G_2$ and $(G_1)_{i_1,j_1} = (G_2)_{i_2,j_2}$. Then $((G_1)_{i_1,j_1-'1})_{\mathbf{2}} \leqslant ((G_2)_{i_2,j_2-'1})_{\mathbf{2}}$.

(18) Let G_1, G_2 be Go-boards. Suppose $\mathrm{Values}\, G_1 \subseteq \mathrm{Values}\, G_2$ and $\langle i_1, j_1\rangle \in$ the indices of G_1 and $\langle i_2, j_2\rangle \in$ the indices of G_2 and $(G_1)_{i_1,j_1} = (G_2)_{i_2,j_2}$. Then $\mathrm{cell}(G_2, i_2, j_2) \subseteq \mathrm{cell}(G_1, i_1, j_1)$.

(19) Let G_1, G_2 be Go-boards. Suppose $\mathrm{Values}\, G_1 \subseteq \mathrm{Values}\, G_2$ and $\langle i_1, j_1\rangle \in$

the indices of G_1 and $\langle i_2, j_2\rangle \in$ the indices of G_2 and $(G_1)_{i_1,j_1} = (G_2)_{i_2,j_2}$. Then $\mathrm{cell}(G_2, i_2 -' 1, j_2) \subseteq \mathrm{cell}(G_1, i_1 -' 1, j_1)$.

(20) Let G_1, G_2 be Go-boards. Suppose $\mathrm{Values}\, G_1 \subseteq \mathrm{Values}\, G_2$ and $\langle i_1, j_1\rangle \in$ the indices of G_1 and $\langle i_2, j_2\rangle \in$ the indices of G_2 and $(G_1)_{i_1,j_1} = (G_2)_{i_2,j_2}$. Then $\mathrm{cell}(G_2, i_2, j_2 -' 1) \subseteq \mathrm{cell}(G_1, i_1, j_1 -' 1)$.

(21) Let f be a standard special circular sequence. Suppose f is a sequence which elements belong to G. Then Values the Go-board of $f \subseteq \mathrm{Values}\, G$.

Let us consider f, G, k. Let us assume that $1 \leqslant k$ and $k + 1 \leqslant \mathrm{len}\, f$ and f is a sequence which elements belong to G. The functor right_cell(f, k, G) yields a subset of $\mathcal{E}^2_{\mathrm{T}}$ and is defined by the condition (Def. 2).

(Def. 2) Let i_1, j_1, i_2, j_2 be natural numbers. Suppose $\langle i_1, j_1\rangle \in$ the indices of G and $\langle i_2, j_2\rangle \in$ the indices of G and $\pi_k f = G_{i_1,j_1}$ and $\pi_{k+1} f = G_{i_2,j_2}$. Then
- (i) $i_1 = i_2$ and $j_1 + 1 = j_2$ and right_cell$(f, k, G) = \mathrm{cell}(G, i_1, j_1)$, or
- (ii) $i_1 + 1 = i_2$ and $j_1 = j_2$ and right_cell$(f, k, G) = \mathrm{cell}(G, i_1, j_1 -' 1)$, or
- (iii) $i_1 = i_2 + 1$ and $j_1 = j_2$ and right_cell$(f, k, G) = \mathrm{cell}(G, i_2, j_2)$, or
- (iv) $i_1 = i_2$ and $j_1 = j_2 + 1$ and right_cell$(f, k, G) = \mathrm{cell}(G, i_1 -' 1, j_2)$.

The functor left_cell(f, k, G) yields a subset of $\mathcal{E}^2_{\mathrm{T}}$ and is defined by the condition (Def. 3).

(Def. 3) Let i_1, j_1, i_2, j_2 be natural numbers. Suppose $\langle i_1, j_1\rangle \in$ the indices of G and $\langle i_2, j_2\rangle \in$ the indices of G and $\pi_k f = G_{i_1,j_1}$ and $\pi_{k+1} f = G_{i_2,j_2}$. Then
- (i) $i_1 = i_2$ and $j_1 + 1 = j_2$ and left_cell$(f, k, G) = \mathrm{cell}(G, i_1 -' 1, j_1)$, or
- (ii) $i_1 + 1 = i_2$ and $j_1 = j_2$ and left_cell$(f, k, G) = \mathrm{cell}(G, i_1, j_1)$, or
- (iii) $i_1 = i_2 + 1$ and $j_1 = j_2$ and left_cell$(f, k, G) = \mathrm{cell}(G, i_2, j_2 -' 1)$, or
- (iv) $i_1 = i_2$ and $j_1 = j_2 + 1$ and left_cell$(f, k, G) = \mathrm{cell}(G, i_1, j_2)$.

We now state a number of propositions:

(22) Suppose that
$1 \leqslant k$ and $k + 1 \leqslant \mathrm{len}\, f$ and f is a sequence which elements belong to G and $\langle i, j\rangle \in$ the indices of G and $\langle i, j + 1\rangle \in$ the indices of G and $\pi_k f = G_{i,j}$ and $\pi_{k+1} f = G_{i,j+1}$. Then left_cell$(f, k, G) = \mathrm{cell}(G, i -' 1, j)$.

(23) Suppose that
$1 \leqslant k$ and $k + 1 \leqslant \mathrm{len}\, f$ and f is a sequence which elements belong to G and $\langle i, j\rangle \in$ the indices of G and $\langle i, j + 1\rangle \in$ the indices of G and $\pi_k f = G_{i,j}$ and $\pi_{k+1} f = G_{i,j+1}$. Then right_cell$(f, k, G) = \mathrm{cell}(G, i, j)$.

(24) Suppose that
$1 \leqslant k$ and $k + 1 \leqslant \mathrm{len}\, f$ and f is a sequence which elements belong to G and $\langle i, j\rangle \in$ the indices of G and $\langle i + 1, j\rangle \in$ the indices of G and $\pi_k f = G_{i,j}$ and $\pi_{k+1} f = G_{i+1,j}$. Then left_cell$(f, k, G) = \mathrm{cell}(G, i, j)$.

(25) Suppose that
$1 \leqslant k$ and $k+1 \leqslant \mathrm{len}\, f$ and f is a sequence which elements belong to G and $\langle i, j\rangle \in$ the indices of G and $\langle i + 1, j\rangle \in$ the indices of G and $\pi_k f = G_{i,j}$

and $\pi_{k+1} f = G_{i+1,j}$. Then right_cell$(f, k, G) = $ cell$(G, i, j -' 1)$.

(26) Suppose that
$1 \leqslant k$ and $k + 1 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G and $\langle i, j\rangle \in$ the indices of G and $\langle i + 1, j\rangle \in$ the indices of G and $\pi_k f = G_{i+1,j}$ and $\pi_{k+1} f = G_{i,j}$. Then left_cell$(f, k, G) = $ cell$(G, i, j -' 1)$.

(27) Suppose that
$1 \leqslant k$ and $k + 1 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G and $\langle i, j\rangle \in$ the indices of G and $\langle i + 1, j\rangle \in$ the indices of G and $\pi_k f = G_{i+1,j}$ and $\pi_{k+1} f = G_{i,j}$. Then right_cell$(f, k, G) = $ cell(G, i, j).

(28) Suppose that
$1 \leqslant k$ and $k + 1 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G and $\langle i, j + 1\rangle \in$ the indices of G and $\langle i, j\rangle \in$ the indices of G and $\pi_k f = G_{i,j+1}$ and $\pi_{k+1} f = G_{i,j}$. Then left_cell$(f, k, G) = $ cell(G, i, j).

(29) Suppose that
$1 \leqslant k$ and $k+1 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G and $\langle i, j+1\rangle \in$ the indices of G and $\langle i, j\rangle \in$ the indices of G and $\pi_k f = G_{i,j+1}$ and $\pi_{k+1} f = G_{i,j}$. Then right_cell$(f, k, G) = $ cell$(G, i -' 1, j)$.

(30) If $1 \leqslant k$ and $k + 1 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G, then left_cell$(f, k, G) \cap$ right_cell$(f, k, G) = \mathcal{L}(f, k)$.

(31) If $1 \leqslant k$ and $k + 1 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G, then right_cell(f, k, G) is closed.

(32) Suppose $1 \leqslant k$ and $k + 1 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G and $k + 1 \leqslant n$. Then left_cell$(f, k, G) = $ left_cell$(f{\restriction}n, k, G)$ and right_cell$(f, k, G) = $ right_cell$(f{\restriction}n, k, G)$.

(33) Suppose $1 \leqslant k$ and $k + 1 \leqslant \operatorname{len}(f_{\lfloor n})$ and $n \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G. Then left_cell$(f, k+n, G) = $ left_cell$(f_{\lfloor n}, k, G)$ and right_cell$(f, k + n, G) = $ right_cell$(f_{\lfloor n}, k, G)$.

(34) Let G be a Go-board and f be a standard special circular sequence. Suppose $1 \leqslant n$ and $n + 1 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G. Then left_cell$(f, n, G) \subseteq$ leftcell(f, n) and right_cell$(f, n, G) \subseteq$ rightcell(f, n).

Let us consider f, G, k. Let us assume that $1 \leqslant k$ and $k + 1 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G. The functor front_right_cell(f, k, G) yielding a subset of $\mathcal{E}^2_{\mathrm{T}}$ is defined by the condition (Def. 4).

(Def. 4) Let i_1, j_1, i_2, j_2 be natural numbers. Suppose $\langle i_1, j_1\rangle \in$ the indices of G and $\langle i_2, j_2\rangle \in$ the indices of G and $\pi_k f = G_{i_1,j_1}$ and $\pi_{k+1} f = G_{i_2,j_2}$. Then
(i) $i_1 = i_2$ and $j_1 + 1 = j_2$ and front_right_cell$(f, k, G) = $ cell(G, i_2, j_2), or
(ii) $i_1 + 1 = i_2$ and $j_1 = j_2$ and front_right_cell$(f, k, G) = $ cell$(G, i_2, j_2 -' 1)$, or

(iii) $i_1 = i_2+1$ and $j_1 = j_2$ and $\text{front\_right\_cell}(f,k,G) = \text{cell}(G, i_2 -' 1, j_2)$, or

(iv) $i_1 = i_2$ and $j_1 = j_2+1$ and $\text{front\_right\_cell}(f,k,G) = \text{cell}(G, i_2 -' 1, j_2 -' 1)$.

The functor $\text{front\_left\_cell}(f,k,G)$ yields a subset of $\mathcal{E}^2_{\rm T}$ and is defined by the condition (Def. 5).

(Def. 5) Let i_1, j_1, i_2, j_2 be natural numbers. Suppose $\langle i_1, j_1\rangle \in$ the indices of G and $\langle i_2, j_2\rangle \in$ the indices of G and $\pi_k f = G_{i_1,j_1}$ and $\pi_{k+1} f = G_{i_2,j_2}$. Then

(i) $i_1 = i_2$ and $j_1 + 1 = j_2$ and $\text{front\_left\_cell}(f,k,G) = \text{cell}(G, i_2 -' 1, j_2)$, or

(ii) $i_1 + 1 = i_2$ and $j_1 = j_2$ and $\text{front\_left\_cell}(f,k,G) = \text{cell}(G, i_2, j_2)$, or

(iii) $i_1 = i_2+1$ and $j_1 = j_2$ and $\text{front\_left\_cell}(f,k,G) = \text{cell}(G, i_2 -' 1, j_2 -' 1)$, or

(iv) $i_1 = i_2$ and $j_1 = j_2 + 1$ and $\text{front\_left\_cell}(f,k,G) = \text{cell}(G, i_2, j_2 -' 1)$.

Next we state several propositions:

(35) Suppose that
$1 \leqslant k$ and $k+1 \leqslant \text{len}\, f$ and f is a sequence which elements belong to G and $\langle i, j\rangle \in$ the indices of G and $\langle i, j+1\rangle \in$ the indices of G and $\pi_k f = G_{i,j}$ and $\pi_{k+1} f = G_{i,j+1}$. Then $\text{front\_left\_cell}(f,k,G) = \text{cell}(G, i -' 1, j+1)$.

(36) Suppose that
$1 \leqslant k$ and $k+1 \leqslant \text{len}\, f$ and f is a sequence which elements belong to G and $\langle i, j\rangle \in$ the indices of G and $\langle i, j+1\rangle \in$ the indices of G and $\pi_k f = G_{i,j}$ and $\pi_{k+1} f = G_{i,j+1}$. Then $\text{front\_right\_cell}(f,k,G) = \text{cell}(G, i, j+1)$.

(37) Suppose that
$1 \leqslant k$ and $k+1 \leqslant \text{len}\, f$ and f is a sequence which elements belong to G and $\langle i, j\rangle \in$ the indices of G and $\langle i+1, j\rangle \in$ the indices of G and $\pi_k f = G_{i,j}$ and $\pi_{k+1} f = G_{i+1,j}$. Then $\text{front\_left\_cell}(f,k,G) = \text{cell}(G, i+1, j)$.

(38) Suppose that
$1 \leqslant k$ and $k+1 \leqslant \text{len}\, f$ and f is a sequence which elements belong to G and $\langle i, j\rangle \in$ the indices of G and $\langle i+1, j\rangle \in$ the indices of G and $\pi_k f = G_{i,j}$ and $\pi_{k+1} f = G_{i+1,j}$. Then $\text{front\_right\_cell}(f,k,G) = \text{cell}(G, i+1, j -' 1)$.

(39) Suppose that
$1 \leqslant k$ and $k+1 \leqslant \text{len}\, f$ and f is a sequence which elements belong to G and $\langle i, j\rangle \in$ the indices of G and $\langle i+1, j\rangle \in$ the indices of G and $\pi_k f = G_{i+1,j}$ and $\pi_{k+1} f = G_{i,j}$. Then $\text{front\_left\_cell}(f,k,G) = \text{cell}(G, i -' 1, j -' 1)$.

(40) Suppose that
$1 \leqslant k$ and $k+1 \leqslant \text{len}\, f$ and f is a sequence which elements belong to G and $\langle i, j\rangle \in$ the indices of G and $\langle i+1, j\rangle \in$ the indices of G and $\pi_k f = G_{i+1,j}$ and $\pi_{k+1} f = G_{i,j}$. Then $\text{front\_right\_cell}(f,k,G) = \text{cell}(G, i -' 1, j)$.

(41) Suppose that

$1 \leqslant k$ and $k+1 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G and $\langle i, j+1\rangle \in$ the indices of G and $\langle i, j\rangle \in$ the indices of G and $\pi_k f = G_{i,j+1}$ and $\pi_{k+1} f = G_{i,j}$. Then front_left_cell$(f, k, G) = $ cell$(G, i, j -' 1)$.

(42) Suppose that
$1 \leqslant k$ and $k+1 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G and $\langle i, j+1\rangle \in$ the indices of G and $\langle i, j\rangle \in$ the indices of G and $\pi_k f = G_{i,j+1}$ and $\pi_{k+1} f = G_{i,j}$. Then front_right_cell$(f, k, G) = $ cell$(G, i -' 1, j -' 1)$.

(43) Suppose $1 \leqslant k$ and $k + 1 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G and $k + 1 \leqslant n$. Then front_left_cell$(f, k, G) =$ front_left_cell$(f{\restriction}n, k, G)$ and front_right_cell$(f, k, G) =$ front_right_cell$(f{\restriction}n, k, G)$.

Let us consider f, G, k. We say that f turns right k, G if and only if the condition (Def. 6) is satisfied.

(Def. 6) Let i_1, j_1, i_2, j_2 be natural numbers. Suppose $\langle i_1, j_1\rangle \in$ the indices of G and $\langle i_2, j_2\rangle \in$ the indices of G and $\pi_k f = G_{i_1,j_1}$ and $\pi_{k+1} f = G_{i_2,j_2}$. Then

(i) $i_1 = i_2$ and $j_1 + 1 = j_2$ and $\langle i_2 + 1, j_2\rangle \in$ the indices of G and $\pi_{k+2} f = G_{i_2+1,j_2}$, or

(ii) $i_1 + 1 = i_2$ and $j_1 = j_2$ and $\langle i_2, j_2 -' 1\rangle \in$ the indices of G and $\pi_{k+2} f = G_{i_2,j_2-'1}$, or

(iii) $i_1 = i_2 + 1$ and $j_1 = j_2$ and $\langle i_2, j_2 + 1\rangle \in$ the indices of G and $\pi_{k+2} f = G_{i_2,j_2+1}$, or

(iv) $i_1 = i_2$ and $j_1 = j_2 + 1$ and $\langle i_2 -' 1, j_2\rangle \in$ the indices of G and $\pi_{k+2} f = G_{i_2-'1,j_2}$.

We say that f turns left k, G if and only if the condition (Def. 7) is satisfied.

(Def. 7) Let i_1, j_1, i_2, j_2 be natural numbers. Suppose $\langle i_1, j_1\rangle \in$ the indices of G and $\langle i_2, j_2\rangle \in$ the indices of G and $\pi_k f = G_{i_1,j_1}$ and $\pi_{k+1} f = G_{i_2,j_2}$. Then

(i) $i_1 = i_2$ and $j_1 + 1 = j_2$ and $\langle i_2 -' 1, j_2\rangle \in$ the indices of G and $\pi_{k+2} f = G_{i_2-'1,j_2}$, or

(ii) $i_1 + 1 = i_2$ and $j_1 = j_2$ and $\langle i_2, j_2 + 1\rangle \in$ the indices of G and $\pi_{k+2} f = G_{i_2,j_2+1}$, or

(iii) $i_1 = i_2 + 1$ and $j_1 = j_2$ and $\langle i_2, j_2 -' 1\rangle \in$ the indices of G and $\pi_{k+2} f = G_{i_2,j_2-'1}$, or

(iv) $i_1 = i_2$ and $j_1 = j_2 + 1$ and $\langle i_2 + 1, j_2\rangle \in$ the indices of G and $\pi_{k+2} f = G_{i_2+1,j_2}$.

We say that f goes straight k, G if and only if the condition (Def. 8) is satisfied.

(Def. 8) Let i_1, j_1, i_2, j_2 be natural numbers. Suppose $\langle i_1, j_1\rangle \in$ the indices of G and $\langle i_2, j_2\rangle \in$ the indices of G and $\pi_k f = G_{i_1,j_1}$ and $\pi_{k+1} f = G_{i_2,j_2}$. Then

(i) $i_1 = i_2$ and $j_1 + 1 = j_2$ and $\langle i_2, j_2 + 1\rangle \in$ the indices of G and $\pi_{k+2} f = G_{i_2,j_2+1}$, or

(ii) $i_1 + 1 = i_2$ and $j_1 = j_2$ and $\langle i_2 + 1, j_2\rangle \in$ the indices of G and $\pi_{k+2} f = G_{i_2+1,j_2}$, or

(iii) $i_1 = i_2 + 1$ and $j_1 = j_2$ and $\langle i_2 -' 1, j_2\rangle \in$ the indices of G and $\pi_{k+2} f = G_{i_2 -' 1, j_2}$, or

(iv) $i_1 = i_2$ and $j_1 = j_2 + 1$ and $\langle i_2, j_2 -' 1\rangle \in$ the indices of G and $\pi_{k+2} f = G_{i_2, j_2 -' 1}$.

One can prove the following propositions:

(44) Suppose $1 \leqslant k$ and $k + 2 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G and $k + 2 \leqslant n$ and $f \restriction n$ turns right k, G. Then f turns right k, G.

(45) Suppose $1 \leqslant k$ and $k + 2 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G and $k + 2 \leqslant n$ and $f \restriction n$ turns left k, G. Then f turns left k, G.

(46) Suppose $1 \leqslant k$ and $k + 2 \leqslant \operatorname{len} f$ and f is a sequence which elements belong to G and $k+2 \leqslant n$ and $f \restriction n$ goes straight k, G. Then f goes straight k, G.

(47) Suppose that
$1 < k$ and $k + 1 \leqslant \operatorname{len} f_1$ and $k + 1 \leqslant \operatorname{len} f_2$ and f_1 is a sequence which elements belong to G and f_2 is a sequence which elements belong to G and $f_1 \restriction k = f_2 \restriction k$ and f_1 turns right $k -' 1$, G and f_2 turns right $k -' 1$, G. Then $f_1 \restriction (k + 1) = f_2 \restriction (k + 1)$.

(48) Suppose that
$1 < k$ and $k + 1 \leqslant \operatorname{len} f_1$ and $k + 1 \leqslant \operatorname{len} f_2$ and f_1 is a sequence which elements belong to G and f_2 is a sequence which elements belong to G and $f_1 \restriction k = f_2 \restriction k$ and f_1 turns left $k -' 1$, G and f_2 turns left $k -' 1$, G. Then $f_1 \restriction (k + 1) = f_2 \restriction (k + 1)$.

(49) Suppose that
$1 < k$ and $k + 1 \leqslant \operatorname{len} f_1$ and $k + 1 \leqslant \operatorname{len} f_2$ and f_1 is a sequence which elements belong to G and f_2 is a sequence which elements belong to G and $f_1 \restriction k = f_2 \restriction k$ and f_1 goes straight $k -' 1$, G and f_2 goes straight $k -' 1$, G. Then $f_1 \restriction (k + 1) = f_2 \restriction (k + 1)$.

References

[1] Grzegorz Bancerek. Cardinal arithmetics. *Formalized Mathematics*, 1(**3**):543–547, 1990.

[2] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.

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

[4] Grzegorz Bancerek. Cartesian product of functions. *Formalized Mathematics*, 2(**4**):547–552, 1991.

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

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

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

[8] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[9] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[10] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[11] Agata Darmochwał. The Euclidean space. *Formalized Mathematics*, 2(**4**):599–603, 1991.
[12] Agata Darmochwał and Yatsuka Nakamura. The topological space $\mathcal{E}^2_{\rm T}$. Arcs, line segments and special polygonal arcs. *Formalized Mathematics*, 2(**5**):617–621, 1991.
[13] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[14] Katarzyna Jankowska. Matrices. Abelian group of matrices. *Formalized Mathematics*, 2(**4**):475–480, 1991.
[15] Jarosław Kotowicz. Functions and finite sequences of real numbers. *Formalized Mathematics*, 3(**2**):275–278, 1992.
[16] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part I. *Formalized Mathematics*, 3(**1**):107–115, 1992.
[17] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part II. *Formalized Mathematics*, 3(**1**):117–121, 1992.
[18] Yatsuka Nakamura and Andrzej Trybulec. Decomposing a Go-board into cells. *Formalized Mathematics*, 5(**3**):323–328, 1996.
[19] Andrzej Nędzusiak. σ-fields and probability. *Formalized Mathematics*, 1(**2**):401–407, 1990.
[20] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[21] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[22] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.
[23] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[24] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[25] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[26] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received April 23, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Propositional Calculus for Boolean Valued Functions. Part III

Shunichi Kobayashi
Shinshu University
Nagano

Summary. In this paper, we have proved some elementary propositional calculus formulae for Boolean valued functions.

MML Identifier: BVFUNC_7.

The articles [6], [8], [9], [2], [3], [5], [1], [7], and [4] provide the terminology and notation for this paper.

In this paper Y is a non empty set.

Next we state a number of propositions:

(1) For all elements a, b of $\mathrm{BVF}(Y)$ holds $(a \Rightarrow b) \wedge (\neg a \Rightarrow b) = b$.

(2) For all elements a, b of $\mathrm{BVF}(Y)$ holds $(a \Rightarrow b) \wedge (a \Rightarrow \neg b) = \neg a$.

(3) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \vee c = (a \Rightarrow b) \vee (a \Rightarrow c)$.

(4) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \wedge c = (a \Rightarrow b) \wedge (a \Rightarrow c)$.

(5) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \vee b \Rightarrow c = (a \Rightarrow c) \wedge (b \Rightarrow c)$.

(6) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \wedge b \Rightarrow c = (a \Rightarrow c) \vee (b \Rightarrow c)$.

(7) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \wedge b \Rightarrow c = a \Rightarrow b \Rightarrow c$.

(8) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \wedge b \Rightarrow c = a \Rightarrow \neg b \vee c$.

(9) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \vee c = a \wedge \neg b \Rightarrow c$.

(10) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \wedge (a \Rightarrow b) = a \wedge b$.

(11) For all elements a, b of $\mathrm{BVF}(Y)$ holds $(a \Rightarrow b) \wedge \neg b = \neg a \wedge \neg b$.

(12) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $(a \Rightarrow b) \wedge (b \Rightarrow c) = (a \Rightarrow b) \wedge (b \Rightarrow c) \wedge (a \Rightarrow c)$.

(13) For every element a of $\mathrm{BVF}(Y)$ holds $\mathit{true}(Y) \Rightarrow a = a$.

(14) For every element a of $\mathrm{BVF}(Y)$ holds $a \Rightarrow \mathit{false}(Y) = \neg a$.

ISSN 1426–2630

(15) For every element a of BVF(Y) holds $\mathit{false}(Y) \Rightarrow a = \mathit{true}(Y)$.

(16) For every element a of BVF(Y) holds $a \Rightarrow \mathit{true}(Y) = \mathit{true}(Y)$.

(17) For every element a of BVF(Y) holds $a \Rightarrow \neg a = \neg a$.

(18) For all elements a, b, c of BVF(Y) holds $a \Rightarrow b \Subset c \Rightarrow a \Rightarrow c \Rightarrow b$.

(19) For all elements a, b, c of BVF(Y) holds $a \Leftrightarrow b \Subset a \Leftrightarrow c \Leftrightarrow b \Leftrightarrow c$.

(20) For all elements a, b, c of BVF(Y) holds $a \Leftrightarrow b \Subset a \Rightarrow c \Leftrightarrow b \Rightarrow c$.

(21) For all elements a, b, c of BVF(Y) holds $a \Leftrightarrow b \Subset c \Rightarrow a \Leftrightarrow c \Rightarrow b$.

(22) For all elements a, b, c of BVF(Y) holds $a \Leftrightarrow b \Subset a \wedge c \Leftrightarrow b \wedge c$.

(23) For all elements a, b, c of BVF(Y) holds $a \Leftrightarrow b \Subset a \vee c \Leftrightarrow b \vee c$.

(24) For all elements a, b of BVF(Y) holds $a \Subset a \Leftrightarrow b \Leftrightarrow b \Leftrightarrow a \Leftrightarrow a$.

(25) For all elements a, b of BVF(Y) holds $a \Subset a \Rightarrow b \Leftrightarrow b$.

(26) For all elements a, b of BVF(Y) holds $a \Subset b \Rightarrow a \Leftrightarrow a$.

(27) For all elements a, b of BVF(Y) holds $a \Subset a \wedge b \Leftrightarrow b \wedge a \Leftrightarrow a$.

References

[1] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.
[2] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[3] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[4] Shunichi Kobayashi and Kui Jia. A theory of Boolean valued functions and partitions. *Formalized Mathematics*, 7(**2**):249–254, 1998.
[5] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[6] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[7] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[8] Edmund Woronowicz. Many–argument relations. *Formalized Mathematics*, 1(**4**):733–737, 1990.
[9] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received April 23, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Propositional Calculus for Boolean Valued Functions. Part IV

Shunichi Kobayashi
Shinshu University
Nagano

Summary. In this paper, we have proved some elementary propositional calculus formulae for Boolean valued functions.

MML Identifier: BVFUNC_8.

The notation and terminology used here are introduced in the following articles: [6], [7], [8], [2], [3], [5], [1], and [4].

In this paper Y denotes a non empty set.

One can prove the following propositions:

(1) For all elements a, b, c, d of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \wedge c \wedge d = (a \Rightarrow b) \wedge (a \Rightarrow c) \wedge (a \Rightarrow d)$.

(2) For all elements a, b, c, d of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \vee c \vee d = (a \Rightarrow b) \vee (a \Rightarrow c) \vee (a \Rightarrow d)$.

(3) For all elements a, b, c, d of $\mathrm{BVF}(Y)$ holds $a \wedge b \wedge c \Rightarrow d = (a \Rightarrow d) \vee (b \Rightarrow d) \vee (c \Rightarrow d)$.

(4) For all elements a, b, c, d of $\mathrm{BVF}(Y)$ holds $a \vee b \vee c \Rightarrow d = (a \Rightarrow d) \wedge (b \Rightarrow d) \wedge (c \Rightarrow d)$.

(5) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $(a \Rightarrow b) \wedge (b \Rightarrow c) \wedge (c \Rightarrow a) = (a \Rightarrow b) \wedge (b \Rightarrow c) \wedge (c \Rightarrow a) \wedge (b \Rightarrow a) \wedge (a \Rightarrow c)$.

(6) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a = a \wedge b \vee a \wedge \neg b$.

(7) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a = (a \vee b) \wedge (a \vee \neg b)$.

(8) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a = a \wedge b \wedge c \vee a \wedge b \wedge \neg c \vee a \wedge \neg b \wedge c \vee a \wedge \neg b \wedge \neg c$.

(9) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a = (a \vee b \vee c) \wedge (a \vee b \vee \neg c) \wedge (a \vee \neg b \vee c) \wedge (a \vee \neg b \vee \neg c)$.

ISSN 1426–2630

(10) For all elements a, b of BVF(Y) holds $a \wedge b = a \wedge (\neg a \vee b)$.

(11) For all elements a, b of BVF(Y) holds $a \vee b = a \vee \neg a \wedge b$.

(12) For all elements a, b of BVF(Y) holds $a \oplus b = \neg(a \Leftrightarrow b)$.

(13) For all elements a, b of BVF(Y) holds $a \oplus b = (a \vee b) \wedge (\neg a \vee \neg b)$.

(14) For every element a of BVF(Y) holds $a \oplus \mathit{true}(Y) = \neg a$.

(15) For every element a of BVF(Y) holds $a \oplus \mathit{false}(Y) = a$.

(16) For all elements a, b of BVF(Y) holds $a \oplus b = \neg a \oplus \neg b$.

(17) For all elements a, b of BVF(Y) holds $\neg(a \oplus b) = a \oplus \neg b$.

(18) For all elements a, b of BVF(Y) holds $a \Leftrightarrow b = (a \vee \neg b) \wedge (\neg a \vee b)$.

(19) For all elements a, b of BVF(Y) holds $a \Leftrightarrow b = a \wedge b \vee \neg a \wedge \neg b$.

(20) For every element a of BVF(Y) holds $a \Leftrightarrow \mathit{true}(Y) = a$.

(21) For every element a of BVF(Y) holds $a \Leftrightarrow \mathit{false}(Y) = \neg a$.

(22) For all elements a, b of BVF(Y) holds $\neg(a \Leftrightarrow b) = a \Leftrightarrow \neg b$.

(23) For all elements a, b of BVF(Y) holds $\neg a \Subset a \Rightarrow b \Leftrightarrow \neg a$.

(24) For all elements a, b of BVF(Y) holds $\neg a \Subset b \Rightarrow a \Leftrightarrow \neg b$.

(25) For all elements a, b of BVF(Y) holds $a \Subset a \vee b \Leftrightarrow b \vee a \Leftrightarrow a$.

(26) For every element a of BVF(Y) holds $a \Rightarrow \neg a \Leftrightarrow \neg a = \mathit{true}(Y)$.

(27) For all elements a, b of BVF(Y) holds $a \Rightarrow b \Rightarrow a \Rightarrow a = \mathit{true}(Y)$.

(28) For all elements a, b, c, d of BVF(Y) holds $(a \Rightarrow c) \wedge (b \Rightarrow d) \wedge (\neg c \vee \neg d) \Rightarrow \neg a \vee \neg b = \mathit{true}(Y)$.

(29) For all elements a, b, c of BVF(Y) holds $a \Rightarrow b \Rightarrow a \Rightarrow b \Rightarrow c \Rightarrow a \Rightarrow c = \mathit{true}(Y)$.

References

[1] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.
[2] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[3] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[4] Shunichi Kobayashi and Kui Jia. A theory of Boolean valued functions and partitions. *Formalized Mathematics*, 7(**2**):249–254, 1998.
[5] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[6] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[7] Edmund Woronowicz. Many–argument relations. *Formalized Mathematics*, 1(**4**):733–737, 1990.
[8] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received April 23, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Basic Properties of Genetic Algorithm

Akihiko Uchibori
Yamaguchi University
Ube

Noboru Endou
Shinshu University
Nagano

Summary. We defined the set of the gene, the space treated by the genetic algorithm and the individual of the space. Moreover, we defined some genetic operators such as one point crossover and two points crossover, and the validity of many characters were proven.

