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

The Composition of Functors and Transformations in Alternative Categories

Artur Korniłowicz
University of Białystok

MML Identifier: `FUNCTOR3`.

The articles [5], [6], [2], [8], [7], [3], [1], [4], and [9] provide the notation and terminology for this paper.

1. Preliminaries

One can verify that there exists a non empty category structure which is transitive, associative, and strict and has units.

Let A be a non empty transitive category structure and let B be a non empty category structure with units. One can verify that there exists a functor structure from A to B which is strict, comp-preserving, comp-reversing, precovariant, precontravariant, and feasible.

Let A be a transitive non empty category structure with units and let B be a non empty category structure with units. Observe that there exists a functor structure from A to B which is strict, comp-preserving, comp-reversing, precovariant, precontravariant, feasible, and id-preserving.

Let A be a transitive non empty category structure with units and let B be a non empty category structure with units. Observe that there exists a functor from A to B which is strict, feasible, covariant, and contravariant.

Next we state several propositions:

(1) Let C be a category, o_1, o_2, o_3, o_4 be objects of C, a be a morphism from o_1 to o_2, b be a morphism from o_2 to o_3, c be a morphism from o_1 to o_4, and d be a morphism from o_4 to o_3. Suppose $b \cdot a = d \cdot c$ and $a \cdot a^{-1} = \mathrm{id}_{(o_2)}$

ISSN 1426–2630

and $d^{-1} \cdot d = \mathrm{id}_{(o_4)}$ and $\langle o_1, o_2 \rangle \neq \emptyset$ and $\langle o_2, o_1 \rangle \neq \emptyset$ and $\langle o_2, o_3 \rangle \neq \emptyset$ and $\langle o_3, o_4 \rangle \neq \emptyset$ and $\langle o_4, o_3 \rangle \neq \emptyset$. Then $c \cdot a^{-1} = d^{-1} \cdot b$.

(2) Let A be a non empty transitive category structure, B, C be non empty category structures with units, F be a feasible precovariant functor structure from A to B, G be a functor structure from B to C, and o, o_1 be objects of A. Then $\text{Morph-Map}_{G \cdot F}(o, o_1) = \text{Morph-Map}_G(F(o), F(o_1)) \cdot \text{Morph-Map}_F(o, o_1)$.

(3) Let A be a non empty transitive category structure, B, C be non empty category structures with units, F be a feasible precontravariant functor structure from A to B, G be a functor structure from B to C, and o, o_1 be objects of A. Then $\text{Morph-Map}_{G \cdot F}(o, o_1) = \text{Morph-Map}_G(F(o_1), F(o)) \cdot \text{Morph-Map}_F(o, o_1)$.

(4) Let A be a non empty transitive category structure, B be a non empty category structure with units, and F be a feasible precovariant functor structure from A to B. Then $\mathrm{id}_B \cdot F =$ the functor structure of F.

(5) Let A be a transitive non empty category structure with units, B be a non empty category structure with units, and F be a feasible precovariant functor structure from A to B. Then $F \cdot \mathrm{id}_A =$ the functor structure of F.

For simplicity, we use the following convention: A denotes a non empty category structure, B, C denote non empty reflexive category structures, F denotes a feasible precovariant functor structure from A to B, G denotes a feasible precovariant functor structure from B to C, M denotes a feasible precontravariant functor structure from A to B, N denotes a feasible precontravariant functor structure from B to C, o_1, o_2 denote objects of A, and m denotes a morphism from o_1 to o_2.

The following four propositions are true:

(6) If $\langle o_1, o_2 \rangle \neq \emptyset$, then $(G \cdot F)(m) = G(F(m))$.

(7) If $\langle o_1, o_2 \rangle \neq \emptyset$, then $(N \cdot M)(m) = N(M(m))$.

(8) If $\langle o_1, o_2 \rangle \neq \emptyset$, then $(N \cdot F)(m) = N(F(m))$.

(9) If $\langle o_1, o_2 \rangle \neq \emptyset$, then $(G \cdot M)(m) = G(M(m))$.

Let A be a non empty transitive category structure, let B be a transitive non empty category structure with units, let C be a non empty category structure with units, let F be a feasible precovariant comp-preserving functor structure from A to B, and let G be a feasible precovariant comp-preserving functor structure from B to C. One can check that $G \cdot F$ is comp-preserving.

Let A be a non empty transitive category structure, let B be a transitive non empty category structure with units, let C be a non empty category structure with units, let F be a feasible precontravariant comp-reversing functor structure from A to B, and let G be a feasible precontravariant comp-reversing functor structure from B to C. One can check that $G \cdot F$ is comp-preserving.

Let A be a non empty transitive category structure, let B be a transitive non empty category structure with units, let C be a non empty category structure with units, let F be a feasible precovariant comp-preserving functor structure from A to B, and let G be a feasible precontravariant comp-reversing functor structure from B to C. One can verify that $G \cdot F$ is comp-reversing.

Let A be a non empty transitive category structure, let B be a transitive non empty category structure with units, let C be a non empty category structure with units, let F be a feasible precontravariant comp-reversing functor structure from A to B, and let G be a feasible precovariant comp-preserving functor structure from B to C. One can verify that $G \cdot F$ is comp-reversing.

Let A, B be transitive non empty category structures with units, let C be a non empty category structure with units, let F be a covariant functor from A to B, and let G be a covariant functor from B to C. Then $G \cdot F$ is a strict covariant functor from A to C.

Let A, B be transitive non empty category structures with units, let C be a non empty category structure with units, let F be a contravariant functor from A to B, and let G be a contravariant functor from B to C. Then $G \cdot F$ is a strict covariant functor from A to C.

Let A, B be transitive non empty category structures with units, let C be a non empty category structure with units, let F be a covariant functor from A to B, and let G be a contravariant functor from B to C. Then $G \cdot F$ is a strict contravariant functor from A to C.

Let A, B be transitive non empty category structures with units, let C be a non empty category structure with units, let F be a contravariant functor from A to B, and let G be a covariant functor from B to C. Then $G \cdot F$ is a strict contravariant functor from A to C.

For simplicity, we adopt the following convention: A, B, C, D are transitive non empty category structures with units, F_1, F_2, F_3 are covariant functors from A to B, G_1, G_2, G_3 are covariant functors from B to C, H_1, H_2 are covariant functors from C to D, p is a transformation from F_1 to F_2, p_1 is a transformation from F_2 to F_3, q is a transformation from G_1 to G_2, q_1 is a transformation from G_2 to G_3, and r is a transformation from H_1 to H_2.

The following proposition is true

(10) If F_1 is transformable to F_2 and G_1 is transformable to G_2, then $G_1 \cdot F_1$ is transformable to $G_2 \cdot F_2$.

2. The Composition of Functors with Transformations

Let A, B, C be transitive non empty category structures with units, let F_1, F_2 be covariant functors from A to B, let t be a transformation from F_1 to

F_2, and let G be a covariant functor from B to C. Let us assume that F_1 is transformable to F_2. The functor $G \cdot t$ yields a transformation from $G \cdot F_1$ to $G \cdot F_2$ and is defined as follows:

(Def. 1) For every object o of A holds $(G \cdot t)(o) = G(t[o])$.

Next we state the proposition

(11) For every object o of A such that F_1 is transformable to F_2 holds $(G_1 \cdot p)[o] = G_1(p[o])$.

Let A, B, C be transitive non empty category structures with units, let G_1, G_2 be covariant functors from B to C, let F be a covariant functor from A to B, and let s be a transformation from G_1 to G_2. Let us assume that G_1 is transformable to G_2. The functor $s \cdot F$ yielding a transformation from $G_1 \cdot F$ to $G_2 \cdot F$ is defined by:

(Def. 2) For every object o of A holds $(s \cdot F)(o) = s[F(o)]$.

Next we state a number of propositions:

(12) For every object o of A such that G_1 is transformable to G_2 holds $(q \cdot F_1)[o] = q[F_1(o)]$.

(13) If F_1 is transformable to F_2 and F_2 is transformable to F_3, then $G_1 \cdot (p_1 \circ p) = G_1 \cdot p_1 \circ G_1 \cdot p$.

(14) If G_1 is transformable to G_2 and G_2 is transformable to G_3, then $(q_1 \circ q) \cdot F_1 = q_1 \cdot F_1 \circ q \cdot F_1$.

(15) If H_1 is transformable to H_2, then $(r \cdot G_1) \cdot F_1 = r \cdot (G_1 \cdot F_1)$.

(16) If G_1 is transformable to G_2, then $(H_1 \cdot q) \cdot F_1 = H_1 \cdot (q \cdot F_1)$.

(17) If F_1 is transformable to F_2, then $(H_1 \cdot G_1) \cdot p = H_1 \cdot (G_1 \cdot p)$.

(18) $\mathrm{id}_{(G_1)} \cdot F_1 = \mathrm{id}_{G_1 \cdot F_1}$.

(19) $G_1 \cdot \mathrm{id}_{(F_1)} = \mathrm{id}_{G_1 \cdot F_1}$.

(20) If F_1 is transformable to F_2, then $\mathrm{id}_B \cdot p = p$.

(21) If G_1 is transformable to G_2, then $q \cdot \mathrm{id}_B = q$.

3. The Composition of Transformations

Let A, B, C be transitive non empty category structures with units, let F_1, F_2 be covariant functors from A to B, let G_1, G_2 be covariant functors from B to C, let t be a transformation from F_1 to F_2, and let s be a transformation from G_1 to G_2. The functor $s\,t$ yielding a transformation from $G_1 \cdot F_1$ to $G_2 \cdot F_2$ is defined as follows:

(Def. 3) $s\,t = s \cdot F_2 \circ G_1 \cdot t$.

The following propositions are true:

(22) Let q be a natural transformation from G_1 to G_2. Suppose F_1 is transformable to F_2 and G_1 is naturally transformable to G_2. Then $q\,p = G_2 \cdot p \circ q \cdot F_1$.

(23) If F_1 is transformable to F_2, then $\mathrm{id}_{\mathrm{id}_B}\ p = p$.

(24) If G_1 is transformable to G_2, then $q\ \mathrm{id}_{\mathrm{id}_B} = q$.

(25) If F_1 is transformable to F_2, then $G_1 \cdot p = \mathrm{id}_{(G_1)}\ p$.

(26) If G_1 is transformable to G_2, then $q \cdot F_1 = q\ \mathrm{id}_{(F_1)}$.

We use the following convention: A, B, C, D are categories, F_1, F_2, F_3 are covariant functors from A to B, and G_1, G_2, G_3 are covariant functors from B to C.

One can prove the following proposition

(27) Let H_1, H_2 be covariant functors from C to D, t be a transformation from F_1 to F_2, s be a transformation from G_1 to G_2, and u be a transformation from H_1 to H_2. Suppose F_1 is transformable to F_2 and G_1 is transformable to G_2 and H_1 is transformable to H_2. Then $(u\,s)\,t = u\,(s\,t)$.

In the sequel t denotes a natural transformation from F_1 to F_2, s denotes a natural transformation from G_1 to G_2, and s_1 denotes a natural transformation from G_2 to G_3.

One can prove the following propositions:

(28) If F_1 is naturally transformable to F_2, then $G_1 \cdot t$ is a natural transformation from $G_1 \cdot F_1$ to $G_1 \cdot F_2$.

(29) If G_1 is naturally transformable to G_2, then $s \cdot F_1$ is a natural transformation from $G_1 \cdot F_1$ to $G_2 \cdot F_1$.

(30) Suppose F_1 is naturally transformable to F_2 and G_1 is naturally transformable to G_2. Then $G_1 \cdot F_1$ is naturally transformable to $G_2 \cdot F_2$ and $s\,t$ is a natural transformation from $G_1 \cdot F_1$ to $G_2 \cdot F_2$.

(31) Let t be a transformation from F_1 to F_2 and t_1 be a transformation from F_2 to F_3. Suppose that

(i) F_1 is naturally transformable to F_2,

(ii) F_2 is naturally transformable to F_3,

(iii) G_1 is naturally transformable to G_2, and

(iv) G_2 is naturally transformable to G_3.

Then $(s_1 \circ s)\,(t_1 \circ t) = s_1\,t_1 \circ s\,t$.

4. Natural Equivalences

One can prove the following proposition

(32) Suppose F_1 is naturally transformable to F_2 and F_2 is transformable to F_1 and for every object a of A holds $t[a]$ is iso. Then

(i) F_2 is naturally transformable to F_1, and

(ii) there exists a natural transformation f from F_2 to F_1 such that for every object a of A holds $f(a) = t[a]^{-1}$ and $f[a]$ is iso.

Let A, B be categories and let F_1, F_2 be covariant functors from A to B. We say that F_1, F_2 are naturally equivalent if and only if the conditions (Def. 4) are satisfied.

(Def. 4)(i) F_1 is naturally transformable to F_2,

(ii) F_2 is transformable to F_1, and

(iii) there exists a natural transformation t from F_1 to F_2 such that for every object a of A holds $t[a]$ is iso.

Let us notice that the predicate F_1, F_2 are naturally equivalent is reflexive and symmetric.

Let A, B be categories and let F_1, F_2 be covariant functors from A to B. Let us assume that F_1, F_2 are naturally equivalent. A natural transformation from F_1 to F_2 is said to be a natural equivalence of F_1 and F_2 if:

(Def. 5) For every object a of A holds $\mathrm{it}[a]$ is iso.

In the sequel e is a natural equivalence of F_1 and F_2, e_1 is a natural equivalence of F_2 and F_3, and f is a natural equivalence of G_1 and G_2.

One can prove the following propositions:

(33) Suppose F_1, F_2 are naturally equivalent and F_2, F_3 are naturally equivalent. Then F_1, F_3 are naturally equivalent.

(34) Suppose F_1, F_2 are naturally equivalent and F_2, F_3 are naturally equivalent. Then $e_1 \circ e$ is a natural equivalence of F_1 and F_3.

(35) Suppose F_1, F_2 are naturally equivalent. Then $G_1 \cdot F_1$, $G_1 \cdot F_2$ are naturally equivalent and $G_1 \cdot e$ is a natural equivalence of $G_1 \cdot F_1$ and $G_1 \cdot F_2$.

(36) Suppose G_1, G_2 are naturally equivalent. Then $G_1 \cdot F_1$, $G_2 \cdot F_1$ are naturally equivalent and $f \cdot F_1$ is a natural equivalence of $G_1 \cdot F_1$ and $G_2 \cdot F_1$.

(37) Suppose F_1, F_2 are naturally equivalent and G_1, G_2 are naturally equivalent. Then $G_1 \cdot F_1$, $G_2 \cdot F_2$ are naturally equivalent and $f\,e$ is a natural equivalence of $G_1 \cdot F_1$ and $G_2 \cdot F_2$.

Let A, B be categories, let F_1, F_2 be covariant functors from A to B, and let e be a natural equivalence of F_1 and F_2. Let us assume that F_1, F_2 are naturally equivalent. The functor e^{-1} yielding a natural equivalence of F_2 and F_1 is defined as follows:

(Def. 6) For every object a of A holds $e^{-1}(a) = e[a]^{-1}$.

The following propositions are true:

(38) For every object o of A such that F_1, F_2 are naturally equivalent holds $e^{-1}[o] = e[o]^{-1}$.

(39) If F_1, F_2 are naturally equivalent, then $e \circ e^{-1} = \mathrm{id}_{(F_2)}$.

(40) If F_1, F_2 are naturally equivalent, then $e^{-1} \circ e = \mathrm{id}_{(F_1)}$.

Let A, B be categories and let F be a covariant functor from A to B. Then id_F is a natural equivalence of F and F.

The following three propositions are true:

(41) If F_1, F_2 are naturally equivalent, then $(e^{-1})^{-1} = e$.

(42) Let k be a natural equivalence of F_1 and F_3. Suppose $k = e_1 \circ e$ and F_1, F_2 are naturally equivalent and F_2, F_3 are naturally equivalent. Then $k^{-1} = e^{-1} \circ e_1{}^{-1}$.

(43) $(\mathrm{id}_{(F_1)})^{-1} = \mathrm{id}_{(F_1)}$.

References

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

[2] Beata Madras. Basic properties of objects and morphisms. *Formalized Mathematics*, 6(**3**):329–334, 1997.

[3] Robert Nieszczerzewski. Category of functors between alternative categories. *Formalized Mathematics*, 6(**3**):371–375, 1997.

[4] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.

[5] Andrzej Trybulec. Categories without uniqueness of **cod** and **dom**. *Formalized Mathematics*, 5(**2**):259–267, 1996.

[6] Andrzej Trybulec. Examples of category structures. *Formalized Mathematics*, 5(**4**):493–500, 1996.

[7] Andrzej Trybulec. Functors for alternative categories. *Formalized Mathematics*, 5(**4**):595–608, 1996.

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

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

Received January 21, 1998

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

Completely-Irreducible Elements[1]

Robert Milewski
University of Białystok

Summary. The article is a translation of [5, 92–93].

MML Identifier: `WAYBEL16`.

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

1. Preliminaries

The following propositions are true:

(1) For every sup-semilattice L and for all elements x, y of L holds $\bigsqcap_L(\uparrow x \cap \uparrow y) = x \sqcup y$.

(2) For every semilattice L and for all elements x, y of L holds $\bigsqcup_L(\downarrow x \cap \downarrow y) = x \sqcap y$.

(3) Let L be a non empty relational structure and x, y be elements of L. If x is maximal in (the carrier of L) $\setminus \uparrow y$, then $\uparrow x \setminus \{x\} = \uparrow x \cap \uparrow y$.

(4) Let L be a non empty relational structure and x, y be elements of L. If x is minimal in (the carrier of L) $\setminus \downarrow y$, then $\downarrow x \setminus \{x\} = \downarrow x \cap \downarrow y$.

(5) Let L be a poset with l.u.b.'s, X, Y be subsets of L, and X', Y' be subsets of $L^{\rm op}$. If $X = X'$ and $Y = Y'$, then $X \sqcup Y = X' \sqcap Y'$.

(6) Let L be a poset with g.l.b.'s, X, Y be subsets of L, and X', Y' be subsets of $L^{\rm op}$. If $X = X'$ and $Y = Y'$, then $X \sqcap Y = X' \sqcup Y'$.

(7) For every non empty reflexive transitive relational structure L holds $\mathrm{Filt}(L) = \mathrm{Ids}(L^{\rm op})$.

[1]This work has been supported by KBN Grant 8 T11C 018 12.

ISSN 1426–2630

(8) For every non empty reflexive transitive relational structure L holds $\mathrm{Ids}(L) = \mathrm{Filt}(L^{\mathrm{op}})$.

2. Free Generation Set

Let S, T be complete non empty posets. A map from S into T is said to be a CLHomomorphism of S, T if:

(Def. 1) It is directed-sups-preserving and infs-preserving.

Let S be a continuous complete non empty poset and let A be a subset of S. We say that A is a free generator set if and only if the condition (Def. 2) is satisfied.

(Def. 2) Let T be a continuous complete non empty poset and f be a function from A into the carrier of T. Then there exists a CLHomomorphism h of S, T such that $h{\upharpoonright}A = f$ and for every CLHomomorphism h' of S, T such that $h'{\upharpoonright}A = f$ holds $h' = h$.

Let L be an upper-bounded non empty poset. One can check that $\mathrm{Filt}(L)$ is non empty.

The following propositions are true:

(9) For every set X and for every non empty subset Y of $\langle \mathrm{Filt}(2^X_\subseteq), \subseteq \rangle$ holds $\bigcap Y$ is a filter of $2^X_\subseteq$.

(10) For every set X and for every non empty subset Y of $\langle \mathrm{Filt}(2^X_\subseteq), \subseteq \rangle$ holds $\inf Y$ exists in $\langle \mathrm{Filt}(2^X_\subseteq), \subseteq \rangle$ and $\bigsqcap_{(\langle \mathrm{Filt}(2^X_\subseteq), \subseteq \rangle)} Y = \bigcap Y$.

(11) For every set X holds 2^X is a filter of $2^X_\subseteq$.

(12) For every set X holds $\{X\}$ is a filter of $2^X_\subseteq$.

(13) For every set X holds $\langle \mathrm{Filt}(2^X_\subseteq), \subseteq \rangle$ is upper-bounded.

(14) For every set X holds $\langle \mathrm{Filt}(2^X_\subseteq), \subseteq \rangle$ is lower-bounded.

(15) For every set X holds $\top_{\langle \mathrm{Filt}(2^X_\subseteq), \subseteq \rangle} = 2^X$.

(16) For every set X holds $\perp_{\langle \mathrm{Filt}(2^X_\subseteq), \subseteq \rangle} = \{X\}$.

(17) For every non empty set X and for every non empty subset Y of $\langle X, \subseteq \rangle$ such that $\sup Y$ exists in $\langle X, \subseteq \rangle$ holds $\bigcup Y \subseteq \sup Y$.

(18) For every upper-bounded semilattice L holds $\langle \mathrm{Filt}(L), \subseteq \rangle$ is complete.

Let L be an upper-bounded semilattice. Note that $\langle \mathrm{Filt}(L), \subseteq \rangle$ is complete.

3. Completely-Irreducible Elements

Let L be a non empty relational structure and let p be an element of L. We say that p is completely-irreducible if and only if:

(Def. 3) Min $\uparrow p \setminus \{p\}$ exists in L.

We now state the proposition

(19) Let L be a non empty relational structure and p be an element of L. If p is completely-irreducible, then $\bigsqcap_L(\uparrow p \setminus \{p\}) \neq p$.

Let L be a non empty relational structure. The functor $\operatorname{Irr} L$ yielding a subset of L is defined by:

(Def. 4) For every element x of L holds $x \in \operatorname{Irr} L$ iff x is completely-irreducible.

The following propositions are true:

(20) Let L be a non empty poset and p be an element of L. Then p is completely-irreducible if and only if there exists an element q of L such that $p < q$ and for every element s of L such that $p < s$ holds $q \leqslant s$ and $\uparrow p = \{p\} \cup \uparrow q$.

(21) For every upper-bounded non empty poset L holds $\top_L \notin \operatorname{Irr} L$.

(22) For every semilattice L holds $\operatorname{Irr} L \subseteq \operatorname{IRR}(L)$.

(23) For every semilattice L and for every element x of L such that x is completely-irreducible holds x is irreducible.

(24) Let L be a non empty poset and x be an element of L. Suppose x is completely-irreducible. Let X be a subset of L. If $\inf X$ exists in L and $x = \inf X$, then $x \in X$.

(25) For every non empty poset L and for every subset X of L such that X is order-generating holds $\operatorname{Irr} L \subseteq X$.

(26) Let L be a complete lattice and p be an element of L. Given an element k of L such that p is maximal in (the carrier of L) $\setminus \uparrow k$. Then p is completely-irreducible.

(27) Let L be a transitive antisymmetric relational structure with l.u.b.'s and p, q, u be elements of L. Suppose $p < q$ and for every element s of L such that $p < s$ holds $q \leqslant s$ and $u \nleqslant p$. Then $p \sqcup u = q \sqcup u$.

(28) Let L be a distributive lattice and p, q, u be elements of L. Suppose $p < q$ and for every element s of L such that $p < s$ holds $q \leqslant s$ and $u \nleqslant p$. Then $u \sqcap q \nleqslant p$.

(29) Let L be a distributive complete lattice. Suppose L^{op} is meet-continuous. Let p be an element of L. Suppose p is completely-irreducible. Then (the carrier of L) $\setminus \downarrow p$ is an open filter of L.

(30) Let L be a distributive complete lattice. Suppose L^{op} is meet-continuous. Let p be an element of L. Suppose p is completely-irreducible. Then there exists an element k of L such that $k \in$ the carrier of $\operatorname{CompactSublatt}(L)$ and p is maximal in (the carrier of L) $\setminus \uparrow k$.

(31) Let L be a lower-bounded algebraic lattice and x, y be elements of L. Suppose $y \nleqslant x$. Then there exists an element p of L such that p is

completely-irreducible and $x \leqslant p$ and $y \nleqslant p$.

(32) Let L be a lower-bounded algebraic lattice. Then $\mathrm{Irr}\, L$ is order-generating and for every subset X of L such that X is order-generating holds $\mathrm{Irr}\, L \subseteq X$.

(33) For every lower-bounded algebraic lattice L and for every element s of L holds $s = \bigsqcap_L(\uparrow s \cap \mathrm{Irr}\, L)$.

(34) Let L be a complete non empty poset, X be a subset of L, and p be an element of L. If p is completely-irreducible and $p = \inf X$, then $p \in X$.

(35) Let L be a complete algebraic lattice and p be an element of L. Suppose p is completely-irreducible. Then $p = \bigsqcap_L\{x; x$ ranges over elements of L: $x \in \uparrow p \ \wedge \ \bigvee_{k\,:\,\text{element of } L}$ ($k \in$ the carrier of CompactSublatt(L) $\wedge$ x is maximal in (the carrier of L) $\setminus \uparrow k$)$\}$.

(36) Let L be a complete algebraic lattice and p be an element of L. Then there exists an element k of L such that $k \in$ the carrier of CompactSublatt(L) and p is maximal in (the carrier of L) $\setminus \uparrow k$ if and only if p is completely-irreducible.

References

[1] Grzegorz Bancerek. Complete lattices. *Formalized Mathematics*, 2(**5**):719–725, 1991.

[2] Grzegorz Bancerek. Bounds in posets and relational substructures. *Formalized Mathematics*, 6(**1**):81–91, 1997.

[3] Grzegorz Bancerek. Directed sets, nets, ideals, filters, and maps. *Formalized Mathematics*, 6(**1**):93–107, 1997.

[4] Czesław Byliński. Galois connections. *Formalized Mathematics*, 6(**1**):131–143, 1997.

[5] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove, and D.S. Scott. *A Compendium of Continuous Lattices*. Springer-Verlag, Berlin, Heidelberg, New York, 1980.

[6] Adam Grabowski. Auxiliary and approximating relations. *Formalized Mathematics*, 6(**2**):179–188, 1997.

[7] Adam Grabowski and Robert Milewski. Boolean posets, posets under inclusion and products of relational structures. *Formalized Mathematics*, 6(**1**):117–121, 1997.

[8] Artur Korniłowicz. Definitions and properties of the join and meet of subsets. *Formalized Mathematics*, 6(**1**):153–158, 1997.

[9] Artur Korniłowicz. Meet–continuous lattices. *Formalized Mathematics*, 6(**1**):159–167, 1997.

[10] Beata Madras. Irreducible and prime elements. *Formalized Mathematics*, 6(**2**):233–239, 1997.

[11] Robert Milewski. Algebraic lattices. *Formalized Mathematics*, 6(**2**):249–254, 1997.

[12] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.

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

[14] Wojciech A. Trybulec. Partially ordered sets. *Formalized Mathematics*, 1(**2**):313–319, 1990.

[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.

Received February 9, 1998

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

Scott-Continuous Functions[1]

Adam Grabowski
University of Białystok

Summary. The article is a translation of [7, pp. 112–113].

MML Identifier: `WAYBEL17`.

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

1. Preliminaries

Let S be a non empty set and let a, b be elements of S. The functor $a, b, ...$ yields a function from $\mathbb{N}$ into S and is defined by the condition (Def. 1).

(Def. 1) Let i be a natural number. Then

(i) if there exists a natural number k such that $i = 2 \cdot k$, then $(a, b, ...)(i) = a$, and

(ii) if it is not true that there exists a natural number k such that $i = 2 \cdot k$, then $(a, b, ...)(i) = b$.

We now state two propositions:

(1) Let S, T be non empty reflexive relational structures, f be a map from S into T, and P be a lower subset of T. If f is monotone, then $f^{-1}(P)$ is lower.

(2) Let S, T be non empty reflexive relational structures, f be a map from S into T, and P be an upper subset of T. If f is monotone, then $f^{-1}(P)$ is upper.

[1]This work has been supported by KBN Grant 8 T11C 018 12.

ISSN 1426–2630

Let T be an up-complete lattice and let S be an inaccessible subset of T. Note that $-S$ is directly closed.

Next we state the proposition

(3) Let S, T be reflexive antisymmetric non empty relational structures and f be a map from S into T. If f is directed-sups-preserving, then f is monotone.

Let S, T be reflexive antisymmetric non empty relational structures. Observe that every map from S into T which is directed-sups-preserving is also monotone.

Next we state the proposition

(4) Let S, T be up-complete Scott top-lattices and f be a map from S into T. If f is continuous, then f is monotone.

2. Poset of Continuous Maps

Let S be a set and let T be a reflexive relational structure. One can verify that $S \longmapsto T$ is reflexive-yielding.

Let S be a non empty set and let T be a complete lattice. Observe that T^S is complete.

Let S, T be up-complete Scott top-lattices. The functor $\mathrm{SCMaps}(S,T)$ yields a strict full relational substructure of $\mathrm{MonMaps}(S,T)$ and is defined by:

(Def. 2) For every map f from S into T holds $f \in$ the carrier of $\mathrm{SCMaps}(S,T)$ iff f is continuous.

Let S, T be up-complete Scott top-lattices. Note that $\mathrm{SCMaps}(S,T)$ is non empty.

3. Some Special Nets

Let S be a non empty relational structure and let a, b be elements of the carrier of S. The functor $\mathrm{NetStr}(a,b)$ yields a strict non empty net structure over S and is defined by the conditions (Def. 3).

(Def. 3)(i) The carrier of $\mathrm{NetStr}(a,b) = \mathbb{N}$,

(ii) the mapping of $\mathrm{NetStr}(a,b) = a, b, \ldots$, and

(iii) for all elements i, j of the carrier of $\mathrm{NetStr}(a,b)$ and for all natural numbers i', j' such that $i = i'$ and $j = j'$ holds $i \leqslant j$ iff $i' \leqslant j'$.

Let S be a non empty relational structure and let a, b be elements of the carrier of S. Note that $\mathrm{NetStr}(a,b)$ is reflexive transitive directed and antisymmetric.

We now state four propositions:

(5) Let S be a non empty relational structure, a, b be elements of the carrier of S, and i be an element of the carrier of $\mathrm{NetStr}(a,b)$. Then $(\mathrm{NetStr}(a,b))(i) = a$ or $(\mathrm{NetStr}(a,b))(i) = b$.

(6) Let S be a non empty relational structure, a, b be elements of the carrier of S, i, j be elements of the carrier of $\mathrm{NetStr}(a,b)$, and i', j' be natural numbers such that $i' = i$ and $j' = i' + 1$ and $j' = j$. Then

(i) if $(\mathrm{NetStr}(a,b))(i) = a$, then $(\mathrm{NetStr}(a,b))(j) = b$, and

(ii) if $(\mathrm{NetStr}(a,b))(i) = b$, then $(\mathrm{NetStr}(a,b))(j) = a$.

(7) For every poset S with g.l.b.'s and for all elements a, b of the carrier of S holds $\liminf \mathrm{NetStr}(a,b) = a \sqcap b$.

(8) Let S, T be posets with g.l.b.'s, a, b be elements of the carrier of S, and f be a map from S into T. Then $\liminf(f \cdot \mathrm{NetStr}(a,b)) = f(a) \sqcap f(b)$.

Let S be a non empty relational structure and let D be a non empty subset of S. The functor $\mathrm{NetStr}(D)$ yielding a strict net structure over S is defined by:

(Def. 4) $\mathrm{NetStr}(D) = \langle D, (\text{the internal relation of } S) \mid^2 D, \mathrm{id}_{\text{the carrier of } S} \restriction D \rangle$.

We now state the proposition

(9) Let S be a non empty reflexive relational structure and D be a non empty subset of S. Then $\mathrm{NetStr}(D) = \mathrm{NetStr}(D, \mathrm{id}_{\text{the carrier of } S} \restriction D)$.

Let S be a non empty reflexive relational structure and let D be a directed non empty subset of S. Note that $\mathrm{NetStr}(D)$ is non empty directed and reflexive.

Let S be a non empty reflexive transitive relational structure and let D be a directed non empty subset of S. One can check that $\mathrm{NetStr}(D)$ is transitive.

Let S be a non empty reflexive relational structure and let D be a directed non empty subset of S. Observe that $\mathrm{NetStr}(D)$ is monotone.

We now state the proposition

(10) For every up-complete lattice S and for every directed non empty subset D of S holds $\liminf \mathrm{NetStr}(D) = \sup D$.

4. Monotone Maps

We now state several propositions:

(11) Let S, T be lattices and f be a map from S into T. If for every net N in S holds $f(\liminf N) \leqslant \liminf(f \cdot N)$, then f is monotone.

(12) Let S, T be continuous lower-bounded lattices and f be a map from S into T. Suppose f is directed-sups-preserving. Let x be an element of S. Then $f(x) = \bigsqcup_T \{f(w); w \text{ ranges over elements of } S\colon w \ll x\}$.

(13) Let S be a lattice, T be an up-complete lower-bounded lattice, and f be a map from S into T. Suppose that for every element x of S holds $f(x) = \bigsqcup_T \{f(w); w \text{ ranges over elements of } S\colon w \ll x\}$. Then f is monotone.

(14) Let S be an up-complete lower-bounded lattice, T be a continuous lower-bounded lattice, and f be a map from S into T. Suppose that for every element x of S holds $f(x) = \bigsqcup_T\{f(w); w$ ranges over elements of S: $w \ll x\}$. Let x be an element of S and y be an element of T. Then $y \ll f(x)$ if and only if there exists an element w of S such that $w \ll x$ and $y \ll f(w)$.

(15) Let S, T be non empty relational structures, D be a subset of S, and f be a map from S into T. Suppose that

(i) sup D exists in S and sup $f^\circ D$ exists in T, or

(ii) S is complete and antisymmetric and T is complete and antisymmetric.

If f is monotone, then $\sup(f^\circ D) \leqslant f(\sup D)$.

(16) Let S, T be non empty reflexive antisymmetric relational structures, D be a directed non empty subset of S, and f be a map from S into T. Suppose sup D exists in S and sup $f^\circ D$ exists in T or S is up-complete and T is up-complete. If f is monotone, then $\sup(f^\circ D) \leqslant f(\sup D)$.

(17) Let S, T be non empty relational structures, D be a subset of S, and f be a map from S into T. Suppose that

(i) inf D exists in S and inf $f^\circ D$ exists in T, or

(ii) S is complete and antisymmetric and T is complete and antisymmetric.

If f is monotone, then $f(\inf D) \leqslant \inf(f^\circ D)$.

(18) Let S, T be up-complete lattices, f be a map from S into T, and N be a monotone non empty net structure over S. If f is monotone, then $f \cdot N$ is monotone.

Let S, T be up-complete lattices, let f be a monotone map from S into T, and let N be a monotone non empty net structure over S. Observe that $f \cdot N$ is monotone.

The following two propositions are true:

(19) Let S, T be up-complete lattices and f be a map from S into T. Suppose that for every net N in S holds $f(\liminf N) \leqslant \liminf(f \cdot N)$. Let D be a directed non empty subset of S. Then $\sup(f^\circ D) = f(\sup D)$.

(20) Let S, T be complete lattices, f be a map from S into T, N be a net in S, j be an element of the carrier of N, and j' be an element of the carrier of $f \cdot N$. Suppose $j' = j$. Suppose f is monotone. Then $f(\bigsqcap_S\{N(k); k$ ranges over elements of the carrier of N: $k \geqslant j\}) \leqslant \bigsqcap_T\{(f \cdot N)(l); l$ ranges over elements of the carrier of $f \cdot N : l \geqslant j'\}$.

5. Necessary and Sufficient Conditions of Scott-continuity

We now state two propositions:

(21) Let S, T be complete Scott top-lattices and f be a map from S into T. Then f is continuous if and only if for every net N in S holds $f(\liminf N) \leqslant \liminf(f \cdot N)$.

(22) Let S, T be complete Scott top-lattices and f be a map from S into T. Then f is continuous if and only if f is directed-sups-preserving.

Let S, T be complete Scott top-lattices. Observe that every map from S into T which is continuous is also directed-sups-preserving and every map from S into T which is directed-sups-preserving is also continuous.

One can prove the following propositions:

(23) Let S, T be continuous complete Scott top-lattices and f be a map from S into T. Then f is continuous if and only if for every element x of S and for every element y of T holds $y \ll f(x)$ iff there exists an element w of S such that $w \ll x$ and $y \ll f(w)$.

(24) Let S, T be continuous complete Scott top-lattices and f be a map from S into T. Then f is continuous if and only if for every element x of S holds $f(x) = \bigsqcup_T \{f(w); w$ ranges over elements of S: $w \ll x\}$.

(25) Let S be a lattice, T be a complete lattice, and f be a map from S into T. Suppose that for every element x of S holds $f(x) = \bigsqcup_T \{f(w); w$ ranges over elements of S: $w \leqslant x \ \wedge \ w$ is compact$\}$. Then f is monotone.

(26) Let S, T be complete Scott top-lattices and f be a map from S into T. Suppose that for every element x of S holds $f(x) = \bigsqcup_T \{f(w); w$ ranges over elements of S: $w \leqslant x \ \wedge \ w$ is compact$\}$. Let x be an element of S. Then $f(x) = \bigsqcup_T \{f(w); w$ ranges over elements of S: $w \ll x\}$.

(27) Let S, T be complete Scott top-lattices and f be a map from S into T. Suppose S is algebraic and T is algebraic. Then f is continuous if and only if for every element x of S and for every element k of T such that $k \in$ the carrier of CompactSublatt(T) holds $k \leqslant f(x)$ iff there exists an element j of S such that $j \in$ the carrier of CompactSublatt(S) and $j \leqslant x$ and $k \leqslant f(j)$.

(28) Let S, T be complete Scott top-lattices and f be a map from S into T. Suppose S is algebraic and T is algebraic. Then f is continuous if and only if for every element x of S holds $f(x) = \bigsqcup_T \{f(w); w$ ranges over elements of S: $w \leqslant x \ \wedge \ w$ is compact$\}$.

References

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

[2] Grzegorz Bancerek. Complete lattices. *Formalized Mathematics*, 2(**5**):719–725, 1991.
[3] Grzegorz Bancerek. Bounds in posets and relational substructures. *Formalized Mathematics*, 6(**1**):81–91, 1997.
[4] Grzegorz Bancerek. Directed sets, nets, ideals, filters, and maps. *Formalized Mathematics*, 6(**1**):93–107, 1997.
[5] Grzegorz Bancerek. The "way-below" relation. *Formalized Mathematics*, 6(**1**):169–176, 1997.
[6] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[7] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove, and D.S. Scott. *A Compendium of Continuous Lattices*. Springer-Verlag, Berlin, Heidelberg, New York, 1980.
[8] Adam Grabowski. On the category of posets. *Formalized Mathematics*, 5(**4**):501–505, 1996.
[9] Artur Korniłowicz. On the topological properties of meet-continuous lattices. *Formalized Mathematics*, 6(**2**):269–277, 1997.
[10] Robert Milewski. Algebraic lattices. *Formalized Mathematics*, 6(**2**):249–254, 1997.
[11] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[12] Andrzej Trybulec. Natural transformations. Discrete categories. *Formalized Mathematics*, 2(**4**):467–474, 1991.
[13] Andrzej Trybulec. Scott topology. *Formalized Mathematics*, 6(**2**):311–319, 1997.
[14] Wojciech A. Trybulec. Partially ordered sets. *Formalized Mathematics*, 1(**2**):313–319, 1990.
[15] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[16] Mariusz Żynel and Czesław Byliński. Properties of relational structures, posets, lattices and maps. *Formalized Mathematics*, 6(**1**):123–130, 1997.

Received February 13, 1998

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

Natural Numbers

Robert Milewski
University of Białystok

MML Identifier: NAT_2.

The papers [6], [4], [2], [7], [1], [3], [5], and [8] provide the terminology and notation for this paper.

1. Preliminaries

In this article we present several logical schemes. The scheme *NonUniqRecExD* deals with a non empty set $\mathcal{A}$, an element $\mathcal{B}$ of $\mathcal{A}$, and a ternary predicate $\mathcal{P}$, and states that:

There exists a function f from $\mathbb{N}$ into $\mathcal{A}$ such that $f(0) = \mathcal{B}$ and for every element n of $\mathbb{N}$ holds $\mathcal{P}[n, f(n), f(n+1)]$

provided the following condition is satisfied:

- For every natural number n and for every element x of $\mathcal{A}$ there exists an element y of $\mathcal{A}$ such that $\mathcal{P}[n, x, y]$.

The scheme *NonUniqFinRecExD* deals with a non empty set $\mathcal{A}$, an element $\mathcal{B}$ of $\mathcal{A}$, a natural number $\mathcal{C}$, and a ternary predicate $\mathcal{P}$, and states that:

There exists a finite sequence p of elements of $\mathcal{A}$ such that $\operatorname{len} p = \mathcal{C}$ but $p(1) = \mathcal{B}$ or $\mathcal{C} = 0$ but for every natural number n such that $1 \leqslant n$ and $n \leqslant \mathcal{C} - 1$ holds $\mathcal{P}[n, p(n), p(n+1)]$

provided the parameters meet the following requirement:

- Let n be a natural number. Suppose $1 \leqslant n$ and $n \leqslant \mathcal{C} - 1$. Let x be an element of $\mathcal{A}$. Then there exists an element y of $\mathcal{A}$ such that $\mathcal{P}[n, x, y]$.

The scheme *NonUniqPiFinRecExD* deals with a non empty set $\mathcal{A}$, an element $\mathcal{B}$ of $\mathcal{A}$, a natural number $\mathcal{C}$, and a ternary predicate $\mathcal{P}$, and states that:

ISSN 1426–2630

There exists a finite sequence p of elements of $\mathcal{A}$ such that $\operatorname{len} p = \mathcal{C}$ but $\pi_1 p = \mathcal{B}$ or $\mathcal{C} = 0$ but for every natural number n such that $1 \leqslant n$ and $n \leqslant \mathcal{C} - 1$ holds $\mathcal{P}[n, \pi_n p, \pi_{n+1} p]$

provided the following condition is met:

- Let n be a natural number. Suppose $1 \leqslant n$ and $n \leqslant \mathcal{C} - 1$. Let x be an element of $\mathcal{A}$. Then there exists an element y of $\mathcal{A}$ such that $\mathcal{P}[n, x, y]$.