MML Identifier: GENEALG1.

The terminology and notation used in this paper have been introduced in the following articles: [10], [6], [1], [4], [13], [12], [3], [8], [2], [11], [7], [9], and [5].

1. Definitions of Gene-Set, GA-Space and Individual

We follow the rules: D is a non empty set, f_1, f_2 are finite sequences of elements of D, and i, n, n_1, n_2, n_3, n_4, n_5, n_6 are natural numbers.

We now state two propositions:

(1) If $n \leqslant \operatorname{len} f_1$, then $(f_1 \frown f_2)_{\restriction n} = ((f_1)_{\restriction n}) \frown f_2$.

(2) $(f_1 \frown f_2)\restriction(\operatorname{len} f_1 + i) = f_1 \frown (f_2 \restriction i)$.

A Gene-Set is a non-empty non empty finite sequence.

Let S be a Gene-Set. We introduce $\mathrm{GA}-\mathrm{Space}\, S$ as a synonym of $\mathrm{Union}\, S$.

Let f be a non-empty non empty function. Note that $\mathrm{Union}\, f$ is non empty.

Let S be a Gene-Set. A finite sequence of elements of $\mathrm{GA}-\mathrm{Space}\, S$ is said to be a Individual of S if:

(Def. 1) $\operatorname{len} \mathrm{it} = \operatorname{len} S$ and for every i such that $i \in \operatorname{dom} \mathrm{it}$ holds $\mathrm{it}(i) \in S(i)$.

ISSN 1426–2630

2. Definitions of Several Genetic Operators

Let S be a Gene-Set, let p_1, p_2 be finite sequences of elements of $\mathrm{GA}-\mathrm{Space}\,S$, and let us consider n. The functor $\mathrm{crossover}(p_1,p_2,n)$ yields a finite sequence of elements of $\mathrm{GA}-\mathrm{Space}\,S$ and is defined as follows:

(Def. 2) $\mathrm{crossover}(p_1,p_2,n) = (p_1{\restriction}n) ^\frown ((p_2)_{\lfloor n})$.

Let S be a Gene-Set, let p_1, p_2 be finite sequences of elements of $\mathrm{GA}-\mathrm{Space}\,S$, and let us consider n_1, n_2. The functor $\mathrm{crossover}(p_1,p_2,n_1,n_2)$ yields a finite sequence of elements of $\mathrm{GA}-\mathrm{Space}\,S$ and is defined as follows:

(Def. 3) $\mathrm{crossover}(p_1,p_2,n_1,n_2) =$
$\mathrm{crossover}(\mathrm{crossover}(p_1,p_2,n_1),\mathrm{crossover}(p_2,p_1,n_1),n_2)$.

Let S be a Gene-Set, let p_1, p_2 be finite sequences of elements of $\mathrm{GA}-\mathrm{Space}\,S$, and let us consider n_1, n_2, n_3. The functor $\mathrm{crossover}(p_1,p_2,n_1,n_2,n_3)$ yields a finite sequence of elements of $\mathrm{GA}-\mathrm{Space}\,S$ and is defined as follows:

(Def. 4) $\mathrm{crossover}(p_1,p_2,n_1,n_2,n_3) =$
$\mathrm{crossover}(\mathrm{crossover}(p_1,p_2,n_1,n_2),\mathrm{crossover}(p_2,p_1,n_1,n_2),n_3)$.

Let S be a Gene-Set, let p_1, p_2 be finite sequences of elements of $\mathrm{GA}-\mathrm{Space}\,S$, and let us consider n_1, n_2, n_3, n_4. The functor $\mathrm{crossover}(p_1,p_2,n_1,n_2,n_3,n_4)$ yields a finite sequence of elements of $\mathrm{GA}-\mathrm{Space}\,S$ and is defined as follows:

(Def. 5) $\mathrm{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) =$
$\mathrm{crossover}(\mathrm{crossover}(p_1,p_2,n_1,n_2,n_3),\mathrm{crossover}(p_2,p_1,n_1,n_2,n_3),n_4)$.

Let S be a Gene-Set, let p_1, p_2 be finite sequences of elements of $\mathrm{GA}-\mathrm{Space}\,S$, and let us consider n_1, n_2, n_3, n_4, n_5. The functor $\mathrm{crossover}(p_1,p_2,n_1,n_2,n_3,n_4,n_5)$ yielding a finite sequence of elements of $\mathrm{GA}-\mathrm{Space}\,S$ is defined by:

(Def. 6) $\mathrm{crossover}(p_1,p_2,n_1,n_2,n_3,n_4,n_5) =$
$\mathrm{crossover}(\mathrm{crossover}(p_1,p_2,n_1,n_2,n_3,n_4),\mathrm{crossover}(p_2,p_1,n_1,n_2,n_3,n_4),n_5)$.

Let S be a Gene-Set, let p_1, p_2 be finite sequences of elements of $\mathrm{GA}-\mathrm{Space}\,S$, and let us consider n_1, n_2, n_3, n_4, n_5, n_6. The functor $\mathrm{crossover}(p_1,p_2,n_1,n_2,n_3,n_4,n_5,n_6)$ yielding a finite sequence of elements of $\mathrm{GA}-\mathrm{Space}\,S$ is defined as follows:

(Def. 7) $\mathrm{crossover}(p_1,p_2,n_1,n_2,n_3,n_4,n_5,n_6) =$
$\mathrm{crossover}(\mathrm{crossover}(p_1,p_2,n_1,n_2,n_3,n_4,n_5),$
$\mathrm{crossover}(p_2,p_1,n_1,n_2,n_3,n_4,n_5),n_6)$.

3. Properties of 1-point Crossover

In the sequel S denotes a Gene-Set and p_1, p_2 denote Individual of S.

The following proposition is true

(3) $\mathrm{crossover}(p_1, p_2, n)$ is a Individual of S.

Let S be a Gene-Set, let p_1, p_2 be Individual of S, and let us consider n. Then $\mathrm{crossover}(p_1, p_2, n)$ is a Individual of S.

One can prove the following propositions:

(4) $\mathrm{crossover}(p_1, p_2, 0) = p_2$.

(5) If $n \geqslant \mathrm{len}\, p_1$, then $\mathrm{crossover}(p_1, p_2, n) = p_1$.

4. Properties of 2-points Crossover

We now state the proposition

(6) $\mathrm{crossover}(p_1, p_2, n_1, n_2)$ is a Individual of S.

Let S be a Gene-Set, let p_1, p_2 be Individual of S, and let us consider n_1, n_2. Then $\mathrm{crossover}(p_1, p_2, n_1, n_2)$ is a Individual of S.

We now state several propositions:

(7) $\mathrm{crossover}(p_1, p_2, 0, n) = \mathrm{crossover}(p_2, p_1, n)$.

(8) $\mathrm{crossover}(p_1, p_2, n, 0) = \mathrm{crossover}(p_2, p_1, n)$.

(9) If $n_1 \geqslant \mathrm{len}\, p_1$, then $\mathrm{crossover}(p_1, p_2, n_1, n_2) = \mathrm{crossover}(p_1, p_2, n_2)$.

(10) If $n_2 \geqslant \mathrm{len}\, p_1$, then $\mathrm{crossover}(p_1, p_2, n_1, n_2) = \mathrm{crossover}(p_1, p_2, n_1)$.

(11) If $n_1 \geqslant \mathrm{len}\, p_1$ and $n_2 \geqslant \mathrm{len}\, p_1$, then $\mathrm{crossover}(p_1, p_2, n_1, n_2) = p_1$.

(12) $\mathrm{crossover}(p_1, p_2, n_1, n_1) = p_1$.

(13) $\mathrm{crossover}(p_1, p_2, n_1, n_2) = \mathrm{crossover}(p_1, p_2, n_2, n_1)$.

5. Properties of 3-points Crossover

Next we state the proposition

(14) $\mathrm{crossover}(p_1, p_2, n_1, n_2, n_3)$ is a Individual of S.

Let S be a Gene-Set, let p_1, p_2 be Individual of S, and let us consider n_1, n_2, n_3. Then $\mathrm{crossover}(p_1, p_2, n_1, n_2, n_3)$ is a Individual of S.

We now state a number of propositions:

(15) $\mathrm{crossover}(p_1, p_2, 0, n_2, n_3) = \mathrm{crossover}(p_2, p_1, n_2, n_3)$ and $\mathrm{crossover}(p_1, p_2, n_1, 0, n_3) = \mathrm{crossover}(p_2, p_1, n_1, n_3)$ and $\mathrm{crossover}(p_1, p_2, n_1, n_2, 0) = \mathrm{crossover}(p_2, p_1, n_1, n_2)$.

(16) $\text{crossover}(p_1, p_2, 0, 0, n_3) = \text{crossover}(p_1, p_2, n_3)$ and $\text{crossover}(p_1, p_2, n_1, 0, 0) = \text{crossover}(p_1, p_2, n_1)$ and $\text{crossover}(p_1, p_2, 0, n_2, 0) = \text{crossover}(p_1, p_2, n_2)$.

(17) $\text{crossover}(p_1, p_2, 0, 0, 0) = p_2$.

(18) If $n_1 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3) = \text{crossover}(p_1, p_2, n_2, n_3)$.

(19) If $n_2 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3) = \text{crossover}(p_1, p_2, n_1, n_3)$.

(20) If $n_3 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3) = \text{crossover}(p_1, p_2, n_1, n_2)$.

(21) If $n_1 \geqslant \text{len}\, p_1$ and $n_2 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3) = \text{crossover}(p_1, p_2, n_3)$.

(22) If $n_1 \geqslant \text{len}\, p_1$ and $n_3 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3) = \text{crossover}(p_1, p_2, n_2)$.

(23) If $n_2 \geqslant \text{len}\, p_1$ and $n_3 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3) = \text{crossover}(p_1, p_2, n_1)$.

(24) If $n_1 \geqslant \text{len}\, p_1$ and $n_2 \geqslant \text{len}\, p_1$ and $n_3 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3) = p_1$.

(25) $\text{crossover}(p_1, p_2, n_1, n_2, n_3) = \text{crossover}(p_1, p_2, n_2, n_1, n_3)$ and $\text{crossover}(p_1, p_2, n_1, n_2, n_3) = \text{crossover}(p_1, p_2, n_1, n_3, n_2)$.

(26) $\text{crossover}(p_1, p_2, n_1, n_2, n_3) = \text{crossover}(p_1, p_2, n_3, n_1, n_2)$.

(27) $\text{crossover}(p_1, p_2, n_1, n_1, n_3) = \text{crossover}(p_1, p_2, n_3)$ and $\text{crossover}(p_1, p_2, n_1, n_2, n_1) = \text{crossover}(p_1, p_2, n_2)$ and $\text{crossover}(p_1, p_2, n_1, n_2, n_2) = \text{crossover}(p_1, p_2, n_1)$.

6. Properties of 4-points Crossover

Next we state the proposition

(28) $\text{crossover}(p_1, p_2, n_1, n_2, n_3, n_4)$ is a Individual of S.

Let S be a Gene-Set, let p_1, p_2 be Individual of S, and let us consider n_1, n_2, n_3, n_4. Then $\text{crossover}(p_1, p_2, n_1, n_2, n_3, n_4)$ is a Individual of S.

The following propositions are true:

(29) $\text{crossover}(p_1, p_2, 0, n_2, n_3, n_4) = \text{crossover}(p_2, p_1, n_2, n_3, n_4)$ and $\text{crossover}(p_1, p_2, n_1, 0, n_3, n_4) = \text{crossover}(p_2, p_1, n_1, n_3, n_4)$ and $\text{crossover}(p_1, p_2, n_1, n_2, 0, n_4) = \text{crossover}(p_2, p_1, n_1, n_2, n_4)$ and $\text{crossover}(p_1, p_2, n_1, n_2, n_3, 0) = \text{crossover}(p_2, p_1, n_1, n_2, n_3)$.

(30) $\text{crossover}(p_1, p_2, 0, 0, n_3, n_4) = \text{crossover}(p_1, p_2, n_3, n_4)$ and $\text{crossover}(p_1, p_2, 0, n_2, 0, n_4) = \text{crossover}(p_1, p_2, n_2, n_4)$ and $\text{crossover}(p_1, p_2, 0, n_2, n_3, 0) = \text{crossover}(p_1, p_2, n_2, n_3)$ and $\text{crossover}(p_1, p_2, n_1, 0, n_3, 0) = \text{crossover}(p_1, p_2, n_1, n_3)$ and $\text{crossover}(p_1, p_2, n_1, 0, 0, n_4) = \text{crossover}(p_1, p_2, n_1, n_4)$ and

crossover$(p_1, p_2, n_1, n_2, 0, 0)$ = crossover(p_1, p_2, n_1, n_2).

(31) crossover$(p_1, p_2, n_1, 0, 0, 0)$ = crossover(p_2, p_1, n_1) and crossover$(p_1, p_2, 0, n_2, 0, 0)$ = crossover(p_2, p_1, n_2) and crossover$(p_1, p_2, 0, 0, n_3, 0)$ = crossover(p_2, p_1, n_3) and crossover$(p_1, p_2, 0, 0, 0, n_4)$ = crossover(p_2, p_1, n_4).

(32) crossover$(p_1, p_2, 0, 0, 0, 0) = p_1$.

(33)(i) If $n_1 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover$(p_1, p_2, n_2, n_3, n_4)$,

(ii) if $n_2 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover$(p_1, p_2, n_1, n_3, n_4)$,

(iii) if $n_3 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover$(p_1, p_2, n_1, n_2, n_4)$, and

(iv) if $n_4 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover$(p_1, p_2, n_1, n_2, n_3)$.

(34)(i) If $n_1 \geqslant \operatorname{len} p_1$ and $n_2 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover(p_1, p_2, n_3, n_4),

(ii) if $n_1 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover(p_1, p_2, n_2, n_4),

(iii) if $n_1 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover(p_1, p_2, n_2, n_3),

(iv) if $n_2 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover(p_1, p_2, n_1, n_4),

(v) if $n_2 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover(p_1, p_2, n_1, n_3), and

(vi) if $n_3 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover(p_1, p_2, n_1, n_2).

(35)(i) If $n_1 \geqslant \operatorname{len} p_1$ and $n_2 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover(p_1, p_2, n_4),

(ii) if $n_1 \geqslant \operatorname{len} p_1$ and $n_2 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover(p_1, p_2, n_3),

(iii) if $n_1 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover(p_1, p_2, n_2), and

(iv) if $n_2 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover(p_1, p_2, n_1).

(36) If $n_1 \geqslant \operatorname{len} p_1$ and $n_2 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then crossover$(p_1, p_2, n_1, n_2, n_3, n_4) = p_1$.

(37) crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover$(p_1, p_2, n_1, n_2, n_4, n_3)$ and crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover$(p_1, p_2, n_1, n_3, n_2, n_4)$ and crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover$(p_1, p_2, n_1, n_3, n_4, n_2)$ and crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover$(p_1, p_2, n_1, n_4, n_2, n_3)$ and crossover$(p_1, p_2, n_1, n_2, n_3, n_4)$ = crossover$(p_1, p_2, n_1, n_4, n_3, n_2)$ and

$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_2,n_1,n_3,n_4)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_2,n_1,n_4,n_3)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_2,n_3,n_1,n_4)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_2,n_3,n_4,n_1)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_2,n_4,n_1,n_3)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_2,n_4,n_3,n_1)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_3,n_1,n_2,n_4)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_3,n_1,n_4,n_2)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_3,n_2,n_1,n_4)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_3,n_2,n_4,n_1)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_3,n_4,n_1,n_2)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_3,n_4,n_2,n_1)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_4,n_1,n_2,n_3)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_4,n_1,n_3,n_2)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_4,n_2,n_1,n_3)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_4,n_2,n_3,n_1)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_4,n_3,n_1,n_2)$ and
$\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4) = \text{crossover}(p_1,p_2,n_4,n_3,n_2,n_1)$.

(38) $\text{crossover}(p_1,p_2,n_1,n_1,n_3,n_4) = \text{crossover}(p_1,p_2,n_3,n_4)$ and $\text{crossover}(p_1,p_2,n_1,n_2,n_1,n_4) = \text{crossover}(p_1,p_2,n_2,n_4)$ and $\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_1) = \text{crossover}(p_1,p_2,n_2,n_3)$ and $\text{crossover}(p_1,p_2,n_1,n_2,n_2,n_4) = \text{crossover}(p_1,p_2,n_1,n_4)$ and $\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_2) = \text{crossover}(p_1,p_2,n_1,n_3)$ and $\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_3) = \text{crossover}(p_1,p_2,n_1,n_2)$.

(39) $\text{crossover}(p_1,p_2,n_1,n_1,n_3,n_3) = p_1$ and $\text{crossover}(p_1,p_2,n_1,n_2,n_1,n_2) = p_1$ and $\text{crossover}(p_1,p_2,n_1,n_2,n_2,n_1) = p_1$.

7. Properties of 5-points Crossover

Next we state the proposition

(40) $\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4,n_5)$ is a Individual of S.

Let S be a Gene-Set, let p_1, p_2 be Individual of S, and let us consider n_1, n_2, n_3, n_4, n_5. Then $\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4,n_5)$ is a Individual of S.

Next we state a number of propositions:

(41) $\text{crossover}(p_1,p_2,0,n_2,n_3,n_4,n_5) = \text{crossover}(p_2,p_1,n_2,n_3,n_4,n_5)$ and $\text{crossover}(p_1,p_2,n_1,0,n_3,n_4,n_5) = \text{crossover}(p_2,p_1,n_1,n_3,n_4,n_5)$ and $\text{crossover}(p_1,p_2,n_1,n_2,0,n_4,n_5) = \text{crossover}(p_2,p_1,n_1,n_2,n_4,n_5)$ and $\text{crossover}(p_1,p_2,n_1,n_2,n_3,0,n_5) = \text{crossover}(p_2,p_1,n_1,n_2,n_3,n_5)$ and $\text{crossover}(p_1,p_2,n_1,n_2,n_3,n_4,0) = \text{crossover}(p_2,p_1,n_1,n_2,n_3,n_4)$.

(42) $\text{crossover}(p_1, p_2, 0, 0, n_3, n_4, n_5) = \text{crossover}(p_1, p_2, n_3, n_4, n_5)$ and
$\text{crossover}(p_1, p_2, 0, n_2, 0, n_4, n_5) = \text{crossover}(p_1, p_2, n_2, n_4, n_5)$ and
$\text{crossover}(p_1, p_2, 0, n_2, n_3, 0, n_5) = \text{crossover}(p_1, p_2, n_2, n_3, n_5)$ and
$\text{crossover}(p_1, p_2, 0, n_2, n_3, n_4, 0) = \text{crossover}(p_1, p_2, n_2, n_3, n_4)$ and
$\text{crossover}(p_1, p_2, n_1, 0, 0, n_4, n_5) = \text{crossover}(p_1, p_2, n_1, n_4, n_5)$ and
$\text{crossover}(p_1, p_2, n_1, 0, n_3, 0, n_5) = \text{crossover}(p_1, p_2, n_1, n_3, n_5)$ and
$\text{crossover}(p_1, p_2, n_1, 0, n_3, n_4, 0) = \text{crossover}(p_1, p_2, n_1, n_3, n_4)$ and
$\text{crossover}(p_1, p_2, n_1, n_2, 0, 0, n_5) = \text{crossover}(p_1, p_2, n_1, n_2, n_5)$ and
$\text{crossover}(p_1, p_2, n_1, n_2, 0, n_4, 0) = \text{crossover}(p_1, p_2, n_1, n_2, n_4)$ and
$\text{crossover}(p_1, p_2, n_1, n_2, n_3, 0, 0) = \text{crossover}(p_1, p_2, n_1, n_2, n_3)$.

(43) $\text{crossover}(p_1, p_2, 0, 0, 0, n_4, n_5) = \text{crossover}(p_2, p_1, n_4, n_5)$ and
$\text{crossover}(p_1, p_2, 0, 0, n_3, 0, n_5) = \text{crossover}(p_2, p_1, n_3, n_5)$ and
$\text{crossover}(p_1, p_2, 0, 0, n_3, n_4, 0) = \text{crossover}(p_2, p_1, n_3, n_4)$ and
$\text{crossover}(p_1, p_2, 0, n_2, 0, 0, n_5) = \text{crossover}(p_2, p_1, n_2, n_5)$ and
$\text{crossover}(p_1, p_2, 0, n_2, 0, n_4, 0) = \text{crossover}(p_2, p_1, n_2, n_4)$ and
$\text{crossover}(p_1, p_2, 0, n_2, n_3, 0, 0) = \text{crossover}(p_2, p_1, n_2, n_3)$ and
$\text{crossover}(p_1, p_2, n_1, 0, 0, 0, n_5) = \text{crossover}(p_2, p_1, n_1, n_5)$ and
$\text{crossover}(p_1, p_2, n_1, 0, 0, n_4, 0) = \text{crossover}(p_2, p_1, n_1, n_4)$ and
$\text{crossover}(p_1, p_2, n_1, 0, n_3, 0, 0) = \text{crossover}(p_2, p_1, n_1, n_3)$ and
$\text{crossover}(p_1, p_2, n_1, n_2, 0, 0, 0) = \text{crossover}(p_2, p_1, n_1, n_2)$.

(44) $\text{crossover}(p_1, p_2, 0, 0, 0, 0, n_5) = \text{crossover}(p_1, p_2, n_5)$ and
$\text{crossover}(p_1, p_2, 0, 0, 0, n_4, 0) = \text{crossover}(p_1, p_2, n_4)$ and
$\text{crossover}(p_1, p_2, 0, 0, n_3, 0, 0) = \text{crossover}(p_1, p_2, n_3)$ and
$\text{crossover}(p_1, p_2, 0, n_2, 0, 0, 0) = \text{crossover}(p_1, p_2, n_2)$ and
$\text{crossover}(p_1, p_2, n_1, 0, 0, 0, 0) = \text{crossover}(p_1, p_2, n_1)$.

(45) $\text{crossover}(p_1, p_2, 0, 0, 0, 0, 0) = p_2$.

(46)(i) If $n_1 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \text{crossover}(p_1, p_2, n_2, n_3, n_4, n_5)$,

(ii) if $n_2 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \text{crossover}(p_1, p_2, n_1, n_3, n_4, n_5)$,

(iii) if $n_3 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \text{crossover}(p_1, p_2, n_1, n_2, n_4, n_5)$,

(iv) if $n_4 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \text{crossover}(p_1, p_2, n_1, n_2, n_3, n_5)$, and

(v) if $n_5 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \text{crossover}(p_1, p_2, n_1, n_2, n_3, n_4)$.

(47)(i) If $n_1 \geqslant \text{len}\, p_1$ and $n_2 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \text{crossover}(p_1, p_2, n_3, n_4, n_5)$,

(ii) if $n_1 \geqslant \text{len}\, p_1$ and $n_3 \geqslant \text{len}\, p_1$, then $\text{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \text{crossover}(p_1, p_2, n_2, n_4, n_5)$,

(iii) if $n_1 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_2, n_3, n_5)$,

(iv) if $n_1 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_2, n_3, n_4)$,

(v) if $n_2 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_1, n_4, n_5)$,

(vi) if $n_2 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_1, n_3, n_5)$,

(vii) if $n_2 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_1, n_3, n_4)$,

(viii) if $n_3 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_1, n_2, n_5)$,

(ix) if $n_3 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_1, n_2, n_4)$, and

(x) if $n_4 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_1, n_2, n_3)$.

(48)(i) If $n_1 \geqslant \operatorname{len} p_1$ and $n_2 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_4, n_5)$,

(ii) if $n_1 \geqslant \operatorname{len} p_1$ and $n_2 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_3, n_5)$,

(iii) if $n_1 \geqslant \operatorname{len} p_1$ and $n_2 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_3, n_4)$,

(iv) if $n_1 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_2, n_5)$,

(v) if $n_1 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_2, n_4)$,

(vi) if $n_1 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_2, n_3)$,

(vii) if $n_2 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_1, n_5)$,

(viii) if $n_2 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_1, n_4)$,

(ix) if $n_2 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_1, n_3)$, and

(x) if $n_3 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_1, n_2)$.

(49)(i) If $n_1 \geqslant \operatorname{len} p_1$ and $n_2 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_5)$,

(ii) if $n_1 \geqslant \operatorname{len} p_1$ and $n_2 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_4)$,

(iii) if $n_1 \geqslant \operatorname{len} p_1$ and $n_2 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_3)$,

(iv) if $n_1 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_2)$, and

(v) if $n_2 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_1)$.

(50) If $n_1 \geqslant \operatorname{len} p_1$ and $n_2 \geqslant \operatorname{len} p_1$ and $n_3 \geqslant \operatorname{len} p_1$ and $n_4 \geqslant \operatorname{len} p_1$ and $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = p_1$.

(51) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_2, n_1, n_3, n_4, n_5)$ and $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_3, n_2, n_1, n_4, n_5)$ and $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_4, n_2, n_3, n_1, n_5)$ and $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_5, n_2, n_3, n_4, n_1)$.

(52) $\operatorname{crossover}(p_1, p_2, n_1, n_1, n_3, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_3, n_4, n_5)$ and $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_1, n_4, n_5) = \operatorname{crossover}(p_1, p_2, n_2, n_4, n_5)$ and $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_1, n_5) = \operatorname{crossover}(p_1, p_2, n_2, n_3, n_5)$ and $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_1) = \operatorname{crossover}(p_1, p_2, n_2, n_3, n_4)$.

8. Properties of 6-points Crossover

Next we state the proposition

(53) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_6)$ is a Individual of S.

Let S be a Gene-Set, let p_1, p_2 be Individual of S, and let us consider n_1, n_2, n_3, n_4, n_5, n_6. Then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_6)$ is a Individual of S.

We now state four propositions:

(54)(i) $\operatorname{crossover}(p_1, p_2, 0, n_2, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_2, p_1, n_2, n_3, n_4, n_5, n_6)$,

(ii) $\operatorname{crossover}(p_1, p_2, n_1, 0, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_2, p_1, n_1, n_3, n_4, n_5, n_6)$,

(iii) $\operatorname{crossover}(p_1, p_2, n_1, n_2, 0, n_4, n_5, n_6) = \operatorname{crossover}(p_2, p_1, n_1, n_2, n_4, n_5, n_6)$,

(iv) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, 0, n_5, n_6) = \operatorname{crossover}(p_2, p_1, n_1, n_2, n_3, n_5, n_6)$,

(v) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, 0, n_6) = \operatorname{crossover}(p_2, p_1, n_1, n_2, n_3, n_4, n_6)$, and

(vi) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, 0) = \operatorname{crossover}(p_2, p_1, n_1, n_2, n_3, n_4, n_5)$.

(55)(i) If $n_1 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_2, n_3, n_4, n_5, n_6)$,

(ii) if $n_2 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_1, n_3, n_4, n_5, n_6)$,

(iii) if $n_3 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_1, n_2, n_4, n_5, n_6)$,

(iv) if $n_4 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_5, n_6)$,

(v) if $n_5 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_6)$, and

(vi) if $n_6 \geqslant \operatorname{len} p_1$, then $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5)$.

(56)(i) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_2, n_1, n_3, n_4, n_5, n_6)$,

(ii) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_3, n_2, n_1, n_4, n_5, n_6)$,

(iii) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_4, n_2, n_3, n_1, n_5, n_6)$,

(iv) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_5, n_2, n_3, n_4, n_1, n_6)$, and

(v) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_6, n_2, n_3, n_4, n_5, n_1)$.

(57)(i) $\operatorname{crossover}(p_1, p_2, n_1, n_1, n_3, n_4, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_3, n_4, n_5, n_6)$,

(ii) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_1, n_4, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_2, n_4, n_5, n_6)$,

(iii) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_1, n_5, n_6) = \operatorname{crossover}(p_1, p_2, n_2, n_3, n_5, n_6)$,

(iv) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_1, n_6) = \operatorname{crossover}(p_1, p_2, n_2, n_3, n_4, n_6)$, and

(v) $\operatorname{crossover}(p_1, p_2, n_1, n_2, n_3, n_4, n_5, n_1) = \operatorname{crossover}(p_1, p_2, n_2, n_3, n_4, n_5)$.

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[2] Grzegorz Bancerek. The reflection theorem. *Formalized Mathematics*, 1(**5**):973–977, 1990.
[3] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[4] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[5] Agata Darmochwał and Yatsuka Nakamura. The topological space $\mathcal{E}^2_{\mathrm{T}}$. Arcs, line segments and special polygonal arcs. *Formalized Mathematics*, 2(**5**):617–621, 1991.
[6] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[7] Jarosław Kotowicz. Functions and finite sequences of real numbers. *Formalized Mathematics*, 3(**2**):275–278, 1992.
[8] Andrzej Nędzusiak. σ-fields and probability. *Formalized Mathematics*, 1(**2**):401–407, 1990.
[9] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[10] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[11] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[12] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[13] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received April 24, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Propositional Calculus for Boolean Valued Functions. Part V

Shunichi Kobayashi
Shinshu University
Nagano

Summary. In this paper, we have proved some elementary propositional calculus formulae for Boolean valued functions.

MML Identifier: BVFUNC_9.

The terminology and notation used here have been introduced in the following articles: [3], [4], [5], [2], and [1].

In this paper Y denotes a non empty set.

We now state a number of propositions:

(1) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $(a \vee b) \wedge (b \Rightarrow c) \Subset a \vee c$.

(2) For all elements a, b of $\mathrm{BVF}(Y)$ holds $a \wedge (a \Rightarrow b) \Subset b$.

(3) For all elements a, b of $\mathrm{BVF}(Y)$ holds $(a \Rightarrow b) \wedge \neg b \Subset \neg a$.

(4) For all elements a, b of $\mathrm{BVF}(Y)$ holds $(a \vee b) \wedge \neg a \Subset b$.

(5) For all elements a, b of $\mathrm{BVF}(Y)$ holds $(a \Rightarrow b) \wedge (\neg a \Rightarrow b) \Subset b$.

(6) For all elements a, b of $\mathrm{BVF}(Y)$ holds $(a \Rightarrow b) \wedge (a \Rightarrow \neg b) \Subset \neg a$.

(7) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \wedge c \Subset a \Rightarrow b$.

(8) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \vee b \Rightarrow c \Subset a \Rightarrow c$.

(9) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \Subset a \wedge c \Rightarrow b$.

(10) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \Subset a \wedge c \Rightarrow b \wedge c$.

(11) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \Subset a \Rightarrow b \vee c$.

(12) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \Rightarrow b \Subset a \vee c \Rightarrow b \vee c$.

(13) For all elements a, b, c of $\mathrm{BVF}(Y)$ holds $a \wedge b \vee c \Subset a \vee c$.

(14) For all elements a, b, c, d of $\mathrm{BVF}(Y)$ holds $a \wedge b \vee c \wedge d \Subset a \vee c$.

ISSN 1426–2630

(15) For all elements a, b, c of BVF(Y) holds $(a \vee b) \wedge (b \Rightarrow c) \Subset a \vee c$.

(16) For all elements a, b, c of BVF(Y) holds $(a \Rightarrow b) \wedge (\neg a \Rightarrow c) \Subset b \vee c$.

(17) For all elements a, b, c of BVF(Y) holds $(a \Rightarrow c) \wedge (b \Rightarrow \neg c) \Subset \neg a \vee \neg b$.

(18) For all elements a, b, c of BVF(Y) holds $(a \vee b) \wedge (\neg a \vee c) \Subset b \vee c$.

(19) For all elements a, b, c of BVF(Y) holds $(a \Rightarrow b) \wedge (a \Rightarrow c) \Subset a \Rightarrow b \wedge c$.

(20) For all elements a, b, c, d of BVF(Y) holds $(a \Rightarrow b) \wedge (c \Rightarrow d) \Subset a \wedge c \Rightarrow b \wedge d$.

(21) For all elements a, b, c of BVF(Y) holds $(a \Rightarrow c) \wedge (b \Rightarrow c) \Subset a \vee b \Rightarrow c$.

(22) For all elements a, b, c, d of BVF(Y) holds $(a \Rightarrow b) \wedge (c \Rightarrow d) \Subset a \vee c \Rightarrow b \vee d$.