The following two propositions are true:

(1) For every real number x holds $x < \lfloor x \rfloor + 1$.

(2) For all real numbers x, y such that $x \geqslant 0$ and $y > 0$ holds $\frac{x}{\lfloor \frac{x}{y} \rfloor + 1} < y$.

2. Division and Rest of Division

The following propositions are true:

(3) For every natural number n holds n is empty iff $n = 0$.

(4) For every natural number n holds $0 \div n = 0$.

(5) For every non empty natural number n holds $n \div n = 1$.

(6) For every natural number n holds $n \div 1 = n$.

(7) For all natural numbers i, j, k, l such that $i \leqslant j$ and $k \leqslant j$ holds if $i = (j -' k) + l$, then $k = (j -' i) + l$.

(8) For all natural numbers i, n such that $i \in \operatorname{Seg} n$ holds $(n -' i) + 1 \in \operatorname{Seg} n$.

(9) For all natural numbers i, j such that $j < i$ holds $(i -' (j+1)) + 1 = i -' j$.

(10) For all natural numbers i, j such that $i \geqslant j$ holds $j -' i = 0$.

(11) For all non empty natural numbers i, j holds $i -' j < i$.

(12) Let n, k be natural numbers. Suppose $k \leqslant n$. Then the n-th power of 2 $=$ (the k-th power of 2) $\cdot$(the $(n -' k)$-th power of 2).

(13) For all natural numbers n, k such that $k \leqslant n$ holds the k-th power of 2 $\mid$ the n-th power of 2.

(14) For all natural numbers n, k such that $k > 0$ and $n \div k = 0$ holds $n < k$.

(15) For all natural numbers n, k such that $k > 0$ and $k \leqslant n$ holds $n \div k \geqslant 1$.

(16) For all natural numbers n, k such that $k \neq 0$ holds $(n+k) \div k = (n \div k) + 1$.

(17) For all natural numbers n, k, i such that $k \mid n$ and $1 \leqslant n$ and $1 \leqslant i$ and $i \leqslant k$ holds $(n -' i) \div k = (n \div k) - 1$.

(18) Let n, k be natural numbers. Suppose $k \leqslant n$. Then (the n-th power of 2) $\div$(the k-th power of 2) $=$ the $(n -' k)$-th power of 2.

(19) For every natural number n such that $n > 0$ holds (the n-th power of 2) $\operatorname{mod} 2 = 0$.

(20) For every natural number n such that $n > 0$ holds $n \bmod 2 = 0$ iff $(n -' 1) \bmod 2 = 1$.

(21) For every non empty natural number n such that $n \neq 1$ holds $n > 1$.

(22) For all natural numbers n, k such that $n \leqslant k$ and $k < n + n$ holds $k \div n = 1$.

(23) For every natural number n holds n is even iff $n \bmod 2 = 0$.

(24) For every natural number n holds n is odd iff $n \bmod 2 = 1$.

(25) For all natural numbers n, k, t such that $1 \leqslant t$ and $k \leqslant n$ and $2 \cdot t \mid k$ holds $n \div t$ is even iff $(n -' k) \div t$ is even.

(26) For all natural numbers n, m, k such that $n \leqslant m$ holds $n \div k \leqslant m \div k$.

(27) For all natural numbers n, k such that $k \leqslant 2 \cdot n$ holds $(k + 1) \div 2 \leqslant n$.

(28) For every even natural number n holds $n \div 2 = (n + 1) \div 2$.

(29) For all natural numbers n, k, i holds $n \div k \div i = n \div k \cdot i$.

Let n be a natural number. We say that n is non trivial if and only if:

(Def. 1) $n \neq 0$ and $n \neq 1$.

One can verify that every natural number which is non trivial is also non empty.

One can check that there exists a natural number which is non trivial.

The following two propositions are true:

(30) For every natural number k holds k is non trivial iff k is non empty and $k \neq 1$.

(31) For every non trivial natural number k holds $k \geqslant 2$.

The scheme *Ind from 2* concerns a unary predicate $\mathcal{P}$, and states that:

For every non trivial natural number k holds $\mathcal{P}[k]$

provided the following conditions are met:

- $\mathcal{P}[2]$, and
- For every non trivial natural number k such that $\mathcal{P}[k]$ holds $\mathcal{P}[k+1]$.

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

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

[5] Konrad Raczkowski and Andrzej Nędzusiak. Serieses. *Formalized Mathematics*, 2(**4**):449–452, 1991.

[6] Piotr Rudnicki and Andrzej Trybulec. Abian's fixed point theorem. *Formalized Mathematics*, 6(**3**):335–338, 1997.

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

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

Received February 23, 1998

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

Binary Arithmetics. Binary Sequences

Robert Milewski
University of Białystok

MML Identifier: BINARI_3.

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

1. Binary Arithmetics

The following propositions are true:

(1) For every non empty natural number n and for every tuple F of n and *Boolean* holds $\mathrm{Absval}(F) <$ the n-th power of 2.

(2) For every non empty natural number n and for all tuples F_1, F_2 of n and *Boolean* such that $\mathrm{Absval}(F_1) = \mathrm{Absval}(F_2)$ holds $F_1 = F_2$.

(3) For all finite sequences t_1, t_2 such that $\mathrm{Rev}(t_1) = \mathrm{Rev}(t_2)$ holds $t_1 = t_2$.

(4) For every natural number n holds $\langle\underbrace{0,\ldots,0}_{n+1}\rangle = \langle\underbrace{0,\ldots,0}_{n}\rangle \frown \langle 0\rangle$.

(5) For every natural number n holds $\langle\underbrace{0,\ldots,0}_{n}\rangle \in \mathit{Boolean}^*$.

(6) For every natural number n and for every tuple y of n and *Boolean* such that $y = \langle\underbrace{0,\ldots,0}_{n}\rangle$ holds $\neg y = n \longmapsto 1$.

(7) For every non empty natural number n and for every tuple F of n and *Boolean* such that $F = \langle\underbrace{0,\ldots,0}_{n}\rangle$ holds $\mathrm{Absval}(F) = 0$.

(8) Let n be a non empty natural number and F be a tuple of n and *Boolean*. If $F = \langle\underbrace{0,\ldots,0}_{n}\rangle$, then $\mathrm{Absval}(\neg F) =$ (the n-th power of 2)-1.

ISSN 1426–2630

(9) For every natural number n holds $\mathrm{Rev}(\langle\underbrace{0,\ldots,0}_{n}\rangle) = \langle\underbrace{0,\ldots,0}_{n}\rangle$.

(10) For every natural number n and for every tuple y of n and *Boolean* such that $y = \langle\underbrace{0,\ldots,0}_{n}\rangle$ holds $\mathrm{Rev}(\neg y) = \neg y$.

(11) $\mathrm{Bin1}(1) = \langle \mathit{true} \rangle$.

(12) For every non empty natural number n holds $\mathrm{Absval}(\mathrm{Bin1}(n)) = 1$.

(13) For all elements x, y of *Boolean* holds $x \vee y = \mathit{true}$ iff $x = \mathit{true}$ or $y = \mathit{true}$ and $x \vee y = \mathit{false}$ iff $x = \mathit{false}$ and $y = \mathit{false}$.

(14) For all elements x, y of *Boolean* holds $\mathrm{add\_ovfl}(\langle x \rangle, \langle y \rangle) = \mathit{true}$ iff $x = \mathit{true}$ and $y = \mathit{true}$.

(15) $\neg\langle \mathit{false} \rangle = \langle \mathit{true} \rangle$.

(16) $\neg\langle \mathit{true} \rangle = \langle \mathit{false} \rangle$.

(17) $\langle \mathit{false} \rangle + \langle \mathit{false} \rangle = \langle \mathit{false} \rangle$.

(18) $\langle \mathit{false} \rangle + \langle \mathit{true} \rangle = \langle \mathit{true} \rangle$ and $\langle \mathit{true} \rangle + \langle \mathit{false} \rangle = \langle \mathit{true} \rangle$.

(19) $\langle \mathit{true} \rangle + \langle \mathit{true} \rangle = \langle \mathit{false} \rangle$.

(20) Let n be a non empty natural number and x, y be tuples of n and *Boolean*. Suppose $\pi_n x = \mathit{true}$ and $\pi_n \,\mathrm{carry}(x, \mathrm{Bin1}(n)) = \mathit{true}$. Let k be a non empty natural number. If $k \neq 1$ and $k \leqslant n$, then $\pi_k x = \mathit{true}$ and $\pi_k \,\mathrm{carry}(x, \mathrm{Bin1}(n)) = \mathit{true}$.

(21) For every non empty natural number n and for every tuple x of n and *Boolean* such that $\pi_n x = \mathit{true}$ and $\pi_n \,\mathrm{carry}(x, \mathrm{Bin1}(n)) = \mathit{true}$ holds $\mathrm{carry}(x, \mathrm{Bin1}(n)) = \neg\, \mathrm{Bin1}(n)$.

(22) Let n be a non empty natural number and x, y be tuples of n and *Boolean*. If $y = \langle\underbrace{0,\ldots,0}_{n}\rangle$ and $\pi_n x = \mathit{true}$ and $\pi_n \,\mathrm{carry}(x, \mathrm{Bin1}(n)) = \mathit{true}$, then $x = \neg y$.

(23) For every non empty natural number n and for every tuple y of n and *Boolean* such that $y = \langle\underbrace{0,\ldots,0}_{n}\rangle$ holds $\mathrm{carry}(\neg y, \mathrm{Bin1}(n)) = \neg\, \mathrm{Bin1}(n)$.

(24) Let n be a non empty natural number and x, y be tuples of n and *Boolean*. If $y = \langle\underbrace{0,\ldots,0}_{n}\rangle$, then $\mathrm{add\_ovfl}(x, \mathrm{Bin1}(n)) = \mathit{true}$ iff $x = \neg y$.

(25) For every non empty natural number n and for every tuple z of n and *Boolean* such that $z = \langle\underbrace{0,\ldots,0}_{n}\rangle$ holds $\neg z + \mathrm{Bin1}(n) = z$.

2. Binary Sequences

Let n, k be natural numbers. The functor n-BinarySequence(k) yielding a tuple of n and *Boolean* is defined by:

(Def. 1) For every natural number i such that $i \in \mathrm{Seg}\, n$ holds $\pi_i(n\text{-BinarySequence}(k)) = ((k \div (\text{the } (i -' 1)\text{-th power of } 2)) \mod 2 = 0 \to \mathit{false}, \mathit{true})$.

One can prove the following propositions:

(26) For every natural number n holds n-BinarySequence(0) $= \langle\underbrace{0, \ldots, 0}_{n}\rangle$.

(27) For all natural numbers n, k such that $k <$ the n-th power of 2 holds $((n+1)\text{-BinarySequence}(k))(n+1) = \mathit{false}$.

(28) Let n be a non empty natural number and k be a natural number. If $k <$ the n-th power of 2, then $(n+1)$-BinarySequence(k) $=$ (n-BinarySequence(k)) $\frown \langle \mathit{false} \rangle$.

(29) For every non empty natural number n holds $(n+1)$-BinarySequence(the n-th power of 2) $= \langle\underbrace{0, \ldots, 0}_{n}\rangle \frown \langle \mathit{true} \rangle$.

(30) Let n be a non empty natural number and k be a natural number. Suppose the n-th power of $2 \leqslant k$ and $k <$ the $(n+1)$-th power of 2. Then $((n+1)\text{-BinarySequence}(k))(n+1) = \mathit{true}$.

(31) Let n be a non empty natural number and k be a natural number. Suppose the n-th power of $2 \leqslant k$ and $k <$ the $(n+1)$-th power of 2. Then $(n+1)$-BinarySequence(k) $=$ (n-BinarySequence($k -'$ (the n-th power of 2)))$\frown\langle \mathit{true} \rangle$.

(32) Let n be a non empty natural number and k be a natural number. Suppose $k <$ the n-th power of 2. Let x be a tuple of n and *Boolean*. If $x = \langle\underbrace{0, \ldots, 0}_{n}\rangle$, then n-BinarySequence(k) $= \neg x$ iff $k =$ (the n-th power of 2)-1.

(33) Let n be a non empty natural number and k be a natural number. If $k + 1 <$ the n-th power of 2, then add_ovfl(n-BinarySequence(k), Bin1(n)) $= \mathit{false}$.

(34) Let n be a non empty natural number and k be a natural number. If $k+1 <$ the n-th power of 2, then n-BinarySequence($k+1$) $=$ (n-BinarySequence(k)) $+$ Bin1(n).

(35) For all natural numbers n, i holds $(n+1)$-BinarySequence(i) $= \langle i \bmod 2\rangle \frown (n\text{-BinarySequence}(i \div 2))$.

(36) For every non empty natural number n and for every natural number k

such that $k <$ the n-th power of 2 holds Absval(n-BinarySequence(k)) = k.

(37) For every non empty natural number n and for every tuple x of n and *Boolean* holds n-BinarySequence(Absval(x)) = x.

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. Binary operations applied to finite sequences. *Formalized Mathematics*, 1(**4**):643–649, 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. Some properties of restrictions of finite sequences. *Formalized Mathematics*, 5(**2**):241–245, 1996.

[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] Yasuho Mizuhara and Takaya Nishiyama. Binary arithmetics, addition and subtraction of integers. *Formalized Mathematics*, 5(**1**):27–29, 1996.

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

[11] Konrad Raczkowski and Andrzej Nędzusiak. Serieses. *Formalized Mathematics*, 2(**4**):449–452, 1991.

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

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

[14] Edmund Woronowicz. Many–argument relations. *Formalized Mathematics*, 1(**4**):733–737, 1990.

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

Received February 24, 1998

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

Full Trees

Robert Milewski
University of Białystok

MML Identifier: BINTREE2.

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

1. Trees and Binary Trees

One can prove the following propositions:

(1) For every set D and for every finite sequence p and for every natural number n such that $p \in D^*$ holds $p \restriction \operatorname{Seg} n \in D^*$.

(2) For every binary tree T holds every element of T is a finite sequence of elements of *Boolean*.

Let T be a binary tree. We see that the element of T is a finite sequence of elements of *Boolean*.

Next we state several propositions:

(3) For every tree T such that $T = \{0, 1\}^*$ holds T is binary.

(4) For every tree T such that $T = \{0, 1\}^*$ holds $\mathrm{Leaves}(T) = \emptyset$.

(5) Let T be a binary tree, n be a natural number, and t be an element of T. If $t \in T\text{-level}(n)$, then t is a tuple of n and *Boolean*.

(6) For every tree T such that for every element t of T holds $\operatorname{succ} t = \{t \frown \langle 0 \rangle, t \frown \langle 1 \rangle\}$ holds $\mathrm{Leaves}(T) = \emptyset$.

(7) For every tree T such that for every element t of T holds $\operatorname{succ} t = \{t \frown \langle 0 \rangle, t \frown \langle 1 \rangle\}$ holds T is binary.

(8) For every tree T holds $T = \{0, 1\}^*$ iff for every element t of T holds $\operatorname{succ} t = \{t \frown \langle 0 \rangle, t \frown \langle 1 \rangle\}$.

ISSN 1426–2630

In this article we present several logical schemes. The scheme *DecoratedBinTreeEx* deals with a non empty set $\mathcal{A}$, an element $\mathcal{B}$ of $\mathcal{A}$, and a ternary predicate $\mathcal{P}$, and states that:

There exists a binary tree D decorated with elements of $\mathcal{A}$ such that $\operatorname{dom} D = \{0,1\}^*$ and $D(\varepsilon) = \mathcal{B}$ and for every node x of D holds $\mathcal{P}[D(x), D(x \frown \langle 0 \rangle), D(x \frown \langle 1 \rangle)]$

provided the following requirement is met:

- For every element a of $\mathcal{A}$ there exist elements b, c of $\mathcal{A}$ such that $\mathcal{P}[a, b, c]$.

The scheme *DecoratedBinTreeEx1* deals with a non empty set $\mathcal{A}$, an element $\mathcal{B}$ of $\mathcal{A}$, and two binary predicates $\mathcal{P}$, $\mathcal{Q}$, and states that:

There exists a binary tree D decorated with elements of $\mathcal{A}$ such that $\operatorname{dom} D = \{0,1\}^*$ and $D(\varepsilon) = \mathcal{B}$ and for every node x of D holds $\mathcal{P}[D(x), D(x \frown \langle 0 \rangle)]$ and $\mathcal{Q}[D(x), D(x \frown \langle 1 \rangle)]$

provided the following requirements are met:

- For every element a of $\mathcal{A}$ there exists an element b of $\mathcal{A}$ such that $\mathcal{P}[a, b]$, and
- For every element a of $\mathcal{A}$ there exists an element b of $\mathcal{A}$ such that $\mathcal{Q}[a, b]$.

Let T be a binary tree and let n be a non empty natural number. The functor $\operatorname{NumberOnLevel}(n, T)$ yields a function from T-level(n) into $\mathbb{N}$ and is defined as follows:

(Def. 1) For every element t of T such that $t \in T$-level(n) and for every tuple F of n and *Boolean* such that $F = \operatorname{Rev}(t)$ holds $(\operatorname{NumberOnLevel}(n, T))(t) = \operatorname{Absval}(F) + 1$.

Let T be a binary tree and let n be a non empty natural number. Note that $\operatorname{NumberOnLevel}(n, T)$ is one-to-one.

2. Full Trees

Let T be a tree. We say that T is full if and only if:

(Def. 2) $T = \{0,1\}^*$.

We now state three propositions:

(9) $\{0,1\}^*$ is a tree.

(10) For every tree T such that $T = \{0,1\}^*$ and for every natural number n holds $\langle \underbrace{0, \ldots, 0}_{n} \rangle \in T$-level$(n)$.

(11) Let T be a tree. Suppose $T = \{0,1\}^*$. Let n be a non empty natural number and y be a tuple of n and *Boolean*. Then $y \in T$-level(n).

Let T be a binary tree and let n be a natural number. Observe that T-level(n) is finite.

One can check that every tree which is full is also binary.

One can verify that there exists a tree which is full.

One can prove the following proposition

(12) For every full tree T and for every non empty natural number n holds $\mathrm{Seg}\,(\text{the } n\text{-th power of } 2) \subseteq \mathrm{rng}\,\mathrm{NumberOnLevel}(n, T)$.

Let T be a full tree and let n be a non empty natural number. The functor FinSeqLevel(n, T) yielding a finite sequence of elements of T-level(n) is defined by:

(Def. 3) $\mathrm{FinSeqLevel}(n, T) = (\mathrm{NumberOnLevel}(n, T))^{-1}$.

Let T be a full tree and let n be a non empty natural number. Note that FinSeqLevel(n, T) is one-to-one.

Next we state a number of propositions:

(13) For every full tree T and for every non empty natural number n holds $(\mathrm{NumberOnLevel}(n, T))(\langle\underbrace{0, \ldots, 0}_{n}\rangle) = 1$.

(14) Let T be a full tree, n be a non empty natural number, and y be a tuple of n and *Boolean*. If $y = \langle\underbrace{0, \ldots, 0}_{n}\rangle$, then $(\mathrm{NumberOnLevel}(n, T))(\neg y) =$ the n-th power of 2.

(15) For every full tree T and for every non empty natural number n holds $(\mathrm{FinSeqLevel}(n, T))(1) = \langle\underbrace{0, \ldots, 0}_{n}\rangle$.

(16) Let T be a full tree, n be a non empty natural number, and y be a tuple of n and *Boolean*. If $y = \langle\underbrace{0, \ldots, 0}_{n}\rangle$, then $(\mathrm{FinSeqLevel}(n, T))(\text{the } n\text{-th power of } 2) = \neg y$.

(17) Let T be a full tree, n be a non empty natural number, and i be a natural number. If $i \in \mathrm{Seg}\,(\text{the } n\text{-th power of } 2)$, then $(\mathrm{FinSeqLevel}(n, T))(i) = \mathrm{Rev}(n\text{-BinarySequence}(i -' 1))$.

(18) For every full tree T and for every natural number n holds $\overline{\overline{T\text{-level}(n)}} =$ the n-th power of 2.

(19) For every full tree T and for every non empty natural number n holds $\mathrm{len}\,\mathrm{FinSeqLevel}(n, T) =$ the n-th power of 2.

(20) For every full tree T and for every non empty natural number n holds $\mathrm{dom}\,\mathrm{FinSeqLevel}(n, T) = \mathrm{Seg}\,(\text{the } n\text{-th power of } 2)$.

(21) For every full tree T and for every non empty natural number n holds $\mathrm{rng}\,\mathrm{FinSeqLevel}(n, T) = T\text{-level}(n)$.

(22) For every full tree T holds $(\mathrm{FinSeqLevel}(1, T))(1) = \langle 0 \rangle$.

(23) For every full tree T holds $(\mathrm{FinSeqLevel}(1,T))(2) = \langle 1 \rangle$.

(24) Let T be a full tree and n, i be non empty natural numbers. Suppose $i \leqslant$ the $(n+1)$-th power of 2. Let F be a tuple of n and *Boolean*. If $F = (\mathrm{FinSeqLevel}(n,T))((i+1) \div 2)$, then $(\mathrm{FinSeqLevel}(n+1,T))(i) = F \frown \langle (i+1) \bmod 2 \rangle$.

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. Introduction to trees. *Formalized Mathematics*, 1(**2**):421–427, 1990.
[4] Grzegorz Bancerek. König's lemma. *Formalized Mathematics*, 2(**3**):397–402, 1991.
[5] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[6] Grzegorz Bancerek and Piotr Rudnicki. On defining functions on binary trees. *Formalized Mathematics*, 5(**1**):9–13, 1996.
[7] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.
[8] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[9] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[10] Czesław Byliński. Some properties of restrictions of finite sequences. *Formalized Mathematics*, 5(**2**):241–245, 1996.
[11] Agata Darmochwał. The Euclidean space. *Formalized Mathematics*, 2(**4**):599–603, 1991.
[12] Robert Milewski. Binary arithmetics. Binary sequences. *Formalized Mathematics*, 7(**1**):23–26, 1998.
[13] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[14] Konrad Raczkowski and Andrzej Nędzusiak. Serieses. *Formalized Mathematics*, 2(**4**):449–452, 1991.
[15] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.
[16] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[17] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[18] Edmund Woronowicz. Many–argument relations. *Formalized Mathematics*, 1(**4**):733–737, 1990.
[19] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received February 25, 1998

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

On T_1 Reflex of Topological Space

Adam Naumowicz
University of Białystok

Mariusz Łapiński
University of Białystok

Summary. This article contains a definition of T_1 reflex of a topological space as a quotient space which is T_1 and fulfils the condition that every continuous map f from a topological space T into S being T_1 space can be considered as a superposition of two continuous maps: the first from T onto its T_1 reflex and the last from T_1 reflex of T into S.

MML Identifier: `T_1TOPSP`.

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

In this paper X denotes a non empty set and w denotes a set.

One can prove the following propositions:

(1) For every set y and for all functions f, g holds $(f \cdot g)^{-1}(y) = g^{-1}(f^{-1}(y))$.

(2) Let T be a non empty topological space, A be a non empty partition of the carrier of T, and y be a subset of the carrier of the decomposition space of A. Then (the projection onto $A)^{-1}(y) = \bigcup y$.

(3) For every non empty set X and for every partition S of X and for every subset A of S holds $\bigcup S \setminus \bigcup A = \bigcup(S \setminus A)$.

(4) For every non empty set X and for every subset A of X and for every partition S of X such that $A \in S$ holds $\bigcup(S \setminus \{A\}) = X \setminus A$.

(5) Let T be a non empty topological space, S be a non empty partition of the carrier of T, A be a subset of the decomposition space of S, and B be a subset of T. If $B = \bigcup A$, then A is closed iff B is closed.

Let X be a non empty set, let x be an element of X, and let S_1 be a partition of X. The functor $\mathrm{EqClass}(x, S_1)$ yielding a subset of X is defined by:

(Def. 1) $x \in \mathrm{EqClass}(x, S_1)$ and $\mathrm{EqClass}(x, S_1) \in S_1$.

Next we state two propositions:

ISSN 1426–2630

(6) For all partitions S_1, S_2 of X such that for every element x of X holds $\mathrm{EqClass}(x, S_1) = \mathrm{EqClass}(x, S_2)$ holds $S_1 = S_2$.

(7) For every non empty set X holds $\{X\}$ is a partition of X.

Let X be a set. Partition family of X is defined by:

(Def. 2) For every set S such that $S \in$ it holds S is a partition of X.

Let X be a non empty set. One can check that there exists a partition of X which is non empty.

One can prove the following proposition

(8) For every set X and for every partition p of X holds $\{p\}$ is a partition family of X.

Let X be a set. One can check that there exists a partition family of X which is non empty.

Next we state two propositions:

(9) For every partition S_1 of X and for all elements x, y of X such that $\mathrm{EqClass}(x, S_1)$ meets $\mathrm{EqClass}(y, S_1)$ holds $\mathrm{EqClass}(x, S_1) = \mathrm{EqClass}(y, S_1)$.

(10) Let A be a set, X be a non empty set, and S be a partition of X. If $A \in S$, then there exists an element x of X such that $A = \mathrm{EqClass}(x, S)$.

Let X be a non empty set and let F be a non empty partition family of X. The functor $\mathrm{Intersection}\, F$ yields a non empty partition of X and is defined as follows:

(Def. 3) For every element x of X holds $\mathrm{EqClass}(x, \mathrm{Intersection}\, F) = \bigcap\{\mathrm{EqClass}(x, S); S$ ranges over partitions of X: $S \in F\}$.

In the sequel T denotes a non empty topological space.

One can prove the following proposition

(11) $\{A; A$ ranges over partitions of the carrier of T: A is closed$\}$ is a partition family of the carrier of T.

Let us consider T. The functor $\mathrm{ClosedPartitions}\, T$ yields a non empty partition family of the carrier of T and is defined by:

(Def. 4) $\mathrm{ClosedPartitions}\, T = \{A; A$ ranges over partitions of the carrier of T: A is closed$\}$.

Let T be a non empty topological space. The functor $\mathrm{T}_1\text{-reflex}\, T$ yields a topological space and is defined as follows:

(Def. 5) $\mathrm{T}_1\text{-reflex}\, T =$ the decomposition space of $\mathrm{Intersection}\, \mathrm{ClosedPartitions}\, T$.

Let us consider T. Note that $\mathrm{T}_1\text{-reflex}\, T$ is strict and non empty.

Next we state the proposition

(12) For every non empty topological space T holds $\mathrm{T}_1\text{-reflex}\, T$ is a T_1 space.

Let T be a non empty topological space. The functor $\mathrm{T}_1\text{-reflect}\, T$ yielding a continuous map from T into $\mathrm{T}_1\text{-reflex}\, T$ is defined as follows:

(Def. 6) $\mathrm{T_1}$-reflect T = the projection onto Intersection ClosedPartitions T.

The following four propositions are true:

(13) Let T, T_1 be non empty topological spaces and f be a continuous map from T into T_1. Suppose T_1 is a $\mathrm{T_1}$ space. Then

(i) $\{f^{-1}(\{z\}); z$ ranges over elements of T_1: $z \in \operatorname{rng} f\}$ is a partition of the carrier of T, and

(ii) for every subset A of T such that $A \in \{f^{-1}(\{z\}); z$ ranges over elements of T_1: $z \in \operatorname{rng} f\}$ holds A is closed.

(14) Let T, T_1 be non empty topological spaces and f be a continuous map from T into T_1. Suppose T_1 is a $\mathrm{T_1}$ space. Let given w and x be an element of T. If $w = \mathrm{EqClass}(x, \text{Intersection ClosedPartitions}\,T)$, then $w \subseteq f^{-1}(\{f(x)\})$.

(15) Let T, T_1 be non empty topological spaces and f be a continuous map from T into T_1. Suppose T_1 is a $\mathrm{T_1}$ space. Let given w. Suppose $w \in$ the carrier of $\mathrm{T_1}$-reflex T. Then there exists an element z of T_1 such that $z \in \operatorname{rng} f$ and $w \subseteq f^{-1}(\{z\})$.

(16) Let T, T_1 be non empty topological spaces and f be a continuous map from T into T_1. Suppose T_1 is a $\mathrm{T_1}$ space. Then there exists a continuous map h from $\mathrm{T_1}$-reflex T into T_1 such that $f = h \cdot \mathrm{T_1}$-reflect T.

Let T, S be non empty topological spaces and let f be a continuous map from T into S. The functor $\mathrm{T_1}$-reflex f yields a continuous map from $\mathrm{T_1}$-reflex T into $\mathrm{T_1}$-reflex S and is defined as follows:

(Def. 7) $\mathrm{T_1}$-reflect $S \cdot f = \mathrm{T_1}$-reflex $f \cdot \mathrm{T_1}$-reflect T.

References

[1] Józef Białas and Yatsuka Nakamura. Dyadic numbers and $\mathrm{T_4}$ topological spaces. *Formalized Mathematics*, 5(**3**):361–366, 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] Agata Darmochwał. Families of subsets, subspaces and mappings in topological spaces. *Formalized Mathematics*, 1(**2**):257–261, 1990.

[5] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.

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

[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] Andrzej Trybulec. A Borsuk theorem on homotopy types. *Formalized Mathematics*, 2(**4**):535–545, 1991.

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

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

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

Received March 7, 1998

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

Bases and Refinements of Topologies[1]

Grzegorz Bancerek
University of Białystok

MML Identifier: YELLOW_9.

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

1. Subsets as Nets

Let A be a set and let B be a non empty set. Observe that B^A is non empty.

In this article we present several logical schemes. The scheme *FraenkelInvolution* deals with a non empty set $\mathcal{A}$, subsets $\mathcal{B}$, $\mathcal{C}$ of $\mathcal{A}$, and a unary functor $\mathcal{F}$ yielding an element of $\mathcal{A}$, and states that:

$\mathcal{B} = \{\mathcal{F}(a); a$ ranges over elements of $\mathcal{A} : a \in \mathcal{C}\}$

provided the parameters have the following properties:

- $\mathcal{C} = \{\mathcal{F}(a); a$ ranges over elements of $\mathcal{A} : a \in \mathcal{B}\}$, and
- For every element a of $\mathcal{A}$ holds $\mathcal{F}(\mathcal{F}(a)) = a$.

The scheme *FraenkelComplement1* deals with a non empty relational structure $\mathcal{A}$, a family $\mathcal{B}$ of subsets of $\mathcal{A}$, a set $\mathcal{C}$, and a unary functor $\mathcal{F}$ yielding a subset of $\mathcal{A}$, and states that:

$\mathcal{B}^{\rm c} = \{-\mathcal{F}(a); a$ ranges over elements of $\mathcal{A} : a \in \mathcal{C}\}$

provided the parameters meet the following requirement:

- $\mathcal{B} = \{\mathcal{F}(a); a$ ranges over elements of $\mathcal{A} : a \in \mathcal{C}\}$.

The scheme *FraenkelComplement2* deals with a non empty relational structure $\mathcal{A}$, a family $\mathcal{B}$ of subsets of $\mathcal{A}$, a set $\mathcal{C}$, and a unary functor $\mathcal{F}$ yielding a subset of $\mathcal{A}$, and states that:

$\mathcal{B}^{\rm c} = \{\mathcal{F}(a); a$ ranges over elements of $\mathcal{A} : a \in \mathcal{C}\}$

[1]This work has been supported by KBN Grant 8 T11C 018 12.

ISSN 1426–2630

provided the parameters meet the following requirement:

- $\mathcal{B} = \{-\mathcal{F}(a); a$ ranges over elements of $\mathcal{A} : a \in \mathcal{C}\}$.

We now state several propositions:

(1) For every non empty relational structure R and for all elements x, y of R holds $y \in -\uparrow x$ iff $x \not\leq y$.

(2) Let S be a 1-sorted structure, T be a non empty 1-sorted structure, f be a map from S into T, and X be a subset of the carrier of T. Then $-f^{-1}(X) = f^{-1}(-X)$.

(3) For every 1-sorted structure T and for every family F of subsets of T holds $F^{\rm c} = \{-a; a$ ranges over subsets of T: $a \in F\}$.

(4) Let R be a non empty relational structure and F be a subset of R. Then $\uparrow F = \bigcup\{\uparrow x; x$ ranges over elements of R: $x \in F\}$ and $\downarrow F = \bigcup\{\downarrow x; x$ ranges over elements of R: $x \in F\}$.

(5) Let R be a non empty relational structure, A be a family of subsets of R, and F be a subset of R. If $A = \{-\uparrow x; x$ ranges over elements of R: $x \in F\}$, then $\text{Intersect}(A) = -\uparrow F$.

Let us mention that there exists a FR-structure which is strict, trivial, reflexive, non empty, discrete, and finite.

One can check that there exists a top-lattice which is strict, complete, and trivial.

Let S be a non empty relational structure and let T be an upper-bounded non empty reflexive antisymmetric relational structure. Note that there exists a map from S into T which is infs-preserving.

Let S be a non empty relational structure and let T be a lower-bounded non empty reflexive antisymmetric relational structure. Note that there exists a map from S into T which is sups-preserving.

Let R, S be 1-sorted structures. Let us assume that the carrier of $S \subseteq$ the carrier of R. The functor $\text{incl}(S, R)$ yields a map from S into R and is defined as follows:

(Def. 1) $\text{incl}(S, R) = \text{id}_{\text{the carrier of } S}$.

Let R be a non empty relational structure and let S be a non empty relational substructure of R. One can check that $\text{incl}(S, R)$ is monotone.

Let R be a non empty relational structure and let X be a non empty subset of the carrier of R. Note that $\text{sub}(X)$ is non empty.

Let R be a non empty relational structure and let X be a non empty subset of the carrier of R. The functor $\langle X; \text{id}\rangle$ yielding a strict non empty net structure over R is defined as follows:

(Def. 2) $\langle X; \text{id}\rangle = \text{incl}(\text{sub}(X), R) \cdot \langle \text{sub}(X); \text{id}\rangle$.

The functor $\langle X^{\rm op}; \text{id}\rangle$ yielding a strict non empty net structure over R is defined as follows:

(Def. 3) $\langle X^{\mathrm{op}}; \mathrm{id}\rangle = \mathrm{incl}(\mathrm{sub}(X), R) \cdot \langle (\mathrm{sub}(X))^{\mathrm{op}}; \mathrm{id}\rangle$.

One can prove the following propositions:

(6) Let R be a non empty relational structure and X be a non empty subset of R. Then

(i) the carrier of $\langle X; \mathrm{id}\rangle = X$,

(ii) $\langle X; \mathrm{id}\rangle$ is a full relational substructure of R, and

(iii) for every element x of $\langle X; \mathrm{id}\rangle$ holds $\langle X; \mathrm{id}\rangle(x) = x$.

(7) Let R be a non empty relational structure and X be a non empty subset of R. Then

(i) the carrier of $\langle X^{\mathrm{op}}; \mathrm{id}\rangle = X$,

(ii) $\langle X^{\mathrm{op}}; \mathrm{id}\rangle$ is a full relational substructure of R^{op}, and

(iii) for every element x of $\langle X^{\mathrm{op}}; \mathrm{id}\rangle$ holds $\langle X^{\mathrm{op}}; \mathrm{id}\rangle(x) = x$.

Let R be a non empty reflexive relational structure and let X be a non empty subset of R. One can check that $\langle X; \mathrm{id}\rangle$ is reflexive and $\langle X^{\mathrm{op}}; \mathrm{id}\rangle$ is reflexive.

Let R be a non empty transitive relational structure and let X be a non empty subset of R. Observe that $\langle X; \mathrm{id}\rangle$ is transitive and $\langle X^{\mathrm{op}}; \mathrm{id}\rangle$ is transitive.

Let R be a non empty reflexive relational structure and let I be a directed subset of R. Note that $\mathrm{sub}(I)$ is directed.

Let R be a non empty reflexive relational structure and let I be a directed non empty subset of R. Note that $\langle I; \mathrm{id}\rangle$ is directed.

Let R be a non empty reflexive relational structure and let F be a filtered non empty subset of R. One can verify that $\langle (\mathrm{sub}(F))^{\mathrm{op}}; \mathrm{id}\rangle$ is directed.

Let R be a non empty reflexive relational structure and let F be a filtered non empty subset of R. Note that $\langle F^{\mathrm{op}}; \mathrm{id}\rangle$ is directed.

2. Operations on Families of Open Sets

One can prove the following propositions:

(8) For every topological space T such that T is empty holds the topology of $T = \{\emptyset\}$.

(9) Let T be a trivial non empty topological space. Then

(i) the topology of $T = 2^{\text{the carrier of } T}$, and

(ii) for every point x of T holds the carrier of $T = \{x\}$ and the topology of $T = \{\emptyset, \{x\}\}$.

(10) Let T be a trivial non empty topological space. Then $\{$the carrier of $T\}$ is a basis of T and $\emptyset$ is a prebasis of T and $\{\emptyset\}$ is a prebasis of T.

(11) For all sets X, Y and for every family A of subsets of X such that $A = \{Y\}$ holds $\mathrm{FinMeetCl}(A) = \{Y, X\}$ and $\mathrm{UniCl}(A) = \{Y, \emptyset\}$.

(12) For every set X and for all families A, B of subsets of X such that $A = B \cup \{X\}$ or $B = A \setminus \{X\}$ holds $\mathrm{Intersect}(A) = \mathrm{Intersect}(B)$.

(13) For every set X and for all families A, B of subsets of X such that $A = B \cup \{X\}$ or $B = A \setminus \{X\}$ holds $\mathrm{FinMeetCl}(A) = \mathrm{FinMeetCl}(B)$.

(14) Let X be a set and A be a family of subsets of X. Suppose $X \in A$. Let x be a set. Then $x \in \mathrm{FinMeetCl}(A)$ if and only if there exists a finite non empty family Y of subsets of X such that $Y \subseteq A$ and $x = \mathrm{Intersect}(Y)$.

(15) For every set X and for every family A of subsets of X holds $\mathrm{UniCl}(\mathrm{UniCl}(A)) = \mathrm{UniCl}(A)$.

(16) For every set X and for every empty family A of subsets of X holds $\mathrm{UniCl}(A) = \{\emptyset\}$.

(17) For every set X and for every empty family A of subsets of X holds $\mathrm{FinMeetCl}(A) = \{X\}$.

(18) For every set X and for every family A of subsets of X such that $A = \{\emptyset, X\}$ holds $\mathrm{UniCl}(A) = A$ and $\mathrm{FinMeetCl}(A) = A$.

(19) Let X, Y be sets, A be a family of subsets of X, and B be a family of subsets of Y. Then
  (i) if $A \subseteq B$, then $\mathrm{UniCl}(A) \subseteq \mathrm{UniCl}(B)$, and
  (ii) if $A = B$, then $\mathrm{UniCl}(A) = \mathrm{UniCl}(B)$.

(20) Let X, Y be sets, A be a family of subsets of X, and B be a family of subsets of Y. If $A = B$ and $X \in A$ and $X \subseteq Y$, then $\mathrm{FinMeetCl}(B) = \{Y\} \cup \mathrm{FinMeetCl}(A)$.

(21) For every set X and for every family A of subsets of X holds $\mathrm{UniCl}(\mathrm{FinMeetCl}(\mathrm{UniCl}(A))) = \mathrm{UniCl}(\mathrm{FinMeetCl}(A))$.

3. Bases

Next we state a number of propositions:

(22) Let T be a topological space and K be a family of subsets of T. Then the topology of $T = \mathrm{UniCl}(K)$ if and only if K is a basis of T.

(23) Let T be a topological space and K be a family of subsets of the carrier of T. Then K is a prebasis of T if and only if $\mathrm{FinMeetCl}(K)$ is a basis of T.

(24) Let T be a non empty topological space and B be a family of subsets of T. If $\mathrm{UniCl}(B)$ is a prebasis of T, then B is a prebasis of T.

(25) Let T_1, T_2 be topological spaces and B be a basis of T_1. Suppose the carrier of T_1 = the carrier of T_2 and B is a basis of T_2. Then the topology of T_1 = the topology of T_2.

(26) Let T_1, T_2 be topological spaces and P be a prebasis of T_1. Suppose the carrier of T_1 = the carrier of T_2 and P is a prebasis of T_2. Then the topology of T_1 = the topology of T_2.

(27) For every topological space T holds every basis of T is open and is a prebasis of T.

(28) For every topological space T holds every prebasis of T is open.

(29) Let T be a non empty topological space and B be a prebasis of T. Then $B \cup \{$the carrier of $T\}$ is a prebasis of T.

(30) For every topological space T and for every basis B of T holds $B \cup \{$the carrier of $T\}$ is a basis of T.

(31) Let T be a topological space, B be a basis of T, and A be a subset of T. Then A is open if and only if for every point p of T such that $p \in A$ there exists a subset a of T such that $a \in B$ and $p \in a$ and $a \subseteq A$.

(32) Let T be a topological space and B be a family of subsets of T. Suppose that

(i) $B \subseteq$ the topology of T, and

(ii) for every subset A of T such that A is open and for every point p of T such that $p \in A$ there exists a subset a of T such that $a \in B$ and $p \in a$ and $a \subseteq A$.

Then B is a basis of T.

(33) Let S be a topological space, T be a non empty topological space, K be a basis of T, and f be a map from S into T. Then f is continuous if and only if for every subset A of T such that $A \in K$ holds $f^{-1}(-A)$ is closed.

(34) Let S be a topological space, T be a non empty topological space, K be a basis of T, and f be a map from S into T. Then f is continuous if and only if for every subset A of T such that $A \in K$ holds $f^{-1}(A)$ is open.