(23) For all elements a, b, c of BVF(Y) holds $(a \Rightarrow b) \wedge (a \Rightarrow c) \Subset a \Rightarrow b \vee c$.

(24) For all elements a_1, b_1, c_1, a_2, b_2, c_2 of BVF(Y) holds $(b_1 \Rightarrow b_2) \wedge (c_1 \Rightarrow c_2) \wedge (a_1 \vee b_1 \vee c_1) \wedge \neg(a_2 \wedge b_2) \wedge \neg(a_2 \wedge c_2) \Subset a_2 \Rightarrow a_1$.

(25) For all elements a_1, b_1, c_1, a_2, b_2, c_2 of BVF(Y) holds $(a_1 \Rightarrow a_2) \wedge (b_1 \Rightarrow b_2) \wedge (c_1 \Rightarrow c_2) \wedge (a_1 \vee b_1 \vee c_1) \wedge \neg(a_2 \wedge b_2) \wedge \neg(a_2 \wedge c_2) \wedge \neg(b_2 \wedge c_2) \Subset (a_2 \Rightarrow a_1) \wedge (b_2 \Rightarrow b_1) \wedge (c_2 \Rightarrow c_1)$.

(26) For all elements a_1, b_1, a_2, b_2 of BVF(Y) holds $(a_1 \Rightarrow a_2) \wedge (b_1 \Rightarrow b_2) \wedge \neg(a_2 \wedge b_2) \Rightarrow \neg(a_1 \wedge b_1) = \mathit{true}(Y)$.

(27) For all elements a_1, b_1, c_1, a_2, b_2, c_2 of BVF(Y) holds $(a_1 \Rightarrow a_2) \wedge (b_1 \Rightarrow b_2) \wedge (c_1 \Rightarrow c_2) \wedge \neg(a_2 \wedge b_2) \wedge \neg(a_2 \wedge c_2) \wedge \neg(b_2 \wedge c_2) \Subset \neg(a_1 \wedge b_1) \wedge \neg(a_1 \wedge c_1) \wedge \neg(b_1 \wedge c_1)$.

References

[1] Shunichi Kobayashi and Kui Jia. A theory of Boolean valued functions and partitions. *Formalized Mathematics*, 7(**2**):249–254, 1998.

[2] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.

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

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

[5] Edmund Woronowicz. Many–argument relations. *Formalized Mathematics*, 1(**4**):733–737, 1990.

Received May 5, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Properties of Left and Right Components

Artur Korniłowicz[1]
University of Białystok

MML Identifier: GOBRD14.

The notation and terminology used here have been introduced in the following papers: [33], [42], [43], [6], [7], [41], [5], [16], [35], [1], [30], [38], [31], [17], [27], [8], [19], [39], [18], [20], [15], [4], [2], [3], [40], [32], [29], [44], [12], [28], [11], [13], [14], [21], [22], [25], [34], [10], [24], [23], [37], [36], [26], and [9].

1. Components

For simplicity, we adopt the following rules: r denotes a real number, i, j, n denote natural numbers, f denotes a non constant standard special circular sequence, g denotes a clockwise oriented non constant standard special circular sequence, p, q denote points of $\mathcal{E}^2_{\rm T}$, P, Q, R denote subsets of $\mathcal{E}^2_{\rm T}$, C denotes a compact non vertical non horizontal subset of $\mathcal{E}^2_{\rm T}$, and G denotes a Go-board.

Next we state several propositions:

(1) Let T be a topological space, A be a subset of the carrier of T, and B be a subset of T. If B is a component of A, then B is connected.

(2) Let A be a subset of the carrier of $\mathcal{E}^n_{\rm T}$ and B be a subset of $\mathcal{E}^n_{\rm T}$. If B is inside component of A, then B is connected.

(3) Let A be a subset of the carrier of $\mathcal{E}^n_{\rm T}$ and B be a subset of $\mathcal{E}^n_{\rm T}$. If B is outside component of A, then B is connected.

(4) For every subset A of the carrier of $\mathcal{E}^n_{\rm T}$ and for every subset B of $\mathcal{E}^n_{\rm T}$ such that B is a component of $A^{\rm c}$ holds $A \cap B = \emptyset$.

(5) If P is outside component of Q and R is inside component of Q, then $P \cap R = \emptyset$.

[1]This paper was written while the author visited Shinshu University, winter 1999.

ISSN 1426–2630

(6) Let A, B be subsets of $\mathcal{E}^2_{\mathrm{T}}$. Suppose A is outside component of $\widetilde{\mathcal{L}}(f)$ and B is outside component of $\widetilde{\mathcal{L}}(f)$. Then $A = B$.

(7) Let p be a point of $\mathcal{E}^2$. Suppose $p = 0_{\mathcal{E}^2_{\mathrm{T}}}$ and P is outside component of $\widetilde{\mathcal{L}}(f)$. Then there exists a real number r such that $r > 0$ and $\mathrm{Ball}(p, r)^{\mathrm{c}} \subseteq P$.

Let C be a closed subset of $\mathcal{E}^2_{\mathrm{T}}$. Observe that $\mathrm{BDD}\, C$ is open and $\mathrm{UBD}\, C$ is open.

Let C be a compact subset of $\mathcal{E}^2_{\mathrm{T}}$. Observe that $\mathrm{UBD}\, C$ is connected.

2. Go-Boards

One can prove the following proposition

(8) For every finite sequence f of elements of $\mathcal{E}^n_{\mathrm{T}}$ such that $\widetilde{\mathcal{L}}(f) \neq \emptyset$ holds $2 \leqslant \mathrm{len}\, f$.

Let n be a natural number and let a, b be points of $\mathcal{E}^n_{\mathrm{T}}$. The functor $\rho(a, b)$ yields a real number and is defined by:

(Def. 1) There exist points p, q of $\mathcal{E}^n$ such that $p = a$ and $q = b$ and $\rho(a, b) = \rho(p, q)$.

Let us notice that the functor $\rho(a, b)$ is commutative.

The following propositions are true:

(9) $\rho(p, q) = \sqrt{(p_{\mathbf{1}} - q_{\mathbf{1}})^{\mathbf{2}} + (p_{\mathbf{2}} - q_{\mathbf{2}})^{\mathbf{2}}}$.

(10) For every point p of $\mathcal{E}^n_{\mathrm{T}}$ holds $\rho(p, p) = 0$.

(11) For all points p, q, r of $\mathcal{E}^n_{\mathrm{T}}$ holds $\rho(p, r) \leqslant \rho(p, q) + \rho(q, r)$.

(12) Let x_1, x_2, y_1, y_2 be real numbers and a, b be points of $\mathcal{E}^2_{\mathrm{T}}$. Suppose $x_1 \leqslant a_{\mathbf{1}}$ and $a_{\mathbf{1}} \leqslant x_2$ and $y_1 \leqslant a_{\mathbf{2}}$ and $a_{\mathbf{2}} \leqslant y_2$ and $x_1 \leqslant b_{\mathbf{1}}$ and $b_{\mathbf{1}} \leqslant x_2$ and $y_1 \leqslant b_{\mathbf{2}}$ and $b_{\mathbf{2}} \leqslant y_2$. Then $\rho(a, b) \leqslant |x_2 - x_1| + |y_2 - y_1|$.

(13) If $1 \leqslant i$ and $i < \mathrm{len}\, G$ and $1 \leqslant j$ and $j < \mathrm{width}\, G$, then $\mathrm{cell}(G, i, j) = \prod[1 \longmapsto [(G_{i,1})_{\mathbf{1}}, (G_{i+1,1})_{\mathbf{1}}], 2 \longmapsto [(G_{1,j})_{\mathbf{2}}, (G_{1,j+1})_{\mathbf{2}}]]$.

(14) If $1 \leqslant i$ and $i < \mathrm{len}\, G$ and $1 \leqslant j$ and $j < \mathrm{width}\, G$, then $\mathrm{cell}(G, i, j)$ is compact.

(15) If $\langle i,\, j \rangle \in$ the indices of G and $\langle i + n,\, j \rangle \in$ the indices of G, then $\rho(G_{i,j}, G_{i+n,j}) = (G_{i+n,j})_{\mathbf{1}} - (G_{i,j})_{\mathbf{1}}$.

(16) If $\langle i,\, j \rangle \in$ the indices of G and $\langle i,\, j + n \rangle \in$ the indices of G, then $\rho(G_{i,j}, G_{i,j+n}) = (G_{i,j+n})_{\mathbf{2}} - (G_{i,j})_{\mathbf{2}}$.

(17) $3 \leqslant \mathrm{len}\, \mathrm{Gauge}(C, n) -' 1$.

(18) Suppose $i \leqslant j$. Let a, b be natural numbers. Suppose $2 \leqslant a$ and $a \leqslant \mathrm{len}\, \mathrm{Gauge}(C, i) - 1$ and $2 \leqslant b$ and $b \leqslant \mathrm{len}\, \mathrm{Gauge}(C, i) - 1$. Then there exist natural numbers c, d such that

$2 \leqslant c$ and $c \leqslant \operatorname{len} \operatorname{Gauge}(C, j) - 1$ and $2 \leqslant d$ and $d \leqslant \operatorname{len} \operatorname{Gauge}(C, j) - 1$ and $\langle c, d\rangle \in$ the indices of $\operatorname{Gauge}(C, j)$ and $(\operatorname{Gauge}(C, i))_{a,b} = (\operatorname{Gauge}(C, j))_{c,d}$ and $c = 2 + 2^{j -' i} \cdot (a -' 2)$ and $d = 2 + 2^{j -' i} \cdot (b -' 2)$.

(19) If $\langle i, j\rangle \in$ the indices of $\operatorname{Gauge}(C, n)$ and $\langle i, j + 1\rangle \in$ the indices of $\operatorname{Gauge}(C, n)$, then $\rho((\operatorname{Gauge}(C, n))_{i,j}, (\operatorname{Gauge}(C, n))_{i,j+1}) = \frac{\text{N-bound}\, C - \text{S-bound}\, C}{2^n}$.

(20) If $\langle i, j\rangle \in$ the indices of $\operatorname{Gauge}(C, n)$ and $\langle i + 1, j\rangle \in$ the indices of $\operatorname{Gauge}(C, n)$, then $\rho((\operatorname{Gauge}(C, n))_{i,j}, (\operatorname{Gauge}(C, n))_{i+1,j}) = \frac{\text{E-bound}\, C - \text{W-bound}\, C}{2^n}$.

(21) If $r > 0$, then there exists a natural number n such that $\rho((\operatorname{Gauge}(C, n))_{1,1}, (\operatorname{Gauge}(C, n))_{1,2}) < r$ and $\rho((\operatorname{Gauge}(C, n))_{1,1}, (\operatorname{Gauge}(C, n))_{2,1}) < r$.

3. LeftComp and RightComp

One can prove the following propositions:

(22) For every subset P of $(\mathcal{E}^2_{\mathrm{T}}) \restriction (\widetilde{\mathcal{L}}(f))^{\mathrm{c}}$ such that P is a component of $(\mathcal{E}^2_{\mathrm{T}}) \restriction (\widetilde{\mathcal{L}}(f))^{\mathrm{c}}$ holds $P = \operatorname{RightComp}(f)$ or $P = \operatorname{LeftComp}(f)$.

(23) Let A_1, A_2 be subsets of $\mathcal{E}^2_{\mathrm{T}}$. Suppose that

(i) $(\widetilde{\mathcal{L}}(f))^{\mathrm{c}} = A_1 \cup A_2$,

(ii) $A_1 \cap A_2 = \emptyset$, and

(iii) for all subsets C_1, C_2 of $(\mathcal{E}^2_{\mathrm{T}}) \restriction (\widetilde{\mathcal{L}}(f))^{\mathrm{c}}$ such that $C_1 = A_1$ and $C_2 = A_2$ holds C_1 is a component of $(\mathcal{E}^2_{\mathrm{T}}) \restriction (\widetilde{\mathcal{L}}(f))^{\mathrm{c}}$ and C_2 is a component of $(\mathcal{E}^2_{\mathrm{T}}) \restriction (\widetilde{\mathcal{L}}(f))^{\mathrm{c}}$.

Then $A_1 = \operatorname{RightComp}(f)$ and $A_2 = \operatorname{LeftComp}(f)$ or $A_1 = \operatorname{LeftComp}(f)$ and $A_2 = \operatorname{RightComp}(f)$.

(24) $\operatorname{LeftComp}(f) \cap \operatorname{RightComp}(f) = \emptyset$.

(25) $\widetilde{\mathcal{L}}(f) \cup \operatorname{RightComp}(f) \cup \operatorname{LeftComp}(f) =$ the carrier of $\mathcal{E}^2_{\mathrm{T}}$.

(26) $p \in \widetilde{\mathcal{L}}(f)$ iff $p \notin \operatorname{LeftComp}(f)$ and $p \notin \operatorname{RightComp}(f)$.

(27) $p \in \operatorname{LeftComp}(f)$ iff $p \notin \widetilde{\mathcal{L}}(f)$ and $p \notin \operatorname{RightComp}(f)$.

(28) $p \in \operatorname{RightComp}(f)$ iff $p \notin \widetilde{\mathcal{L}}(f)$ and $p \notin \operatorname{LeftComp}(f)$.

(29) $\widetilde{\mathcal{L}}(f) = \overline{\operatorname{RightComp}(f)} \setminus \operatorname{RightComp}(f)$.

(30) $\widetilde{\mathcal{L}}(f) = \overline{\operatorname{LeftComp}(f)} \setminus \operatorname{LeftComp}(f)$.

(31) $\overline{\operatorname{RightComp}(f)} = \operatorname{RightComp}(f) \cup \widetilde{\mathcal{L}}(f)$.

(32) $\overline{\operatorname{LeftComp}(f)} = \operatorname{LeftComp}(f) \cup \widetilde{\mathcal{L}}(f)$.

Let f be a non constant standard special circular sequence. One can verify that $\widetilde{\mathcal{L}}(f)$ is Jordan.

The following propositions are true:

(33) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$ and $p \in \text{RightComp}(g)$, then $\text{W-bound}\,\widetilde{\mathcal{L}}(g) < p_{\mathbf{1}}$.

(34) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$ and $p \in \text{RightComp}(g)$, then $\text{E-bound}\,\widetilde{\mathcal{L}}(g) > p_{\mathbf{1}}$.

(35) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$ and $p \in \text{RightComp}(g)$, then $\text{N-bound}\,\widetilde{\mathcal{L}}(g) > p_{\mathbf{2}}$.

(36) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$ and $p \in \text{RightComp}(g)$, then $\text{S-bound}\,\widetilde{\mathcal{L}}(g) < p_{\mathbf{2}}$.

(37) If $p \in \text{RightComp}(f)$ and $q \in \text{LeftComp}(f)$, then $\mathcal{L}(p, q) \cap \widetilde{\mathcal{L}}(f) \neq \emptyset$.

(38) $\overline{\text{RightComp}(\text{SpStSeq}\,C)} = \prod[1 \longmapsto [\text{W-bound}\,\widetilde{\mathcal{L}}(\text{SpStSeq}\,C),$
$\text{E-bound}\,\widetilde{\mathcal{L}}(\text{SpStSeq}\,C)], 2 \longmapsto [\text{S-bound}\,\widetilde{\mathcal{L}}(\text{SpStSeq}\,C),$
$\text{N-bound}\,\widetilde{\mathcal{L}}(\text{SpStSeq}\,C)]]$.

(39) $(\text{proj1})^\circ\widetilde{\mathcal{L}}(f) \subseteq (\text{proj1})^\circ\overline{\text{RightComp}(f)}$ and if $\pi_1 f = \text{N-min}\,\widetilde{\mathcal{L}}(f)$ and f is clockwise oriented, then $(\text{proj1})^\circ\overline{\text{RightComp}(f)} = (\text{proj1})^\circ\widetilde{\mathcal{L}}(f)$.

(40) $(\text{proj2})^\circ\widetilde{\mathcal{L}}(f) \subseteq (\text{proj2})^\circ\overline{\text{RightComp}(f)}$ and if $\pi_1 f = \text{N-min}\,\widetilde{\mathcal{L}}(f)$ and f is clockwise oriented, then $(\text{proj2})^\circ\overline{\text{RightComp}(f)} = (\text{proj2})^\circ\widetilde{\mathcal{L}}(f)$.

(41) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$, then $\text{RightComp}(g) \subseteq \text{RightComp}(\text{SpStSeq}\,\widetilde{\mathcal{L}}(g))$.

(42) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$, then $\overline{\text{RightComp}(g)}$ is compact.

(43) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$, then $\text{LeftComp}(g)$ is non Bounded.

(44) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$, then $\text{LeftComp}(g)$ is outside component of $\widetilde{\mathcal{L}}(g)$.

(45) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$, then $\text{RightComp}(g)$ is inside component of $\widetilde{\mathcal{L}}(g)$.

(46) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$, then $\text{UBD}\,\widetilde{\mathcal{L}}(g) = \text{LeftComp}(g)$.

(47) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$, then $\text{BDD}\,\widetilde{\mathcal{L}}(g) = \text{RightComp}(g)$.

(48) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$ and P is outside component of $\widetilde{\mathcal{L}}(g)$, then $P = \text{LeftComp}(g)$.

(49) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$ and P is inside component of $\widetilde{\mathcal{L}}(g)$, then $P \cap \text{RightComp}(g) \neq \emptyset$.

(50) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$ and P is inside component of $\widetilde{\mathcal{L}}(g)$, then $P = \text{BDD}\,\widetilde{\mathcal{L}}(g)$.

(51) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$, then $\text{W-bound}\,\widetilde{\mathcal{L}}(g) = \text{W-bound}\,\text{RightComp}(g)$.

(52) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$, then $\text{E-bound}\,\widetilde{\mathcal{L}}(g) = \text{E-bound}\,\text{RightComp}(g)$.

(53) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$, then $\text{N-bound}\,\widetilde{\mathcal{L}}(g) = \text{N-bound}\,\text{RightComp}(g)$.

(54) If $\pi_1 g = \text{N-min}\,\widetilde{\mathcal{L}}(g)$, then $\text{S-bound}\,\widetilde{\mathcal{L}}(g) = \text{S-bound}\,\text{RightComp}(g)$.

Acknowledgments

I would like to thank Professor Yatsuka Nakamura for his help in the preparation of the article.

References

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

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

[3] Grzegorz Bancerek. Countable sets and Hessenberg's theorem. *Formalized Mathematics*, 2(**1**):65–69, 1991.
[4] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[5] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(**1**):175–180, 1990.
[6] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[7] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[8] Czesław Byliński. Products and coproducts in categories. *Formalized Mathematics*, 2(**5**):701–709, 1991.
[9] Czesław Byliński. Gauges. *Formalized Mathematics*, 8(**1**):25–27, 1999.
[10] Czesław Byliński and Piotr Rudnicki. Bounding boxes for compact sets in $\mathcal{E}^2$. *Formalized Mathematics*, 6(**3**):427–440, 1997.
[11] Agata Darmochwał. Compact spaces. *Formalized Mathematics*, 1(**2**):383–386, 1990.
[12] Agata Darmochwał. Families of subsets, subspaces and mappings in topological spaces. *Formalized Mathematics*, 1(**2**):257–261, 1990.
[13] Agata Darmochwał. The Euclidean space. *Formalized Mathematics*, 2(**4**):599–603, 1991.
[14] Agata Darmochwał and Yatsuka Nakamura. The topological space $\mathcal{E}^2_{\rm T}$. Arcs, line segments and special polygonal arcs. *Formalized Mathematics*, 2(**5**):617–621, 1991.
[15] Alicia de la Cruz. Totally bounded metric spaces. *Formalized Mathematics*, 2(**4**):559–562, 1991.
[16] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[17] Katarzyna Jankowska. Matrices. Abelian group of matrices. *Formalized Mathematics*, 2(**4**):475–480, 1991.
[18] Stanisława Kanas, Adam Lecko, and Mariusz Startek. Metric spaces. *Formalized Mathematics*, 1(**3**):607–610, 1990.
[19] Jarosław Kotowicz. Convergent real sequences. Upper and lower bound of sets of real numbers. *Formalized Mathematics*, 1(**3**):477–481, 1990.
[20] Jarosław Kotowicz. The limit of a real function at infinity. *Formalized Mathematics*, 2(**1**):17–28, 1991.
[21] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part I. *Formalized Mathematics*, 3(**1**):107–115, 1992.
[22] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part II. *Formalized Mathematics*, 3(**1**):117–121, 1992.
[23] Yatsuka Nakamura. Graph theoretical properties of arcs in the plane and Fashoda Meet Theorem. *Formalized Mathematics*, 7(**2**):193–201, 1998.
[24] Yatsuka Nakamura and Czesław Byliński. Extremal properties of vertices on special polygons. Part I. *Formalized Mathematics*, 5(**1**):97–102, 1996.
[25] Yatsuka Nakamura and Andrzej Trybulec. Decomposing a Go-board into cells. *Formalized Mathematics*, 5(**3**):323–328, 1996.
[26] Yatsuka Nakamura, Andrzej Trybulec, and Czesław Byliński. Bounded domains and unbounded domains. *Formalized Mathematics*, 8(**1**):1–13, 1999.
[27] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[28] Beata Padlewska. Connected spaces. *Formalized Mathematics*, 1(**1**):239–244, 1990.
[29] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[30] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.
[31] Konrad Raczkowski and Andrzej Nędzusiak. Real exponents and logarithms. *Formalized Mathematics*, 2(**2**):213–216, 1991.
[32] Konrad Raczkowski and Paweł Sadowski. Topological properties of subsets in real numbers. *Formalized Mathematics*, 1(**4**):777–780, 1990.
[33] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[34] Andrzej Trybulec. Left and right component of the complement of a special closed curve. *Formalized Mathematics*, 5(**4**):465–468, 1996.

[35] Andrzej Trybulec and Czesław Byliński. Some properties of real numbers. *Formalized Mathematics*, 1(**3**):445–449, 1990.

[36] Andrzej Trybulec and Yatsuka Nakamura. On the order on a special polygon. *Formalized Mathematics*, 6(**4**):541–548, 1997.

[37] Andrzej Trybulec and Yatsuka Nakamura. On the rectangular finite sequences of the points of the plane. *Formalized Mathematics*, 6(**4**):531–539, 1997.

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

[39] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.

[40] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.

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

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

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

[44] Mirosław Wysocki and Agata Darmochwał. Subsets of topological spaces. *Formalized Mathematics*, 1(**1**):231–237, 1990.

Received May 5, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Noetherian Lattices

Christoph Schwarzweller
University of Tuebingen

Summary. In this article we define noetherian and co-noetherian lattices and show how some properties concerning upper and lower neighbours, irreducibility and density can be improved when restricted to these kinds of lattices. In addition we define atomic lattices.

MML Identifier: `LATTICE6`.

The notation and terminology used here are introduced in the following papers: [18], [13], [17], [14], [19], [7], [1], [8], [6], [20], [3], [9], [2], [10], [15], [16], [5], [11], [4], and [12].

Let us observe that there exists a lattice which is finite.

Let us mention that every lattice which is finite is also complete.

Let L be a lattice and let D be a subset of the carrier of L. The functor $D^{\cdot}$ yields a subset of $\mathrm{Poset}(L)$ and is defined by:

(Def. 1) $D^{\cdot} = \{d^{\cdot}; d$ ranges over elements of the carrier of L: $d \in D\}$.

Let L be a lattice and let D be a subset of the carrier of $\mathrm{Poset}(L)$. The functor ${}^{\cdot}D$ yielding a subset of the carrier of L is defined by:

(Def. 2) ${}^{\cdot}D = \{{}^{\cdot}d; d$ ranges over elements of $\mathrm{Poset}(L)$: $d \in D\}$.

Let L be a finite lattice. Note that $\mathrm{Poset}(L)$ is well founded.

Let L be a lattice. We say that L is noetherian if and only if:

(Def. 3) $\mathrm{Poset}(L)$ is well founded.

We say that L is co-noetherian if and only if:

(Def. 4) $\mathrm{Poset}(L)^{\smile}$ is well founded.

One can verify the following observations:

* there exists a lattice which is noetherian and upper-bounded,
* there exists a lattice which is noetherian and lower-bounded, and
* there exists a lattice which is noetherian and complete.

ISSN 1426–2630

One can verify the following observations:

* there exists a lattice which is co-noetherian and upper-bounded,
* there exists a lattice which is co-noetherian and lower-bounded, and
* there exists a lattice which is co-noetherian and complete.

Next we state the proposition

(1) For every lattice L holds L is noetherian iff L° is co-noetherian.

One can check that every lattice which is finite is also noetherian and every lattice which is finite is also co-noetherian.

Let L be a lattice and let a, b be elements of the carrier of L. We say that a is-upper-neighbour-of b if and only if:

(Def. 5) $a \neq b$ and $b \sqsubseteq a$ and for every element c of the carrier of L such that $b \sqsubseteq c$ and $c \sqsubseteq a$ holds $c = a$ or $c = b$.

We introduce b is-lower-neighbour-of a as a synonym of a is-upper-neighbour-of b.

We now state several propositions:

(2) Let L be a lattice, a be an element of the carrier of L, and b, c be elements of the carrier of L such that $b \neq c$. Then

(i) if b is-upper-neighbour-of a and c is-upper-neighbour-of a, then $a = c \sqcap b$, and

(ii) if b is-lower-neighbour-of a and c is-lower-neighbour-of a, then $a = c \sqcup b$.

(3) Let L be a noetherian lattice, a be an element of the carrier of L, and d be an element of the carrier of L. Suppose $a \sqsubseteq d$ and $a \neq d$. Then there exists an element c of the carrier of L such that $c \sqsubseteq d$ and c is-upper-neighbour-of a.

(4) Let L be a co-noetherian lattice, a be an element of the carrier of L, and d be an element of the carrier of L. Suppose $d \sqsubseteq a$ and $a \neq d$. Then there exists an element c of the carrier of L such that $d \sqsubseteq c$ and c is-lower-neighbour-of a.

(5) Let L be an upper-bounded lattice. Then it is not true that there exists an element b of the carrier of L such that b is-upper-neighbour-of $\top_L$.

(6) Let L be a noetherian upper-bounded lattice and a be an element of the carrier of L. Then $a = \top_L$ if and only if it is not true that there exists an element b of the carrier of L such that b is-upper-neighbour-of a.

(7) Let L be a lower-bounded lattice. Then it is not true that there exists an element b of the carrier of L such that b is-lower-neighbour-of $\bot_L$.

(8) Let L be a co-noetherian lower-bounded lattice and a be an element of the carrier of L. Then $a = \bot_L$ if and only if it is not true that there exists an element b of the carrier of L such that b is-lower-neighbour-of a.

Let L be a complete lattice and let a be an element of the carrier of L. The functor a^* yielding an element of the carrier of L is defined by:

(Def. 6) $a^* = \bigsqcap_L \{d; d$ ranges over elements of the carrier of L: $a \sqsubseteq d \wedge d \neq a\}$.

The functor ${}^*a$ yields an element of the carrier of L and is defined as follows:

(Def. 7) ${}^*a = \bigsqcup_L \{d; d$ ranges over elements of the carrier of L: $d \sqsubseteq a \wedge d \neq a\}$.

Let L be a complete lattice and let a be an element of the carrier of L. We say that a is completely-meet-irreducible if and only if:

(Def. 8) $a^* \neq a$.

We say that a is completely-join-irreducible if and only if:

(Def. 9) ${}^*a \neq a$.

The following propositions are true:

(9) For every complete lattice L and for every element a of the carrier of L holds $a \sqsubseteq a^*$ and ${}^*a \sqsubseteq a$.

(10) For every complete lattice L holds $(\top_L)^* = \top_L$ and $(\top_L)^{\cdot}$ is meet-irreducible.

(11) For every complete lattice L holds ${}^*(\bot_L) = \bot_L$ and $(\bot_L)^{\cdot}$ is join-irreducible.

(12) Let L be a complete lattice and a be an element of the carrier of L. Suppose a is completely-meet-irreducible. Then

(i) a^* is-upper-neighbour-of a, and

(ii) for every element c of the carrier of L such that c is-upper-neighbour-of a holds $c = a^*$.

(13) Let L be a complete lattice and a be an element of the carrier of L. Suppose a is completely-join-irreducible. Then

(i) ${}^*a$ is-lower-neighbour-of a, and

(ii) for every element c of the carrier of L such that c is-lower-neighbour-of a holds $c = {}^*a$.

(14) Let L be a noetherian complete lattice and a be an element of the carrier of L. Suppose $a \neq \top_L$. Then a is completely-meet-irreducible if and only if there exists an element b of the carrier of L such that b is-upper-neighbour-of a and for every element c of the carrier of L such that c is-upper-neighbour-of a holds $c = b$.

(15) Let L be a co-noetherian complete lattice and a be an element of the carrier of L. Suppose $a \neq \bot_L$. Then a is completely-join-irreducible if and only if there exists an element b of the carrier of L such that b is-lower-neighbour-of a and for every element c of the carrier of L such that c is-lower-neighbour-of a holds $c = b$.

(16) Let L be a complete lattice and a be an element of the carrier of L. If a is completely-meet-irreducible, then $a^{\cdot}$ is meet-irreducible.

(17) Let L be a complete noetherian lattice and a be an element of the carrier of L. Suppose $a \neq \top_L$. Then a is completely-meet-irreducible if and only if $a^{\cdot}$ is meet-irreducible.

(18) Let L be a complete lattice and a be an element of the carrier of L. If a is completely-join-irreducible, then $a^{\cdot}$ is join-irreducible.

(19) Let L be a complete co-noetherian lattice and a be an element of the carrier of L. Suppose $a \neq \bot_L$. Then a is completely-join-irreducible if and only if $a^{\cdot}$ is join-irreducible.

(20) Let L be a finite lattice and a be an element of the carrier of L such that $a \neq \bot_L$ and $a \neq \top_L$. Then

(i) a is completely-meet-irreducible iff $a^{\cdot}$ is meet-irreducible, and

(ii) a is completely-join-irreducible iff $a^{\cdot}$ is join-irreducible.

Let L be a lattice and let a be an element of the carrier of L. We say that a is atomic if and only if:

(Def. 10) a is-upper-neighbour-of $\bot_L$.

We say that a is co-atomic if and only if:

(Def. 11) a is-lower-neighbour-of $\top_L$.

One can prove the following propositions:

(21) Let L be a complete lattice and a be an element of the carrier of L. If a is atomic, then a is completely-join-irreducible.

(22) Let L be a complete lattice and a be an element of the carrier of L. If a is co-atomic, then a is completely-meet-irreducible.

Let L be a lattice. We say that L is atomic if and only if the condition (Def. 12) is satisfied.

(Def. 12) Let a be an element of the carrier of L. Then there exists a subset X of the carrier of L such that for every element x of the carrier of L such that $x \in X$ holds x is atomic and $a = \bigsqcup_L X$.

One can verify that there exists a lattice which is atomic and complete.

Let L be a complete lattice and let D be a subset of L. We say that D is supremum-dense if and only if:

(Def. 13) For every element a of the carrier of L there exists a subset D' of D such that $a = \bigsqcup_L D'$.

We say that D is infimum-dense if and only if:

(Def. 14) For every element a of the carrier of L there exists a subset D' of D such that $a = \bigsqcap_L D'$.

One can prove the following propositions:

(23) Let L be a complete lattice and D be a subset of L. Then D is supremum-dense if and only if for every element a of the carrier of L holds $a = \bigsqcup_L \{d; d$ ranges over elements of the carrier of L: $d \in D \ \wedge \ d \sqsubseteq a\}$.

(24) Let L be a complete lattice and D be a subset of L. Then D is infimum-dense if and only if for every element a of the carrier of L holds $a = \bigsqcap_L\{d; d$ ranges over elements of the carrier of L: $d \in D \ \wedge \ a \sqsubseteq d\}$.

(25) Let L be a complete lattice and D be a subset of L. Then D is infimum-dense if and only if $D^{\cdot}$ is order-generating.

Let L be a complete lattice. The functor MIRRS L yields a subset of L and is defined by:

(Def. 15) MIRRS $L = \{a; a$ ranges over elements of the carrier of L: a is completely-meet-irreducible$\}$.

The functor JIRRS L yielding a subset of L is defined by:

(Def. 16) JIRRS $L = \{a; a$ ranges over elements of the carrier of L: a is completely-join-irreducible$\}$.