(35) Let S be a topological space, T be a non empty topological space, K be a prebasis of T, and f be a map from S into T. Then f is continuous if and only if for every subset A of T such that $A \in K$ holds $f^{-1}(-A)$ is closed.

(36) Let S be a topological space, T be a non empty topological space, K be a prebasis of T, and f be a map from S into T. Then f is continuous if and only if for every subset A of T such that $A \in K$ holds $f^{-1}(A)$ is open.

(37) Let T be a non empty topological space, x be a point of T, X be a subset of T, and K be a basis of T. Suppose that for every subset A of T such that $A \in K$ and $x \in A$ holds A meets X. Then $x \in \overline{X}$.

(38) Let T be a non empty topological space, x be a point of T, X be a subset of T, and K be a prebasis of T. Suppose that for every finite family Z of subsets of T such that $Z \subseteq K$ and $x \in \mathrm{Intersect}(Z)$ holds $\mathrm{Intersect}(Z)$ meets X. Then $x \in \overline{X}$.

(39) Let T be a non empty topological space, K be a prebasis of T, x be a point of T, and N be a net in T. Suppose that for every subset A of T such that $A \in K$ and $x \in A$ holds N is eventually in A. Let S be a subset of T. If $\operatorname{rng} \operatorname{netmap}(N, T) \subseteq S$, then $x \in \overline{S}$.

4. Product Topologies

The following four propositions are true:

(40) Let T_1, T_2 be non empty topological spaces, B_1 be a basis of T_1, and B_2 be a basis of T_2. Then $\{[\!: a, b :\!]; a$ ranges over subsets of T_1, b ranges over subsets of T_2: $a \in B_1 \ \wedge \ b \in B_2\}$ is a basis of $[\!: T_1, T_2 :\!]$.

(41) Let T_1, T_2 be non empty topological spaces, B_1 be a prebasis of T_1, and B_2 be a prebasis of T_2. Then $\{[\!:$ the carrier of T_1, $b :\!]; b$ ranges over subsets of T_2: $b \in B_2\} \cup \{[\!: a,$ the carrier of $T_2 :\!]; a$ ranges over subsets of T_1: $a \in B_1\}$ is a prebasis of $[\!: T_1, T_2 :\!]$.

(42) Let X_1, X_2 be sets, A be a family of subsets of $[\!: X_1, X_2 :\!]$, A_1 be a non empty family of subsets of X_1, and A_2 be a non empty family of subsets of X_2. Suppose $A = \{[\!: a, b :\!]; a$ ranges over subsets of X_1, b ranges over subsets of X_2: $a \in A_1 \ \wedge \ b \in A_2\}$. Then $\operatorname{Intersect}(A) = [\!: \operatorname{Intersect}(A_1), \operatorname{Intersect}(A_2) :\!]$.

(43) Let T_1, T_2 be non empty topological spaces, B_1 be a prebasis of T_1, and B_2 be a prebasis of T_2. Suppose $\bigcup B_1 =$ the carrier of T_1 and $\bigcup B_2 =$ the carrier of T_2. Then $\{[\!: a, b :\!]; a$ ranges over subsets of T_1, b ranges over subsets of T_2: $a \in B_1 \ \wedge \ b \in B_2\}$ is a prebasis of $[\!: T_1, T_2 :\!]$.

5. Topological Augmentations

Let R be a relational structure. A FR-structure is called a topological augmentation of R if:

(Def. 4) The relational structure of it $=$ the relational structure of R.

Let R be a relational structure and let T be a topological augmentation of R. We introduce T is correct as a synonym of T is topological space-like.

Let R be a relational structure. Note that there exists a topological augmentation of R which is correct, discrete, and strict.

We now state three propositions:

(44) Every FR-structure T is a topological augmentation of T.

(45) Let S be a FR-structure and T be a topological augmentation of S. Then S is a topological augmentation of T.

(46) Let R be a relational structure and T_1 be a topological augmentation of R. Then every topological augmentation of T_1 is a topological augmentation of R.

Let R be a non empty relational structure. One can check that every topological augmentation of R is non empty.

Let R be a reflexive relational structure. Note that every topological augmentation of R is reflexive.

Let R be a transitive relational structure. One can check that every topological augmentation of R is transitive.

Let R be an antisymmetric relational structure. One can verify that every topological augmentation of R is antisymmetric.

Let R be a complete non empty relational structure. Observe that every topological augmentation of R is complete.

We now state three propositions:

(47) Let S be a complete reflexive antisymmetric non empty relational structure and T be a non empty reflexive relational structure. Suppose the relational structure of $S =$ the relational structure of T. Let A be a subset of S and C be a subset of T. If $A = C$ and A is inaccessible, then C is inaccessible.

(48) Let S be a non empty reflexive relational structure and T be a topological augmentation of S. If the topology of $T = \sigma(S)$, then T is correct.

(49) Let S be a complete lattice and T be a topological augmentation of S. If the topology of $T = \sigma(S)$, then T is Scott.

Let R be a complete lattice. One can verify that there exists a topological augmentation of R which is Scott, strict, and correct.

The following three propositions are true:

(50) Let S, T be complete Scott non empty reflexive transitive antisymmetric FR-structures. Suppose the relational structure of $S =$ the relational structure of T. Let F be a subset of S and G be a subset of T. If $F = G$, then if F is open, then G is open.

(51) For every complete lattice S and for every Scott topological augmentation T of S holds the topology of $T = \sigma(S)$.

(52) Let S, T be complete lattices. Suppose the relational structure of $S =$ the relational structure of T. Then $\sigma(S) = \sigma(T)$.

Let R be a complete lattice. Observe that every topological augmentation of R which is Scott is also correct.

6. Refinements

Let T be a topological structure. A topological space is said to be a topological extension of T if:

(Def. 5) The carrier of T = the carrier of it and the topology of $T \subseteq$ the topology of it.

One can prove the following proposition

(53) Let S be a topological structure. Then there exists a topological extension T of S such that T is strict and the topology of S is a prebasis of T.

Let T be a topological structure. Note that there exists a topological extension of T which is strict and discrete.

Let T_1, T_2 be topological structures. A topological space is said to be a refinement of T_1 and T_2 if it satisfies the conditions (Def. 6).

(Def. 6)(i) The carrier of it = (the carrier of T_1) $\cup$ (the carrier of T_2), and

(ii) (the topology of T_1) $\cup$ (the topology of T_2) is a prebasis of it.

Let T_1 be a non empty topological structure and let T_2 be a topological structure. Observe that every refinement of T_1 and T_2 is non empty and every refinement of T_2 and T_1 is non empty.

The following propositions are true:

(54) Let T_1, T_2 be topological structures and T, T' be refinements of T_1 and T_2. Then the topological structure of T = the topological structure of T'.

(55) For all topological structures T_1, T_2 holds every refinement of T_1 and T_2 is a refinement of T_2 and T_1.

(56) Let T_1, T_2 be topological structures, T be a refinement of T_1 and T_2, and X be a family of subsets of T. Suppose X = (the topology of T_1) $\cup$ (the topology of T_2). Then the topology of T = UniCl(FinMeetCl(X)).

(57) Let T_1, T_2 be topological structures. Suppose the carrier of T_1 = the carrier of T_2. Then every refinement of T_1 and T_2 is a topological extension of T_1 and a topological extension of T_2.

(58) Let T_1, T_2 be non empty topological spaces, T be a refinement of T_1 and T_2, B_1 be a prebasis of T_1, and B_2 be a prebasis of T_2. Then $B_1 \cup B_2 \cup \{$the carrier of T_1, the carrier of $T_2\}$ is a prebasis of T.

(59) Let T_1, T_2 be non empty topological spaces, B_1 be a basis of T_1, B_2 be a basis of T_2, and T be a refinement of T_1 and T_2. Then $B_1 \cup B_2 \cup B_1 \Cap B_2$ is a basis of T.

(60) Let T_1, T_2 be non empty topological spaces. Suppose the carrier of T_1 = the carrier of T_2. Let T be a refinement of T_1 and T_2. Then (the topology of T_1) $\Cap$ (the topology of T_2) is a basis of T.

(61) Let L be a non empty relational structure, T_1, T_2 be correct topological augmentations of L, and T be a refinement of T_1 and T_2. Then (the topology of T_1) $\Cap$ (the topology of T_2) is a basis of T.

References

[1] Grzegorz Bancerek. Complete lattices. *Formalized Mathematics*, 2(**5**):719–725, 1991.
[2] Grzegorz Bancerek. Bounds in posets and relational substructures. *Formalized Mathematics*, 6(**1**):81–91, 1997.
[3] Grzegorz Bancerek. Directed sets, nets, ideals, filters, and maps. *Formalized Mathematics*, 6(**1**):93–107, 1997.
[4] Józef Białas. Group and field definitions. *Formalized Mathematics*, 1(**3**):433–439, 1990.
[5] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 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] Artur Korniłowicz. On the topological properties of meet-continuous lattices. *Formalized Mathematics*, 6(**2**):269–277, 1997.
[9] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[10] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[11] Alexander Yu. Shibakov and Andrzej Trybulec. The Cantor set. *Formalized Mathematics*, 5(**2**):233–236, 1996.
[12] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[13] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[14] Andrzej Trybulec. A Borsuk theorem on homotopy types. *Formalized Mathematics*, 2(**4**):535–545, 1991.
[15] Andrzej Trybulec. Scott topology. *Formalized Mathematics*, 6(**2**):311–319, 1997.
[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 defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received March 9, 1998

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

The Properties of Product of Relational Structures[1]

Artur Korniłowicz
University of Białystok

Summary. This work contains useful facts about the product of relational structures. It continues the formalization of [6].

MML Identifier: `YELLOW10`.

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

1. On the Elements of Product of Relational Structures

Let S, T be non empty upper-bounded relational structures. One can check that $[\!: S, T :\!]$ is upper-bounded.

Let S, T be non empty lower-bounded relational structures. Observe that $[\!: S, T :\!]$ is lower-bounded.

The following propositions are true:

(1) Let S, T be non empty relational structures. If $[\!: S, T :\!]$ is upper-bounded, then S is upper-bounded and T is upper-bounded.

(2) Let S, T be non empty relational structures. If $[\!: S, T :\!]$ is lower-bounded, then S is lower-bounded and T is lower-bounded.

(3) For all upper-bounded antisymmetric non empty relational structures S, T holds $\top_{[\!: S, T :\!]} = \langle \top_S, \top_T \rangle$.

(4) For all lower-bounded antisymmetric non empty relational structures S, T holds $\bot_{[\!: S, T :\!]} = \langle \bot_S, \bot_T \rangle$.

[1]This work has been supported by KBN Grant 8 T11C 018 12.

ISSN 1426–2630

(5) Let S, T be lower-bounded antisymmetric non empty relational structures and D be a subset of $[\!:S, T\,:\!]$. If $[\!:S, T\,:\!]$ is complete or sup D exists in $[\!:S, T\,:\!]$, then $\sup D = \langle \sup \pi_1(D), \sup \pi_2(D) \rangle$.

(6) Let S, T be upper-bounded antisymmetric non empty relational structures and D be a subset of $[\!:S, T\,:\!]$. If $[\!:S, T\,:\!]$ is complete or inf D exists in $[\!:S, T\,:\!]$, then $\inf D = \langle \inf \pi_1(D), \inf \pi_2(D) \rangle$.

(7) Let S, T be non empty relational structures and x, y be elements of $[\!:S, T\,:\!]$. Then $x \leqslant \{y\}$ if and only if the following conditions are satisfied:

(i) $x_{\mathbf{1}} \leqslant \{y_{\mathbf{1}}\}$, and

(ii) $x_{\mathbf{2}} \leqslant \{y_{\mathbf{2}}\}$.

(8) Let S, T be non empty relational structures and x, y, z be elements of $[\!:S, T\,:\!]$. Then $x \leqslant \{y, z\}$ if and only if the following conditions are satisfied:

(i) $x_{\mathbf{1}} \leqslant \{y_{\mathbf{1}}, z_{\mathbf{1}}\}$, and

(ii) $x_{\mathbf{2}} \leqslant \{y_{\mathbf{2}}, z_{\mathbf{2}}\}$.

(9) Let S, T be non empty relational structures and x, y be elements of $[\!:S, T\,:\!]$. Then $x \geqslant \{y\}$ if and only if the following conditions are satisfied:

(i) $x_{\mathbf{1}} \geqslant \{y_{\mathbf{1}}\}$, and

(ii) $x_{\mathbf{2}} \geqslant \{y_{\mathbf{2}}\}$.

(10) Let S, T be non empty relational structures and x, y, z be elements of $[\!:S, T\,:\!]$. Then $x \geqslant \{y, z\}$ if and only if the following conditions are satisfied:

(i) $x_{\mathbf{1}} \geqslant \{y_{\mathbf{1}}, z_{\mathbf{1}}\}$, and

(ii) $x_{\mathbf{2}} \geqslant \{y_{\mathbf{2}}, z_{\mathbf{2}}\}$.

(11) Let S, T be non empty antisymmetric relational structures and x, y be elements of $[\!:S, T\,:\!]$. Then inf $\{x, y\}$ exists in $[\!:S, T\,:\!]$ if and only if inf $\{x_{\mathbf{1}}, y_{\mathbf{1}}\}$ exists in S and inf $\{x_{\mathbf{2}}, y_{\mathbf{2}}\}$ exists in T.

(12) Let S, T be non empty antisymmetric relational structures and x, y be elements of $[\!:S, T\,:\!]$. Then sup $\{x, y\}$ exists in $[\!:S, T\,:\!]$ if and only if sup $\{x_{\mathbf{1}}, y_{\mathbf{1}}\}$ exists in S and sup $\{x_{\mathbf{2}}, y_{\mathbf{2}}\}$ exists in T.

(13) Let S, T be antisymmetric relational structures with g.l.b.'s and x, y be elements of $[\!:S, T\,:\!]$. Then $(x \sqcap y)_{\mathbf{1}} = x_{\mathbf{1}} \sqcap y_{\mathbf{1}}$ and $(x \sqcap y)_{\mathbf{2}} = x_{\mathbf{2}} \sqcap y_{\mathbf{2}}$.

(14) Let S, T be antisymmetric relational structures with l.u.b.'s and x, y be elements of $[\!:S, T\,:\!]$. Then $(x \sqcup y)_{\mathbf{1}} = x_{\mathbf{1}} \sqcup y_{\mathbf{1}}$ and $(x \sqcup y)_{\mathbf{2}} = x_{\mathbf{2}} \sqcup y_{\mathbf{2}}$.

(15) Let S, T be antisymmetric relational structures with g.l.b.'s, x_1, y_1 be elements of S, and x_2, y_2 be elements of T. Then $\langle x_1 \sqcap y_1, x_2 \sqcap y_2 \rangle = \langle x_1, x_2 \rangle \sqcap \langle y_1, y_2 \rangle$.

(16) Let S, T be antisymmetric relational structures with l.u.b.'s, x_1, y_1 be elements of S, and x_2, y_2 be elements of T. Then $\langle x_1 \sqcup y_1, x_2 \sqcup y_2 \rangle = \langle x_1, x_2 \rangle \sqcup \langle y_1, y_2 \rangle$.

Let S be an antisymmetric relational structure with l.u.b.'s and g.l.b.'s and let x, y be elements of S. Let us note that the predicate y is a complement of x is symmetric.

One can prove the following propositions:

(17) Let S, T be bounded antisymmetric relational structures with l.u.b.'s and g.l.b.'s and x, y be elements of $[\!:S, T:\!]$. Then x is a complement of y if and only if $x_{\mathbf{1}}$ is a complement of $y_{\mathbf{1}}$ and $x_{\mathbf{2}}$ is a complement of $y_{\mathbf{2}}$.

(18) Let S, T be antisymmetric up-complete non empty reflexive relational structures, a, c be elements of S, and b, d be elements of T. If $\langle a, b\rangle \ll \langle c, d\rangle$, then $a \ll c$ and $b \ll d$.

(19) Let S, T be up-complete non empty posets, a, c be elements of S, and b, d be elements of T. Then $\langle a, b\rangle \ll \langle c, d\rangle$ if and only if $a \ll c$ and $b \ll d$.

(20) Let S, T be antisymmetric up-complete non empty reflexive relational structures and x, y be elements of $[\!:S, T:\!]$. If $x \ll y$, then $x_{\mathbf{1}} \ll y_{\mathbf{1}}$ and $x_{\mathbf{2}} \ll y_{\mathbf{2}}$.

(21) Let S, T be up-complete non empty posets and x, y be elements of $[\!:S, T:\!]$. Then $x \ll y$ if and only if the following conditions are satisfied:

(i) $x_{\mathbf{1}} \ll y_{\mathbf{1}}$, and

(ii) $x_{\mathbf{2}} \ll y_{\mathbf{2}}$.

(22) Let S, T be antisymmetric up-complete non empty reflexive relational structures and x be an element of $[\!:S, T:\!]$. If x is compact, then $x_{\mathbf{1}}$ is compact and $x_{\mathbf{2}}$ is compact.

(23) Let S, T be up-complete non empty posets and x be an element of $[\!:S, T:\!]$. If $x_{\mathbf{1}}$ is compact and $x_{\mathbf{2}}$ is compact, then x is compact.

2. On the Subsets of Product of Relational Structures

The following propositions are true:

(24) Let S, T be antisymmetric relational structures with g.l.b.'s and X, Y be subsets of $[\!:S, T:\!]$. Then $\pi_1(X \sqcap Y) = \pi_1(X) \sqcap \pi_1(Y)$ and $\pi_2(X \sqcap Y) = \pi_2(X) \sqcap \pi_2(Y)$.

(25) Let S, T be antisymmetric relational structures with l.u.b.'s and X, Y be subsets of $[\!:S, T:\!]$. Then $\pi_1(X \sqcup Y) = \pi_1(X) \sqcup \pi_1(Y)$ and $\pi_2(X \sqcup Y) = \pi_2(X) \sqcup \pi_2(Y)$.

(26) For all relational structures S, T and for every subset X of $[\!:S, T:\!]$ holds $\downarrow X \subseteq [\!:\downarrow\pi_1(X), \downarrow\pi_2(X):\!]$.

(27) For all relational structures S, T and for every subset X of S and for every subset Y of T holds $[\!:\downarrow X, \downarrow Y:\!] = \downarrow[\!:X, Y:\!]$.

(28) For all relational structures S, T and for every subset X of $[\!: S, T :\!]$ holds $\pi_1(\downarrow X) \subseteq \downarrow\pi_1(X)$ and $\pi_2(\downarrow X) \subseteq \downarrow\pi_2(X)$.

(29) Let S be a relational structure, T be a reflexive relational structure, and X be a subset of $[\!: S, T :\!]$. Then $\pi_1(\downarrow X) = \downarrow\pi_1(X)$.

(30) Let S be a reflexive relational structure, T be a relational structure, and X be a subset of $[\!: S, T :\!]$. Then $\pi_2(\downarrow X) = \downarrow\pi_2(X)$.

(31) For all relational structures S, T and for every subset X of $[\!: S, T :\!]$ holds $\uparrow X \subseteq [\!: \uparrow\pi_1(X), \uparrow\pi_2(X) :\!]$.

(32) For all relational structures S, T and for every subset X of S and for every subset Y of T holds $[\!: \uparrow X, \uparrow Y :\!] = \uparrow[\!: X, Y :\!]$.

(33) For all relational structures S, T and for every subset X of $[\!: S, T :\!]$ holds $\pi_1(\uparrow X) \subseteq \uparrow\pi_1(X)$ and $\pi_2(\uparrow X) \subseteq \uparrow\pi_2(X)$.

(34) Let S be a relational structure, T be a reflexive relational structure, and X be a subset of $[\!: S, T :\!]$. Then $\pi_1(\uparrow X) = \uparrow\pi_1(X)$.

(35) Let S be a reflexive relational structure, T be a relational structure, and X be a subset of $[\!: S, T :\!]$. Then $\pi_2(\uparrow X) = \uparrow\pi_2(X)$.

(36) Let S, T be non empty relational structures, s be an element of S, and t be an element of T. Then $[\!: \downarrow s, \downarrow t :\!] = \downarrow\langle s, t\rangle$.

(37) For all non empty relational structures S, T and for every element x of $[\!: S, T :\!]$ holds $\pi_1(\downarrow x) \subseteq \downarrow(x_{\mathbf{1}})$ and $\pi_2(\downarrow x) \subseteq \downarrow(x_{\mathbf{2}})$.

(38) Let S be a non empty relational structure, T be a non empty reflexive relational structure, and x be an element of $[\!: S, T :\!]$. Then $\pi_1(\downarrow x) = \downarrow(x_{\mathbf{1}})$.

(39) Let S be a non empty reflexive relational structure, T be a non empty relational structure, and x be an element of $[\!: S, T :\!]$. Then $\pi_2(\downarrow x) = \downarrow(x_{\mathbf{2}})$.