One can prove the following two propositions:

(26) For every complete lattice L and for every subset D of L such that D is supremum-dense holds JIRRS $L \subseteq D$.

(27) For every complete lattice L and for every subset D of L such that D is infimum-dense holds MIRRS $L \subseteq D$.

Let L be a co-noetherian complete lattice. Note that MIRRS L is infimum-dense.

Let L be a noetherian complete lattice. One can check that JIRRS L is supremum-dense.

References

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

[2] Grzegorz Bancerek. The well ordering relations. *Formalized Mathematics*, 1(**1**):123–129, 1990.

[3] Grzegorz Bancerek. Complete lattices. *Formalized Mathematics*, 2(**5**):719–725, 1991.

[4] Grzegorz Bancerek. Bounds in posets and relational substructures. *Formalized Mathematics*, 6(**1**):81–91, 1997.

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

[6] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(**1**):175–180, 1990.

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

[8] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.

[9] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.

[10] Beata Madras. Irreducible and prime elements. *Formalized Mathematics*, 6(**2**):233–239, 1997.

[11] Piotr Rudnicki and Andrzej Trybulec. On same equivalents of well-foundedness. *Formalized Mathematics*, 6(**3**):339–343, 1997.

[12] Andrzej Trybulec. Finite join and finite meet and dual lattices. *Formalized Mathematics*, 1(**5**):983–988, 1990.

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

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

[15] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.
[16] Wojciech A. Trybulec. Partially ordered sets. *Formalized Mathematics*, 1(**2**):313–319, 1990.
[17] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[18] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[19] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[20] Stanisław Żukowski. Introduction to lattice theory. *Formalized Mathematics*, 1(**1**):215–222, 1990.

Received June 9, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

A Small Computer Model with Push-Down Stack[1]

Jing-Chao Chen
Shanghai Jiaotong University

Summary. The SCMFSA computer can prove the correctness of many algorithms. Unfortunately, it cannot prove the correctness of recursive algorithms. For this reason, this article improves the SCMFSA computer and presents a Small Computer Model with Push-Down Stack (called SCMPDS for short). In addition to conventional arithmetic and "goto" instructions, we increase two new instructions such as "return" and "save instruction-counter" in order to be able to design recursive programs.

MML Identifier: `SCMPDS_1`.

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

1. Preliminaries

For simplicity, we follow the rules: x_1, x_2, x_3, x_4, x_5 are sets, i, j, k are natural numbers, I, I_2, I_3, I_4 are elements of $\mathbb{Z}_{14}$, i_1 is an element of Instr-Loc$_{\rm SCM}$, d_1, d_2, d_3, d_4, d_5 are elements of Data-Loc$_{\rm SCM}$, and k_1, k_2, k_3, k_4, k_5, k_6 are integers.

Let x_1, x_2, x_3, x_4 be sets. The functor $<*x_1, x_2, x_3, x_4*>$ yields a set and is defined as follows:

(Def. 1) $<*x_1, x_2, x_3, x_4*> = \langle x_1, x_2, x_3 \rangle \frown \langle x_4 \rangle$.

Let x_5 be a set. The functor $<*x_1, x_2, x_3, x_4, x_5*>$ yielding a set is defined by:

(Def. 2) $<*x_1, x_2, x_3, x_4, x_5*> = \langle x_1, x_2, x_3 \rangle \frown \langle x_4, x_5 \rangle$.

[1]This work was done while the author visited Shinshu University March–April 1999.

ISSN 1426–2630

Let x_1, x_2, x_3, x_4 be sets. One can verify that $< *x_1, x_2, x_3, x_4* >$ is function-like and relation-like. Let x_5 be a set. One can verify that $< *x_1, x_2, x_3, x_4, x_5* >$ is function-like and relation-like.

Let x_1, x_2, x_3, x_4 be sets. One can verify that $< *x_1, x_2, x_3, x_4* >$ is finite sequence-like. Let x_5 be a set. One can check that $< *x_1, x_2, x_3, x_4, x_5* >$ is finite sequence-like.

Let D be a non empty set and let x_1, x_2, x_3, x_4 be elements of D. Then $< *x_1, x_2, x_3, x_4* >$ is a finite sequence of elements of D.

Let D be a non empty set and let x_1, x_2, x_3, x_4, x_5 be elements of D. Then $< *x_1, x_2, x_3, x_4, x_5* >$ is a finite sequence of elements of D.

One can prove the following propositions:

(1) $< *x_1, x_2, x_3, x_4* >= \langle x_1, x_2, x_3 \rangle ^\frown \langle x_4 \rangle$ and $< *x_1, x_2, x_3, x_4* >= \langle x_1, x_2 \rangle ^\frown \langle x_3, x_4 \rangle$ and $< *x_1, x_2, x_3, x_4* >= \langle x_1 \rangle ^\frown \langle x_2, x_3, x_4 \rangle$ and $< *x_1, x_2, x_3, x_4* >= \langle x_1 \rangle ^\frown \langle x_2 \rangle ^\frown \langle x_3 \rangle ^\frown \langle x_4 \rangle$.

(2) $< *x_1, x_2, x_3, x_4, x_5* >= \langle x_1, x_2, x_3 \rangle ^\frown \langle x_4, x_5 \rangle$ and $< *x_1, x_2, x_3, x_4, x_5* > =< *x_1, x_2, x_3, x_4* > ^\frown \langle x_5 \rangle$ and $< *x_1, x_2, x_3, x_4, x_5* >= \langle x_1 \rangle ^\frown \langle x_2 \rangle ^\frown \langle x_3 \rangle ^\frown \langle x_4 \rangle ^\frown \langle x_5 \rangle$ and $< *x_1, x_2, x_3, x_4, x_5* >= \langle x_1, x_2 \rangle ^\frown \langle x_3, x_4, x_5 \rangle$ and $< *x_1, x_2, x_3, x_4, x_5* >= \langle x_1 \rangle ^\frown < *x_2, x_3, x_4, x_5* >$.

We adopt the following rules: N_1 is a non empty set, y_1, y_2, y_3, y_4, y_5 are elements of N_1, and p is a finite sequence.

We now state several propositions:

(3) $p =< *x_1, x_2, x_3, x_4* >$ iff $\operatorname{len} p = 4$ and $p(1) = x_1$ and $p(2) = x_2$ and $p(3) = x_3$ and $p(4) = x_4$.

(4) $\operatorname{dom} < *x_1, x_2, x_3, x_4* >= \operatorname{Seg} 4$.

(5) $p =< *x_1, x_2, x_3, x_4, x_5* >$ iff $\operatorname{len} p = 5$ and $p(1) = x_1$ and $p(2) = x_2$ and $p(3) = x_3$ and $p(4) = x_4$ and $p(5) = x_5$.

(6) $\operatorname{dom} < *x_1, x_2, x_3, x_4, x_5* >= \operatorname{Seg} 5$.

(7) $\pi_1 < *y_1, y_2, y_3, y_4* >= y_1$ and $\pi_2 < *y_1, y_2, y_3, y_4* >= y_2$ and $\pi_3 < *y_1, y_2, y_3, y_4* >= y_3$ and $\pi_4 < *y_1, y_2, y_3, y_4* >= y_4$.

(8) $\pi_1 < *y_1, y_2, y_3, y_4, y_5* >= y_1$ and $\pi_2 < *y_1, y_2, y_3, y_4, y_5* >= y_2$ and $\pi_3 < *y_1, y_2, y_3, y_4, y_5* >= y_3$ and $\pi_4 < *y_1, y_2, y_3, y_4, y_5* >= y_4$ and $\pi_5 < *y_1, y_2, y_3, y_4, y_5* >= y_5$.

(9) For every integer k holds $k \in \bigcup\{\mathbb{Z}\} \cup \mathbb{N}$.

(10) For every integer k holds $k \in \text{Data-Loc}_{\text{SCM}} \cup \mathbb{Z}$.

(11) For every element d of $\text{Data-Loc}_{\text{SCM}}$ holds $d \in \text{Data-Loc}_{\text{SCM}} \cup \mathbb{Z}$.

2. The Construction of SCM with Push-Down Stack

The subset SCMPDS − Instr of $[: \mathbb{Z}_{14}, (\bigcup\{\mathbb{Z}\} \cup \mathbb{N})^* :]$ is defined by the condition (Def. 3).

(Def. 3) $\mathrm{SCMPDS}-\mathrm{Instr} = \{\langle 0, \langle l \rangle\rangle : l$ ranges over integers$\} \cup \{\langle 1, \langle s_1 \rangle\rangle : s_1$ ranges over elements of $\text{Data-Loc}_{\mathrm{SCM}}\} \cup \{\langle I, \langle v, c\rangle\rangle; I$ ranges over elements of $\mathbb{Z}_{14}$, v ranges over elements of $\text{Data-Loc}_{\mathrm{SCM}}$, c ranges over integers: $I \in \{2, 3\}\} \cup \{\langle I, \langle v, c_1, c_2\rangle\rangle; I$ ranges over elements of $\mathbb{Z}_{14}$, v ranges over elements of $\text{Data-Loc}_{\mathrm{SCM}}$, c_1 ranges over integers, c_2 ranges over integers: $I \in \{4, 5, 6, 7, 8\}\} \cup \{\langle I, < *v_1, v_2, c_1, c_2* > \rangle; I$ ranges over elements of $\mathbb{Z}_{14}$, v_1 ranges over elements of $\text{Data-Loc}_{\mathrm{SCM}}$, v_2 ranges over elements of $\text{Data-Loc}_{\mathrm{SCM}}$, c_1 ranges over integers, c_2 ranges over integers: $I \in \{9, 10, 11, 12, 13\}\}$.

We now state two propositions:

(12) $\mathrm{SCMPDS}-\mathrm{Instr} = \{\langle 0, \langle k_1\rangle\rangle\} \cup \{\langle 1, \langle d_1\rangle\rangle\} \cup \{\langle I_2, \langle d_2, k_2\rangle\rangle : I_2 \in \{2, 3\}\} \cup \{\langle I_3, \langle d_3, k_3, k_4\rangle\rangle : I_3 \in \{4, 5, 6, 7, 8\}\} \cup \{\langle I_4, < *d_4, d_5, k_5, k_6* > \rangle : I_4 \in \{9, 10, 11, 12, 13\}\}$.

(13) $\langle 0, \langle 0\rangle\rangle \in \mathrm{SCMPDS}-\mathrm{Instr}$.

One can verify that SCMPDS − Instr is non empty.

We now state three propositions:

(14) $k = 0$ or there exists j such that $k = 2 \cdot j + 1$ or there exists j such that $k = 2 \cdot j + 2$.

(15) If $k = 0$, then it is not true that there exists j such that $k = 2 \cdot j + 1$ and it is not true that there exists j such that $k = 2 \cdot j + 2$.

(16)(i) If there exists j such that $k = 2 \cdot j + 1$, then $k \neq 0$ and it is not true that there exists j such that $k = 2 \cdot j + 2$, and

(ii) if there exists j such that $k = 2 \cdot j + 2$, then $k \neq 0$ and it is not true that there exists j such that $k = 2 \cdot j + 1$.

The function SCMPDS − OK from $\mathbb{N}$ into $\{\mathbb{Z}\} \cup \{\mathrm{SCMPDS}-\mathrm{Instr}, \text{Instr-Loc}_{\mathrm{SCM}}\}$ is defined as follows:

(Def. 4) $(\mathrm{SCMPDS}-\mathrm{OK})(0) = \text{Instr-Loc}_{\mathrm{SCM}}$ and for every natural number k holds $(\mathrm{SCMPDS}-\mathrm{OK})(2 \cdot k + 1) = \mathbb{Z}$ and $(\mathrm{SCMPDS}-\mathrm{OK})(2 \cdot k + 2) = \mathrm{SCMPDS}-\mathrm{Instr}$.

A SCMPDS-State is an element of $\prod \mathrm{SCMPDS}-\mathrm{OK}$.

Next we state several propositions:

(17) $\text{Instr-Loc}_{\mathrm{SCM}} \neq \mathrm{SCMPDS}-\mathrm{Instr}$ and $\mathrm{SCMPDS}-\mathrm{Instr} \neq \mathbb{Z}$.

(18) $(\mathrm{SCMPDS}-\mathrm{OK})(i) = \text{Instr-Loc}_{\mathrm{SCM}}$ iff $i = 0$.

(19) $(\mathrm{SCMPDS}-\mathrm{OK})(i) = \mathbb{Z}$ iff there exists k such that $i = 2 \cdot k + 1$.

(20) $(\mathrm{SCMPDS}-\mathrm{OK})(i) = \mathrm{SCMPDS}-\mathrm{Instr}$ iff there exists k such that $i = 2 \cdot k + 2$.

(21) $(\mathrm{SCMPDS}-\mathrm{OK})(d_1) = \mathbb{Z}$.

(22) $(\mathrm{SCMPDS}-\mathrm{OK})(i_1) = \mathrm{SCMPDS}-\mathrm{Instr}$.

(23) $\pi_0 \prod \mathrm{SCMPDS}-\mathrm{OK} = \text{Instr-Loc}_{\mathrm{SCM}}$.

(24) $\pi_{d_1} \prod \mathrm{SCMPDS} - \mathrm{OK} = \mathbb{Z}$.

(25) $\pi_{i_1} \prod \mathrm{SCMPDS} - \mathrm{OK} = \mathrm{SCMPDS} - \mathrm{Instr}$.

Let s be a SCMPDS-State. The functor $\mathbf{IC}_s$ yielding an element of Instr-Loc$_{\mathrm{SCM}}$ is defined as follows:

(Def. 5) $\mathbf{IC}_s = s(0)$.

Let s be a SCMPDS-State and let u be an element of Instr-Loc$_{\mathrm{SCM}}$. The functor $\mathrm{Chg}_{\mathrm{SCM}}(s, u)$ yielding a SCMPDS-State is defined as follows:

(Def. 6) $\mathrm{Chg}_{\mathrm{SCM}}(s, u) = s{+}\cdot(0 \dot{\longmapsto} u)$.

We now state three propositions:

(26) For every SCMPDS-State s and for every element u of Instr-Loc$_{\mathrm{SCM}}$ holds $(\mathrm{Chg}_{\mathrm{SCM}}(s, u))(0) = u$.

(27) For every SCMPDS-State s and for every element u of Instr-Loc$_{\mathrm{SCM}}$ and for every element m_1 of Data-Loc$_{\mathrm{SCM}}$ holds $(\mathrm{Chg}_{\mathrm{SCM}}(s, u))(m_1) = s(m_1)$.

(28) For every SCMPDS-State s and for all elements u, v of Instr-Loc$_{\mathrm{SCM}}$ holds $(\mathrm{Chg}_{\mathrm{SCM}}(s, u))(v) = s(v)$.

Let s be a SCMPDS-State, let t be an element of Data-Loc$_{\mathrm{SCM}}$, and let u be an integer. The functor $\mathrm{Chg}_{\mathrm{SCM}}(s, t, u)$ yields a SCMPDS-State and is defined as follows:

(Def. 7) $\mathrm{Chg}_{\mathrm{SCM}}(s, t, u) = s{+}\cdot(t \dot{\longmapsto} u)$.

The following propositions are true:

(29) For every SCMPDS-State s and for every element t of Data-Loc$_{\mathrm{SCM}}$ and for every integer u holds $(\mathrm{Chg}_{\mathrm{SCM}}(s, t, u))(0) = s(0)$.

(30) For every SCMPDS-State s and for every element t of Data-Loc$_{\mathrm{SCM}}$ and for every integer u holds $(\mathrm{Chg}_{\mathrm{SCM}}(s, t, u))(t) = u$.

(31) Let s be a SCMPDS-State, t be an element of Data-Loc$_{\mathrm{SCM}}$, u be an integer, and m_1 be an element of Data-Loc$_{\mathrm{SCM}}$. If $m_1 \neq t$, then $(\mathrm{Chg}_{\mathrm{SCM}}(s, t, u))(m_1) = s(m_1)$.

(32) Let s be a SCMPDS-State, t be an element of Data-Loc$_{\mathrm{SCM}}$, u be an integer, and v be an element of Instr-Loc$_{\mathrm{SCM}}$. Then $(\mathrm{Chg}_{\mathrm{SCM}}(s, t, u))(v) = s(v)$.

Let s be a SCMPDS-State and let a be an element of Data-Loc$_{\mathrm{SCM}}$. Then $s(a)$ is an integer.

Let s be a SCMPDS-State, let a be an element of Data-Loc$_{\mathrm{SCM}}$, and let n be an integer. The functor Address_Add(s, a, n) yields an element of Data-Loc$_{\mathrm{SCM}}$ and is defined by:

(Def. 8) $\mathrm{Address\_Add}(s, a, n) = 2 \cdot |s(a) + n| + 1$.

Let s be a SCMPDS-State and let n be an integer. The functor jump_address(s, n) yielding an element of Instr-Loc$_{\mathrm{SCM}}$ is defined as follows:

(Def. 9) $\mathrm{jump\_address}(s, n) = |((\mathbf{IC}_s \ \mathbf{qua} \ \text{natural number}) - 2) + 2 \cdot n| + 2$.

Let d be an element of $\text{Data-Loc}_{\text{SCM}}$ and let s be an integer. Then $\langle d, s\rangle$ is a finite sequence of elements of $\text{Data-Loc}_{\text{SCM}} \cup \mathbb{Z}$.

Let x be an element of SCMPDS – Instr. Let us assume that there exist an element m_1 of $\text{Data-Loc}_{\text{SCM}}$ and I such that $x = \langle I, \langle m_1\rangle\rangle$. The functor $x\,\text{address}_1$ yielding an element of $\text{Data-Loc}_{\text{SCM}}$ is defined as follows:

(Def. 10) There exists a finite sequence f of elements of $\text{Data-Loc}_{\text{SCM}}$ such that $f = x_{\mathbf{2}}$ and $x\,\text{address}_1 = \pi_1 f$.

The following proposition is true

(33) For every element x of SCMPDS – Instr and for every element m_1 of $\text{Data-Loc}_{\text{SCM}}$ such that $x = \langle I, \langle m_1\rangle\rangle$ holds $x\,\text{address}_1 = m_1$.

Let x be an element of SCMPDS – Instr. Let us assume that there exist an integer r and I such that $x = \langle I, \langle r\rangle\rangle$. The functor $x\,\text{const\_INT}$ yielding an integer is defined by:

(Def. 11) There exists a finite sequence f of elements of $\mathbb{Z}$ such that $f = x_{\mathbf{2}}$ and $x\,\text{const\_INT} = \pi_1 f$.

The following proposition is true

(34) For every element x of SCMPDS – Instr and for every integer k such that $x = \langle I, \langle k\rangle\rangle$ holds $x\,\text{const\_INT} = k$.

Let x be an element of SCMPDS – Instr. Let us assume that there exist an element m_1 of $\text{Data-Loc}_{\text{SCM}}$, an integer r, and I such that $x = \langle I, \langle m_1, r\rangle\rangle$. The functor $x\,\text{P21address}$ yielding an element of $\text{Data-Loc}_{\text{SCM}}$ is defined as follows:

(Def. 12) There exists a finite sequence f of elements of $\text{Data-Loc}_{\text{SCM}} \cup \mathbb{Z}$ such that $f = x_{\mathbf{2}}$ and $x\,\text{P21address} = \pi_1 f$.

The functor $x\,\text{P22const}$ yielding an integer is defined as follows:

(Def. 13) There exists a finite sequence f of elements of $\text{Data-Loc}_{\text{SCM}} \cup \mathbb{Z}$ such that $f = x_{\mathbf{2}}$ and $x\,\text{P22const} = \pi_2 f$.

The following proposition is true

(35) Let x be an element of SCMPDS – Instr, m_1 be an element of $\text{Data-Loc}_{\text{SCM}}$, and r be an integer. If $x = \langle I, \langle m_1, r\rangle\rangle$, then $x\,\text{P21address} = m_1$ and $x\,\text{P22const} = r$.

Let x be an element of SCMPDS – Instr. Let us assume that there exist an element m_2 of $\text{Data-Loc}_{\text{SCM}}$, integers k_1, k_2, and I such that $x = \langle I, \langle m_2, k_1, k_2\rangle\rangle$. The functor $x\,\text{P31address}$ yielding an element of $\text{Data-Loc}_{\text{SCM}}$ is defined as follows:

(Def. 14) There exists a finite sequence f of elements of $\text{Data-Loc}_{\text{SCM}} \cup \mathbb{Z}$ such that $f = x_{\mathbf{2}}$ and $x\,\text{P31address} = \pi_1 f$.

The functor $x\,\text{P32const}$ yielding an integer is defined as follows:

(Def. 15) There exists a finite sequence f of elements of $\text{Data-Loc}_{\text{SCM}} \cup \mathbb{Z}$ such that $f = x_{\mathbf{2}}$ and $x\,\text{P32const} = \pi_2 f$.

The functor $x\,$P33const yields an integer and is defined by:

(Def. 16) There exists a finite sequence f of elements of $\text{Data-Loc}_{\text{SCM}} \cup \mathbb{Z}$ such that $f = x_{\mathbf{2}}$ and $x\,\text{P33const} = \pi_3 f$.

We now state the proposition

(36) Let x be an element of SCMPDS − Instr, d_1 be an element of $\text{Data-Loc}_{\text{SCM}}$, and k_1, k_2 be integers. If $x = \langle I, \langle d_1, k_1, k_2 \rangle \rangle$, then $x\,\text{P31address} = d_1$ and $x\,\text{P32const} = k_1$ and $x\,\text{P33const} = k_2$.

Let x be an element of SCMPDS − Instr. Let us assume that there exist elements m_2, m_3 of $\text{Data-Loc}_{\text{SCM}}$, integers k_1, k_2, and I such that $x = \langle I, < *m_2, m_3, k_1, k_2* > \rangle$. The functor $x\,$P41address yields an element of $\text{Data-Loc}_{\text{SCM}}$ and is defined by:

(Def. 17) There exists a finite sequence f of elements of $\text{Data-Loc}_{\text{SCM}} \cup \mathbb{Z}$ such that $f = x_{\mathbf{2}}$ and $x\,\text{P41address} = \pi_1 f$.

The functor $x\,$P42address yields an element of $\text{Data-Loc}_{\text{SCM}}$ and is defined as follows:

(Def. 18) There exists a finite sequence f of elements of $\text{Data-Loc}_{\text{SCM}} \cup \mathbb{Z}$ such that $f = x_{\mathbf{2}}$ and $x\,\text{P42address} = \pi_2 f$.

The functor $x\,$P43const yielding an integer is defined as follows:

(Def. 19) There exists a finite sequence f of elements of $\text{Data-Loc}_{\text{SCM}} \cup \mathbb{Z}$ such that $f = x_{\mathbf{2}}$ and $x\,\text{P43const} = \pi_3 f$.

The functor $x\,$P44const yielding an integer is defined as follows:

(Def. 20) There exists a finite sequence f of elements of $\text{Data-Loc}_{\text{SCM}} \cup \mathbb{Z}$ such that $f = x_{\mathbf{2}}$ and $x\,\text{P44const} = \pi_4 f$.

We now state the proposition

(37) Let x be an element of SCMPDS − Instr, d_1, d_2 be elements of $\text{Data-Loc}_{\text{SCM}}$, and k_1, k_2 be integers. If $x = \langle I, < *d_1, d_2, k_1, k_2* > \rangle$, then $x\,\text{P41address} = d_1$ and $x\,\text{P42address} = d_2$ and $x\,\text{P43const} = k_1$ and $x\,\text{P44const} = k_2$.

Let s be a SCMPDS-State and let a be an element of $\text{Data-Loc}_{\text{SCM}}$. The functor $\text{PopInstrLoc}(s, a)$ yielding an element of $\text{Instr-Loc}_{\text{SCM}}$ is defined as follows:

(Def. 21) $\text{PopInstrLoc}(s, a) = 2 \cdot (|s(a)| \div 2) + 4$.

The natural number RetSP is defined as follows:

(Def. 22) $\text{RetSP} = 0$.

The natural number RetIC is defined as follows:

(Def. 23) $\text{RetIC} = 1$.

Let x be an element of SCMPDS − Instr and let s be a SCMPDS-State. The functor $\text{Exec-Res}_{\text{SCM}}(x, s)$ yielding a SCMPDS-State is defined as follows:

(Def. 24) $\text{Exec-Res}_{\text{SCM}}(x, s) =$

$$\begin{cases} \text{Chg}_{\text{SCM}}(s, \text{jump\_address}(s, x\,\text{const\_INT})), \text{ if there exists } k_1 \text{ such that } x = \langle 0, \langle k_1 \rangle \rangle, \\ \text{Chg}_{\text{SCM}}(\text{Chg}_{\text{SCM}}(s, x\,\text{P21address}, x\,\text{P22const}), \text{Next}(\mathbf{IC}_s)), \text{ if there exist } d_1, k_1 \text{ such that } x = \langle 2, \langle d_1, k_1 \rangle \rangle, \\ \text{Chg}_{\text{SCM}}(\text{Chg}_{\text{SCM}}(s, \text{Address\_Add}(s, x\,\text{P21address}, x\,\text{P22const}), (\mathbf{IC}_s \ \mathbf{qua} \text{ natural number})), \text{Next}(\mathbf{IC}_s)), \text{ if there exist } d_1, k_1 \text{ such that } x = \langle 3, \langle d_1, k_1 \rangle \rangle, \\ \text{Chg}_{\text{SCM}}(\text{Chg}_{\text{SCM}}(s, x\,\text{address}_1, s(\text{Address\_Add}(s, x\,\text{address}_1, \text{RetSP}))), \text{PopInstrLoc}(s, \text{Address\_Add}(s, x\,\text{address}_1, \text{RetIC}))), \text{ if there exists } d_1 \text{ such that } x = \langle 1, \langle d_1 \rangle \rangle, \\ \text{Chg}_{\text{SCM}}(s, (s(\text{Address\_Add}(s, x\,\text{P31address}, x\,\text{P32const})) = 0 \rightarrow \text{Next}(\mathbf{IC}_s), \text{jump\_address}(s, x\,\text{P33const}))), \text{ if there exist } d_1, k_1, k_2 \text{ such that } x = \langle 4, \langle d_1, k_1, k_2 \rangle \rangle, \\ \text{Chg}_{\text{SCM}}(s, (s(\text{Address\_Add}(s, x\,\text{P31address}, x\,\text{P32const})) > 0 \rightarrow \text{Next}(\mathbf{IC}_s), \text{jump\_address}(s, x\,\text{P33const}))), \text{ if there exist } d_1, k_1, k_2 \text{ such that } x = \langle 5, \langle d_1, k_1, k_2 \rangle \rangle, \\ \text{Chg}_{\text{SCM}}(s, (0 > s(\text{Address\_Add}(s, x\,\text{P31address}, x\,\text{P32const})) \rightarrow \text{Next}(\mathbf{IC}_s), \text{jump\_address}(s, x\,\text{P33const}))), \text{ if there exist } d_1, k_1, k_2 \text{ such that } x = \langle 6, \langle d_1, k_1, k_2 \rangle \rangle, \\ \text{Chg}_{\text{SCM}}(\text{Chg}_{\text{SCM}}(s, \text{Address\_Add}(s, x\,\text{P31address}, x\,\text{P32const}), x\,\text{P33const}), \text{Next}(\mathbf{IC}_s)), \text{ if there exist } d_1, k_1, k_2 \text{ such that } x = \langle 7, \langle d_1, k_1, k_2 \rangle \rangle, \\ \text{Chg}_{\text{SCM}}(\text{Chg}_{\text{SCM}}(s, \text{Address\_Add}(s, x\,\text{P31address}, x\,\text{P32const}), s(\text{Address\_Add}(s, x\,\text{P31address}, x\,\text{P32const})) + x\,\text{P33const}), \text{Next}(\mathbf{IC}_s)), \text{ if there exist } d_1, k_1, k_2 \text{ such that } x = \langle 8, \langle d_1, k_1, k_2 \rangle \rangle, \\ \text{Chg}_{\text{SCM}}(\text{Chg}_{\text{SCM}}(s, \text{Address\_Add}(s, x\,\text{P41address}, x\,\text{P43const}), s(\text{Address\_Add}(s, x\,\text{P41address}, x\,\text{P43const})) + s(\text{Address\_Add}(s, x\,\text{P42address}, x\,\text{P44const}))), \text{Next}(\mathbf{IC}_s)), \text{ if there exist } d_1, d_2, k_1, k_2 \text{ such that } x = \langle 9, <*d_1, d_2, k_1, k_2*> \rangle, \\ \text{Chg}_{\text{SCM}}(\text{Chg}_{\text{SCM}}(s, \text{Address\_Add}(s, x\,\text{P41address}, x\,\text{P43const}), s(\text{Address\_Add}(s, x\,\text{P41address}, x\,\text{P43const})) - s(\text{Address\_Add}(s, x\,\text{P42address}, x\,\text{P44const}))), \text{Next}(\mathbf{IC}_s)), \text{ if there exist } d_1, d_2, k_1, k_2 \text{ such that } x = \langle 10, <*d_1, d_2, k_1, k_2*> \rangle, \\ \text{Chg}_{\text{SCM}}(\text{Chg}_{\text{SCM}}(s, \text{Address\_Add}(s, x\,\text{P41address}, x\,\text{P43const}), s(\text{Address\_Add}(s, x\,\text{P41address}, x\,\text{P43const})) \cdot s(\text{Address\_Add}(s, x\,\text{P42address}, x\,\text{P44const}))), \text{Next}(\mathbf{IC}_s)), \text{ if there exist } d_1, d_2, k_1, k_2 \text{ such that } x = \langle 11, <*d_1, d_2, k_1, k_2*> \rangle, \\ \text{Chg}_{\text{SCM}}(\text{Chg}_{\text{SCM}}(s, \text{Address\_Add}(s, x\,\text{P41address}, x\,\text{P43const}), s(\text{Address\_Add}(s, x\,\text{P42address}, x\,\text{P44const}))), \text{Next}(\mathbf{IC}_s)), \text{ if there exist } d_1, d_2, k_1, k_2 \text{ such that } x = \langle 13, <*d_1, d_2, k_1, k_2*> \rangle, \\ \text{Chg}_{\text{SCM}}(\text{Chg}_{\text{SCM}}(\text{Chg}_{\text{SCM}}(s, \text{Address\_Add}(s, x\,\text{P41address}, x\,\text{P43const}), s(\text{Address\_Add}(s, x\,\text{P41address}, x\,\text{P43const})) \div s(\text{Address\_Add}(s, x\,\text{P42address}, x\,\text{P44const}))), \text{Address\_Add}(s, x\,\text{P42address}, x\,\text{P44const}), s(\text{Address\_Add}(s, x\,\text{P41address}, x\,\text{P43const})) \bmod s(\text{Address\_Add}(s, x\,\text{P42address}, x\,\text{P44const}))), \text{Next}(\mathbf{IC}_s)), \text{ if there exist } d_1, d_2, k_1, k_2 \text{ such that } x = \langle 12, <*d_1, d_2, k_1, k_2*> \rangle, \\ s, \text{ otherwise.} \end{cases}$$

Let f be a function from SCMPDS − Instr into $(\prod \text{SCMPDS} - \text{OK})^{\prod \text{SCMPDS}-\text{OK}}$ and let x be an element of SCMPDS − Instr. Note that $f(x)$ is function-like and relation-like.

The function SCMPDS − Exec from SCMPDS − Instr into

$(\prod \mathrm{SCMPDS} - \mathrm{OK})^{\Pi\,\mathrm{SCMPDS}-\mathrm{OK}}$ is defined by:

(Def. 25) For every element x of $\mathrm{SCMPDS} - \mathrm{Instr}$ and for every SCMPDS-State y holds $(\mathrm{SCMPDS} - \mathrm{Exec})(x)(y) = \text{Exec-Res}_{\mathrm{SCM}}(x, y)$.

Acknowledgments

We wish to thank Prof. Y. Nakamura for many helpful suggestions.

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[2] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[3] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[4] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.
[5] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[6] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[7] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[8] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[9] Czesław Byliński. Subcategories and products of categories. *Formalized Mathematics*, 1(**4**):725–732, 1990.
[10] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[11] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.
[12] Dariusz Surowik. Cyclic groups and some of their properties - part I. *Formalized Mathematics*, 2(**5**):623–627, 1991.
[13] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[14] Andrzej Trybulec. Function domains and Frænkel operator. *Formalized Mathematics*, 1(**3**):495–500, 1990.
[15] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[16] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[17] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[18] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.
[19] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 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 15, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

The SCMPDS Computer and the Basic Semantics of its Instructions[1]

Jing-Chao Chen
Shanghai Jiaotong University

Summary. The article defines the SCMPDS computer and its instructions. The SCMPDS computer consists of such instructions as conventional arithmetic, "goto", "return" and "save instruction-counter" ("saveIC" for short). The address used in the "goto" instruction is an offset value rather than a pointer in the standard sense. Thus, we don't define halting instruction directly but define it by "goto 0" instruction. The "saveIC" and "return" equal almost call and return statements in the usual high programming language. Theoretically, the SCMPDS computer can implement all algorithms described by the usual high programming language including recursive routine. In addition, we describe the execution semantics and halting properties of each instruction.

MML Identifier: SCMPDS_2.

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

1. The SCMPDS Computer

In this paper x denotes a set and i, k denote natural numbers.

The strict AMI SCMPDS over $\{\mathbb{Z}\}$ is defined as follows:

(Def. 1) $\mathrm{SCMPDS} = \langle \mathbb{N}, 0, \text{Instr-Loc}_{\mathrm{SCM}}, \mathbb{Z}_{14}, \mathrm{SCMPDS} - \mathrm{Instr}, \mathrm{SCMPDS} - \mathrm{OK}, \mathrm{SCMPDS} - \mathrm{Exec}\rangle$.

Next we state three propositions:

[1]This work was done while the author visited Shinshu University March–April 1999.

ISSN 1426–2630

(1) There exists k such that $x = 2 \cdot k + 2$ iff $x \in \text{Instr-Loc}_{\text{SCM}}$.

(2) SCMPDS is data-oriented.

(3) SCMPDS is definite.

Let us note that SCMPDS is von Neumann data-oriented and definite.

The following two propositions are true:

(4)(i) The instruction locations of SCMPDS $\neq \mathbb{Z}$,

(ii) the instructions of SCMPDS $\neq \mathbb{Z}$, and

(iii) the instruction locations of SCMPDS $\neq$ the instructions of SCMPDS.

(5) $\mathbb{N} = \{0\} \cup \text{Data-Loc}_{\text{SCM}} \cup \text{Instr-Loc}_{\text{SCM}}$.

In the sequel s is a state of SCMPDS.

One can prove the following propositions:

(6) $\mathbf{IC}_{\text{SCMPDS}} = 0$.

(7) For every SCMPDS-State S such that $S = s$ holds $\mathbf{IC}_s = \mathbf{IC}_S$.

2. The Memory Structure

An object of SCMPDS is called a Int position if:

(Def. 2) It $\in \text{Data-Loc}_{\text{SCM}}$.

In the sequel d_1 denotes a Int position.

The following propositions are true:

(8) $d_1 \in \text{Data-Loc}_{\text{SCM}}$.

(9) If $x \in \text{Data-Loc}_{\text{SCM}}$, then x is a Int position.

(10) $\text{Data-Loc}_{\text{SCM}}$ misses the instruction locations of SCMPDS.

(11) The instruction locations of SCMPDS are infinite.

(12) Every Int position is a data-location.

(13) For every Int position l holds $\text{ObjectKind}(l) = \mathbb{Z}$.

(14) For every set x such that $x \in \text{Instr-Loc}_{\text{SCM}}$ holds x is an instruction-location of SCMPDS.

3. The Instruction Structure

We use the following convention: d_2, d_3, d_4, d_5, d_6 are elements of $\text{Data-Loc}_{\text{SCM}}$ and k_1, k_2, k_3, k_4, k_5, k_6 are integers.

Let I be an instruction of SCMPDS. The functor $\text{InsCode}(I)$ yields a natural number and is defined by:

(Def. 3) $\text{InsCode}(I) = I_{\mathbf{1}}$.

In the sequel I is an instruction of SCMPDS.

Next we state the proposition

(15) For every instruction I of SCMPDS holds $\mathrm{InsCode}(I) \leqslant 13$.

Let s be a state of SCMPDS and let d be a Int position. Then $s(d)$ is an integer.

Let m, n be integers. The functor $\mathrm{DataLoc}(m,n)$ yields a Int position and is defined as follows:

(Def. 4) $\mathrm{DataLoc}(m,n) = 2 \cdot |m+n| + 1$.

One can prove the following propositions:

(16) $\langle 0, \langle k_1 \rangle \rangle \in \mathrm{SCMPDS} - \mathrm{Instr}$.

(17) $\langle 1, \langle d_2 \rangle \rangle \in \mathrm{SCMPDS} - \mathrm{Instr}$.

(18) If $x \in \{2,3\}$, then $\langle x, \langle d_3, k_2 \rangle \rangle \in \mathrm{SCMPDS} - \mathrm{Instr}$.

(19) If $x \in \{4,5,6,7,8\}$, then $\langle x, \langle d_4, k_3, k_4 \rangle \rangle \in \mathrm{SCMPDS} - \mathrm{Instr}$.

(20) If $x \in \{9,10,11,12,13\}$, then $\langle x, < *d_5, d_6, k_5, k_6* > \rangle \in \mathrm{SCMPDS} - \mathrm{Instr}$.

In the sequel a, b, c are Int position.

Let us consider k_1. The functor goto k_1 yielding an instruction of SCMPDS is defined as follows:

(Def. 5) $\mathrm{goto}\ k_1 = \langle 0, \langle k_1 \rangle \rangle$.

Let us consider a. The functor return a yields an instruction of SCMPDS and is defined by:

(Def. 6) $\mathrm{return}\, a = \langle 1, \langle a \rangle \rangle$.

Let us consider a, k_1. The functor $a{:=}k_1$ yields an instruction of SCMPDS and is defined as follows:

(Def. 7) $a{:=}k_1 = \langle 2, \langle a, k_1 \rangle \rangle$.

The functor $\mathrm{saveIC}(a, k_1)$ yields an instruction of SCMPDS and is defined as follows:

(Def. 8) $\mathrm{saveIC}(a, k_1) = \langle 3, \langle a, k_1 \rangle \rangle$.

Let us consider a, k_1, k_2. The functor $(a, k_1) <> 0\_gotok_2$ yields an instruction of SCMPDS and is defined as follows:

(Def. 9) $(a, k_1) <> 0\_gotok_2 = \langle 4, \langle a, k_1, k_2 \rangle \rangle$.

The functor $(a, k_1) <= 0\_gotok_2$ yielding an instruction of SCMPDS is defined as follows:

(Def. 10) $(a, k_1) <= 0\_gotok_2 = \langle 5, \langle a, k_1, k_2 \rangle \rangle$.

The functor $(a, k_1) >= 0\_gotok_2$ yielding an instruction of SCMPDS is defined by:

(Def. 11) $(a, k_1) >= 0\_gotok_2 = \langle 6, \langle a, k_1, k_2 \rangle \rangle$.

The functor $a_{k_1}{:=}k_2$ yielding an instruction of SCMPDS is defined as follows:

(Def. 12) $a_{k_1}{:=}k_2 = \langle 7, \langle a, k_1, k_2\rangle\rangle$.

The functor $\mathrm{AddTo}(a, k_1, k_2)$ yielding an instruction of SCMPDS is defined by:

(Def. 13) $\mathrm{AddTo}(a, k_1, k_2) = \langle 8, \langle a, k_1, k_2\rangle\rangle$.

Let us consider a, b, k_1, k_2. The functor $\mathrm{AddTo}(a, k_1, b, k_2)$ yields an instruction of SCMPDS and is defined by:

(Def. 14) $\mathrm{AddTo}(a, k_1, b, k_2) = \langle 9, < *a, b, k_1, k_2* > \rangle$.

The functor $\mathrm{SubFrom}(a, k_1, b, k_2)$ yielding an instruction of SCMPDS is defined by:

(Def. 15) $\mathrm{SubFrom}(a, k_1, b, k_2) = \langle 10, < *a, b, k_1, k_2* > \rangle$.

The functor $\mathrm{MultBy}(a, k_1, b, k_2)$ yielding an instruction of SCMPDS is defined as follows:

(Def. 16) $\mathrm{MultBy}(a, k_1, b, k_2) = \langle 11, < *a, b, k_1, k_2* > \rangle$.

The functor $\mathrm{Divide}(a, k_1, b, k_2)$ yielding an instruction of SCMPDS is defined by:

(Def. 17) $\mathrm{Divide}(a, k_1, b, k_2) = \langle 12, < *a, b, k_1, k_2* > \rangle$.

The functor $(a, k_1) := (b, k_2)$ yielding an instruction of SCMPDS is defined by:

(Def. 18) $(a, k_1) := (b, k_2) = \langle 13, < *a, b, k_1, k_2* > \rangle$.

One can prove the following propositions:

(21) $\mathrm{InsCode}(\mathrm{goto}\ k_1) = 0$.

(22) $\mathrm{InsCode}(\mathrm{return}\, a) = 1$.

(23) $\mathrm{InsCode}(a{:=}k_1) = 2$.

(24) $\mathrm{InsCode}(\mathrm{saveIC}(a, k_1)) = 3$.

(25) $\mathrm{InsCode}((a, k_1) <> 0\_goto k_2) = 4$.

(26) $\mathrm{InsCode}((a, k_1) <= 0\_goto k_2) = 5$.

(27) $\mathrm{InsCode}((a, k_1) >= 0\_goto k_2) = 6$.

(28) $\mathrm{InsCode}(a_{k_1}{:=}k_2) = 7$.

(29) $\mathrm{InsCode}(\mathrm{AddTo}(a, k_1, k_2)) = 8$.

(30) $\mathrm{InsCode}(\mathrm{AddTo}(a, k_1, b, k_2)) = 9$.

(31) $\mathrm{InsCode}(\mathrm{SubFrom}(a, k_1, b, k_2)) = 10$.

(32) $\mathrm{InsCode}(\mathrm{MultBy}(a, k_1, b, k_2)) = 11$.

(33) $\mathrm{InsCode}(\mathrm{Divide}(a, k_1, b, k_2)) = 12$.

(34) $\mathrm{InsCode}((a, k_1) := (b, k_2)) = 13$.

(35) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 0$ there exists k_1 such that $i_1 = \mathrm{goto}\ k_1$.

(36) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 1$ there exists a such that $i_1 = \mathrm{return}\, a$.

(37) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 2$ there exist a, k_1 such that $i_1 = a{:=}k_1$.

(38) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 3$ there exist a, k_1 such that $i_1 = \mathrm{saveIC}(a, k_1)$.

(39) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 4$ there exist a, k_1, k_2 such that $i_1 = (a, k_1) <> 0\_gotok_2$.

(40) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 5$ there exist a, k_1, k_2 such that $i_1 = (a, k_1) <= 0\_gotok_2$.

(41) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 6$ there exist a, k_1, k_2 such that $i_1 = (a, k_1) >= 0\_gotok_2$.

(42) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 7$ there exist a, k_1, k_2 such that $i_1 = a_{k_1}{:=}k_2$.

(43) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 8$ there exist a, k_1, k_2 such that $i_1 = \mathrm{AddTo}(a, k_1, k_2)$.

(44) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 9$ there exist a, b, k_1, k_2 such that $i_1 = \mathrm{AddTo}(a, k_1, b, k_2)$.

(45) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 10$ there exist a, b, k_1, k_2 such that $i_1 = \mathrm{SubFrom}(a, k_1, b, k_2)$.

(46) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 11$ there exist a, b, k_1, k_2 such that $i_1 = \mathrm{MultBy}(a, k_1, b, k_2)$.

(47) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 12$ there exist a, b, k_1, k_2 such that $i_1 = \mathrm{Divide}(a, k_1, b, k_2)$.

(48) For every instruction i_1 of SCMPDS such that $\mathrm{InsCode}(i_1) = 13$ there exist a, b, k_1, k_2 such that $i_1 = (a, k_1) := (b, k_2)$.

(49) For every state s of SCMPDS and for every Int position d holds $d \in \mathrm{dom}\, s$.

(50) For every state s of SCMPDS holds $\text{Data-Loc}_{\mathrm{SCM}} \subseteq \mathrm{dom}\, s$.

(51) For every state s of SCMPDS holds $\mathrm{dom}(s \restriction \text{Data-Loc}_{\mathrm{SCM}}) = \text{Data-Loc}_{\mathrm{SCM}}$.

(52) For every Int position d_7 holds $d_7 \neq \mathbf{IC}_{\mathrm{SCMPDS}}$.

(53) For every instruction-location i_2 of SCMPDS and for every Int position d_7 holds $i_2 \neq d_7$.

(54) Let s_1, s_2 be states of SCMPDS. Suppose $\mathbf{IC}_{(s_1)} = \mathbf{IC}_{(s_2)}$ and for every Int position a holds $s_1(a) = s_2(a)$ and for every instruction-location i of SCMPDS holds $s_1(i) = s_2(i)$. Then $s_1 = s_2$.

Let l_1 be an instruction-location of SCMPDS. The functor $\mathrm{Next}(l_1)$ yields an instruction-location of SCMPDS and is defined by:

(Def. 19) There exists an element m_1 of $\text{Instr-Loc}_{\mathrm{SCM}}$ such that $m_1 = l_1$ and $\mathrm{Next}(l_1) = \mathrm{Next}(m_1)$.

One can prove the following propositions:

(55) For every instruction-location l_1 of SCMPDS and for every element m_1 of Instr-Loc$_{\rm SCM}$ such that $m_1 = l_1$ holds Next(m_1) = Next(l_1).

(56) For every element i of SCMPDS − Instr such that $i = I$ and for every SCMPDS-State S such that $S = s$ holds Exec(I, s) = Exec-Res$_{\rm SCM}(i, S)$.

4. Execution Semantics of the SCMPDS instructions

The following propositions are true:

(57) (Exec(a:=k_1, s))($\mathbf{IC}_{\rm SCMPDS}$) = Next($\mathbf{IC}_s$) and (Exec(a:=k_1, s))(a) = k_1 and for every b such that $b \neq a$ holds (Exec(a:=k_1, s))(b) = $s(b)$.

(58) (Exec(a_{k_1}:=k_2, s))($\mathbf{IC}_{\rm SCMPDS}$) = Next($\mathbf{IC}_s$) and (Exec(a_{k_1}:=k_2, s)) (DataLoc($s(a)$, k_1)) = k_2 and for every b such that $b \neq$ DataLoc($s(a)$, k_1) holds (Exec(a_{k_1}:=k_2, s))(b) = $s(b)$.

(59) (Exec((a, k_1) := (b, k_2), s))($\mathbf{IC}_{\rm SCMPDS}$) = Next($\mathbf{IC}_s$) and (Exec((a, k_1) := (b, k_2), s))(DataLoc($s(a)$, k_1)) = s(DataLoc($s(b)$, k_2)) and for every c such that $c \neq$ DataLoc($s(a)$, k_1) holds (Exec((a, k_1) := (b, k_2), s))(c) = $s(c)$.

(60) (Exec(AddTo(a, k_1, k_2), s))($\mathbf{IC}_{\rm SCMPDS}$) = Next($\mathbf{IC}_s$) and (Exec(AddTo (a, k_1, k_2), s))(DataLoc($s(a)$, k_1)) = s(DataLoc($s(a)$, k_1)) + k_2 and for every b such that $b \neq$ DataLoc($s(a)$, k_1) holds (Exec(AddTo(a, k_1, k_2), s))(b) = $s(b)$.

(61) (Exec(AddTo(a, k_1, b, k_2), s))($\mathbf{IC}_{\rm SCMPDS}$) = Next($\mathbf{IC}_s$) and (Exec(AddTo (a, k_1, b, k_2), s))(DataLoc($s(a)$, k_1)) = s(DataLoc($s(a)$, k_1))+s(DataLoc($s(b)$, k_2)) and for every c such that $c \neq$ DataLoc($s(a)$, k_1) holds (Exec(AddTo(a, k_1, b, k_2), s))(c) = $s(c)$.

(62) (Exec(SubFrom(a, k_1, b, k_2), s))($\mathbf{IC}_{\rm SCMPDS}$) = Next($\mathbf{IC}_s$) and (Exec (SubFrom(a, k_1, b, k_2), s))(DataLoc($s(a)$, k_1)) = s(DataLoc($s(a)$, k_1)) − s(DataLoc($s(b)$, k_2)) and for every c such that $c \neq$ DataLoc($s(a)$, k_1) holds (Exec(SubFrom(a, k_1, b, k_2), s))(c) = $s(c)$.

(63) (Exec(MultBy(a, k_1, b, k_2), s))($\mathbf{IC}_{\rm SCMPDS}$) = Next($\mathbf{IC}_s$) and (Exec (MultBy(a, k_1, b, k_2), s))(DataLoc($s(a)$, k_1)) = s(DataLoc($s(a)$, k_1)) $\cdot$ s(DataLoc($s(b)$, k_2)) and for every c such that $c \neq$ DataLoc($s(a)$, k_1) holds (Exec(MultBy(a, k_1, b, k_2), s))(c) = $s(c)$.

(64)(i) (Exec(Divide(a, k_1, b, k_2), s))($\mathbf{IC}_{\rm SCMPDS}$) = Next($\mathbf{IC}_s$),

(ii) if DataLoc($s(a)$, k_1) $\neq$ DataLoc($s(b)$, k_2), then (Exec(Divide(a, k_1, b, k_2), s))(DataLoc($s(a)$, k_1)) = s(DataLoc($s(a)$, k_1)) $\div$ s(DataLoc($s(b)$, k_2)),

(iii) (Exec(Divide(a, k_1, b, k_2), s))(DataLoc($s(b)$, k_2)) = s(DataLoc($s(a)$, k_1)) mod s(DataLoc($s(b)$, k_2)), and

(iv) for every c such that $c \neq \mathrm{DataLoc}(s(a), k_1)$ and $c \neq \mathrm{DataLoc}(s(b), k_2)$ holds $(\mathrm{Exec}(\mathrm{Divide}(a, k_1, b, k_2), s))(c) = s(c)$.

(65) $(\mathrm{Exec}(\mathrm{Divide}(a, k_1, a, k_1), s))(\mathbf{IC}_{\mathrm{SCMPDS}}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(\mathrm{Divide}(a, k_1, a, k_1), s))(\mathrm{DataLoc}(s(a), k_1)) = s(\mathrm{DataLoc}(s(a), k_1)) \bmod s(\mathrm{DataLoc}(s(a), k_1))$ and for every c such that $c \neq \mathrm{DataLoc}(s(a), k_1)$ holds $(\mathrm{Exec}(\mathrm{Divide}(a, k_1, a, k_1), s))(c) = s(c)$.

Let s be a state of SCMPDS and let c be an integer. The functor $\mathrm{ICplusConst}(s, c)$ yields an instruction-location of SCMPDS and is defined by:

(Def. 20) There exists a natural number m such that $m = \mathbf{IC}_s$ and $\mathrm{ICplusConst}(s, c) = |(m - 2) + 2 \cdot c| + 2$.

The following propositions are true:

(66) $(\mathrm{Exec}(\mathrm{goto}\ k_1, s))(\mathbf{IC}_{\mathrm{SCMPDS}}) = \mathrm{ICplusConst}(s, k_1)$ and for every a holds $(\mathrm{Exec}(\mathrm{goto}\ k_1, s))(a) = s(a)$.

(67) If $s(\mathrm{DataLoc}(s(a), k_1)) \neq 0$, then $(\mathrm{Exec}((a, k_1) <> 0\_goto k_2, s))(\mathbf{IC}_{\mathrm{SCMPDS}}) = \mathrm{ICplusConst}(s, k_2)$ and if $s(\mathrm{DataLoc}(s(a), k_1)) = 0$, then $(\mathrm{Exec}((a, k_1) <> 0\_goto k_2, s))(\mathbf{IC}_{\mathrm{SCMPDS}}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}((a, k_1) <> 0\_goto k_2, s))(b) = s(b)$.

(68) If $s(\mathrm{DataLoc}(s(a), k_1)) \leqslant 0$, then $(\mathrm{Exec}((a, k_1) <= 0\_goto k_2, s))(\mathbf{IC}_{\mathrm{SCMPDS}}) = \mathrm{ICplusConst}(s, k_2)$ and if $s(\mathrm{DataLoc}(s(a), k_1)) > 0$, then $(\mathrm{Exec}((a, k_1) <= 0\_goto k_2, s))(\mathbf{IC}_{\mathrm{SCMPDS}}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}((a, k_1) <= 0\_goto k_2, s))(b) = s(b)$.

(69) If $s(\mathrm{DataLoc}(s(a), k_1)) \geqslant 0$, then $(\mathrm{Exec}((a, k_1) >= 0\_goto k_2, s))(\mathbf{IC}_{\mathrm{SCMPDS}}) = \mathrm{ICplusConst}(s, k_2)$ and if $s(\mathrm{DataLoc}(s(a), k_1)) < 0$, then $(\mathrm{Exec}((a, k_1) >= 0\_goto k_2, s))(\mathbf{IC}_{\mathrm{SCMPDS}}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}((a, k_1) >= 0\_goto k_2, s))(b) = s(b)$.

(70) $(\mathrm{Exec}(\mathrm{return}\, a, s))(\mathbf{IC}_{\mathrm{SCMPDS}}) = 2 \cdot (|s(\mathrm{DataLoc}(s(a), \mathrm{RetIC}))| \div 2) + 4$ and $(\mathrm{Exec}(\mathrm{return}\, a, s))(a) = s(\mathrm{DataLoc}(s(a), \mathrm{RetSP}))$ and for every b such that $a \neq b$ holds $(\mathrm{Exec}(\mathrm{return}\, a, s))(b) = s(b)$.

(71) $(\mathrm{Exec}(\mathrm{saveIC}(a, k_1), s))(\mathbf{IC}_{\mathrm{SCMPDS}}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(\mathrm{saveIC}(a, k_1), s))(\mathrm{DataLoc}(s(a), k_1)) = \mathbf{IC}_s$ and for every b such that $\mathrm{DataLoc}(s(a), k_1) \neq b$ holds $(\mathrm{Exec}(\mathrm{saveIC}(a, k_1), s))(b) = s(b)$.

(72) For every integer k there exists a function f from $\mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$ into $\mathbb{Z}$ such that for every element x of $\mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$ holds $f(x) = k$.

(73) For every integer k there exists a state s of SCMPDS such that for every Int position d holds $s(d) = k$.

(74) Let k be an integer and l_1 be an instruction-location of SCMPDS. Then there exists a state s of SCMPDS such that $s(0) = l_1$ and for every Int position d holds $s(d) = k$.

(75) goto 0 is halting.

(76) For every instruction I of SCMPDS such that there exists s such that $(\mathrm{Exec}(I,s))(\mathbf{IC}_{\mathrm{SCMPDS}}) = \mathrm{Next}(\mathbf{IC}_s)$ holds I is non halting.

(77) $a{:=}k_1$ is non halting.

(78) $a_{k_1}{:=}k_2$ is non halting.

(79) $(a,k_1) := (b,k_2)$ is non halting.

(80) $\mathrm{AddTo}(a,k_1,k_2)$ is non halting.

(81) $\mathrm{AddTo}(a,k_1,b,k_2)$ is non halting.

(82) $\mathrm{SubFrom}(a,k_1,b,k_2)$ is non halting.

(83) $\mathrm{MultBy}(a,k_1,b,k_2)$ is non halting.

(84) $\mathrm{Divide}(a,k_1,b,k_2)$ is non halting.

(85) If $k_1 \neq 0$, then goto k_1 is non halting.

(86) $(a,k_1) <> 0\_gotok_2$ is non halting.

(87) $(a,k_1) <= 0\_gotok_2$ is non halting.

(88) $(a,k_1) >= 0\_gotok_2$ is non halting.

(89) return a is non halting.

(90) $\mathrm{saveIC}(a,k_1)$ is non halting.

(91) Let I be a set. Then I is an instruction of SCMPDS if and only if one of the following conditions is satisfied:
there exists k_1 such that $I =$ goto k_1 or there exists a such that $I =$ return a or there exist a, k_1 such that $I = \mathrm{saveIC}(a,k_1)$ or there exist a, k_1 such that $I = a{:=}k_1$ or there exist a, k_1, k_2 such that $I = a_{k_1}{:=}k_2$ or there exist a, k_1, k_2 such that $I = (a,k_1) <> 0\_gotok_2$ or there exist a, k_1, k_2 such that $I = (a,k_1) <= 0\_gotok_2$ or there exist a, k_1, k_2 such that $I = (a,k_1) >= 0\_gotok_2$ or there exist a, b, k_1, k_2 such that $I = \mathrm{AddTo}(a,k_1,k_2)$ or there exist a, b, k_1, k_2 such that $I = \mathrm{AddTo}(a,k_1,b,k_2)$ or there exist a, b, k_1, k_2 such that $I = \mathrm{SubFrom}(a,k_1,b,k_2)$ or there exist a, b, k_1, k_2 such that $I = \mathrm{MultBy}(a,k_1,b,k_2)$ or there exist a, b, k_1, k_2 such that $I = \mathrm{Divide}(a,k_1,b,k_2)$ or there exist a, b, k_1, k_2 such that $I = (a,k_1) := (b,k_2)$.

Let us observe that SCMPDS is halting.

We now state several propositions:

(92) For every instruction I of SCMPDS such that I is halting holds $I = \mathbf{halt}_{\mathrm{SCMPDS}}$.

(93) $\mathbf{halt}_{\mathrm{SCMPDS}} = \mathrm{goto}\ 0$.

(94) $\mathrm{Exec}(\mathbf{halt}_{\mathrm{SCMPDS}}, s) = s$.

(95) For every state s of SCMPDS and for every instruction-location i of SCMPDS holds $s(i)$ is an instruction of SCMPDS.

(96) For every state s of SCMPDS and for every instruction i of SCMPDS and for every instruction-location l of SCMPDS holds $(\mathrm{Exec}(i,s))(l) = s(l)$.

(97) SCMPDS is realistic.

Let us observe that SCMPDS is steady-programmed and realistic.

One can prove the following propositions:

(98) $\mathbf{IC}_{\rm SCMPDS} \neq \mathbf{d}_i$ and $\mathbf{IC}_{\rm SCMPDS} \neq \mathbf{i}_i$.

(99) For every instruction I of SCMPDS such that I = goto 0 holds I is halting.

Acknowledgments

We wish to thank Prof. Y. Nakamura for many helpful suggestions.

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[2] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[3] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[4] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.
[5] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[6] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[7] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[8] Jing-Chao Chen. A small computer model with push-down stack. *Formalized Mathematics*, 8(**1**):175–182, 1999.
[9] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[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] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.
[13] Dariusz Surowik. Cyclic groups and some of their properties - part I. *Formalized Mathematics*, 2(**5**):623–627, 1991.
[14] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[15] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[16] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[17] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[18] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[19] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 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 15, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Computation and Program Shift in the SCMPDS Computer[1]

Jing-Chao Chen
Shanghai Jiaotong University

Summary. A finite partial state is said to be autonomic if the computation results in any two states containing it are same on its domain. On the basis of this definition, this article presents some computation results about autonomic finite partial states of the SCMPDS computer. Because the instructions of the SCMPDS computer are more complicated than those of the SCMFSA computer, the results given by this article are weaker than those reported previously by the article on the SCMFSA computer. The second task of this article is to define the notion of program shift. The importance of this notion is that the computation of some program blocks can be simplified by shifting a program block to the initial position.

MML Identifier: `SCMPDS_3`.

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

1. Preliminaries

In this paper k, m, n denote natural numbers.

Next we state several propositions:

(1) Suppose $n \leqslant 13$. Then $n = 0$ or $n = 1$ or $n = 2$ or $n = 3$ or $n = 4$ or $n = 5$ or $n = 6$ or $n = 7$ or $n = 8$ or $n = 9$ or $n = 10$ or $n = 11$ or $n = 12$ or $n = 13$.

[1]This work was done while the author visited Shinshu University March–April 1999.

ISSN 1426–2630

(2) For every integer k_1 and for all states s_1, s_2 of SCMPDS such that $\mathbf{IC}_{(s_1)} = \mathbf{IC}_{(s_2)}$ holds $\mathrm{ICplusConst}(s_1, k_1) = \mathrm{ICplusConst}(s_2, k_1)$.

(3) Let k_1 be an integer, a be a Int position, and s_1, s_2 be states of SCMPDS. If $s_1 \restriction \text{Data-Loc}_{\text{SCM}} = s_2 \restriction \text{Data-Loc}_{\text{SCM}}$, then $s_1(\mathrm{DataLoc}(s_1(a), k_1)) = s_2(\mathrm{DataLoc}(s_2(a), k_1))$.

(4) For every Int position a and for all states s_1, s_2 of SCMPDS such that $s_1 \restriction \text{Data-Loc}_{\text{SCM}} = s_2 \restriction \text{Data-Loc}_{\text{SCM}}$ holds $s_1(a) = s_2(a)$.

(5) The objects of SCMPDS $= \{\mathbf{IC}_{\text{SCMPDS}}\} \cup \text{Data-Loc}_{\text{SCM}} \cup$ the instruction locations of SCMPDS.

(6) $\mathbf{IC}_{\text{SCMPDS}} \notin \text{Data-Loc}_{\text{SCM}}$.

(7) For all states s_1, s_2 of SCMPDS such that $s_1 \restriction (\text{Data-Loc}_{\text{SCM}} \cup \{\mathbf{IC}_{\text{SCMPDS}}\}) = s_2 \restriction (\text{Data-Loc}_{\text{SCM}} \cup \{\mathbf{IC}_{\text{SCMPDS}}\})$ and for every instruction l of SCMPDS holds $\mathrm{Exec}(l, s_1) \restriction (\text{Data-Loc}_{\text{SCM}} \cup \{\mathbf{IC}_{\text{SCMPDS}}\}) = \mathrm{Exec}(l, s_2) \restriction (\text{Data-Loc}_{\text{SCM}} \cup \{\mathbf{IC}_{\text{SCMPDS}}\})$.

(8) For every instruction i of SCMPDS and for every state s of SCMPDS holds $\mathrm{Exec}(i, s) \restriction \text{Instr-Loc}_{\text{SCM}} = s \restriction \text{Instr-Loc}_{\text{SCM}}$.

2. Finite Partial States of SCMPDS

Next we state two propositions:

(9) For every finite partial state p of SCMPDS holds $\mathrm{DataPart}(p) = p \restriction \text{Data-Loc}_{\text{SCM}}$.

(10) For every finite partial state p of SCMPDS holds p is data-only iff $\operatorname{dom} p \subseteq \text{Data-Loc}_{\text{SCM}}$.

Let us mention that there exists a finite partial state of SCMPDS which is data-only.

Next we state two propositions:

(11) For every finite partial state p of SCMPDS holds $\operatorname{dom} \mathrm{DataPart}(p) \subseteq \text{Data-Loc}_{\text{SCM}}$.

(12) For every finite partial state p of SCMPDS holds $\operatorname{dom} \mathrm{ProgramPart}(p) \subseteq$ the instruction locations of SCMPDS.

Let I_1 be a partial function from FinPartSt(SCMPDS) to FinPartSt(SCMPDS). We say that I_1 is data-only if and only if the condition (Def. 1) is satisfied.

(Def. 1) Let p be a finite partial state of SCMPDS. Suppose $p \in \operatorname{dom} I_1$. Then p is data-only and for every finite partial state q of SCMPDS such that $q = I_1(p)$ holds q is data-only.

Let us observe that there exists a partial function from FinPartSt(SCMPDS) to FinPartSt(SCMPDS) which is data-only.

Next we state three propositions:

(13) Let i be an instruction of SCMPDS, s be a state of SCMPDS, and p be a programmed finite partial state of SCMPDS. Then $\mathrm{Exec}(i, s{+\cdot}p) = \mathrm{Exec}(i, s){+\cdot}p$.

(14) For every state s of SCMPDS and for every instruction-location i_1 of SCMPDS and for every Int position a holds $s(a) = (s{+\cdot}\,\text{Start-At}(i_1))(a)$.

(15) For all states s, t of SCMPDS holds $s{+\cdot}t{\restriction}\text{Data-Loc}_{\mathrm{SCM}}$ is a state of SCMPDS.