(40) Let S, T be non empty relational structures, s be an element of S, and t be an element of T. Then $[\!: \uparrow s, \uparrow t :\!] = \uparrow\langle s, t\rangle$.

(41) For all non empty relational structures S, T and for every element x of $[\!: S, T :\!]$ holds $\pi_1(\uparrow x) \subseteq \uparrow(x_{\mathbf{1}})$ and $\pi_2(\uparrow x) \subseteq \uparrow(x_{\mathbf{2}})$.

(42) Let S be a non empty relational structure, T be a non empty reflexive relational structure, and x be an element of $[\!: S, T :\!]$. Then $\pi_1(\uparrow x) = \uparrow(x_{\mathbf{1}})$.

(43) Let S be a non empty reflexive relational structure, T be a non empty relational structure, and x be an element of $[\!: S, T :\!]$. Then $\pi_2(\uparrow x) = \uparrow(x_{\mathbf{2}})$.

(44) For all up-complete non empty posets S, T and for every element s of S and for every element t of T holds $[\!: \twoheaddownarrow s, \twoheaddownarrow t :\!] = \twoheaddownarrow\langle s, t\rangle$.

(45) Let S, T be antisymmetric up-complete non empty reflexive relational structures and x be an element of $[\!: S, T :\!]$. Then $\pi_1(\twoheaddownarrow x) \subseteq \twoheaddownarrow(x_{\mathbf{1}})$ and $\pi_2(\twoheaddownarrow x) \subseteq \twoheaddownarrow(x_{\mathbf{2}})$.

(46) Let S be an up-complete non empty poset, T be an up-complete lower-bounded non empty poset, and x be an element of $[\!: S, T :\!]$. Then $\pi_1(\twoheaddownarrow x) =$

$\downarrow(x_{\mathbf{1}})$.

(47) Let S be an up-complete lower-bounded non empty poset, T be an up-complete non empty poset, and x be an element of $[: S, T :]$. Then $\pi_2(\downarrow x) = \downarrow(x_{\mathbf{2}})$.

(48) For all up-complete non empty posets S, T and for every element s of S and for every element t of T holds $[: \uparrow s, \uparrow t :] = \uparrow\langle s, t\rangle$.

(49) Let S, T be antisymmetric up-complete non empty reflexive relational structures and x be an element of $[: S, T :]$. Then $\pi_1(\uparrow x) \subseteq \uparrow(x_{\mathbf{1}})$ and $\pi_2(\uparrow x) \subseteq \uparrow(x_{\mathbf{2}})$.

(50) For all up-complete non empty posets S, T and for every element s of S and for every element t of T holds $[: \mathrm{compactbelow}(s), \mathrm{compactbelow}(t) :] = \mathrm{compactbelow}(\langle s, t\rangle)$.

(51) Let S, T be antisymmetric up-complete non empty reflexive relational structures and x be an element of $[: S, T :]$. Then $\pi_1(\mathrm{compactbelow}(x)) \subseteq \mathrm{compactbelow}(x_{\mathbf{1}})$ and $\pi_2(\mathrm{compactbelow}(x)) \subseteq \mathrm{compactbelow}(x_{\mathbf{2}})$.

(52) Let S be an up-complete non empty poset, T be an up-complete lower-bounded non empty poset, and x be an element of $[: S, T :]$. Then $\pi_1(\mathrm{compactbelow}(x)) = \mathrm{compactbelow}(x_{\mathbf{1}})$.

(53) Let S be an up-complete lower-bounded non empty poset, T be an up-complete non empty poset, and x be an element of $[: S, T :]$. Then $\pi_2(\mathrm{compactbelow}(x)) = \mathrm{compactbelow}(x_{\mathbf{2}})$.

Let S be a non empty reflexive relational structure. One can verify that every subset of S which is empty is also open.

The following propositions are true:

(54) Let S, T be antisymmetric up-complete non empty reflexive relational structures and X be a subset of $[: S, T :]$. If X is open, then $\pi_1(X)$ is open and $\pi_2(X)$ is open.

(55) Let S, T be up-complete non empty posets, X be a subset of S, and Y be a subset of T. If X is open and Y is open, then $[: X, Y :]$ is open.

(56) Let S, T be antisymmetric up-complete non empty reflexive relational structures and X be a subset of $[: S, T :]$. If X is inaccessible, then $\pi_1(X)$ is inaccessible and $\pi_2(X)$ is inaccessible.

(57) Let S, T be antisymmetric up-complete non empty reflexive relational structures, X be an upper subset of S, and Y be an upper subset of T. If X is inaccessible and Y is inaccessible, then $[: X, Y :]$ is inaccessible.

(58) Let S, T be antisymmetric up-complete non empty reflexive relational structures, X be a subset of S, and Y be a subset of T such that $[: X, Y :]$ is directly closed. Then

(i) if $Y \neq \emptyset$, then X is directly closed, and

(ii) if $X \neq \emptyset$, then Y is directly closed.

(59) Let S, T be antisymmetric up-complete non empty reflexive relational structures, X be a subset of S, and Y be a subset of T. Suppose X is directly closed and Y is directly closed. Then $[: X, Y :]$ is directly closed.

(60) Let S, T be antisymmetric up-complete non empty reflexive relational structures and X be a subset of $[: S, T :]$. If X has the property (S), then $\pi_1(X)$ has the property (S) and $\pi_2(X)$ has the property (S).

(61) Let S, T be up-complete non empty posets, X be a subset of S, and Y be a subset of T. If X has the property (S) and Y has the property (S), then $[: X, Y :]$ has the property (S).

3. On the Products of Relational Structures

We now state the proposition

(62) Let S, T be non empty reflexive relational structures. Suppose the relational structure of S = the relational structure of T and S is inf-complete. Then T is inf-complete.

Let S be an inf-complete non empty reflexive relational structure. Observe that the relational structure of S is inf-complete.

Let S, T be inf-complete non empty reflexive relational structures. Observe that $[: S, T :]$ is inf-complete.

The following proposition is true

(63) Let S, T be non empty reflexive relational structures. If $[: S, T :]$ is inf-complete, then S is inf-complete and T is inf-complete.

Let S, T be complemented bounded antisymmetric non empty relational structures with g.l.b.'s and l.u.b.'s. Observe that $[: S, T :]$ is complemented.

Next we state the proposition

(64) Let S, T be bounded antisymmetric relational structures with g.l.b.'s and l.u.b.'s. If $[: S, T :]$ is complemented, then S is complemented and T is complemented.

Let S, T be distributive antisymmetric non empty relational structures with g.l.b.'s and l.u.b.'s. Observe that $[: S, T :]$ is distributive.

The following propositions are true:

(65) Let S be an antisymmetric relational structure with g.l.b.'s and l.u.b.'s and T be a reflexive antisymmetric relational structure with g.l.b.'s and l.u.b.'s. If $[: S, T :]$ is distributive, then S is distributive.

(66) Let S be a reflexive antisymmetric relational structure with g.l.b.'s and l.u.b.'s and T be an antisymmetric relational structure with g.l.b.'s and l.u.b.'s. If $[: S, T :]$ is distributive, then T is distributive.

Let S, T be meet-continuous semilattices. Observe that $[\!: S, T :\!]$ satisfies MC.

We now state the proposition

(67) For all semilattices S, T such that $[\!: S, T :\!]$ is meet-continuous holds S is meet-continuous and T is meet-continuous.

Let S, T be up-complete inf-complete non empty posets satisfying axiom of approximation. Note that $[\!: S, T :\!]$ satisfies axiom of approximation.

Let S, T be continuous inf-complete non empty posets. Observe that $[\!: S, T :\!]$ is continuous.

Next we state the proposition

(68) Let S, T be up-complete lower-bounded non empty posets. If $[\!: S, T :\!]$ is continuous, then S is continuous and T is continuous.

Let S, T be up-complete lower-bounded sup-semilattices satisfying axiom K. Note that $[\!: S, T :\!]$ satisfies axiom K.

Let S, T be complete algebraic lower-bounded sup-semilattices. Note that $[\!: S, T :\!]$ is algebraic.

The following proposition is true

(69) For all lower-bounded non empty posets S, T such that $[\!: S, T :\!]$ is algebraic holds S is algebraic and T is algebraic.

Let S, T be arithmetic lower-bounded lattices. Note that $[\!: S, T :\!]$ is arithmetic.

Next we state the proposition

(70) For all lower-bounded lattices S, T such that $[\!: S, T :\!]$ is arithmetic holds S is arithmetic and T is arithmetic.

References

[1] Grzegorz Bancerek. Complete lattices. *Formalized Mathematics*, 2(**5**):719–725, 1991.

[2] Grzegorz Bancerek. Bounds in posets and relational substructures. *Formalized Mathematics*, 6(**1**):81–91, 1997.

[3] Grzegorz Bancerek. Directed sets, nets, ideals, filters, and maps. *Formalized Mathematics*, 6(**1**):93–107, 1997.

[4] Grzegorz Bancerek. The "way-below" relation. *Formalized Mathematics*, 6(**1**):169–176, 1997.

[5] Czesław Byliński. Galois connections. *Formalized Mathematics*, 6(**1**):131–143, 1997.

[6] G. Gierz, K.H. Hofmann, K. Keimel, J.D. Lawson, M. Mislove, and D.S. Scott. *A Compendium of Continuous Lattices*. Springer-Verlag, Berlin, Heidelberg, New York, 1980.

[7] Artur Korniłowicz. Cartesian products of relations and relational structures. *Formalized Mathematics*, 6(**1**):145–152, 1997.

[8] Artur Korniłowicz. Definitions and properties of the join and meet of subsets. *Formalized Mathematics*, 6(**1**):153–158, 1997.

[9] Artur Korniłowicz. Meet–continuous lattices. *Formalized Mathematics*, 6(**1**):159–167, 1997.

[10] Beata Madras. Irreducible and prime elements. *Formalized Mathematics*, 6(**2**):233–239, 1997.

[11] Robert Milewski. Algebraic lattices. *Formalized Mathematics*, 6(**2**):249–254, 1997.

[12] Andrzej Trybulec. Scott topology. *Formalized Mathematics*, 6(**2**):311–319, 1997.

[13] Wojciech A. Trybulec. Partially ordered sets. *Formalized Mathematics*, 1(**2**):313–319, 1990.

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

Received March 27, 1998

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

On the Characterization of Modular and Distributive Lattices[1]

Adam Naumowicz
University of Białystok

Summary. This article contains definitions of the "pentagon" lattice N_5 and the "diamond" lattice M_3. It is followed by the characterization of modular and distributive lattices depending on the possible shape of substructures. The last part treats of interval-like sublattices of any lattice.

MML Identifier: YELLOW11.

The papers [8], [5], [1], [7], [6], [3], [4], and [2] provide the notation and terminology for this paper.

1. Preliminaries

One can prove the following propositions:

(1) $3 = \{0, 1, 2\}$.

(2) $2 \setminus 1 = \{1\}$.

(3) $3 \setminus 1 = \{1, 2\}$.

(4) $3 \setminus 2 = \{2\}$.

(5) Let L be an antisymmetric reflexive relational structure with g.l.b.'s and l.u.b.'s and a, b be elements of L. Then $a \sqcap b = b$ if and only if $a \sqcup b = a$.

(6) For every lattice L and for all elements a, b, c of L holds $(a \sqcap b) \sqcup (a \sqcap c) \leqslant a \sqcap (b \sqcup c)$.

(7) For every lattice L and for all elements a, b, c of L holds $a \sqcup (b \sqcap c) \leqslant (a \sqcup b) \sqcap (a \sqcup c)$.

[1]This work has been supported by KBN Grant 8 T11C 018 12.

ISSN 1426–2630

(8) For every lattice L and for all elements a, b, c of L such that $a \leqslant c$ holds $a \sqcup (b \sqcap c) \leqslant (a \sqcup b) \sqcap c$.

2. Diamond and Pentagon

The relational structure N_5 is defined as follows:

(Def. 1) $N_5 = \langle \{0, 3 \setminus 1, 2, 3 \setminus 2, 3\}, \subseteq \rangle$.

Let us note that N_5 is strict reflexive transitive and antisymmetric and N_5 has g.l.b.'s and l.u.b.'s.

The relational structure M_3 is defined by:

(Def. 2) $M_3 = \langle \{0, 1, 2 \setminus 1, 3 \setminus 2, 3\}, \subseteq \rangle$.

Let us note that M_3 is strict reflexive transitive and antisymmetric and M_3 has g.l.b.'s and l.u.b.'s.

One can prove the following two propositions:

(9) Let L be a lattice. Then the following statements are equivalent
 (i) there exists a full sublattice K of L such that N_5 and K are isomorphic,
 (ii) there exist elements a, b, c, d, e of L such that $a \neq b$ and $a \neq c$ and $a \neq d$ and $a \neq e$ and $b \neq c$ and $b \neq d$ and $b \neq e$ and $c \neq d$ and $c \neq e$ and $d \neq e$ and $a \sqcap b = a$ and $a \sqcap c = a$ and $c \sqcap e = c$ and $d \sqcap e = d$ and $b \sqcap c = a$ and $b \sqcap d = b$ and $c \sqcap d = a$ and $b \sqcup c = e$ and $c \sqcup d = e$.

(10) Let L be a lattice. Then the following statements are equivalent
 (i) there exists a full sublattice K of L such that M_3 and K are isomorphic,
 (ii) there exist elements a, b, c, d, e of L such that $a \neq b$ and $a \neq c$ and $a \neq d$ and $a \neq e$ and $b \neq c$ and $b \neq d$ and $b \neq e$ and $c \neq d$ and $c \neq e$ and $d \neq e$ and $a \sqcap b = a$ and $a \sqcap c = a$ and $a \sqcap d = a$ and $b \sqcap e = b$ and $c \sqcap e = c$ and $d \sqcap e = d$ and $b \sqcap c = a$ and $b \sqcap d = a$ and $c \sqcap d = a$ and $b \sqcup c = e$ and $b \sqcup d = e$ and $c \sqcup d = e$.

Let L be a non empty relational structure. We say that L is modular if and only if:

(Def. 3) For all elements a, b, c of L such that $a \leqslant c$ holds $a \sqcup (b \sqcap c) = (a \sqcup b) \sqcap c$.

Let us note that every non empty antisymmetric reflexive relational structure with g.l.b.'s which is distributive is also modular.

Next we state two propositions:

(11) Let L be a lattice. Then L is modular if and only if it is not true that there exists a full sublattice K of L such that N_5 and K are isomorphic.

(12) Let L be a lattice. Suppose L is modular. Then L is distributive if and only if it is not true that there exists a full sublattice K of L such that M_3 and K are isomorphic.

3. Intervals of a Lattice

Let L be a non empty relational structure and let a, b be elements of L. The functor $[a,b]$ yielding a subset of L is defined as follows:

(Def. 4) For every element c of L holds $c \in [a,b]$ iff $a \leqslant c$ and $c \leqslant b$.

Let L be a non empty relational structure and let I_1 be a subset of L. We say that I_1 is interval if and only if:

(Def. 5) There exist elements a, b of L such that $I_1 = [a,b]$.

Let L be a non empty reflexive transitive relational structure. One can check that every subset of L which is non empty and interval is also directed and every subset of L which is non empty and interval is also filtered.

Let L be a non empty relational structure and let a, b be elements of L. Observe that $[a,b]$ is interval.

Next we state the proposition

(13) For every non empty reflexive transitive relational structure L and for all elements a, b of L holds $[a,b] = \uparrow a \cap \downarrow b$.

Let L be a poset with g.l.b.'s and let a, b be elements of L. Observe that $\mathrm{sub}([a,b])$ is meet-inheriting.

Let L be a poset with l.u.b.'s and let a, b be elements of L. Note that $\mathrm{sub}([a,b])$ is join-inheriting.

One can prove the following proposition

(14) Let L be a lattice and a, b be elements of L. If L is modular, then $\mathrm{sub}([b, a \sqcup b])$ and $\mathrm{sub}([a \sqcap b, a])$ are isomorphic.

Let us mention that there exists a lattice which is finite and non empty.

Let us note that every semilattice which is finite is also lower-bounded.

Let us note that every lattice which is finite is also complete.

References

[1] Grzegorz Bancerek. Complete lattices. *Formalized Mathematics*, 2(**5**):719–725, 1991.
[2] Grzegorz Bancerek. Bounds in posets and relational substructures. *Formalized Mathematics*, 6(**1**):81–91, 1997.
[3] Grzegorz Bancerek. Directed sets, nets, ideals, filters, and maps. *Formalized Mathematics*, 6(**1**):93–107, 1997.
[4] Czesław Byliński. Galois connections. *Formalized Mathematics*, 6(**1**):131–143, 1997.
[5] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[6] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[7] Wojciech A. Trybulec. Partially ordered sets. *Formalized Mathematics*, 1(**2**):313–319, 1990.
[8] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.

Received April 3, 1998

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

Injective Spaces[1]

Jarosław Gryko
University of Białystok

MML Identifier: WAYBEL18.

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

1. Product Topologies

The following propositions are true:

(1) Let x, y, z, Z be sets. Then $Z \subseteq \{x, y, z\}$ if and only if one of the following conditions is satisfied:

(i) $Z = \emptyset$, or
(ii) $Z = \{x\}$, or
(iii) $Z = \{y\}$, or
(iv) $Z = \{z\}$, or
(v) $Z = \{x, y\}$, or
(vi) $Z = \{y, z\}$, or
(vii) $Z = \{x, z\}$, or
(viii) $Z = \{x, y, z\}$.

(2) For every set X and for all families A, B of subsets of X such that $B = A \setminus \{\emptyset\}$ or $A = B \cup \{\emptyset\}$ holds $\mathrm{UniCl}(A) = \mathrm{UniCl}(B)$.

(3) Let T be a topological space and K be a family of subsets of T. Then K is a basis of T if and only if $K \setminus \{\emptyset\}$ is a basis of T.

Let F be a binary relation. We say that F is topological space yielding if and only if:

[1]This work has been supported by KBN Grant 8 T11C 018 12.

ISSN 1426–2630

(Def. 1) For every set x such that $x \in \operatorname{rng} F$ holds x is a topological space.

One can verify that every function which is topological space yielding is also 1-sorted yielding.

Let I be a set. Note that there exists a many sorted set indexed by I which is topological space yielding.

Let I be a set. One can check that there exists a many sorted set indexed by I which is topological space yielding and nonempty.

Let J be a non empty set, let A be a topological space yielding many sorted set indexed by J, and let j be an element of J. Then $A(j)$ is a topological space.

Let I be a set and let J be a topological space yielding many sorted set indexed by I. The product prebasis for J is a family of subsets of $\prod$ (the support of J) and is defined by the condition (Def. 2).

(Def. 2) Let x be a subset of $\prod$ (the support of J). Then $x \in$ the product prebasis for J if and only if there exists a set i and there exists a topological space T and there exists a subset V of T such that $i \in I$ and V is open and $T = J(i)$ and $x = \prod(($the support of $J) +\cdot (i, V))$.

Next we state the proposition

(4) For every set X and for every family A of subsets of X holds $\langle X, \mathrm{UniCl}(\mathrm{FinMeetCl}(A))\rangle$ is topological space-like.

Let I be a set and let J be a topological space yielding nonempty many sorted set indexed by I. The functor $\prod J$ yielding a strict topological space is defined by:

(Def. 3) The carrier of $\prod J = \prod$ (the support of J) and the product prebasis for J is a prebasis of $\prod J$.

Let I be a set and let J be a topological space yielding nonempty many sorted set indexed by I. One can check that $\prod J$ is non empty.

Let I be a non empty set, let J be a topological space yielding nonempty many sorted set indexed by I, and let i be an element of I. Then $J(i)$ is a non empty topological space.

Let I be a set and let J be a topological space yielding nonempty many sorted set indexed by I. Observe that every element of the carrier of $\prod J$ is function-like and relation-like.

Let I be a non empty set, let J be a topological space yielding nonempty many sorted set indexed by I, let x be an element of the carrier of $\prod J$, and let i be an element of I. Then $x(i)$ is an element of $J(i)$.

Let I be a non empty set, let J be a topological space yielding nonempty many sorted set indexed by I, and let i be an element of I. The functor $\mathrm{proj}(J, i)$ yielding a map from $\prod J$ into $J(i)$ is defined as follows:

(Def. 4) $\mathrm{proj}(J, i) = \mathrm{proj}($the support of J, $i)$.

One can prove the following propositions:

(5) Let I be a non empty set, J be a topological space yielding nonempty many sorted set indexed by I, i be an element of I, and P be a subset of the carrier of $J(i)$. Then $(\mathrm{proj}(J,i))^{-1}(P) = \prod((\text{the support of } J) +\cdot (i, P))$.

(6) Let I be a non empty set, J be a topological space yielding nonempty many sorted set indexed by I, and i be an element of I. Then $\mathrm{proj}(J,i)$ is continuous.

(7) Let X be a non empty topological space, I be a non empty set, J be a topological space yielding nonempty many sorted set indexed by I, and f be a map from X into $\prod J$. Then f is continuous if and only if for every element i of I holds $\mathrm{proj}(J,i) \cdot f$ is continuous.

2. Injective Spaces

Let Z be a topological structure. We say that Z is injective if and only if the condition (Def. 5) is satisfied.

(Def. 5) Let X be a non empty topological space and f be a map from X into Z. Suppose f is continuous. Let Y be a non empty topological space. Suppose X is a subspace of Y. Then there exists a map g from Y into Z such that g is continuous and $g{\restriction}\text{the carrier of } X = f$.

One can prove the following two propositions:

(8) Let I be a non empty set and J be a topological space yielding nonempty many sorted set indexed by I. If for every element i of I holds $J(i)$ is injective, then $\prod J$ is injective.

(9) Let T be a non empty topological space. Suppose T is injective. Let S be a non empty subspace of T. If S is a retract of T, then S is injective.

Let X be a 1-sorted structure, let Y be a topological structure, and let f be a map from X into Y. The functor $\mathrm{Im}\, f$ yielding a subspace of Y is defined as follows:

(Def. 6) $\mathrm{Im}\, f = Y {\restriction} \mathrm{rng}\, f$.

Let X be a non empty 1-sorted structure, let Y be a non empty topological structure, and let f be a map from X into Y. Note that $\mathrm{Im}\, f$ is non empty.

One can prove the following proposition

(10) Let X be a 1-sorted structure, Y be a topological structure, and f be a map from X into Y. Then the carrier of $\mathrm{Im}\, f = \mathrm{rng}\, f$.

Let X be a 1-sorted structure, let Y be a non empty topological structure, and let f be a map from X into Y. The functor f° yielding a map from X into $\mathrm{Im}\, f$ is defined by:

(Def. 7) $f^{\circ} = f$.

Next we state the proposition

(11) Let X, Y be non empty topological spaces and f be a map from X into Y. If f is continuous, then f° is continuous.

Let X be a 1-sorted structure, let Y be a non empty topological structure, and let f be a map from X into Y. One can verify that f° is onto.

Let X, Y be topological structures. We say that X is a topological retract of Y if and only if:

(Def. 8) There exists a map f from Y into Y such that f is continuous and $f \cdot f = f$ and $\mathrm{Im}\, f$ and X are homeomorphic.

The following proposition is true

(12) Let T, S be non empty topological spaces. Suppose T is injective. Let f be a map from T into S. If f° is a homeomorphism, then T is a topological retract of S.

The Sierpiński space is a strict topological structure and is defined by the conditions (Def. 9).

(Def. 9)(i) The carrier of the Sierpiński space $= \{0, 1\}$, and

(ii) the topology of the Sierpiński space $= \{\emptyset, \{1\}, \{0, 1\}\}$.

Let us note that the Sierpiński space is non empty and topological space-like.

One can check that the Sierpiński space is discernible.

Let us note that the Sierpiński space is injective.

Let I be a set and let S be a non empty 1-sorted structure. One can verify that $I \longmapsto S$ is nonempty.

Let I be a set and let T be a topological space. One can check that $I \longmapsto T$ is topological space yielding.

Let I be a set and let L be a reflexive relational structure. One can check that $I \longmapsto L$ is reflexive-yielding.

Let I be a non empty set and let L be a non empty antisymmetric relational structure. Note that $\prod(I \longmapsto L)$ is antisymmetric.

Let I be a non empty set and let L be a non empty transitive relational structure. One can check that $\prod(I \longmapsto L)$ is transitive.

The following two propositions are true:

(13) Let T be a Scott topological augmentation of $2^{1}_{\subseteq}$. Then the topology of $T =$ the topology of the Sierpiński space.

(14) Let I be a non empty set. Then $\{\prod(($the support of $I \longmapsto$ the Sierpiński space$)+\cdot(i, \{1\})) : i$ ranges over elements of $I\}$ is a prebasis of $\prod(I \longmapsto$ the Sierpiński space$)$.

Let I be a non empty set and let L be a complete lattice. One can check that $\prod(I \longmapsto L)$ is complete and has l.u.b.'s.

Let I be a non empty set and let X be an algebraic lower-bounded lattice. One can check that $\prod(I \longmapsto X)$ is algebraic.

Next we state several propositions:

(15) Let X be a non empty set. Then there exists a map f from $2^X_\subseteq$ into $\prod(X \longmapsto 2^1_\subseteq)$ such that f is isomorphic and for every subset Y of X holds $f(Y) = \chi_{Y,X}$.

(16) Let I be a non empty set and T be a Scott topological augmentation of $\prod(I \longmapsto 2^1_\subseteq)$. Then the topology of T = the topology of $\prod(I \longmapsto$ the Sierpiński space).

(17) Let T, S be non empty topological spaces. Suppose the carrier of T = the carrier of S and the topology of T = the topology of S and T is injective. Then S is injective.

(18) For every non empty set I holds every Scott topological augmentation of $\prod I \longmapsto 2^1_\subseteq$ is injective.

(19) Let T be a T_0-space. Then there exists a non empty set M and there exists a map f from T into $\prod(M \longmapsto$ the Sierpiński space) such that f° is a homeomorphism.

(20) Let T be a T_0-space. Suppose T is injective. Then there exists a non empty set M such that T is a topological retract of $\prod(M \longmapsto$ the Sierpiński space).

References

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

[2] Grzegorz Bancerek. Directed sets, nets, ideals, filters, and maps. *Formalized Mathematics*, 6(**1**):93–107, 1997.

[3] Grzegorz Bancerek. The "way-below" relation. *Formalized Mathematics*, 6(**1**):169–176, 1997.

[4] Grzegorz Bancerek. Bases and refinements of topologies. *Formalized Mathematics*, 7(**1**):35–43, 1998.

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

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

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

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

[9] Adam Grabowski and Robert Milewski. Boolean posets, posets under inclusion and products of relational structures. *Formalized Mathematics*, 6(**1**):117–121, 1997.

[10] Beata Madras. Product of family of universal algebras. *Formalized Mathematics*, 4(**1**):103–108, 1993.

[11] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, I. *Formalized Mathematics*, 5(**2**):167–172, 1996.

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

[13] Alexander Yu. Shibakov and Andrzej Trybulec. The Cantor set. *Formalized Mathematics*, 5(**2**):233–236, 1996.

[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. A Borsuk theorem on homotopy types. *Formalized Mathematics*, 2(**4**):535–545, 1991.
[17] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[18] Andrzej Trybulec. Scott topology. *Formalized Mathematics*, 6(**2**):311–319, 1997.
[19] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[20] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[21] Mariusz Żynel and Czesław Byliński. Properties of relational structures, posets, lattices and maps. *Formalized Mathematics*, 6(**1**):123–130, 1997.
[22] Mariusz Żynel and Adam Guzowski. T_0 topological spaces. *Formalized Mathematics*, 5(**1**):75–77, 1996.

Received April 17, 1998

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

On the Characterization of Hausdorff Spaces[1]

Artur Korniłowicz
University of Białystok

MML Identifier: YELLOW12.

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

1. The Properties of Some Functions

In this paper A, B, X, Y denote sets.

Let X be an empty set. Note that $\bigcup X$ is empty.

Next we state several propositions:

(1) $(\delta_X)^\circ A \subseteq [\!: A, A :\!]$.

(2) $(\delta_X)^{-1}([\!: A, A :\!]) \subseteq A$.

(3) For every subset A of X holds $(\delta_X)^{-1}([\!: A, A :\!]) = A$.

(4) $\mathrm{dom}\langle \pi_2(X \times Y), \pi_1(X \times Y)\rangle = [\!: X, Y :\!]$ and $\mathrm{rng}\langle \pi_2(X \times Y), \pi_1(X \times Y)\rangle = [\!: Y, X :\!]$.

(5) $\langle \pi_2(X \times Y), \pi_1(X \times Y)\rangle^\circ [\!: A, B :\!] \subseteq [\!: B, A :\!]$.

(6) For every subset A of X and for every subset B of Y holds $\langle \pi_2(X \times Y), \pi_1(X \times Y)\rangle^\circ [\!: A, B :\!] = [\!: B, A :\!]$.

(7) $\langle \pi_2(X \times Y), \pi_1(X \times Y)\rangle$ is one-to-one.

Let X, Y be sets. One can verify that $\langle \pi_2(X \times Y), \pi_1(X \times Y)\rangle$ is one-to-one.

The following proposition is true

(8) $\langle \pi_2(X \times Y), \pi_1(X \times Y)\rangle^{-1} = \langle \pi_2(Y \times X), \pi_1(Y \times X)\rangle$.

---

[1]This work has been supported by KBN Grant 8 T11C 018 12.

ISSN 1426–2630

2. The Properties of the Relational Structures

Next we state a number of propositions:

(9) Let L_1 be a semilattice, L_2 be a non empty relational structure, x, y be elements of L_1, and x_1, y_1 be elements of L_2. Suppose the relational structure of L_1 = the relational structure of L_2 and $x = x_1$ and $y = y_1$. Then $x \sqcap y = x_1 \sqcap y_1$.

(10) Let L_1 be a sup-semilattice, L_2 be a non empty relational structure, x, y be elements of L_1, and x_1, y_1 be elements of L_2. Suppose the relational structure of L_1 = the relational structure of L_2 and $x = x_1$ and $y = y_1$. Then $x \sqcup y = x_1 \sqcup y_1$.

(11) Let L_1 be a semilattice, L_2 be a non empty relational structure, X, Y be subsets of L_1, and X_1, Y_1 be subsets of L_2. Suppose the relational structure of L_1 = the relational structure of L_2 and $X = X_1$ and $Y = Y_1$. Then $X \sqcap Y = X_1 \sqcap Y_1$.

(12) Let L_1 be a sup-semilattice, L_2 be a non empty relational structure, X, Y be subsets of L_1, and X_1, Y_1 be subsets of L_2. Suppose the relational structure of L_1 = the relational structure of L_2 and $X = X_1$ and $Y = Y_1$. Then $X \sqcup Y = X_1 \sqcup Y_1$.

(13) Let L_1 be an antisymmetric up-complete non empty reflexive relational structure, L_2 be a non empty reflexive relational structure, x be an element of L_1, and y be an element of L_2. Suppose the relational structure of L_1 = the relational structure of L_2 and $x = y$. Then $\Downarrow x = \Downarrow y$ and $\Uparrow x = \Uparrow y$.

(14) Let L_1 be a meet-continuous semilattice and L_2 be a non empty reflexive relational structure. Suppose the relational structure of L_1 = the relational structure of L_2. Then L_2 is meet-continuous.

(15) Let L_1 be a continuous antisymmetric non empty reflexive relational structure and L_2 be a non empty reflexive relational structure. Suppose the relational structure of L_1 = the relational structure of L_2. Then L_2 is continuous.

(16) Let L_1, L_2 be relational structures, A be a subset of L_1, and J be a subset of L_2. Suppose the relational structure of L_1 = the relational structure of L_2 and $A = J$. Then $\mathrm{sub}(A) = \mathrm{sub}(J)$.

(17) Let L_1, L_2 be non empty relational structures, A be a relational substructure of L_1, and J be a relational substructure of L_2. Suppose that

(i) the relational structure of L_1 = the relational structure of L_2,

(ii) the relational structure of A = the relational structure of J, and

(iii) A is meet-inheriting.

Then J is meet-inheriting.

(18) Let L_1, L_2 be non empty relational structures, A be a relational substructure of L_1, and J be a relational substructure of L_2. Suppose that

(i) the relational structure of L_1 = the relational structure of L_2,

(ii) the relational structure of A = the relational structure of J, and

(iii) A is join-inheriting.

Then J is join-inheriting.

(19) Let L_1 be an up-complete antisymmetric non empty reflexive relational structure, L_2 be a non empty reflexive relational structure, X be a subset of L_1, and Y be a subset of L_2 such that the relational structure of L_1 = the relational structure of L_2 and $X = Y$ and X has the property (S). Then Y has the property (S).

(20) Let L_1 be an up-complete antisymmetric non empty reflexive relational structure, L_2 be a non empty reflexive relational structure, X be a subset of L_1, and Y be a subset of L_2. Suppose the relational structure of L_1 = the relational structure of L_2 and $X = Y$ and X is directly closed. Then Y is directly closed.

(21) Let N be an antisymmetric relational structure with g.l.b.'s, D, E be subsets of N, and X be an upper subset of N. If $D \cap X = \emptyset$, then $(D \sqcap E) \cap X = \emptyset$.

(22) Let R be a reflexive non empty relational structure. Then $\triangle_{\text{the carrier of } R} \subseteq$ (the internal relation of R) $\cap$ (the internal relation of $R^{\smile}$).

(23) Let R be an antisymmetric relational structure. Then (the internal relation of R) $\cap$ (the internal relation of $R^{\smile}$) $\subseteq \triangle_{\text{the carrier of } R}$.

(24) Let R be an upper-bounded semilattice and X be a subset of $[\!: R, R :\!]$. If $\inf (\sqcap_R)^{\circ}X$ exists in R, then $\sqcap_R$ preserves inf of X.

Let R be a complete semilattice. One can verify that $\sqcap_R$ is infs-preserving.

Next we state the proposition

(25) Let R be a lower-bounded sup-semilattice and X be a subset of $[\!: R, R :\!]$. If $\sup (\sqcup_R)^{\circ}X$ exists in R, then $\sqcup_R$ preserves sup of X.

Let R be a complete sup-semilattice. Note that $\sqcup_R$ is sups-preserving.

One can prove the following propositions:

(26) For every semilattice N and for every subset A of N such that $\mathrm{sub}(A)$ is meet-inheriting holds A is filtered.

(27) For every sup-semilattice N and for every subset A of N such that $\mathrm{sub}(A)$ is join-inheriting holds A is directed.

(28) Let N be a transitive relational structure and A, J be subsets of N. If A is coarser than $\uparrow J$, then $\uparrow A \subseteq \uparrow J$.

(29) For every transitive relational structure N and for all subsets A, J of N such that A is finer than $\downarrow J$ holds $\downarrow A \subseteq \downarrow J$.

(30) Let N be a non empty reflexive relational structure, x be an element of N, and X be a subset of N. If $x \in X$, then $\uparrow x \subseteq \uparrow X$.

(31) Let N be a non empty reflexive relational structure, x be an element of N, and X be a subset of N. If $x \in X$, then $\downarrow x \subseteq \downarrow X$.

3. On the Hausdorff Spaces

In the sequel R, S, T denote non empty topological spaces.

Let T be a non empty topological structure. One can verify that the topological structure of T is non empty.

Let T be a topological space. Observe that the topological structure of T is topological space-like.

Next we state three propositions:

(32) Let S, T be topological structures and B be a basis of S. Suppose the topological structure of S = the topological structure of T. Then B is a basis of T.

(33) Let S, T be topological structures and B be a prebasis of S. Suppose the topological structure of S = the topological structure of T. Then B is a prebasis of T.

(34) Every basis of T is non empty.

Let T be a non empty topological space. Note that every basis of T is non empty.

The following proposition is true

(35) For every point x of T holds every basis of x is non empty.

Let T be a non empty topological space and let x be a point of T. One can check that every basis of x is non empty.

Next we state a number of propositions:

(36) Let S_1, T_1, S_2, T_2 be non empty topological spaces, f be a map from S_1 into S_2, and g be a map from T_1 into T_2. Suppose that
- (i) the topological structure of S_1 = the topological structure of T_1,
- (ii) the topological structure of S_2 = the topological structure of T_2,
- (iii) $f = g$, and
- (iv) f is continuous.

Then g is continuous.

(37) $\triangle_{\text{the carrier of } T} = \{p; p$ ranges over points of $[\!: T, T :\!]$: π_1((the carrier of T) $\times$ the carrier of T)$(p) = \pi_2$((the carrier of T) $\times$ the carrier of T)$(p)\}$.

(38) $\delta_{\text{the carrier of } T}$ is a continuous map from T into $[\!: T, T :\!]$.

(39) π_1((the carrier of S) $\times$ the carrier of T) is a continuous map from $[\!: S, T :\!]$ into S.

(40) π_2((the carrier of S) $\times$ the carrier of T) is a continuous map from $[: S, T :]$ into T.

(41) Let f be a continuous map from T into S and g be a continuous map from T into R. Then $\langle f, g \rangle$ is a continuous map from T into $[: S, R :]$.

(42) $\langle \pi_2$((the carrier of S)$\times$the carrier of T)$, \pi_1$((the carrier of S)$\times$the carrier of T))$\rangle$ is a continuous map from $[: S, T :]$ into $[: T, S :]$.

(43) Let f be a map from $[: S, T :]$ into $[: T, S :]$. Suppose $f = \langle \pi_2$((the carrier of S) $\times$ the carrier of T)$, \pi_1$((the carrier of S) $\times$ the carrier of T))$\rangle$. Then f is a homeomorphism.

(44) $[: S, T :]$ and $[: T, S :]$ are homeomorphic.

(45) Let T be a Hausdorff non empty topological space and f, g be continuous maps from S into T. Then

(i) for every subset X of S such that $X = \{p; p$ ranges over points of S: $f(p) \neq g(p)\}$ holds X is open, and

(ii) for every subset X of S such that $X = \{p; p$ ranges over points of S: $f(p) = g(p)\}$ holds X is closed.

(46) T is Hausdorff iff for every subset A of $[: T, T :]$ such that $A = \triangle_{\text{the carrier of } T}$ holds A is closed.

Let S, T be topological structures. Note that there exists a refinement of S and T which is strict.

Let S be a non empty topological structure and let T be a topological structure. Observe that there exists a refinement of S and T which is strict and non empty and there exists a refinement of T and S which is strict and non empty.

We now state the proposition

(47) Let R, S, T be topological structures. Then R is a refinement of S and T if and only if the topological structure of R is a refinement of S and T.

For simplicity, we adopt the following convention: S_1, S_2, T_1, T_2 are non empty topological spaces, R is a refinement of $[: S_1, T_1 :]$ and $[: S_2, T_2 :]$, R_1 is a refinement of S_1 and S_2, and R_2 is a refinement of T_1 and T_2.

The following three propositions are true:

(48) Suppose the carrier of S_1 = the carrier of S_2 and the carrier of T_1 = the carrier of T_2. Then $\{[: U_1, V_1 :] \cap [: U_2, V_2 :]; U_1$ ranges over subsets of S_1, U_2 ranges over subsets of S_2, V_1 ranges over subsets of T_1, V_2 ranges over subsets of T_2: U_1 is open $\wedge$ U_2 is open $\wedge$ V_1 is open $\wedge$ V_2 is open$\}$ is a basis of R.

(49) Suppose the carrier of S_1 = the carrier of S_2 and the carrier of T_1 = the carrier of T_2. Then the carrier of $[: R_1, R_2 :]$ = the carrier of R and the topology of $[: R_1, R_2 :]$ = the topology of R.

(50) Suppose the carrier of S_1 = the carrier of S_2 and the carrier of T_1 = the carrier of T_2. Then $[: R_1, R_2 :]$ is a refinement of $[: S_1, T_1 :]$ and $[: S_2, T_2 :]$.

References

[1] Grzegorz Bancerek. Complete lattices. *Formalized Mathematics*, 2(**5**):719–725, 1991.
[2] Grzegorz Bancerek. Bounds in posets and relational substructures. *Formalized Mathematics*, 6(**1**):81–91, 1997.
[3] Grzegorz Bancerek. Directed sets, nets, ideals, filters, and maps. *Formalized Mathematics*, 6(**1**):93–107, 1997.
[4] Grzegorz Bancerek. The "way-below" relation. *Formalized Mathematics*, 6(**1**):169–176, 1997.
[5] Grzegorz Bancerek. Bases and refinements of topologies. *Formalized Mathematics*, 7(**1**):35–43, 1998.
[6] Czesław Byliński. Basic functions and operations on functions. *Formalized Mathematics*, 1(**1**):245–254, 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. Compact spaces. *Formalized Mathematics*, 1(**2**):383–386, 1990.
[11] Agata Darmochwał. Families of subsets, subspaces and mappings in topological spaces. *Formalized Mathematics*, 1(**2**):257–261, 1990.
[12] Artur Korniłowicz. Cartesian products of relations and relational structures. *Formalized Mathematics*, 6(**1**):145–152, 1997.
[13] Artur Korniłowicz. Definitions and properties of the join and meet of subsets. *Formalized Mathematics*, 6(**1**):153–158, 1997.
[14] Artur Korniłowicz. Meet–continuous lattices. *Formalized Mathematics*, 6(**1**):159–167, 1997.
[15] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[16] Konrad Raczkowski and Paweł Sadowski. Equivalence relations and classes of abstraction. *Formalized Mathematics*, 1(**3**):441–444, 1990.
[17] Alexander Yu. Shibakov and Andrzej Trybulec. The Cantor set. *Formalized Mathematics*, 5(**2**):233–236, 1996.
[18] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[19] Andrzej Trybulec. A Borsuk theorem on homotopy types. *Formalized Mathematics*, 2(**4**):535–545, 1991.
[20] Andrzej Trybulec. Baire spaces, Sober spaces. *Formalized Mathematics*, 6(**2**):289–294, 1997.
[21] Andrzej Trybulec. Scott topology. *Formalized Mathematics*, 6(**2**):311–319, 1997.
[22] Wojciech A. Trybulec. Partially ordered sets. *Formalized Mathematics*, 1(**2**):313–319, 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.
[26] Mariusz Żynel and Adam Guzowski. T_0 topological spaces. *Formalized Mathematics*, 5(**1**):75–77, 1996.