3. Autonomic Finite Partial States of SCMPDS and its Computation

Let l_1 be a Int position and let a be an integer. Then $l_1 \mathop{\longmapsto}\limits^{\cdot} a$ is a finite partial state of SCMPDS.

Next we state the proposition

(16) For every autonomic finite partial state p of SCMPDS such that $\mathrm{DataPart}(p) \neq \emptyset$ holds $\mathbf{IC}_{\mathrm{SCMPDS}} \in \mathrm{dom}\, p$.

Let us observe that there exists a finite partial state of SCMPDS which is autonomic and non programmed.

One can prove the following propositions:

(17) For every autonomic non programmed finite partial state p of SCMPDS holds $\mathbf{IC}_{\mathrm{SCMPDS}} \in \mathrm{dom}\, p$.

(18) Let s_1, s_2 be states of SCMPDS and k_1, k_2, m be integers. If $\mathbf{IC}_{(s_1)} = \mathbf{IC}_{(s_2)}$ and $k_1 \neq k_2$ and $m = \mathbf{IC}_{(s_1)}$ and $(m-2)+2\cdot k_1 \geqslant 0$ and $(m-2)+2\cdot k_2 \geqslant 0$, then $\mathrm{ICplusConst}(s_1, k_1) \neq \mathrm{ICplusConst}(s_2, k_2)$.

(19) For all states s_1, s_2 of SCMPDS and for all natural numbers k_1, k_2 such that $\mathbf{IC}_{(s_1)} = \mathbf{IC}_{(s_2)}$ and $k_1 \neq k_2$ holds $\mathrm{ICplusConst}(s_1, k_1) \neq \mathrm{ICplusConst}(s_2, k_2)$.

(20) For every state s of SCMPDS holds $\mathrm{Next}(\mathbf{IC}_s) = \mathrm{ICplusConst}(s, 1)$.

(21) For every autonomic finite partial state p of SCMPDS such that $\mathbf{IC}_{\mathrm{SCMPDS}} \in \mathrm{dom}\, p$ holds $\mathbf{IC}_p \in \mathrm{dom}\, p$.

(22) Let p be an autonomic non programmed finite partial state of SCMPDS and s be a state of SCMPDS. If $p \subseteq s$, then for every natural number i holds $\mathbf{IC}_{(\mathrm{Computation}(s))(i)} \in \mathrm{dom}\,\mathrm{ProgramPart}(p)$.

(23) Let p be an autonomic non programmed finite partial state of SCMPDS and s_1, s_2 be states of SCMPDS. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number. Then $\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)} = \mathbf{IC}_{(\mathrm{Computation}(s_2))(i)}$ and $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{CurInstr}((\mathrm{Computation}(s_2))(i))$.

(24) Let p be an autonomic non programmed finite partial state of SCMPDS and s_1, s_2 be states of SCMPDS. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, k_1, k_2 be integers, and a, b be Int position. Suppose $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = (a, k_1) := (b, k_2)$ and $a \in \mathrm{dom}\, p$ and $\mathrm{DataLoc}((\mathrm{Computation}(s_1))(i)(a), k_1) \in \mathrm{dom}\, p$. Then $(\mathrm{Computation}(s_1))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_1))(i)(b), k_2)) = (\mathrm{Computation}(s_2))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_2))(i)(b), k_2))$.

(25) Let p be an autonomic non programmed finite partial state of SCMPDS and s_1, s_2 be states of SCMPDS. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, k_1, k_2 be integers, and a, b be Int position. Suppose $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{AddTo}(a, k_1, b, k_2)$ and $a \in \mathrm{dom}\, p$ and $\mathrm{DataLoc}((\mathrm{Computation}(s_1))(i)(a), k_1) \in \mathrm{dom}\, p$. Then $(\mathrm{Computation}(s_1))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_1))(i)(b), k_2)) = (\mathrm{Computation}(s_2))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_2))(i)(b), k_2))$.

(26) Let p be an autonomic non programmed finite partial state of SCMPDS and s_1, s_2 be states of SCMPDS. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, k_1, k_2 be integers, and a, b be Int position. Suppose $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{SubFrom}(a, k_1, b, k_2)$ and $a \in \mathrm{dom}\, p$ and $\mathrm{DataLoc}((\mathrm{Computation}(s_1))(i)(a), k_1) \in \mathrm{dom}\, p$. Then $(\mathrm{Computation}(s_1))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_1))(i)(b), k_2)) = (\mathrm{Computation}(s_2))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_2))(i)(b), k_2))$.

(27) Let p be an autonomic non programmed finite partial state of SCMPDS and s_1, s_2 be states of SCMPDS. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, k_1, k_2 be integers, and a, b be Int position. Suppose $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{MultBy}(a, k_1, b, k_2)$ and $a \in \mathrm{dom}\, p$ and $\mathrm{DataLoc}((\mathrm{Computation}(s_1))(i)(a), k_1) \in \mathrm{dom}\, p$. Then $(\mathrm{Computation}(s_1))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_1))(i)(a), k_1)) \cdot (\mathrm{Computation}(s_1))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_1))(i)(b), k_2)) = (\mathrm{Computation}(s_2))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_2))(i)(a), k_1)) \cdot (\mathrm{Computation}(s_2))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_2))(i)(b), k_2))$.

(28) Let p be an autonomic non programmed finite partial state of SCMPDS and s_1, s_2 be states of SCMPDS. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i, m be natural numbers, a be a Int position, and k_1, k_2 be integers. Suppose $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = (a, k_1) <> 0\_gotok_2$ and $m = \mathbf{IC}_{(\mathrm{Computation}(s_1))(i)}$ and $(m - 2) + 2 \cdot k_2 \geqslant 0$ and $k_2 \neq 1$. Then $(\mathrm{Computation}(s_1))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_1))(i)(a), k_1)) = 0$ if and only if $(\mathrm{Computation}(s_2))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_2))(i)(a), k_1)) = 0$.

(29) Let p be an autonomic non programmed finite partial state of SCMPDS and s_1, s_2 be states of SCMPDS. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i, m be natural numbers, a be a Int position, and k_1, k_2 be integers. Suppose $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = (a, k_1) <= 0\_gotok_2$ and

$m = \mathbf{IC}_{(\mathrm{Computation}(s_1))(i)}$ and $(m - 2) + 2 \cdot k_2 \geqslant 0$ and $k_2 \neq 1$. Then $(\mathrm{Computation}(s_1))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_1))(i)(a), k_1)) > 0$ if and only if $(\mathrm{Computation}(s_2))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_2))(i)(a), k_1)) > 0$.

(30) Let p be an autonomic non programmed finite partial state of SCMPDS and s_1, s_2 be states of SCMPDS. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i, m be natural numbers, a be a Int position, and k_1, k_2 be integers. Suppose $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = (a, k_1) >= 0\_goto k_2$ and $m = \mathbf{IC}_{(\mathrm{Computation}(s_1))(i)}$ and $(m - 2) + 2 \cdot k_2 \geqslant 0$ and $k_2 \neq 1$. Then $(\mathrm{Computation}(s_1))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_1))(i)(a), k_1)) < 0$ if and only if $(\mathrm{Computation}(s_2))(i)(\mathrm{DataLoc}((\mathrm{Computation}(s_2))(i)(a), k_1)) < 0$.

4. Program Shift in the SCMPDS Computer

Let us consider k. The functor $\mathrm{inspos}\, k$ yielding an instruction-location of SCMPDS is defined by:

(Def. 2) $\mathrm{inspos}\, k = \mathbf{i}_k$.

One can prove the following two propositions:

(31) For all natural numbers k_1, k_2 such that $k_1 \neq k_2$ holds $\mathrm{inspos}\, k_1 \neq \mathrm{inspos}\, k_2$.

(32) For every instruction-location i_2 of SCMPDS there exists a natural number i such that $i_2 = \mathrm{inspos}\, i$.

Let l_2 be an instruction-location of SCMPDS and let k be a natural number. The functor $l_2 + k$ yields an instruction-location of SCMPDS and is defined as follows:

(Def. 3) There exists a natural number m such that $l_2 = \mathrm{inspos}\, m$ and $l_2 + k = \mathrm{inspos}\, m + k$.

The functor $l_2 -' k$ yielding an instruction-location of SCMPDS is defined as follows:

(Def. 4) There exists a natural number m such that $l_2 = \mathrm{inspos}\, m$ and $l_2 -' k = \mathrm{inspos}\, m -' k$.

Next we state four propositions:

(33) For every instruction-location l of SCMPDS and for all m, n holds $(l + m) + n = l + (m + n)$.

(34) For every instruction-location l_2 of SCMPDS and for every natural number k holds $(l_2 + k) -' k = l_2$.

(35) For all instructions-locations l_3, l_4 of SCMPDS and for every natural number k holds $\mathrm{Start\text{-}At}(l_3 + k) = \mathrm{Start\text{-}At}(l_4 + k)$ iff $\mathrm{Start\text{-}At}(l_3) = \mathrm{Start\text{-}At}(l_4)$.

(36) For all instructions-locations l_3, l_4 of SCMPDS and for every natural number k such that $\text{Start-At}(l_3) = \text{Start-At}(l_4)$ holds $\text{Start-At}(l_3 -' k) = \text{Start-At}(l_4 -' k)$.

Let I_1 be a finite partial state of SCMPDS. We say that I_1 is initial if and only if:

(Def. 5) For all m, n such that $\text{inspos}\, n \in \text{dom}\, I_1$ and $m < n$ holds $\text{inspos}\, m \in \text{dom}\, I_1$.

The finite partial state SCMPDS − Stop of SCMPDS is defined as follows:

(Def. 6) $\text{SCMPDS} - \text{Stop} = \text{inspos}\, 0 \mapsto \mathbf{halt}_{\text{SCMPDS}}$.

Let us observe that SCMPDS − Stop is non empty initial and programmed.

Let us observe that there exists a finite partial state of SCMPDS which is initial, programmed, and non empty.

Let p be a programmed finite partial state of SCMPDS and let k be a natural number. The functor $\text{Shift}(p, k)$ yielding a programmed finite partial state of SCMPDS is defined as follows:

(Def. 7) $\text{dom}\, \text{Shift}(p, k) = \{\text{inspos}\, m+k : \text{inspos}\, m \in \text{dom}\, p\}$ and for every m such that $\text{inspos}\, m \in \text{dom}\, p$ holds $(\text{Shift}(p, k))(\text{inspos}\, m + k) = p(\text{inspos}\, m)$.

We now state several propositions:

(37) Let l be an instruction-location of SCMPDS, k be a natural number, and p be a programmed finite partial state of SCMPDS. If $l \in \text{dom}\, p$, then $(\text{Shift}(p, k))(l + k) = p(l)$.

(38) Let p be a programmed finite partial state of SCMPDS and k be a natural number. Then $\text{dom}\, \text{Shift}(p, k) = \{i_2+k; i_2$ ranges over instructions-locations of SCMPDS: $i_2 \in \text{dom}\, p\}$.

(39) For every programmed finite partial state I of SCMPDS holds $\text{Shift}(\text{Shift}(I, m), n) = \text{Shift}(I, m + n)$.

(40) Let s be a programmed finite partial state of SCMPDS, f be a function from the instructions of SCMPDS into the instructions of SCMPDS, and given n. Then $\text{Shift}(f \cdot s, n) = f \cdot \text{Shift}(s, n)$.

(41) For all programmed finite partial states I, J of SCMPDS holds $\text{Shift}(I{+}\!\cdot J, n) = \text{Shift}(I, n){+}\!\cdot \text{Shift}(J, n)$.

Acknowledgments

We wish to thank Prof. Y. Nakamura for many helpful suggestions.

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[2] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[3] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.

[4] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[5] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[6] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[7] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.
[8] Czesław Byliński. Products and coproducts in categories. *Formalized Mathematics*, 2(**5**):701–709, 1991.
[9] Jing-Chao Chen. The SCMPDS computer and the basic semantics of its instructions. *Formalized Mathematics*, 8(**1**):183–191, 1999.
[10] Jing-Chao Chen. A small computer model with push-down stack. *Formalized Mathematics*, 8(**1**):175–182, 1999.
[11] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[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] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[16] Dariusz Surowik. Cyclic groups and some of their properties - part I. *Formalized Mathematics*, 2(**5**):623–627, 1991.
[17] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[18] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[19] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[20] Andrzej Trybulec and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):571–576, 1996.
[21] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[22] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.
[23] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[24] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[25] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received June 15, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

The Construction and Shiftability of Program Blocks for SCMPDS[1]

Jing-Chao Chen
Shanghai Jiaotong University

Summary. In this article,a program block is defined as a finite sequence of instructions stored consecutively on initial positions. Based on this definition,any program block with more than two instructions can be viewed as the combination of two smaller program blocks. To describe the computation of a program block by the result of its two sub-blocks, we introduce the notions of paraclosed, parahalting, valid, and shiftable, the meaning of which may be stated as follows:

- a program is paraclosed if and only if any state containing it is closed,
- a program is parahalting if and only if any state containing it is halting,
- in a program block, a jumping instruction is valid if its jumping offset is valid,
- a program block is shiftable if it does not contain any return and saveIC instructions,and each instruction in it is valid.

When a program block is shiftable, its computing result does not depend on its storage position.

MML Identifier: `SCMPDS_4`.

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

[1]This research is partially supported by the National Natural Science Foundation of China Grant No. 69873033.

ISSN 1426–2630

1. Definition of a Program Block and its Basic Properties

A Program-block is an initial programmed finite partial state of SCMPDS.

We adopt the following convention: m, n are natural numbers, i, j, k are instructions of SCMPDS, and I, J, K are Program-block.

Let us consider i. The functor Load(i) yielding a Program-block is defined as follows:

(Def. 1) $\mathrm{Load}(i) = \mathrm{inspos}\,0 \dot{\longmapsto} i$.

Let us consider i. Note that Load(i) is non empty.

Next we state the proposition

(1) For every Program-block P and for every n holds $n < \mathrm{card}\,P$ iff $\mathrm{inspos}\,n \in \mathrm{dom}\,P$.

Let I be an initial finite partial state of SCMPDS. Note that ProgramPart(I) is initial.

Next we state four propositions:

(2) $\mathrm{dom}\,I$ misses $\mathrm{dom}\,\mathrm{Shift}(J, \mathrm{card}\,I)$.

(3) For every programmed finite partial state I of SCMPDS holds $\mathrm{card}\,\mathrm{Shift}(I, m) = \mathrm{card}\,I$.

(4) For all finite partial states I, J of SCMPDS holds $\mathrm{ProgramPart}(I {+\cdot} J) = \mathrm{ProgramPart}(I) {+\cdot}\, \mathrm{ProgramPart}(J)$.

(5) For all finite partial states I, J of SCMPDS holds $\mathrm{Shift}(\mathrm{ProgramPart}(I {+\cdot} J), n) = \mathrm{Shift}(\mathrm{ProgramPart}(I), n) {+\cdot}\, \mathrm{Shift}(\mathrm{ProgramPart}(J), n)$.

We use the following convention: a, b are Int position, s, s_1, s_2 are states of SCMPDS, and k_1, k_2 are integers.

Let us consider I. The functor Initialized(I) yields a finite partial state of SCMPDS and is defined as follows:

(Def. 2) $\mathrm{Initialized}(I) = I {+\cdot}\, \text{Start-At}(\mathrm{inspos}\,0)$.

We now state a number of propositions:

(6) $\mathrm{InsCode}(i) \in \{0, 1, 4, 5, 6\}$ or $(\mathrm{Exec}(i, s))(\mathbf{IC}_{\mathrm{SCMPDS}}) = \mathrm{Next}(\mathbf{IC}_s)$.

(7) $\mathbf{IC}_{\mathrm{SCMPDS}} \in \mathrm{dom}\,\mathrm{Initialized}(I)$.

(8) $\mathbf{IC}_{\mathrm{Initialized}(I)} = \mathrm{inspos}\,0$.

(9) $I \subseteq \mathrm{Initialized}(I)$.

(10) s and $s {+\cdot} I$ are equal outside the instruction locations of SCMPDS.

(11) Let s_1, s_2 be states of SCMPDS. Suppose $\mathbf{IC}_{(s_1)} = \mathbf{IC}_{(s_2)}$ and for every Int position a holds $s_1(a) = s_2(a)$. Then s_1 and s_2 are equal outside the instruction locations of SCMPDS.

(13)[2] Suppose s_1 and s_2 are equal outside the instruction locations of SCMPDS. Let a be a Int position. Then $s_1(a) = s_2(a)$.

(14) If s_1 and s_2 are equal outside the instruction locations of SCMPDS, then $s_1(\mathrm{DataLoc}(s_1(a), k_1)) = s_2(\mathrm{DataLoc}(s_2(a), k_1))$.

(15) Suppose s_1 and s_2 are equal outside the instruction locations of SCMPDS. Then $\mathrm{Exec}(i, s_1)$ and $\mathrm{Exec}(i, s_2)$ are equal outside the instruction locations of SCMPDS.

(16) $\mathrm{Initialized}(I)\restriction$the instruction locations of SCMPDS $= I$.

(17) For all natural numbers k_1, k_2 such that $k_1 \neq k_2$ holds $\mathrm{DataLoc}(k_1, 0) \neq \mathrm{DataLoc}(k_2, 0)$.

(18) For every Int position d_1 there exists a natural number i such that $d_1 = \mathrm{DataLoc}(i, 0)$.

The scheme *SCMPDSEx* deals with a unary functor $\mathcal{F}$ yielding an instruction of SCMPDS, a unary functor $\mathcal{G}$ yielding an integer, and an instruction-location $\mathcal{A}$ of SCMPDS, and states that:

> There exists a state S of SCMPDS such that $\mathbf{IC}_S = \mathcal{A}$ and for every natural number i holds $S(\mathrm{inspos}\, i) = \mathcal{F}(i)$ and $S(\mathrm{DataLoc}(i, 0)) = \mathcal{G}(i)$

for all values of the parameters.

Next we state a number of propositions:

(19) For every state s of SCMPDS holds $\mathrm{dom}\, s = \{\mathbf{IC}_{\mathrm{SCMPDS}}\} \cup \text{Data-Loc}_{\mathrm{SCM}} \cup$ the instruction locations of SCMPDS.

(20) Let s be a state of SCMPDS and x be a set. Suppose $x \in \mathrm{dom}\, s$. Then x is a Int position or $x = \mathbf{IC}_{\mathrm{SCMPDS}}$ or x is an instruction-location of SCMPDS.

(21) Let s_1, s_2 be states of SCMPDS. Then for every instruction-location l of SCMPDS holds $s_1(l) = s_2(l)$ if and only if $s_1\restriction$the instruction locations of SCMPDS $= s_2\restriction$the instruction locations of SCMPDS.

(22) For every instruction-location i of SCMPDS holds $i \notin \text{Data-Loc}_{\mathrm{SCM}}$.

(23) For all states s_1, s_2 of SCMPDS holds for every Int position a holds $s_1(a) = s_2(a)$ iff $s_1\restriction\text{Data-Loc}_{\mathrm{SCM}} = s_2\restriction\text{Data-Loc}_{\mathrm{SCM}}$.

(24) Let s_1, s_2 be states of SCMPDS. Suppose s_1 and s_2 are equal outside the instruction locations of SCMPDS. Then $s_1\restriction\text{Data-Loc}_{\mathrm{SCM}} = s_2\restriction\text{Data-Loc}_{\mathrm{SCM}}$.

(25) For all states s, s_3 of SCMPDS and for every set A holds $(s_3 {+\cdot} s\restriction A)\restriction A = s\restriction A$.

(26) For all Program-block I, J holds I and J are equal outside the instruction locations of SCMPDS.

[2]The proposition (12) has been removed.

(27) For every Program-block I holds dom Initialized(I) $=$ dom $I \cup \{\mathbf{IC}_{\mathrm{SCMPDS}}\}$.

(28) For every Program-block I and for every set x such that $x \in$ dom Initialized(I) holds $x \in$ dom I or $x = \mathbf{IC}_{\mathrm{SCMPDS}}$.

(29) For every Program-block I holds (Initialized(I))($\mathbf{IC}_{\mathrm{SCMPDS}}$) $=$ inspos 0.

(30) For every Program-block I holds $\mathbf{IC}_{\mathrm{SCMPDS}} \notin$ dom I.

(31) For every Program-block I and for every Int position a holds $a \notin$ dom Initialized(I).

In the sequel x denotes a set.

The following propositions are true:

(32) If $x \in$ dom I, then $I(x) = (I +\cdot$ Start-At(inspos n))(x).

(33) For every Program-block I and for every set x such that $x \in$ dom I holds $I(x) =$ (Initialized(I))(x).

(34) For all Program-block I, J and for every state s of SCMPDS such that Initialized(J) $\subseteq s$ holds $s +\cdot$ Initialized(I) $= s +\cdot I$.

(35) For all Program-block I, J and for every state s of SCMPDS such that Initialized(J) $\subseteq s$ holds Initialized(I) $\subseteq s +\cdot I$.

(36) Let I, J be Program-block and s be a state of SCMPDS. Then $s +\cdot$ Initialized(I) and $s +\cdot$ Initialized(J) are equal outside the instruction locations of SCMPDS.

2. Combining two Consecutive Blocks into One Program Block

Let I, J be Program-block. The functor $I;J$ yields a Program-block and is defined by:

(Def. 3) $I;J = I +\cdot$ Shift(J, card I).

One can prove the following propositions:

(37) For all Program-block I, J and for every instruction-location l of SCMPDS such that $l \in$ dom I holds $(I;J)(l) = I(l)$.

(38) For all Program-block I, J and for every instruction-location l of SCMPDS such that $l \in$ dom J holds $(I;J)(l + \mathrm{card}\, I) = J(l)$.

(39) For all Program-block I, J holds dom $I \subseteq$ dom($I;J$).

(40) For all Program-block I, J holds $I \subseteq I;J$.

(41) For all Program-block I, J holds $I +\cdot (I;J) = I;J$.

(42) For all Program-block I, J holds Initialized(I)$+\cdot(I;J) =$ Initialized($I;J$).

3. Combining a Block and a Instruction into One Program Block

Let us consider i, J. The functor i;J yielding a Program-block is defined by:

(Def. 4) $i;J = \mathrm{Load}(i);J$.

Let us consider I, j. The functor I;j yields a Program-block and is defined by:

(Def. 5) $I;j = I;\mathrm{Load}(j)$.

Let us consider i, j. The functor i;j yielding a Program-block is defined as follows:

(Def. 6) $i;j = \mathrm{Load}(i);\mathrm{Load}(j)$.

The following propositions are true:

(43) $i;j = \mathrm{Load}(i);j$.

(44) $i;j = i;\mathrm{Load}(j)$.

(45) $\mathrm{card}(I;J) = \mathrm{card}\, I + \mathrm{card}\, J$.

(46) $(I;J);K = I;(J;K)$.

(47) $(I;J);k = I;(J;k)$.

(48) $(I;j);K = I;(j;K)$.

(49) $(I;j);k = I;(j;k)$.

(50) $(i;J);K = i;(J;K)$.

(51) $(i;J);k = i;(J;k)$.

(52) $(i;j);K = i;(j;K)$.

(53) $(i;j);k = i;(j;k)$.

(54) $\mathrm{dom}\, I \cap \mathrm{dom}\, \mathrm{Start\text{-}At}(\mathrm{inspos}\, n) = \emptyset$.

(55) $\mathrm{Start\text{-}At}(\mathrm{inspos}\, 0) \subseteq \mathrm{Initialized}(I)$.

(56) If $I +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\, n) \subseteq s$, then $I \subseteq s$.

(57) If $\mathrm{Initialized}(I) \subseteq s$, then $I \subseteq s$.

(58) $(I +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\, n)) \restriction \text{the instruction locations of SCMPDS} = I$.

In the sequel l, l_1 denote instructions-locations of SCMPDS.

Next we state four propositions:

(59) $a \notin \mathrm{dom}\, \mathrm{Start\text{-}At}(l)$.

(60) $l_1 \notin \mathrm{dom}\, \mathrm{Start\text{-}At}(l)$.

(61) $a \notin \mathrm{dom}(I +\cdot \mathrm{Start\text{-}At}(l))$.

(62) $s +\cdot I +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\, 0) = s +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\, 0) +\cdot I$.

Let s be a state of SCMPDS, let l_2 be a Int position, and let k be an integer. Then $s +\cdot (l_2, k)$ is a state of SCMPDS.

4. The Notions of Paraclosed, Parahalting and their Basic Properties

Let I be a Program-block. The functor $\mathrm{stop}\, I$ yielding a Program-block is defined as follows:

(Def. 7) $\mathrm{stop}\, I = I; \mathrm{SCMPDS} - \mathrm{Stop}$.

Let I be a Program-block and let s be a state of SCMPDS. The functor $\mathrm{IExec}(I, s)$ yielding a state of SCMPDS is defined as follows:

(Def. 8) $\mathrm{IExec}(I, s) = \mathrm{Result}(s {+\cdot} \mathrm{Initialized}(\mathrm{stop}\, I)) {+\cdot} s {\restriction}$the instruction locations of SCMPDS.

Let I be a Program-block. We say that I is paraclosed if and only if:

(Def. 9) For every state s of SCMPDS and for every natural number n such that $\mathrm{Initialized}(\mathrm{stop}\, I) \subseteq s$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(n)} \in \mathrm{dom}\, \mathrm{stop}\, I$.

We say that I is parahalting if and only if:

(Def. 10) $\mathrm{Initialized}(\mathrm{stop}\, I)$ is halting.

Let us note that there exists a Program-block which is parahalting.

One can prove the following proposition

(63) For every parahalting Program-block I such that $\mathrm{Initialized}(\mathrm{stop}\, I) \subseteq s$ holds s is halting.

Let I be a parahalting Program-block. Note that $\mathrm{Initialized}(\mathrm{stop}\, I)$ is halting.

Let l_3, l_4 be instructions-locations of SCMPDS and let a, b be instructions of SCMPDS. Then $[l_3 \longmapsto a, l_4 \longmapsto b]$ is a finite partial state of SCMPDS.

One can prove the following propositions:

(64) For every integer k such that $k \neq 0$ holds $\mathrm{goto}\, k \neq \mathbf{halt}_{\mathrm{SCMPDS}}$.

(65) $\mathbf{IC}_s \neq \mathrm{Next}(\mathbf{IC}_s)$.

(66) $s_2 {+\cdot} [\mathbf{IC}_{(s_2)} \longmapsto \mathrm{goto}\ 1, \mathrm{Next}(\mathbf{IC}_{(s_2)}) \longmapsto \mathrm{goto}\ (-1)]$ is not halting.

(67) Suppose that

(i) s_1 and s_2 are equal outside the instruction locations of SCMPDS,

(ii) $I \subseteq s_1$,

(iii) $I \subseteq s_2$, and

(iv) for every m such that $m < n$ holds $\mathbf{IC}_{(\mathrm{Computation}(s_2))(m)} \in \mathrm{dom}\, I$.

Let given m. Suppose $m \leqslant n$. Then $(\mathrm{Computation}(s_1))(m)$ and $(\mathrm{Computation}(s_2))(m)$ are equal outside the instruction locations of SCMPDS.

(68) For every state s of SCMPDS and for every instruction-location l of SCMPDS holds $l \in \mathrm{dom}\, s$.

In the sequel l_1, l_5 are instructions-locations of SCMPDS and i_1, i_2 are instructions of SCMPDS.

The following propositions are true:

(69) $s{+}\cdot[l_1 \longmapsto i_1, l_5 \longmapsto i_2] = s +\cdot (l_1, i_1) +\cdot (l_5, i_2)$.

(70) $\mathrm{Next}(\mathrm{inspos}\, n) = \mathrm{inspos}\, n + 1$.

(71) If $\mathbf{IC}_s \notin \mathrm{dom}\, I$, then $\mathrm{Next}(\mathbf{IC}_s) \notin \mathrm{dom}\, I$.

Let us mention that every Program-block which is parahalting is also paraclosed.

We now state several propositions:

(72) $\mathrm{dom}\, \mathrm{SCMPDS} - \mathrm{Stop} = \{\mathrm{inspos}\, 0\}$.

(73) $\mathrm{inspos}\, 0 \in \mathrm{dom}\, \mathrm{SCMPDS} - \mathrm{Stop}$ and $(\mathrm{SCMPDS} - \mathrm{Stop})(\mathrm{inspos}\, 0) = \mathbf{halt}_{\mathrm{SCMPDS}}$.

(74) $\mathrm{card}\, \mathrm{SCMPDS} - \mathrm{Stop} = 1$.

(75) $\mathrm{inspos}\, 0 \in \mathrm{dom}\, \mathrm{stop}\, I$.

(76) Let p be a programmed finite partial state of SCMPDS, k be a natural number, and i_3 be an instruction-location of SCMPDS. If $i_3 \in \mathrm{dom}\, p$, then $i_3 + k \in \mathrm{dom}\, \mathrm{Shift}(p, k)$.

5. Shiftability of Program Blocks and Instructions

Let i be an instruction of SCMPDS and let n be a natural number. We say that i valid at n if and only if the conditions (Def. 11) are satisfied.

(Def. 11)(i) If $\mathrm{InsCode}(i) = 0$, then there exists k_1 such that $i = \mathrm{goto}\, k_1$ and $n + k_1 \geqslant 0$,

(ii) if $\mathrm{InsCode}(i) = 4$, then there exist a, k_1, k_2 such that $i = (a, k_1) <> 0\_gotok_2$ and $n + k_2 \geqslant 0$,

(iii) if $\mathrm{InsCode}(i) = 5$, then there exist a, k_1, k_2 such that $i = (a, k_1) <= 0\_gotok_2$ and $n + k_2 \geqslant 0$, and

(iv) if $\mathrm{InsCode}(i) = 6$, then there exist a, k_1, k_2 such that $i = (a, k_1) >= 0\_gotok_2$ and $n + k_2 \geqslant 0$.

One can prove the following proposition

(77) Let i be an instruction of SCMPDS and m, n be natural numbers. If i valid at m and $m \leqslant n$, then i valid at n.

Let I_1 be a finite partial state of SCMPDS. We say that I_1 is shiftable if and only if:

(Def. 12) For all n, i such that $\mathrm{inspos}\, n \in \mathrm{dom}\, I_1$ and $i = I_1(\mathrm{inspos}\, n)$ holds $\mathrm{InsCode}(i) \neq 1$ and $\mathrm{InsCode}(i) \neq 3$ and i valid at n.

Let us mention that there exists a Program-block which is parahalting and shiftable.

Let i be an instruction of SCMPDS. We say that i is shiftable if and only if:

(Def. 13) $\mathrm{InsCode}(i) = 2$ or $\mathrm{InsCode}(i) > 6$.

One can check that there exists an instruction of SCMPDS which is shiftable.

Let us consider a, k_1. Observe that $a{:=}k_1$ is shiftable.

Let us consider a, k_1, k_2. One can check that $a_{k_1}{:=}k_2$ is shiftable.

Let us consider a, k_1, k_2. Observe that $\mathrm{AddTo}(a, k_1, k_2)$ is shiftable.

Let us consider a, b, k_1, k_2. One can check the following observations:

* $\mathrm{AddTo}(a, k_1, b, k_2)$ is shiftable,
* $\mathrm{SubFrom}(a, k_1, b, k_2)$ is shiftable,
* $\mathrm{MultBy}(a, k_1, b, k_2)$ is shiftable,
* $\mathrm{Divide}(a, k_1, b, k_2)$ is shiftable, and
* $(a, k_1) := (b, k_2)$ is shiftable.

Let I, J be shiftable Program-block. Observe that $I;J$ is shiftable.

Let i be a shiftable instruction of SCMPDS. Observe that $\mathrm{Load}(i)$ is shiftable.

Let i be a shiftable instruction of SCMPDS and let J be a shiftable Program-block. Observe that $i;J$ is shiftable.

Let I be a shiftable Program-block and let j be a shiftable instruction of SCMPDS. Observe that $I;j$ is shiftable.

Let i, j be shiftable instructions of SCMPDS. Note that $i;j$ is shiftable.

Let us note that SCMPDS $-$ Stop is parahalting and shiftable.

Let I be a shiftable Program-block. One can verify that $\mathrm{stop}\, I$ is shiftable.

Next we state the proposition

(78) For every shiftable Program-block I and for every integer k_1 such that $\mathrm{card}\, I + k_1 \geqslant 0$ holds $I;\mathrm{goto}\ k_1$ is shiftable.

Let n be a natural number. Note that $\mathrm{Load}(\mathrm{goto}\ n)$ is shiftable.

One can prove the following proposition

(79) Let I be a shiftable Program-block, k_1, k_2 be integers, and a be a Int position. If $\mathrm{card}\, I + k_2 \geqslant 0$, then $I;((a, k_1) <> 0\_goto k_2)$ is shiftable.

Let k_1 be an integer, let a be a Int position, and let n be a natural number. Note that $\mathrm{Load}((a, k_1) <> 0\_goto n)$ is shiftable.

Next we state the proposition

(80) Let I be a shiftable Program-block, k_1, k_2 be integers, and a be a Int position. If $\mathrm{card}\, I + k_2 \geqslant 0$, then $I;((a, k_1) <= 0\_goto k_2)$ is shiftable.

Let k_1 be an integer, let a be a Int position, and let n be a natural number. Observe that $\mathrm{Load}((a, k_1) <= 0\_goto n)$ is shiftable.

One can prove the following proposition

(81) Let I be a shiftable Program-block, k_1, k_2 be integers, and a be a Int position. If $\mathrm{card}\, I + k_2 \geqslant 0$, then $I;((a, k_1) >= 0\_goto k_2)$ is shiftable.

Let k_1 be an integer, let a be a Int position, and let n be a natural number. Observe that $\mathrm{Load}((a, k_1) >= 0\_goto n)$ is shiftable.

We now state three propositions:

(82) Let s_1, s_2 be states of SCMPDS, n, m be natural numbers, and k_1 be an integer. If $\mathbf{IC}_{(s_1)} = \mathrm{inspos}\, m$ and $m + k_1 \geqslant 0$ and $\mathbf{IC}_{(s_1)} + n = \mathbf{IC}_{(s_2)}$, then $\mathrm{ICplusConst}(s_1, k_1) + n = \mathrm{ICplusConst}(s_2, k_1)$.

(83) Let s_1, s_2 be states of SCMPDS, n, m be natural numbers, and i be an instruction of SCMPDS. Suppose $\mathbf{IC}_{(s_1)} = \mathrm{inspos}\, m$ and i valid at m and $\mathrm{InsCode}(i) \neq 1$ and $\mathrm{InsCode}(i) \neq 3$ and $\mathbf{IC}_{(s_1)} + n = \mathbf{IC}_{(s_2)}$ and $s_1 \restriction \text{Data-Loc}_{\mathrm{SCM}} = s_2 \restriction \text{Data-Loc}_{\mathrm{SCM}}$. Then $\mathbf{IC}_{\mathrm{Exec}(i, s_1)} + n = \mathbf{IC}_{\mathrm{Exec}(i, s_2)}$ and $\mathrm{Exec}(i, s_1) \restriction \text{Data-Loc}_{\mathrm{SCM}} = \mathrm{Exec}(i, s_2) \restriction \text{Data-Loc}_{\mathrm{SCM}}$.

(84) Let J be a parahalting shiftable Program-block. Suppose Initialized(stop $J) \subseteq s_1$. Let n be a natural number. Suppose $\mathrm{Shift}(\mathrm{stop}\, J, n) \subseteq s_2$ and $\mathbf{IC}_{(s_2)} = \mathrm{inspos}\, n$ and $s_1 \restriction \text{Data-Loc}_{\mathrm{SCM}} = s_2 \restriction \text{Data-Loc}_{\mathrm{SCM}}$. Let i be a natural number. Then $\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)} + n = \mathbf{IC}_{(\mathrm{Computation}(s_2))(i)}$ and $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{CurInstr}((\mathrm{Computation}(s_2))(i))$ and $(\mathrm{Computation}(s_1))(i) \restriction \text{Data-Loc}_{\mathrm{SCM}} = (\mathrm{Computation}(s_2))(i) \restriction \text{Data-Loc}_{\mathrm{SCM}}$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.

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

[3] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.

[4] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.

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

[6] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.

[7] Czesław Byliński. Products and coproducts in categories. *Formalized Mathematics*, 2(**5**):701–709, 1991.

[8] Jing-Chao Chen. Computation and program shift in the SCMPDS computer. *Formalized Mathematics*, 8(**1**):193–199, 1999.

[9] Jing-Chao Chen. The SCMPDS computer and the basic semantics of its instructions. *Formalized Mathematics*, 8(**1**):183–191, 1999.

[10] Jing-Chao Chen. A small computer model with push-down stack. *Formalized Mathematics*, 8(**1**):175–182, 1999.

[11] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.

[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] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.

[16] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.

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

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

[19] Andrzej Trybulec and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\rm FSA}$. *Formalized Mathematics*, 5(**4**):571–576, 1996.
[20] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[21] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.
[22] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[23] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[24] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received June 15, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

Computation of Two Consecutive Program Blocks for SCMPDS[1]

Jing-Chao Chen
Shanghai Jiaotong University

Summary. In this article, a program block without halting instructions is called No-StopCode program block. If a program consists of two blocks, where the first block is parahalting (i.e. halt for all states) and No-StopCode, and the second block is parahalting and shiftable, it can be computed by combining the computation results of the two blocks. For a program which consists of a instruction and a block, we obtain a similar conclusion. For a large amount of programs, the computation method given in the article is useful, but it is not suitable to recursive programs.

MML Identifier: SCMPDS_5.

The terminology and notation used here have been introduced in the following articles: [16], [20], [11], [21], [5], [6], [18], [2], [12], [13], [17], [14], [4], [10], [9], [19], [7], [1], [15], [8], and [3].

1. Preliminaries

For simplicity, we use the following convention: x denotes a set, m, n denote natural numbers, a, b denote Int position, i denotes an instruction of SCMPDS, s, s_1, s_2 denote states of SCMPDS, k_1, k_2 denote integers, l_1 denotes an instruction-location of SCMPDS, I, J denote Program-block, and N denotes a set with non empty elements.

One can prove the following propositions:

[1]This research is partially supported by the National Natural Science Foundation of China Grant No. 69873033.

ISSN 1426–2630

(1) Let S be a halting von Neumann definite AMI over N and s be a state of S. If $s = \mathrm{Following}(s)$, then for every n holds $(\mathrm{Computation}(s))(n) = s$.

(2) $x \in \mathrm{dom}\,\mathrm{Load}(i)$ iff $x = \mathrm{inspos}\,0$.

(3) If $l_1 \in \mathrm{dom}\,\mathrm{stop}\,I$ and $(\mathrm{stop}\,I)(l_1) \neq \mathbf{halt}_{\mathrm{SCMPDS}}$, then $l_1 \in \mathrm{dom}\,I$.

(4) $\mathrm{dom}\,\mathrm{Load}(i) = \{\mathrm{inspos}\,0\}$ and $(\mathrm{Load}(i))(\mathrm{inspos}\,0) = i$.

(5) $\mathrm{inspos}\,0 \in \mathrm{dom}\,\mathrm{Load}(i)$.

(6) $\mathrm{card}\,\mathrm{Load}(i) = 1$.

(7) $\mathrm{card}\,\mathrm{stop}\,I = \mathrm{card}\,I + 1$.

(8) $\mathrm{card}\,\mathrm{stop}\,\mathrm{Load}(i) = 2$.

(9) $\mathrm{inspos}\,0 \in \mathrm{dom}\,\mathrm{stop}\,\mathrm{Load}(i)$ and $\mathrm{inspos}\,1 \in \mathrm{dom}\,\mathrm{stop}\,\mathrm{Load}(i)$.

(10) $(\mathrm{stop}\,\mathrm{Load}(i))(\mathrm{inspos}\,0) = i$ and $(\mathrm{stop}\,\mathrm{Load}(i))(\mathrm{inspos}\,1) = \mathbf{halt}_{\mathrm{SCMPDS}}$.

(11) $x \in \mathrm{dom}\,\mathrm{stop}\,\mathrm{Load}(i)$ iff $x = \mathrm{inspos}\,0$ or $x = \mathrm{inspos}\,1$.

(12) $\mathrm{dom}\,\mathrm{stop}\,\mathrm{Load}(i) = \{\mathrm{inspos}\,0, \mathrm{inspos}\,1\}$.

(13) $\mathrm{inspos}\,0 \in \mathrm{dom}\,\mathrm{Initialized}(\mathrm{stop}\,\mathrm{Load}(i))$ and $\mathrm{inspos}\,1 \in \mathrm{dom}\,\mathrm{Initialized}(\mathrm{stop}\,\mathrm{Load}(i))$ and $(\mathrm{Initialized}(\mathrm{stop}\,\mathrm{Load}(i)))(\mathrm{inspos}\,0) = i$ and $(\mathrm{Initialized}(\mathrm{stop}\,\mathrm{Load}(i)))(\mathrm{inspos}\,1) = \mathbf{halt}_{\mathrm{SCMPDS}}$.

(14) For all Program-block I, J holds $\mathrm{Initialized}(\mathrm{stop}\,I;J) = (I;(J;\mathrm{SCMPDS} - \mathrm{Stop})) +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\,0)$.

(15) For all Program-block I, J holds $\mathrm{Initialized}(I) \subseteq \mathrm{Initialized}(\mathrm{stop}\,I;J)$.

(16) $\mathrm{dom}\,\mathrm{stop}\,I \subseteq \mathrm{dom}\,\mathrm{stop}\,I;J$.

(17) For all Program-block I, J holds $\mathrm{Initialized}(\mathrm{stop}\,I) +\cdot \mathrm{Initialized}(\mathrm{stop}\,I;J) = \mathrm{Initialized}(\mathrm{stop}\,I;J)$.

(18) If $\mathrm{Initialized}(I) \subseteq s$, then $\mathbf{IC}_s = \mathrm{inspos}\,0$.

(19) $(s +\cdot \mathrm{Initialized}(I))(a) = s(a)$.

(20) Let I be a parahalting Program-block. Suppose $\mathrm{Initialized}(\mathrm{stop}\,I) \subseteq s_1$ and $\mathrm{Initialized}(\mathrm{stop}\,I) \subseteq s_2$ and s_1 and s_2 are equal outside the instruction locations of SCMPDS. Let k be a natural number. Then $(\mathrm{Computation}(s_1))(k)$ and $(\mathrm{Computation}(s_2))(k)$ are equal outside the instruction locations of SCMPDS and $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(k)) = \mathrm{CurInstr}((\mathrm{Computation}(s_2))(k))$.

(21) Let I be a parahalting Program-block. Suppose $\mathrm{Initialized}(\mathrm{stop}\,I) \subseteq s_1$ and $\mathrm{Initialized}(\mathrm{stop}\,I) \subseteq s_2$ and s_1 and s_2 are equal outside the instruction locations of SCMPDS. Then $\mathrm{LifeSpan}(s_1) = \mathrm{LifeSpan}(s_2)$ and $\mathrm{Result}(s_1)$ and $\mathrm{Result}(s_2)$ are equal outside the instruction locations of SCMPDS.

(22) For every Program-block I holds $\mathbf{IC}_{\mathrm{IExec}(I,s)} = \mathbf{IC}_{\mathrm{Result}(s +\cdot \mathrm{Initialized}(\mathrm{stop}\,I))}$.

(23) Let I be a parahalting Program-block and J be a Program-block. Suppose $\mathrm{Initialized}(\mathrm{stop}\,I) \subseteq s$. Let given m. Suppose $m \leqslant \mathrm{LifeSpan}(s)$. Then $(\mathrm{Computation}(s))(m)$ and $(\mathrm{Computation}(s +\cdot (I;J)))(m)$ are equal outside the instruction locations of SCMPDS.

(24) Let I be a parahalting Program-block and J be a Program-block. Suppose Initialized(stop I) $\subseteq s$. Let given m. Suppose $m \leqslant$ LifeSpan(s). Then (Computation(s))(m) and (Computation(s+· Initialized(stop I;J)))(m) are equal outside the instruction locations of SCMPDS.

2. Non Halting Instructions and Parahalting Instructions

Let i be an instruction of SCMPDS. We say that i is No-StopCode if and only if:

(Def. 1) $i \neq \mathbf{halt}_{\mathrm{SCMPDS}}$.

Let i be an instruction of SCMPDS. We say that i is parahalting if and only if:

(Def. 2) Load(i) is parahalting.

One can verify that there exists an instruction of SCMPDS which is No-StopCode, shiftable, and parahalting.

One can prove the following proposition

(25) If $k_1 \neq 0$, then goto k_1 is No-StopCode.

Let us consider a. Observe that return a is No-StopCode.

Let us consider a, k_1. Note that a:=k_1 is No-StopCode and saveIC(a, k_1) is No-StopCode.

Let us consider a, k_1, k_2. One can check the following observations:

* (a, k_1) <> 0_*goto*k_2 is No-StopCode,
* (a, k_1) <= 0_*goto*k_2 is No-StopCode,
* (a, k_1) >= 0_*goto*k_2 is No-StopCode, and
* a_{k_1}:=k_2 is No-StopCode.

Let us consider a, k_1, k_2. Note that AddTo(a, k_1, k_2) is No-StopCode.

Let us consider a, b, k_1, k_2. One can verify the following observations:

* AddTo(a, k_1, b, k_2) is No-StopCode,
* SubFrom(a, k_1, b, k_2) is No-StopCode,
* MultBy(a, k_1, b, k_2) is No-StopCode,
* Divide(a, k_1, b, k_2) is No-StopCode, and
* $(a, k_1) := (b, k_2)$ is No-StopCode.

Let us note that $\mathbf{halt}_{\mathrm{SCMPDS}}$ is parahalting.

Let i be a parahalting instruction of SCMPDS. Observe that Load(i) is parahalting.

Let us consider a, k_1. Observe that a:=k_1 is parahalting.

Let us consider a, k_1, k_2. Note that a_{k_1}:=k_2 is parahalting and AddTo(a, k_1, k_2) is parahalting.

Let us consider a, b, k_1, k_2. One can check the following observations:

* AddTo(a, k_1, b, k_2) is parahalting,
* SubFrom(a, k_1, b, k_2) is parahalting,
* MultBy(a, k_1, b, k_2) is parahalting,
* Divide(a, k_1, b, k_2) is parahalting, and
* $(a, k_1) := (b, k_2)$ is parahalting.

Next we state the proposition

(26) If InsCode$(i) = 1$, then i is not parahalting.

Let I_1 be a finite partial state of SCMPDS. We say that I_1 is No-StopCode if and only if:

(Def. 3) For every instruction-location x of SCMPDS such that $x \in \operatorname{dom} I_1$ holds $I_1(x) \neq \mathbf{halt}_{\mathrm{SCMPDS}}$.

Let us observe that there exists a Program-block which is parahalting, shiftable, and No-StopCode.

Let I, J be No-StopCode Program-block. Observe that I;J is No-StopCode.

Let i be a No-StopCode instruction of SCMPDS. Observe that Load(i) is No-StopCode.

Let i be a No-StopCode instruction of SCMPDS and let J be a No-StopCode Program-block. Note that i;J is No-StopCode.

Let I be a No-StopCode Program-block and let j be a No-StopCode instruction of SCMPDS. Observe that I;j is No-StopCode.

Let i, j be No-StopCode instructions of SCMPDS. Observe that i;j is No-StopCode.

Next we state several propositions:

(27) For every parahalting No-StopCode Program-block I such that $\mathrm{Initialized}(\mathrm{stop}\, I) \subseteq s$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s +\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))} = \mathrm{inspos}\, \mathrm{card}\, I$.

(28) For every parahalting Program-block I and for every natural number k such that $k < \mathrm{LifeSpan}(s +\cdot \mathrm{Initialized}(\mathrm{stop}\, I))$ holds $\mathbf{IC}_{(\mathrm{Computation}(s +\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))(k)} \in \operatorname{dom} I$.

(29) Let I be a parahalting Program-block and k be a natural number. Suppose $\mathrm{Initialized}(I) \subseteq s$ and $k \leqslant \mathrm{LifeSpan}(s +\cdot \mathrm{Initialized}(\mathrm{stop}\, I))$. Then $(\mathrm{Computation}(s))(k)$ and $(\mathrm{Computation}(s +\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))(k)$ are equal outside the instruction locations of SCMPDS.

(30) For every parahalting No-StopCode Program-block I such that $\mathrm{Initialized}(I) \subseteq s$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s +\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))} = \mathrm{inspos}\, \mathrm{card}\, I$.

(31) For every parahalting Program-block I such that $\mathrm{Initialized}(I) \subseteq s$ holds $\mathrm{CurInstr}((\mathrm{Computation}(s))(\mathrm{LifeSpan}(s +\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))) = \mathbf{halt}_{\mathrm{SCMPDS}}$ or $\mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s +\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))} = \mathrm{inspos}\, \mathrm{card}\, I$.

(32) Let I be a parahalting No-StopCode Program-block and k be a natural number. If Initialized(I) $\subseteq s$ and $k <$ LifeSpan(s+· Initialized(stop I)), then CurInstr((Computation(s))(k)) $\neq$ $\mathbf{halt}_{\mathrm{SCMPDS}}$.

(33) Let I be a parahalting Program-block, J be a Program-block, and k be a natural number. Suppose $k \leqslant$ LifeSpan(s+· Initialized(stop I)). Then (Computation(s+· Initialized(stop I)))(k) and (Computation(s+·((I;J)+· Start-At(inspos 0))))(k) are equal outside the instruction locations of SCMPDS.

(34) Let I be a parahalting Program-block, J be a Program-block, and k be a natural number. Suppose $k \leqslant$ LifeSpan(s+· Initialized(stop I)). Then (Computation(s+· Initialized(stop I)))(k) and (Computation(s+· Initialized (stop I;J)))(k) are equal outside the instruction locations of SCMPDS.

Let I be a parahalting Program-block and let J be a parahalting shiftable Program-block. One can verify that I;J is parahalting.

Let i be a parahalting instruction of SCMPDS and let J be a parahalting shiftable Program-block. Note that i;J is parahalting.

Let I be a parahalting Program-block and let j be a parahalting shiftable instruction of SCMPDS. Observe that I;j is parahalting.

Let i be a parahalting instruction of SCMPDS and let j be a parahalting shiftable instruction of SCMPDS. One can check that i;j is parahalting.

Next we state the proposition

(35) Let s, s_1 be states of SCMPDS and J be a parahalting shiftable Program-block. If $s =$ (Computation(s_1+· Initialized(stop J)))(m), then Exec(CurInstr(s), s+· Start-At($\mathbf{IC}_s + n$)) = Following(s)+· Start-At($\mathbf{IC}_{\mathrm{Following}(s)} + n$).

3. Computation of two Consecutive Program Blocks

The following propositions are true:

(36) Let I be a parahalting No-StopCode Program-block, J be a parahalting shiftable Program-block, and k be a natural number. Suppose Initialized(stop I;J) $\subseteq s$. Then (Computation(Result(s+· Initialized (stop I))+· Initialized(stop J)))(k)+· Start-At ($\mathbf{IC}_{(\mathrm{Computation}(\mathrm{Result}(s+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,I))+\cdot\,\mathrm{Initialized}(\mathrm{stop}\,J)))(k)}$ + card I) and (Computation(s+· Initialized(stop I;J)))(LifeSpan(s+· Initialized(stop I))+ k) are equal outside the instruction locations of SCMPDS.

(37) Let I be a parahalting No-StopCode Program-block and J be a parahalting shiftable Program-block. Then LifeSpan(s+· Initialized(stop I;J)) = LifeSpan(s+· Initialized(stop I))+LifeSpan(Result(s+· Initialized(stop I))+· Initialized(stop J)).

(38) Let I be a parahalting No-StopCode Program-block and J be a parahalting shiftable Program-block. Then $\mathrm{IExec}(I;J,s) = \mathrm{IExec}(J,\mathrm{IExec}(I,s)) +\cdot \text{Start-At}(\mathbf{IC}_{\mathrm{IExec}(J,\mathrm{IExec}(I,s))} + \mathrm{card}\, I)$.

(39) Let I be a parahalting No-StopCode Program-block and J be a parahalting shiftable Program-block. Then $(\mathrm{IExec}(I;J,s))(a) = (\mathrm{IExec}(J,\mathrm{IExec}(I,s)))(a)$.

4. Computation of the Program Consisting of a Instruction and a Block

Let s be a state of SCMPDS. The functor Initialized(s) yields a state of SCMPDS and is defined by:

(Def. 4) $\mathrm{Initialized}(s) = s +\cdot \text{Start-At}(\mathrm{inspos}\, 0)$.

Next we state several propositions:

(40) $\mathbf{IC}_{\mathrm{Initialized}(s)} = \mathrm{inspos}\, 0$ and $(\mathrm{Initialized}(s))(a) = s(a)$ and $(\mathrm{Initialized}(s))(l_1) = s(l_1)$.

(41) s_1 and s_2 are equal outside the instruction locations of SCMPDS iff $s_1 \restriction (\text{Data-Loc}_{\mathrm{SCM}} \cup \{\mathbf{IC}_{\mathrm{SCMPDS}}\}) = s_2 \restriction (\text{Data-Loc}_{\mathrm{SCM}} \cup \{\mathbf{IC}_{\mathrm{SCMPDS}}\})$.

(42) If $s_1 \restriction \text{Data-Loc}_{\mathrm{SCM}} = s_2 \restriction \text{Data-Loc}_{\mathrm{SCM}}$, then $s_1(\mathrm{DataLoc}(s_1(a), k_1)) = s_2(\mathrm{DataLoc}(s_2(a), k_1))$.

(43) If $s_1 \restriction \text{Data-Loc}_{\mathrm{SCM}} = s_2 \restriction \text{Data-Loc}_{\mathrm{SCM}}$ and $\mathrm{InsCode}(i) \neq 3$, then $\mathrm{Exec}(i, s_1) \restriction \text{Data-Loc}_{\mathrm{SCM}} = \mathrm{Exec}(i, s_2) \restriction \text{Data-Loc}_{\mathrm{SCM}}$.

(44) For every shiftable instruction i of SCMPDS such that $s_1 \restriction \text{Data-Loc}_{\mathrm{SCM}} = s_2 \restriction \text{Data-Loc}_{\mathrm{SCM}}$ holds $\mathrm{Exec}(i, s_1) \restriction \text{Data-Loc}_{\mathrm{SCM}} = \mathrm{Exec}(i, s_2) \restriction \text{Data-Loc}_{\mathrm{SCM}}$.

(45) For every parahalting instruction i of SCMPDS holds $\mathrm{Exec}(i, \mathrm{Initialized}(s)) = \mathrm{IExec}(\mathrm{Load}(i), s)$.

(46) Let I be a parahalting No-StopCode Program-block and j be a parahalting shiftable instruction of SCMPDS. Then $(\mathrm{IExec}(I;j,s))(a) = (\mathrm{Exec}(j,\mathrm{IExec}(I,s)))(a)$.

(47) Let i be a No-StopCode parahalting instruction of SCMPDS and j be a shiftable parahalting instruction of SCMPDS. Then $(\mathrm{IExec}(i;j,s))(a) = (\mathrm{Exec}(j,\mathrm{Exec}(i,\mathrm{Initialized}(s))))(a)$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.

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

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

[4] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.
[5] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[6] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[7] Jing-Chao Chen. Computation and program shift in the SCMPDS computer. *Formalized Mathematics*, 8(**1**):193–199, 1999.
[8] Jing-Chao Chen. The construction and shiftability of program blocks for SCMPDS. *Formalized Mathematics*, 8(**1**):201–210, 1999.
[9] Jing-Chao Chen. The SCMPDS computer and the basic semantics of its instructions. *Formalized Mathematics*, 8(**1**):183–191, 1999.
[10] Jing-Chao Chen. A small computer model with push-down stack. *Formalized Mathematics*, 8(**1**):175–182, 1999.
[11] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[12] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[13] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.
[14] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[15] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[16] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[17] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[18] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[19] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.
[20] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[21] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received June 15, 1999

FORMALIZED MATHEMATICS
Volume 8, Number 1, 1999
University of Białystok

The Construction and Computation of Conditional Statements for SCMPDS[1]

Jing-Chao Chen
Shanghai Jiaotong University

Summary. We construct conditional statements like the usual high level program language by program blocks of SCMPDS. Roughly speaking, the article justifies such a fact that when the condition of a conditional statement is true (false), and the true (false) branch is shiftable, parahalting and does not contain any halting instruction, and the false branch is shiftable, then it is halting and its computation result equals that of the true (false) branch. The parahalting means some program halts for all states, this is strong condition. For this reason, we introduce the notions of "is_closed_on" and "is_halting_on". The predicate "A is_closed_on B" denotes program A is closed on state B, and "A is_halting_on B" denotes program A is halting on state B. We obtain a similar theorem to the above fact by replacing parahalting by "is_closed_on" and "is_halting_on".

MML Identifier: `SCMPDS_6`.

The terminology and notation used in this paper are introduced in the following papers: [16], [19], [11], [14], [20], [5], [6], [18], [2], [12], [13], [17], [15], [4], [10], [7], [1], [9], [3], and [8].

1. Preliminaries

For simplicity, we follow the rules: a denotes a Int position, i denotes an instruction of SCMPDS, s, s_1, s_2 denote states of SCMPDS, k_1 denotes an integer, l_1 denotes an instruction-location of SCMPDS, and I, J denote Program-block.

One can prove the following propositions:

[1]This research is partially supported by the National Natural Science Foundation of China Grant No. 69873033.

ISSN 1426–2630

(1) For every state s of SCMPDS holds dom($s\restriction$the instruction locations of SCMPDS) = the instruction locations of SCMPDS.

(2) For every state s of SCMPDS such that s is halting and for every natural number k such that LifeSpan(s) $\leqslant k$ holds CurInstr((Computation(s))(k)) = $\mathbf{halt}_{\mathrm{SCMPDS}}$.

(3) For every state s of SCMPDS such that s is halting and for every natural number k such that LifeSpan(s) $\leqslant k$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s))}$.

(4) Let s_1, s_2 be states of SCMPDS. Then s_1 and s_2 are equal outside the instruction locations of SCMPDS if and only if $\mathbf{IC}_{(s_1)} = \mathbf{IC}_{(s_2)}$ and $s_1\restriction\mathrm{Data\text{-}Loc}_{\mathrm{SCM}} = s_2\restriction\mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$.

(5) For every state s of SCMPDS and for every Program-block I holds Initialized(s)+· Initialized(I) = s+· Initialized(I).

(6) For every Program-block I and for every instruction-location l of SCMPDS holds $I \subseteq I$+· Start-At(l).

(7) For every state s of SCMPDS and for every instruction-location l of SCMPDS holds $s\restriction\mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$ = (s+· Start-At(l))$\restriction\mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$.

(8) For every state s of SCMPDS and for every Program-block I and for every instruction-location l of SCMPDS holds $s\restriction\mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$ = (s+·(I+· Start-At(l)))$\restriction\mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$.

(9) For every state s of SCMPDS and for every Program-block I holds $s\restriction\mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$ = (s+· Initialized(I))$\restriction\mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$.

(10) Let s be a state of SCMPDS and l be an instruction-location of SCMPDS. Then dom($s\restriction$the instruction locations of SCMPDS) misses dom Start-At(l).

(11) Let s be a state of SCMPDS, I, J be Program-block, and l be an instruction-location of SCMPDS. Then s+·(I+· Start-At(l)) and s+·(J+· Start-At(l)) are equal outside the instruction locations of SCMPDS.

(12) Let s_1, s_2 be states of SCMPDS and I, J be Program-block. Suppose $s_1\restriction\mathrm{Data\text{-}Loc}_{\mathrm{SCM}} = s_2\restriction\mathrm{Data\text{-}Loc}_{\mathrm{SCM}}$. Then s_1+· Initialized(I) and s_2+· Initialized(J) are equal outside the instruction locations of SCMPDS.

(13) Let I be a programmed finite partial state of SCMPDS and x be a set. If $x \in \mathrm{dom}\, I$, then $I(x)$ is an instruction of SCMPDS.

(14) For every state s of SCMPDS and for all instructions-locations l_2, l_3 of SCMPDS holds s+· Start-At(l_2)+· Start-At(l_3) = s+· Start-At(l_3).

(15) card(i;I) = card $I + 1$.

(16) (i;I)(inspos 0) = i.

(17) $I \subseteq$ Initialized(stop I).

(18) If $l_1 \in \operatorname{dom} I$, then $l_1 \in \operatorname{dom} \operatorname{stop} I$.

(19) If $l_1 \in \operatorname{dom} I$, then $(\operatorname{stop} I)(l_1) = I(l_1)$.

(20) If $l_1 \in \operatorname{dom} I$, then $(\mathrm{Initialized}(\operatorname{stop} I))(l_1) = I(l_1)$.

(21) $\mathbf{IC}_{s+\cdot\,\mathrm{Initialized}(I)} = \mathrm{inspos}\,0$.

(22) $\mathrm{CurInstr}(s+\cdot\,\mathrm{Initialized}(\operatorname{stop} i;I)) = i$.

(23) For every state s of SCMPDS and for all natural numbers m_1, m_2 such that $\mathbf{IC}_s = \mathrm{inspos}\,m_1$ holds $\mathrm{ICplusConst}(s, m_2) = \mathrm{inspos}\,m_1 + m_2$.

(24) For all Program-block I, J holds $\mathrm{Shift}(\operatorname{stop} J, \operatorname{card} I) \subseteq \operatorname{stop} I;J$.

(25) $\mathrm{inspos}\,\operatorname{card} I \in \operatorname{dom} \operatorname{stop} I$ and $(\operatorname{stop} I)(\mathrm{inspos}\,\operatorname{card} I) = \mathbf{halt}_{\mathrm{SCMPDS}}$.

(26) For all instructions-locations x, l of SCMPDS holds $(\mathrm{IExec}(J, s))(x) = (\mathrm{IExec}(I, s)+\cdot\,\mathrm{Start\text{-}At}(l))(x)$.

(27) For all instructions-locations x, l of SCMPDS holds $(\mathrm{IExec}(I, s))(x) = (s+\cdot\,\mathrm{Start\text{-}At}(l))(x)$.

(28) Let s be a state of SCMPDS, i be a No-StopCode parahalting instruction of SCMPDS, J be a parahalting shiftable Program-block, and a be a Int position. Then $(\mathrm{IExec}(i;J, s))(a) = (\mathrm{IExec}(J, \mathrm{Exec}(i, \mathrm{Initialized}(s))))(a)$.

(29) For every Int position a and for all integers k_1, k_2 holds $(a, k_1) <> 0\_goto k_2 \neq \mathbf{halt}_{\mathrm{SCMPDS}}$.

(30) For every Int position a and for all integers k_1, k_2 holds $(a, k_1) <= 0\_goto k_2 \neq \mathbf{halt}_{\mathrm{SCMPDS}}$.

(31) For every Int position a and for all integers k_1, k_2 holds $(a, k_1) >= 0\_goto k_2 \neq \mathbf{halt}_{\mathrm{SCMPDS}}$.

Let us consider k_1. The functor $\mathrm{Goto}(k_1)$ yielding a Program-block is defined as follows:

(Def. 1) $\mathrm{Goto}(k_1) = \mathrm{Load}(\mathrm{goto}\ k_1)$.

Let n be a natural number. One can verify that goto $(n+1)$ is No-StopCode and goto $(-(n+1))$ is No-StopCode.

Let n be a natural number. Observe that $\mathrm{Goto}(n+1)$ is No-StopCode and $\mathrm{Goto}(-(n+1))$ is No-StopCode.

The following two propositions are true:

(32) $\operatorname{card} \mathrm{Goto}(k_1) = 1$.

(33) $\mathrm{inspos}\,0 \in \operatorname{dom} \mathrm{Goto}(k_1)$ and $(\mathrm{Goto}(k_1))(\mathrm{inspos}\,0) = \mathrm{goto}\ k_1$.

2. The Predicates of is_closed_on and is_halting_on

Let I be a Program-block and let s be a state of SCMPDS. We say that I is closed on s if and only if:

(Def. 2) For every natural number k holds $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))(k)} \in \mathrm{dom}\, \mathrm{stop}\, I$.

We say that I is halting on s if and only if:

(Def. 3) $s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I)$ is halting.

We now state a number of propositions:

(34) For every Program-block I holds I is paraclosed iff for every state s of SCMPDS holds I is closed on s.

(35) For every Program-block I holds I is parahalting iff for every state s of SCMPDS holds I is halting on s.

(36) Let s_1, s_2 be states of SCMPDS and I be a Program-block. If $s_1 \restriction \text{Data-Loc}_{\text{SCM}} = s_2 \restriction \text{Data-Loc}_{\text{SCM}}$, then if I is closed on s_1, then I is closed on s_2.

(37) Let s_1, s_2 be states of SCMPDS and I be a Program-block. Suppose $s_1 \restriction \text{Data-Loc}_{\text{SCM}} = s_2 \restriction \text{Data-Loc}_{\text{SCM}}$. Suppose I is closed on s_1 and halting on s_1. Then I is closed on s_2 and halting on s_2.

(38) For every state s of SCMPDS and for all Program-block I, J holds I is closed on s iff I is closed on $s+\cdot \mathrm{Initialized}(J)$.

(39) Let I, J be Program-block and s be a state of SCMPDS. Suppose I is closed on s and halting on s. Then

(i) for every natural number k such that $k \leqslant \mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I))$ holds $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))(k)} = \mathbf{IC}_{(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I;J)))(k)}$, and

(ii) $(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))(\mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I))) \restriction \text{Data-Loc}_{\text{SCM}} = (\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I;J)))(\mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I))) \restriction \text{Data-Loc}_{\text{SCM}}$.

(40) Let I be a Program-block and k be a natural number. If I is closed on s and halting on s and $k < \mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I))$, then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))(k)} \in \mathrm{dom}\, I$.

(41) Let I, J be Program-block, s be a state of SCMPDS, and k be a natural number. Suppose I is closed on s and halting on s and $k < \mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I))$. Then $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))(k)) = \mathrm{CurInstr}((\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I;J)))(k))$.

(42) Let I be a No-StopCode Program-block, s be a state of SCMPDS, and k be a natural number. If I is closed on s and halting on s and $k < \mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I))$, then $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))(k)) \neq \mathbf{halt}_{\text{SCMPDS}}$.

(43) Let I be a No-StopCode Program-block and s be a state of SCMPDS. If I is closed on s and halting on s, then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))(\mathrm{LifeSpan}(s+\cdot \mathrm{Initialized}(\mathrm{stop}\, I)))} = \mathrm{inspos}\, \mathrm{card}\, I$.

(44) Let I, J be Program-block and s be a state of SCMPDS. Suppose I is closed on s and halting on s. Then $I;\mathrm{Goto}(\mathrm{card}\,J+1);J$ is halting on s and $I;\mathrm{Goto}(\mathrm{card}\,J+1);J$ is closed on s.

(45) Let I be a shiftable Program-block. Suppose $\mathrm{Initialized}(\mathrm{stop}\,I) \subseteq s_1$ and I is closed on s_1. Let n be a natural number. Suppose $\mathrm{Shift}(\mathrm{stop}\,I, n) \subseteq s_2$ and $\mathbf{IC}_{(s_2)} = \mathrm{inspos}\,n$ and $s_1 \restriction \text{Data-Loc}_{\mathrm{SCM}} = s_2 \restriction \text{Data-Loc}_{\mathrm{SCM}}$. Let i be a natural number. Then $\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)} + n = \mathbf{IC}_{(\mathrm{Computation}(s_2))(i)}$ and $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{CurInstr}((\mathrm{Computation}(s_2))(i))$ and $(\mathrm{Computation}(s_1))(i) \restriction \text{Data-Loc}_{\mathrm{SCM}} = (\mathrm{Computation}(s_2))(i) \restriction \text{Data-Loc}_{\mathrm{SCM}}$.

(46) Let s be a state of SCMPDS, I be a No-StopCode Program-block, and J be a Program-block. If I is closed on s and halting on s, then $\mathbf{IC}_{\mathrm{IExec}(I;\,\mathrm{Goto}(\mathrm{card}\,J+1);J,s)} = \mathrm{inspos}\,\mathrm{card}\,I + \mathrm{card}\,J + 1$.

(47) Let s be a state of SCMPDS, I be a No-StopCode Program-block, and J be a Program-block. If I is closed on s and halting on s, then $\mathrm{IExec}(I;\mathrm{Goto}(\mathrm{card}\,J+1);J,s) = \mathrm{IExec}(I,s) +\cdot\, \text{Start-At}(\mathrm{inspos}\,\mathrm{card}\,I + \mathrm{card}\,J + 1)$.

(48) Let s be a state of SCMPDS and I be a No-StopCode Program-block. If I is closed on s and halting on s, then $\mathbf{IC}_{\mathrm{IExec}(I,s)} = \mathrm{inspos}\,\mathrm{card}\,I$.

3. The Construction of Conditional Statements

Let a be a Int position, let k be an integer, and let I, J be Program-block. The functor **if** $a = k$ **then** I **else** J yielding a Program-block is defined by:

(Def. 4) **if** $a = k$ **then** I **else** $J = ((a,k) <> 0\_goto\,\mathrm{card}\,I + 2);I;\mathrm{Goto}(\mathrm{card}\,J + 1);J$.

The functor **if** $a > k$ **then** I **else** J yielding a Program-block is defined by:

(Def. 5) **if** $a > k$ **then** I **else** $J = ((a,k) <= 0\_goto\,\mathrm{card}\,I + 2);I;\mathrm{Goto}(\mathrm{card}\,J + 1);J$.

The functor **if** $a < k$ **then** I **else** J yielding a Program-block is defined by:

(Def. 6) **if** $a < k$ **then** I **else** $J = ((a,k) >= 0\_goto\,\mathrm{card}\,I + 2);I;\mathrm{Goto}(\mathrm{card}\,J + 1);J$.

Let a be a Int position, let k be an integer, and let I be a Program-block. The functor **if** $a = 0$ **then** k **else** I yields a Program-block and is defined as follows:

(Def. 7) **if** $a = 0$ **then** k **else** $I = ((a,k) <> 0\_goto\,\mathrm{card}\,I + 1);I$.

The functor **if** $a \neq 0$ **then** k **else** I yielding a Program-block is defined by:

(Def. 8) **if** $a \neq 0$ **then** k **else** $I = ((a,k) <> 0\_goto2);\mathrm{goto}\ (\mathrm{card}\,I + 1);I$.

The functor **if** $a > 0$ **then** k **else** I yielding a Program-block is defined as follows:

(Def. 9) **if** $a > 0$ **then** k **else** $I = ((a, k) <= 0\_goto \operatorname{card} I + 1);I$.

The functor **if** $a \leqslant 0$ **then** k **else** I yields a Program-block and is defined as follows:

(Def. 10) **if** $a \leqslant 0$ **then** k **else** $I = ((a, k) <= 0\_goto 2);\text{goto } (\operatorname{card} I + 1);I$.

The functor **if** $a < 0$ **then** k **else** I yields a Program-block and is defined as follows:

(Def. 11) **if** $a < 0$ **then** k **else** $I = ((a, k) >= 0\_goto \operatorname{card} I + 1);I$.

The functor **if** $a \geqslant 0$ **then** k **else** I yields a Program-block and is defined as follows:

(Def. 12) **if** $a \geqslant 0$ **then** k **else** $I = ((a, k) >= 0\_goto 2);\text{goto } (\operatorname{card} I + 1);I$.

4. The Computation of "if var=0 then block1 else block2"

One can prove the following propositions:

(49) card(**if** $a = k_1$ **then** I **else** J) $= \operatorname{card} I + \operatorname{card} J + 2$.

(50) inspos 0 $\in$ dom(**if** $a = k_1$ **then** I **else** J) and inspos 1 $\in$ dom(**if** $a = k_1$ **then** I **else** J).

(51) (**if** $a = k_1$ **then** I **else** J)(inspos 0) $= (a, k_1) <> 0\_goto \operatorname{card} I + 2$.

(52) Let s be a state of SCMPDS, I, J be shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\text{DataLoc}(s(a), k_1)) = 0$ and I is closed on s and halting on s. Then **if** $a = k_1$ **then** I **else** J is closed on s and **if** $a = k_1$ **then** I **else** J is halting on s.

(53) Let s be a state of SCMPDS, I be a Program-block, J be a shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\text{DataLoc}(s(a), k_1)) \neq 0$ and J is closed on s and halting on s. Then **if** $a = k_1$ **then** I **else** J is closed on s and **if** $a = k_1$ **then** I **else** J is halting on s.

(54) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, J be a shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\text{DataLoc}(s(a), k_1)) = 0$ and I is closed on s and halting on s. Then IExec(**if** $a = k_1$ **then** I **else** J, s) $= \text{IExec}(I, s) +\cdot \text{Start-At}(\text{inspos} \operatorname{card} I + \operatorname{card} J + 2)$.

(55) Let s be a state of SCMPDS, I be a Program-block, J be a No-StopCode shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\text{DataLoc}(s(a), k_1)) \neq 0$ and J is closed on s and halting on s. Then IExec(**if** $a = k_1$ **then** I **else** J, s) $= \text{IExec}(J, s) +\cdot \text{Start-At}(\text{inspos} \operatorname{card} I + \operatorname{card} J + 2)$.

Let I, J be shiftable parahalting Program-block, let a be a Int position, and let k_1 be an integer. Observe that **if** $a = k_1$ **then** I **else** J is shiftable and parahalting.

Let I, J be No-StopCode Program-block, let a be a Int position, and let k_1 be an integer. Note that **if** $a = k_1$ **then** I **else** J is No-StopCode.

We now state three propositions:

(56) Let s be a state of SCMPDS, I, J be No-StopCode shiftable parahalting Program-block, a be a Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a=k_1\ \mathbf{then}\ I\ \mathbf{else}\ J,s)} = \mathrm{inspos}\,\mathrm{card}\,I + \mathrm{card}\,J + 2$.

(57) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, J be a shiftable Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) = 0$, then $(\mathrm{IExec}(\mathbf{if}\ a = k_1\ \mathbf{then}\ I\ \mathbf{else}\ J, s))(b) = (\mathrm{IExec}(I, s))(b)$.

(58) Let s be a state of SCMPDS, I be a Program-block, J be a No-StopCode parahalting shiftable Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \neq 0$, then $(\mathrm{IExec}(\mathbf{if}\ a = k_1\ \mathbf{then}\ I\ \mathbf{else}\ J, s))(b) = (\mathrm{IExec}(J, s))(b)$.

5. The Computation of "if var=0 then block"

One can prove the following propositions:

(59) $\mathrm{card}(\mathbf{if}\ a = 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I) = \mathrm{card}\,I + 1$.

(60) $\mathrm{inspos}\,0 \in \mathrm{dom}(\mathbf{if}\ a = 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I)$.

(61) $(\mathbf{if}\ a = 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I)(\mathrm{inspos}\,0) = (a, k_1) <> 0\_goto\,\mathrm{card}\,I + 1$.

(62) Let s be a state of SCMPDS, I be a shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) = 0$ and I is closed on s and halting on s. Then **if** $a = 0$ **then** k_1 **else** I is closed on s and **if** $a = 0$ **then** k_1 **else** I is halting on s.

(63) Let s be a state of SCMPDS, I be a Program-block, a be a Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \neq 0$, then **if** $a = 0$ **then** k_1 **else** I is closed on s and **if** $a = 0$ **then** k_1 **else** I is halting on s.

(64) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) = 0$ and I is closed on s and halting on s. Then $\mathrm{IExec}(\mathbf{if}\ a = 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I, s) = \mathrm{IExec}(I, s) +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\,\mathrm{card}\,I + 1)$.

(65) Let s be a state of SCMPDS, I be a Program-block, a be a Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \neq 0$, then $\mathrm{IExec}(\mathbf{if}\ a =$

0 **then** k_1 **else** $I, s) = s +\cdot$ Start-At(inspos card $I + 1$).

Let I be a shiftable parahalting Program-block, let a be a Int position, and let k_1 be an integer. One can verify that **if** $a = 0$ **then** k_1 **else** I is shiftable and parahalting.

Let I be a No-StopCode Program-block, let a be a Int position, and let k_1 be an integer. Observe that **if** $a = 0$ **then** k_1 **else** I is No-StopCode.

Next we state three propositions:

(66) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a be a Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a=0\ \mathbf{then}\ k_1\ \mathbf{else}\ I,s)} = \mathrm{inspos}\,\mathrm{card}\,I + 1$.

(67) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) = 0$, then (IExec(**if** $a = 0$ **then** k_1 **else** $I, s))(b) =$ (IExec$(I, s))(b)$.

(68) Let s be a state of SCMPDS, I be a Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \neq 0$, then (IExec(**if** $a =$ 0 **then** k_1 **else** $I, s))(b) = s(b)$.

6. The Computation of "if var<>0 then block"

One can prove the following propositions:

(69) card(**if** $a \neq 0$ **then** k_1 **else** I) = card $I + 2$.

(70) inspos 0 $\in$ dom(**if** $a \neq 0$ **then** k_1 **else** I) and inspos 1 $\in$ dom(**if** $a \neq$ 0 **then** k_1 **else** I).

(71) (**if** $a \neq 0$ **then** k_1 **else** I)(inspos 0) $= (a, k_1)$ <> 0_*goto*2 and (**if** $a \neq$ 0 **then** k_1 **else** I)(inspos 1) = goto (card $I + 1$).

(72) Let s be a state of SCMPDS, I be a shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) \neq 0$ and I is closed on s and halting on s. Then **if** $a \neq 0$ **then** k_1 **else** I is closed on s and **if** $a \neq 0$ **then** k_1 **else** I is halting on s.

(73) Let s be a state of SCMPDS, I be a Program-block, a be a Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) = 0$, then **if** $a \neq$ 0 **then** k_1 **else** I is closed on s and **if** $a \neq 0$ **then** k_1 **else** I is halting on s.

(74) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) \neq 0$ and I is closed on s and halting on s. Then IExec(**if** $a \neq 0$ **then** k_1 **else** $I, s) = \mathrm{IExec}(I, s) +\cdot$ Start-At(inspos card $I +$ 2).

(75) Let s be a state of SCMPDS, I be a Program-block, a be a Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) = 0$, then IExec(**if** $a \neq 0$ **then** k_1 **else** $I, s) = s +\cdot$ Start-At(inspos card $I + 2$).

Let I be a shiftable parahalting Program-block, let a be a Int position, and let k_1 be an integer. Observe that **if** $a \neq 0$ **then** k_1 **else** I is shiftable and parahalting.

Let I be a No-StopCode Program-block, let a be a Int position, and let k_1 be an integer. One can verify that **if** $a \neq 0$ **then** k_1 **else** I is No-StopCode.

One can prove the following three propositions:

(76) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a be a Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a \neq 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I, s)} = \mathrm{inspos}\,\mathrm{card}\, I + 2$.

(77) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \neq 0$, then (IExec(**if** $a \neq 0$ **then** k_1 **else** $I, s))(b) = (\mathrm{IExec}(I, s))(b)$.

(78) Let s be a state of SCMPDS, I be a Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) = 0$, then (IExec(**if** $a \neq 0$ **then** k_1 **else** $I, s))(b) = s(b)$.

7. The Computation of "if var>0 then block1 else block2"

We now state several propositions:

(79) card(**if** $a > k_1$ **then** I **else** J) = card I + card $J + 2$.

(80) inspos $0 \in$ dom(**if** $a > k_1$ **then** I **else** J) and inspos $1 \in$ dom(**if** $a > k_1$ **then** I **else** J).

(81) (**if** $a > k_1$ **then** I **else** J)(inspos 0) = (a, k_1) <= 0_*goto* card $I + 2$.

(82) Let s be a state of SCMPDS, I, J be shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) > 0$ and I is closed on s and halting on s. Then **if** $a > k_1$ **then** I **else** J is closed on s and **if** $a > k_1$ **then** I **else** J is halting on s.

(83) Let s be a state of SCMPDS, I be a Program-block, J be a shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) \leqslant 0$ and J is closed on s and halting on s. Then **if** $a > k_1$ **then** I **else** J is closed on s and **if** $a > k_1$ **then** I **else** J is halting on s.

(84) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, J be a shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) > 0$ and I is closed

on s and halting on s. Then IExec(**if** $a > k_1$ **then** I **else** J, s) $=$ IExec(I, s)+· Start-At(inspos card I + card $J + 2$).

(85) Let s be a state of SCMPDS, I be a Program-block, J be a No-StopCode shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) \leqslant 0$ and J is closed on s and halting on s. Then IExec(**if** $a > k_1$ **then** I **else** J, s) $=$ IExec(J, s)+· Start-At(inspos card I + card $J + 2$).

Let I, J be shiftable parahalting Program-block, let a be a Int position, and let k_1 be an integer. Note that **if** $a > k_1$ **then** I **else** J is shiftable and parahalting.

Let I, J be No-StopCode Program-block, let a be a Int position, and let k_1 be an integer. Note that **if** $a > k_1$ **then** I **else** J is No-StopCode.

Next we state three propositions:

(86) Let s be a state of SCMPDS, I, J be No-StopCode shiftable parahalting Program-block, a be a Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a>k_1\ \mathbf{then}\ I\ \mathbf{else}\ J,s)} = \mathrm{inspos}\,\mathrm{card}\,I + \mathrm{card}\,J + 2$.

(87) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, J be a shiftable Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) > 0$, then (IExec(**if** $a > k_1$ **then** I **else** J, s))(b) = (IExec(I, s))(b).

(88) Let s be a state of SCMPDS, I be a Program-block, J be a No-StopCode parahalting shiftable Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \leqslant 0$, then (IExec(**if** $a > k_1$ **then** I **else** J, s))(b) = (IExec(J, s))(b).

8. The Computation of "if var>0 then block"

The following propositions are true:

(89) card(**if** $a > 0$ **then** k_1 **else** I) = card $I + 1$.

(90) inspos $0 \in$ dom(**if** $a > 0$ **then** k_1 **else** I).

(91) (**if** $a > 0$ **then** k_1 **else** I)(inspos 0) = (a, k_1) <= 0_*goto* card $I + 1$.

(92) Let s be a state of SCMPDS, I be a shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) > 0$ and I is closed on s and halting on s. Then **if** $a > 0$ **then** k_1 **else** I is closed on s and **if** $a > 0$ **then** k_1 **else** I is halting on s.

(93) Let s be a state of SCMPDS, I be a Program-block, a be a Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \leqslant 0$, then **if** $a > 0$ **then** k_1 **else** I is closed on s and **if** $a > 0$ **then** k_1 **else** I is halting on s.

(94) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) > 0$ and I is closed on s and halting on s. Then $\mathrm{IExec}(\textbf{if } a > 0 \textbf{ then } k_1 \textbf{ else } I, s) = \mathrm{IExec}(I, s) +\cdot \text{Start-At}(\text{inspos card } I + 1)$.

(95) Let s be a state of SCMPDS, I be a Program-block, a be a Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \leqslant 0$, then $\mathrm{IExec}(\textbf{if } a > 0 \textbf{ then } k_1 \textbf{ else } I, s) = s +\cdot \text{Start-At}(\text{inspos card } I + 1)$.

Let I be a shiftable parahalting Program-block, let a be a Int position, and let k_1 be an integer. Observe that **if** $a > 0$ **then** k_1 **else** I is shiftable and parahalting.

Let I be a No-StopCode Program-block, let a be a Int position, and let k_1 be an integer. Observe that **if** $a > 0$ **then** k_1 **else** I is No-StopCode.

The following propositions are true:

(96) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a be a Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\textbf{if } a>0 \textbf{ then } k_1 \textbf{ else } I, s)} = \text{inspos card } I + 1$.

(97) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) > 0$, then $(\mathrm{IExec}(\textbf{if } a > 0 \textbf{ then } k_1 \textbf{ else } I, s))(b) = (\mathrm{IExec}(I, s))(b)$.

(98) Let s be a state of SCMPDS, I be a Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \leqslant 0$, then $(\mathrm{IExec}(\textbf{if } a > 0 \textbf{ then } k_1 \textbf{ else } I, s))(b) = s(b)$.

9. The Computation of "if var<=0 then block"

We now state several propositions:

(99) $\mathrm{card}(\textbf{if } a \leqslant 0 \textbf{ then } k_1 \textbf{ else } I) = \mathrm{card}\, I + 2$.

(100) $\text{inspos } 0 \in \mathrm{dom}(\textbf{if } a \leqslant 0 \textbf{ then } k_1 \textbf{ else } I)$ and $\text{inspos } 1 \in \mathrm{dom}(\textbf{if } a \leqslant 0 \textbf{ then } k_1 \textbf{ else } I)$.

(101) $(\textbf{if } a \leqslant 0 \textbf{ then } k_1 \textbf{ else } I)(\text{inspos } 0) = (a, k_1) <= 0\_goto 2$ and $(\textbf{if } a \leqslant 0 \textbf{ then } k_1 \textbf{ else } I)(\text{inspos } 1) = \text{goto } (\mathrm{card}\, I + 1)$.

(102) Let s be a state of SCMPDS, I be a shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) \leqslant 0$ and I is closed on s and halting on s. Then **if** $a \leqslant 0$ **then** k_1 **else** I is closed on s and **if** $a \leqslant 0$ **then** k_1 **else** I is halting on s.

(103) Let s be a state of SCMPDS, I be a Program-block, a be a Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) > 0$, then **if** $a \leqslant$

0 **then** k_1 **else** I is closed on s and **if** $a \leqslant 0$ **then** k_1 **else** I is halting on s.

(104) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) \leqslant 0$ and I is closed on s and halting on s. Then IExec(**if** $a \leqslant 0$ **then** k_1 **else** $I, s) = \mathrm{IExec}(I, s) +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\,\mathrm{card}\, I + 2)$.

(105) Let s be a state of SCMPDS, I be a Program-block, a be a Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) > 0$, then IExec(**if** $a \leqslant 0$ **then** k_1 **else** $I, s) = s +\cdot \mathrm{Start\text{-}At}(\mathrm{inspos}\,\mathrm{card}\, I + 2)$.

Let I be a shiftable parahalting Program-block, let a be a Int position, and let k_1 be an integer. Observe that **if** $a \leqslant 0$ **then** k_1 **else** I is shiftable and parahalting.

Let I be a No-StopCode Program-block, let a be a Int position, and let k_1 be an integer. Note that **if** $a \leqslant 0$ **then** k_1 **else** I is No-StopCode.

We now state three propositions:

(106) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a be a Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a \leqslant 0\ \mathbf{then}\ k_1\ \mathbf{else}\ I, s)} = \mathrm{inspos}\,\mathrm{card}\, I + 2$.

(107) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \leqslant 0$, then (IExec(**if** $a \leqslant 0$ **then** k_1 **else** $I, s))(b) = (\mathrm{IExec}(I, s))(b)$.

(108) Let s be a state of SCMPDS, I be a Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) > 0$, then (IExec(**if** $a \leqslant 0$ **then** k_1 **else** $I, s))(b) = s(b)$.

10. The Computation of "if var<0 then block1 else block2"

One can prove the following propositions:

(109) card(**if** $a < k_1$ **then** I **else** $J) = \mathrm{card}\, I + \mathrm{card}\, J + 2$.

(110) $\mathrm{inspos}\, 0 \in$ dom(**if** $a < k_1$ **then** I **else** J) and $\mathrm{inspos}\, 1 \in$ dom(**if** $a < k_1$ **then** I **else** J).

(111) (**if** $a < k_1$ **then** I **else** $J)(\mathrm{inspos}\, 0) = (a, k_1) >= 0\_goto\ \mathrm{card}\, I + 2$.

(112) Let s be a state of SCMPDS, I, J be shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) < 0$ and I is closed on s and halting on s. Then **if** $a < k_1$ **then** I **else** J is closed on s and **if** $a < k_1$ **then** I **else** J is halting on s.

(113) Let s be a state of SCMPDS, I be a Program-block, J be a shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) \geqslant 0$ and J is closed on s and halting on s. Then **if** $a < k_1$ **then** I **else** J is closed on s and **if** $a < k_1$ **then** I **else** J is halting on s.

(114) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, J be a shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) < 0$ and I is closed on s and halting on s. Then IExec(**if** $a < k_1$ **then** I **else** J, s) $=$ $\mathrm{IExec}(I, s) +\cdot\, \text{Start-At}(\mathrm{inspos}\,\mathrm{card}\,I + \mathrm{card}\,J + 2)$.

(115) Let s be a state of SCMPDS, I be a Program-block, J be a No-StopCode shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) \geqslant 0$ and J is closed on s and halting on s. Then IExec(**if** $a < k_1$ **then** I **else** J, s) $=$ $\mathrm{IExec}(J, s) +\cdot\, \text{Start-At}(\mathrm{inspos}\,\mathrm{card}\,I + \mathrm{card}\,J + 2)$.

Let I, J be shiftable parahalting Program-block, let a be a Int position, and let k_1 be an integer. Observe that **if** $a < k_1$ **then** I **else** J is shiftable and parahalting.

Let I, J be No-StopCode Program-block, let a be a Int position, and let k_1 be an integer. Note that **if** $a < k_1$ **then** I **else** J is No-StopCode.

Next we state three propositions:

(116) Let s be a state of SCMPDS, I, J be No-StopCode shiftable parahalting Program-block, a be a Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a<k_1\ \mathbf{then}\ I\ \mathbf{else}\ J, s)} = \mathrm{inspos}\,\mathrm{card}\,I + \mathrm{card}\,J + 2$.

(117) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, J be a shiftable Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) < 0$, then (IExec(**if** $a < k_1$ **then** I **else** J, s))$(b) = (\mathrm{IExec}(I, s))(b)$.

(118) Let s be a state of SCMPDS, I be a Program-block, J be a No-StopCode parahalting shiftable Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \geqslant 0$, then (IExec(**if** $a < k_1$ **then** I **else** J, s))$(b) = (\mathrm{IExec}(J, s))(b)$.

11. The Computation of "if var<0 then block"

One can prove the following propositions:

(119) card(**if** $a < 0$ **then** k_1 **else** I) $= \mathrm{card}\,I + 1$.

(120) $\mathrm{inspos}\,0 \in$ dom(**if** $a < 0$ **then** k_1 **else** I).

(121) (**if** $a < 0$ **then** k_1 **else** I)$(\mathrm{inspos}\,0) = (a, k_1) >= 0\_goto\,\mathrm{card}\,I + 1$.

(122) Let s be a state of SCMPDS, I be a shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) < 0$ and I is closed on s and halting on s. Then **if** $a < 0$ **then** k_1 **else** I is closed on s and **if** $a < 0$ **then** k_1 **else** I is halting on s.

(123) Let s be a state of SCMPDS, I be a Program-block, a be a Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \geqslant 0$, then **if** $a < 0$ **then** k_1 **else** I is closed on s and **if** $a < 0$ **then** k_1 **else** I is halting on s.

(124) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\mathrm{DataLoc}(s(a), k_1)) < 0$ and I is closed on s and halting on s. Then IExec(**if** $a < 0$ **then** k_1 **else** I, s) = IExec(I, s)+· Start-At(inspos card I + 1).

(125) Let s be a state of SCMPDS, I be a Program-block, a be a Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \geqslant 0$, then IExec(**if** $a < 0$ **then** k_1 **else** I, s) = s+· Start-At(inspos card $I + 1$).

Let I be a shiftable parahalting Program-block, let a be a Int position, and let k_1 be an integer. Note that **if** $a < 0$ **then** k_1 **else** I is shiftable and parahalting.

Let I be a No-StopCode Program-block, let a be a Int position, and let k_1 be an integer. One can check that **if** $a < 0$ **then** k_1 **else** I is No-StopCode.

Next we state three propositions:

(126) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a be a Int position, and k_1 be an integer. Then $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a<0\ \mathbf{then}\ k_1\ \mathbf{else}\ I,s)}$ = inspos card $I + 1$.

(127) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) < 0$, then (IExec(**if** $a < 0$ **then** k_1 **else** I, s))(b) = (IExec(I, s))(b).

(128) Let s be a state of SCMPDS, I be a Program-block, a, b be Int position, and k_1 be an integer. If $s(\mathrm{DataLoc}(s(a), k_1)) \geqslant 0$, then (IExec(**if** $a < 0$ **then** k_1 **else** I, s))$(b) = s(b)$.

12. The Computation of "if var>=0 then block"

The following propositions are true:

(129) card(**if** $a \geqslant 0$ **then** k_1 **else** I) = card $I + 2$.

(130) inspos 0 $\in$ dom(**if** $a \geqslant 0$ **then** k_1 **else** I) and inspos 1 $\in$ dom(**if** $a \geqslant 0$ **then** k_1 **else** I).

(131) (**if** $a \geqslant 0$ **then** k_1 **else** I)(inspos 0) = (a, k_1) >= 0_*goto*2 and (**if** $a \geqslant 0$ **then** k_1 **else** I)(inspos 1) = goto (card $I + 1$).

(132) Let s be a state of SCMPDS, I be a shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\text{DataLoc}(s(a), k_1)) \geqslant 0$ and I is closed on s and halting on s. Then **if** $a \geqslant 0$ **then** k_1 **else** I is closed on s and **if** $a \geqslant 0$ **then** k_1 **else** I is halting on s.

(133) Let s be a state of SCMPDS, I be a Program-block, a be a Int position, and k_1 be an integer. If $s(\text{DataLoc}(s(a), k_1)) < 0$, then **if** $a \geqslant 0$ **then** k_1 **else** I is closed on s and **if** $a \geqslant 0$ **then** k_1 **else** I is halting on s.

(134) Let s be a state of SCMPDS, I be a No-StopCode shiftable Program-block, a be a Int position, and k_1 be an integer. Suppose $s(\text{DataLoc}(s(a), k_1)) \geqslant 0$ and I is closed on s and halting on s. Then IExec(**if** $a \geqslant 0$ **then** k_1 **else** I, s) = IExec(I, s)+· Start-At(inspos card $I+2$).

(135) Let s be a state of SCMPDS, I be a Program-block, a be a Int position, and k_1 be an integer. If $s(\text{DataLoc}(s(a), k_1)) < 0$, then IExec(**if** $a \geqslant 0$ **then** k_1 **else** I, s) = s+· Start-At(inspos card $I + 2$).

Let I be a shiftable parahalting Program-block, let a be a Int position, and let k_1 be an integer. Note that **if** $a \geqslant 0$ **then** k_1 **else** I is shiftable and parahalting.

Let I be a No-StopCode Program-block, let a be a Int position, and let k_1 be an integer. Observe that **if** $a \geqslant 0$ **then** k_1 **else** I is No-StopCode.

We now state three propositions:

(136) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a be a Int position, and k_1 be an integer. Then $\mathbf{IC}_{\text{IExec}(\textbf{if } a \geqslant 0 \textbf{ then } k_1 \textbf{ else } I, s)} = \text{inspos card } I + 2$.

(137) Let s be a state of SCMPDS, I be a No-StopCode shiftable parahalting Program-block, a, b be Int position, and k_1 be an integer. If $s(\text{DataLoc}(s(a), k_1)) \geqslant 0$, then (IExec(**if** $a \geqslant 0$ **then** k_1 **else** I, s))(b) = (IExec(I, s))(b).

(138) Let s be a state of SCMPDS, I be a Program-block, a, b be Int position, and k_1 be an integer. If $s(\text{DataLoc}(s(a), k_1)) < 0$, then (IExec(**if** $a \geqslant 0$ **then** k_1 **else** I, s))(b) = $s(b)$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.
[2] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[3] Grzegorz Bancerek and Piotr Rudnicki. Development of terminology for **scm**. *Formalized Mathematics*, 4(**1**):61–67, 1993.

[4] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.
[5] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[6] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[7] Jing-Chao Chen. Computation and program shift in the SCMPDS computer. *Formalized Mathematics*, 8(**1**):193–199, 1999.
[8] Jing-Chao Chen. Computation of two consecutive program blocks for SCMPDS. *Formalized Mathematics*, 8(**1**):211–217, 1999.
[9] Jing-Chao Chen. The construction and shiftability of program blocks for SCMPDS. *Formalized Mathematics*, 8(**1**):201–210, 1999.
[10] Jing-Chao Chen. The SCMPDS computer and the basic semantics of its instructions. *Formalized Mathematics*, 8(**1**):183–191, 1999.
[11] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[12] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[13] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.
[14] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.
[15] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[16] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[17] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[18] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[19] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[20] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received June 15, 1999