Received April 18, 1998

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

The Field of Quotients Over an Integral Domain

Christoph Schwarzweller
University of Tübingen

Summary. We introduce the field of quotients over an integral domain following the well-known construction using pairs over integral domains. In addition we define ring homomorphisms and prove some basic facts about fields of quotients including their universal property.

MML Identifier: `QUOFIELD`.

The papers [1], [13], [10], [2], [3], [7], [9], [11], [12], [5], [6], [8], and [4] provide the terminology and notation for this paper.

1. Preliminaries

Let I be a non empty zero structure. The functor $\mathrm{Q}(I)$ is a subset of $[\!:$ the carrier of I, the carrier of $I\, :\!]$ and is defined by:

(Def. 1) For every set u holds $u \in \mathrm{Q}(I)$ iff there exist elements a, b of the carrier of I such that $u = \langle a, b \rangle$ and $b \neq 0_I$.

Next we state the proposition

(1) For every non degenerated non empty multiplicative loop with zero structure I holds $\mathrm{Q}(I)$ is non empty.

The following two propositions are true:

(2) Let I be a non degenerated non empty multiplicative loop with zero structure and u be an element of $\mathrm{Q}(I)$. Then $u_{\mathbf{2}} \neq 0_I$.

(3) Let I be a non degenerated non empty multiplicative loop with zero structure and u be an element of $\mathrm{Q}(I)$. Then $u_{\mathbf{1}}$ is an element of the carrier of I and $u_{\mathbf{2}}$ is an element of the carrier of I.

ISSN 1426–2630

Let I be a non degenerated integral domain-like non empty double loop structure and let u, v be elements of $\mathrm{Q}(I)$. The functor $u+v$ yielding an element of $\mathrm{Q}(I)$ is defined by:

(Def. 2) $u+v=\langle u_{\mathbf{1}}\cdot v_{\mathbf{2}}+v_{\mathbf{1}}\cdot u_{\mathbf{2}},\, u_{\mathbf{2}}\cdot v_{\mathbf{2}}\rangle$.

Let I be a non degenerated integral domain-like non empty double loop structure and let u, v be elements of $\mathrm{Q}(I)$. The functor $u\cdot v$ yielding an element of $\mathrm{Q}(I)$ is defined as follows:

(Def. 3) $u\cdot v=\langle u_{\mathbf{1}}\cdot v_{\mathbf{1}},\, u_{\mathbf{2}}\cdot v_{\mathbf{2}}\rangle$.

The following two propositions are true:

(4) Let I be a non degenerated integral domain-like associative commutative Abelian add-associative distributive non empty double loop structure and u, v, w be elements of $\mathrm{Q}(I)$. Then $u+(v+w)=(u+v)+w$ and $u+v=v+u$.

(5) Let I be a non degenerated integral domain-like associative commutative Abelian non empty double loop structure and u, v, w be elements of $\mathrm{Q}(I)$. Then $u\cdot(v\cdot w)=(u\cdot v)\cdot w$ and $u\cdot v=v\cdot u$.

Let I be a non degenerated integral domain-like associative commutative Abelian add-associative distributive non empty double loop structure and let u, v be elements of $\mathrm{Q}(I)$. Let us notice that the functor $u+v$ is commutative.

Let I be a non degenerated integral domain-like associative commutative Abelian non empty double loop structure and let u, v be elements of $\mathrm{Q}(I)$. Let us note that the functor $u\cdot v$ is commutative.

Let I be a non degenerated non empty multiplicative loop with zero structure and let u be an element of $\mathrm{Q}(I)$. The functor $\mathrm{QClass}(u)$ is a subset of $\mathrm{Q}(I)$ and is defined as follows:

(Def. 4) For every element z of $\mathrm{Q}(I)$ holds $z\in\mathrm{QClass}(u)$ iff $z_{\mathbf{1}}\cdot u_{\mathbf{2}}=z_{\mathbf{2}}\cdot u_{\mathbf{1}}$.

The following proposition is true

(6) Let I be a non degenerated commutative non empty multiplicative loop with zero structure and u be an element of $\mathrm{Q}(I)$. Then $u\in\mathrm{QClass}(u)$.

Let I be a non degenerated commutative non empty multiplicative loop with zero structure and let u be an element of $\mathrm{Q}(I)$. Observe that $\mathrm{QClass}(u)$ is non empty.

Let I be a non degenerated non empty multiplicative loop with zero structure. The functor $\mathrm{Quot}(I)$ is a family of subsets of $\mathrm{Q}(I)$ and is defined by:

(Def. 5) For every subset A of $\mathrm{Q}(I)$ holds $A\in\mathrm{Quot}(I)$ iff there exists an element u of $\mathrm{Q}(I)$ such that $A=\mathrm{QClass}(u)$.

Next we state the proposition

(7) For every non degenerated non empty multiplicative loop with zero structure I holds $\mathrm{Quot}(I)$ is non empty.

Next we state two propositions:

(8) Let I be a non degenerated integral domain-like ring and u, v be elements of $\mathrm{Q}(I)$. If there exists an element w of $\mathrm{Quot}(I)$ such that $u \in w$ and $v \in w$, then $u_{\mathbf{1}} \cdot v_{\mathbf{2}} = v_{\mathbf{1}} \cdot u_{\mathbf{2}}$.

(9) For every non degenerated integral domain-like ring I and for all elements u, v of $\mathrm{Quot}(I)$ such that $u \cap v \neq \emptyset$ holds $u = v$.

2. Defining the Operations

Let I be a non degenerated integral domain-like ring and let u, v be elements of $\mathrm{Quot}(I)$. The functor $u +_{\mathrm{q}} v$ yielding an element of $\mathrm{Quot}(I)$ is defined by the condition (Def. 6).

(Def. 6) Let z be an element of $\mathrm{Q}(I)$. Then $z \in u +_{\mathrm{q}} v$ if and only if there exist elements a, b of $\mathrm{Q}(I)$ such that $a \in u$ and $b \in v$ and $z_{\mathbf{1}} \cdot (a_{\mathbf{2}} \cdot b_{\mathbf{2}}) = z_{\mathbf{2}} \cdot (a_{\mathbf{1}} \cdot b_{\mathbf{2}} + b_{\mathbf{1}} \cdot a_{\mathbf{2}})$.

Let I be a non degenerated integral domain-like ring and let u, v be elements of $\mathrm{Quot}(I)$. The functor $u \cdot_{\mathrm{q}} v$ yielding an element of $\mathrm{Quot}(I)$ is defined by the condition (Def. 7).

(Def. 7) Let z be an element of $\mathrm{Q}(I)$. Then $z \in u \cdot_{\mathrm{q}} v$ if and only if there exist elements a, b of $\mathrm{Q}(I)$ such that $a \in u$ and $b \in v$ and $z_{\mathbf{1}} \cdot (a_{\mathbf{2}} \cdot b_{\mathbf{2}}) = z_{\mathbf{2}} \cdot (a_{\mathbf{1}} \cdot b_{\mathbf{1}})$.

Next we state the proposition

(10) Let I be a non degenerated non empty multiplicative loop with zero structure and u be an element of $\mathrm{Q}(I)$. Then $\mathrm{QClass}(u)$ is an element of $\mathrm{Quot}(I)$.

We now state two propositions:

(11) For every non degenerated integral domain-like ring I and for all elements u, v of $\mathrm{Q}(I)$ holds $\mathrm{QClass}(u) +_{\mathrm{q}} \mathrm{QClass}(v) = \mathrm{QClass}(u + v)$.

(12) For every non degenerated integral domain-like ring I and for all elements u, v of $\mathrm{Q}(I)$ holds $\mathrm{QClass}(u) \cdot_{\mathrm{q}} \mathrm{QClass}(v) = \mathrm{QClass}(u \cdot v)$.

Let I be a non degenerated integral domain-like ring. The functor $0_{\mathrm{q}}(I)$ yielding an element of $\mathrm{Quot}(I)$ is defined by:

(Def. 8) For every element z of $\mathrm{Q}(I)$ holds $z \in 0_{\mathrm{q}}(I)$ iff $z_{\mathbf{1}} = 0_I$.

Let I be a non degenerated integral domain-like ring. The functor $1_{\mathrm{q}}(I)$ yielding an element of $\mathrm{Quot}(I)$ is defined as follows:

(Def. 9) For every element z of $\mathrm{Q}(I)$ holds $z \in 1_{\mathrm{q}}(I)$ iff $z_{\mathbf{1}} = z_{\mathbf{2}}$.

Let I be a non degenerated integral domain-like ring and let u be an element of $\mathrm{Quot}(I)$. The functor $-_{\mathrm{q}}u$ yielding an element of $\mathrm{Quot}(I)$ is defined by:

(Def. 10) For every element z of $\mathrm{Q}(I)$ holds $z \in -_{\mathrm{q}}u$ iff there exists an element a of $\mathrm{Q}(I)$ such that $a \in u$ and $z_{\mathbf{1}} \cdot a_{\mathbf{2}} = z_{\mathbf{2}} \cdot -a_{\mathbf{1}}$.

Let I be a non degenerated integral domain-like ring and let u be an element of $\mathrm{Quot}(I)$. Let us assume that $u \neq 0_{\mathrm{q}}(I)$. The functor u_{q}^{-1} yields an element of $\mathrm{Quot}(I)$ and is defined by:

(Def. 11) For every element z of $\mathrm{Q}(I)$ holds $z \in u_{\mathrm{q}}^{-1}$ iff there exists an element a of $\mathrm{Q}(I)$ such that $a \in u$ and $z_{\mathbf{1}} \cdot a_{\mathbf{1}} = z_{\mathbf{2}} \cdot a_{\mathbf{2}}$.

The following propositions are true:

(13) Let I be a non degenerated integral domain-like ring and u, v, w be elements of $\mathrm{Quot}(I)$. Then $u +_{\mathrm{q}} (v +_{\mathrm{q}} w) = (u +_{\mathrm{q}} v) +_{\mathrm{q}} w$ and $u +_{\mathrm{q}} v = v +_{\mathrm{q}} u$.

(14) For every non degenerated integral domain-like ring I and for every element u of $\mathrm{Quot}(I)$ holds $u +_{\mathrm{q}} 0_{\mathrm{q}}(I) = u$ and $0_{\mathrm{q}}(I) +_{\mathrm{q}} u = u$.

(15) Let I be a non degenerated integral domain-like ring and u, v, w be elements of $\mathrm{Quot}(I)$. Then $u \cdot_{\mathrm{q}} (v \cdot_{\mathrm{q}} w) = (u \cdot_{\mathrm{q}} v) \cdot_{\mathrm{q}} w$ and $u \cdot_{\mathrm{q}} v = v \cdot_{\mathrm{q}} u$.

(16) For every non degenerated integral domain-like ring I and for every element u of $\mathrm{Quot}(I)$ holds $u \cdot_{\mathrm{q}} 1_{\mathrm{q}}(I) = u$ and $1_{\mathrm{q}}(I) \cdot_{\mathrm{q}} u = u$.

(17) For every non degenerated integral domain-like ring I and for all elements u, v, w of $\mathrm{Quot}(I)$ holds $(u +_{\mathrm{q}} v) \cdot_{\mathrm{q}} w = (u \cdot_{\mathrm{q}} w) +_{\mathrm{q}} (v \cdot_{\mathrm{q}} w)$.

(18) For every non degenerated integral domain-like ring I and for all elements u, v, w of $\mathrm{Quot}(I)$ holds $u \cdot_{\mathrm{q}} (v +_{\mathrm{q}} w) = (u \cdot_{\mathrm{q}} v) +_{\mathrm{q}} (u \cdot_{\mathrm{q}} w)$.

(19) For every non degenerated integral domain-like ring I and for every element u of $\mathrm{Quot}(I)$ holds $u +_{\mathrm{q}} -_{\mathrm{q}}u = 0_{\mathrm{q}}(I)$ and $-_{\mathrm{q}}u +_{\mathrm{q}} u = 0_{\mathrm{q}}(I)$.

(20) Let I be a non degenerated integral domain-like ring and u be an element of $\mathrm{Quot}(I)$. If $u \neq 0_{\mathrm{q}}(I)$, then $u \cdot_{\mathrm{q}} u_{\mathrm{q}}^{-1} = 1_{\mathrm{q}}(I)$ and $u_{\mathrm{q}}^{-1} \cdot_{\mathrm{q}} u = 1_{\mathrm{q}}(I)$.

(21) For every non degenerated integral domain-like ring I holds $1_{\mathrm{q}}(I) \neq 0_{\mathrm{q}}(I)$.

Let I be a non degenerated integral domain-like ring. The functor $+_{\mathrm{q}}(I)$ yielding a binary operation on $\mathrm{Quot}(I)$ is defined as follows:

(Def. 12) For all elements u, v of $\mathrm{Quot}(I)$ holds $(+_{\mathrm{q}}(I))(u, v) = u +_{\mathrm{q}} v$.

Let I be a non degenerated integral domain-like ring. The functor $\cdot_{\mathrm{q}}(I)$ yields a binary operation on $\mathrm{Quot}(I)$ and is defined as follows:

(Def. 13) For all elements u, v of $\mathrm{Quot}(I)$ holds $(\cdot_{\mathrm{q}}(I))(u, v) = u \cdot_{\mathrm{q}} v$.

Let I be a non degenerated integral domain-like ring. The functor $-_{\mathrm{q}}(I)$ yields a unary operation on $\mathrm{Quot}(I)$ and is defined as follows:

(Def. 14) For every element u of $\mathrm{Quot}(I)$ holds $(-_{\mathrm{q}}(I))(u) = -_{\mathrm{q}}u$.

Let I be a non degenerated integral domain-like ring. The functor ${}_{\mathrm{q}}^{-1}(I)$ yields a unary operation on $\mathrm{Quot}(I)$ and is defined as follows:

(Def. 15) For every element u of $\mathrm{Quot}(I)$ holds $({}_{\mathrm{q}}^{-1}(I))(u) = u_{\mathrm{q}}^{-1}$.

We now state a number of propositions:

(22) For every non degenerated integral domain-like ring I and for all elements u, v, w of $\mathrm{Quot}(I)$ holds $(+_{\mathrm{q}}(I))((+_{\mathrm{q}}(I))(u, v), w) = (+_{\mathrm{q}}(I))(u, (+_{\mathrm{q}}(I))(v, w))$.

(23) For every non degenerated integral domain-like ring I and for all elements u, v of $\mathrm{Quot}(I)$ holds $(+_{\mathrm{q}}(I))(u, v) = (+_{\mathrm{q}}(I))(v, u)$.

(24) For every non degenerated integral domain-like ring I and for every element u of $\mathrm{Quot}(I)$ holds $(+_{\mathrm{q}}(I))(u, 0_{\mathrm{q}}(I)) = u$ and $(+_{\mathrm{q}}(I))(0_{\mathrm{q}}(I), u) = u$.

(25) For every non degenerated integral domain-like ring I and for all elements u, v, w of $\mathrm{Quot}(I)$ holds $(\cdot_{\mathrm{q}}(I))((\cdot_{\mathrm{q}}(I))(u, v), w) = (\cdot_{\mathrm{q}}(I))(u, (\cdot_{\mathrm{q}}(I))(v, w))$.

(26) For every non degenerated integral domain-like ring I and for all elements u, v of $\mathrm{Quot}(I)$ holds $(\cdot_{\mathrm{q}}(I))(u, v) = (\cdot_{\mathrm{q}}(I))(v, u)$.

(27) For every non degenerated integral domain-like ring I and for every element u of $\mathrm{Quot}(I)$ holds $(\cdot_{\mathrm{q}}(I))(u, 1_{\mathrm{q}}(I)) = u$ and $(\cdot_{\mathrm{q}}(I))(1_{\mathrm{q}}(I), u) = u$.

(28) Let I be a non degenerated integral domain-like ring and u, v, w be elements of $\mathrm{Quot}(I)$. Then $(\cdot_{\mathrm{q}}(I))((+_{\mathrm{q}}(I))(u, v), w) = (+_{\mathrm{q}}(I))((\cdot_{\mathrm{q}}(I))(u, w), (\cdot_{\mathrm{q}}(I))(v, w))$.

(29) Let I be a non degenerated integral domain-like ring and u, v, w be elements of $\mathrm{Quot}(I)$. Then $(\cdot_{\mathrm{q}}(I))(u, (+_{\mathrm{q}}(I))(v, w)) = (+_{\mathrm{q}}(I))((\cdot_{\mathrm{q}}(I))(u, v), (\cdot_{\mathrm{q}}(I))(u, w))$.

(30) Let I be a non degenerated integral domain-like ring and u be an element of $\mathrm{Quot}(I)$. Then $(+_{\mathrm{q}}(I))(u, (-_{\mathrm{q}}(I))(u)) = 0_{\mathrm{q}}(I)$ and $(+_{\mathrm{q}}(I))((-_{\mathrm{q}}(I))(u), u) = 0_{\mathrm{q}}(I)$.

(31) Let I be a non degenerated integral domain-like ring and u be an element of $\mathrm{Quot}(I)$. If $u \neq 0_{\mathrm{q}}(I)$, then $(\cdot_{\mathrm{q}}(I))(u, ({}_{\mathrm{q}}^{-1}(I))(u)) = 1_{\mathrm{q}}(I)$ and $(\cdot_{\mathrm{q}}(I))(({}_{\mathrm{q}}^{-1}(I))(u), u) = 1_{\mathrm{q}}(I)$.

3. Defining the Field of Quotients

Let I be a non degenerated integral domain-like ring. The field of quotients of I yields a strict double loop structure and is defined as follows:

(Def. 16) The field of quotients of $I = \langle \mathrm{Quot}(I), +_{\mathrm{q}}(I), \cdot_{\mathrm{q}}(I), 1_{\mathrm{q}}(I), 0_{\mathrm{q}}(I) \rangle$.

Let I be a non degenerated integral domain-like ring. Observe that the field of quotients of I is non empty.

The following propositions are true:

(32) Let I be a non degenerated integral domain-like ring. Then

(i) the carrier of the field of quotients of $I = \mathrm{Quot}(I)$,

(ii) the addition of the field of quotients of $I = +_{\mathrm{q}}(I)$,

(iii) the multiplication of the field of quotients of $I = \cdot_{\rm q}(I)$,
(iv) the zero of the field of quotients of $I = 0_{\rm q}(I)$, and
(v) the unity of the field of quotients of $I = 1_{\rm q}(I)$.

(33) Let I be a non degenerated integral domain-like ring and u, v be elements of the carrier of the field of quotients of I. Then $(+_{\rm q}(I))(u, v)$ is an element of the carrier of the field of quotients of I.

(34) Let I be a non degenerated integral domain-like ring and u be an element of the carrier of the field of quotients of I. Then $(-_{\rm q}(I))(u)$ is an element of the carrier of the field of quotients of I.

(35) Let I be a non degenerated integral domain-like ring and u, v be elements of the carrier of the field of quotients of I. Then $(\cdot_{\rm q}(I))(u, v)$ is an element of the carrier of the field of quotients of I.

(36) Let I be a non degenerated integral domain-like ring and u be an element of the carrier of the field of quotients of I. Then $({}_{\rm q}^{-1}(I))(u)$ is an element of the carrier of the field of quotients of I.

(37) Let I be a non degenerated integral domain-like ring and u, v be elements of the carrier of the field of quotients of I. Then $u + v = (+_{\rm q}(I))(u, v)$.

Let I be a non degenerated integral domain-like ring. One can verify that the field of quotients of I is add-associative right zeroed and right complementable.

Next we state a number of propositions:

(38) Let I be a non degenerated integral domain-like ring and u be an element of the carrier of the field of quotients of I. Then $-u = (-_{\rm q}(I))(u)$.

(39) Let I be a non degenerated integral domain-like ring and u, v be elements of the carrier of the field of quotients of I. Then $u \cdot v = (\cdot_{\rm q}(I))(u, v)$.

(40) Let I be a non degenerated integral domain-like ring. Then $1_{\text{the field of quotients of } I} = 1_{\rm q}(I)$ and $0_{\text{the field of quotients of } I} = 0_{\rm q}(I)$.

(41) Let I be a non degenerated integral domain-like ring and u, v, w be elements of the carrier of the field of quotients of I. Then $(u + v) + w = u + (v + w)$.

(42) Let I be a non degenerated integral domain-like ring and u, v be elements of the carrier of the field of quotients of I. Then $u + v = v + u$.

(43) Let I be a non degenerated integral domain-like ring and u be an element of the carrier of the field of quotients of I. Then $u + 0_{\text{the field of quotients of } I} = u$.

(44) Let I be a non degenerated integral domain-like ring and u be an element of the carrier of the field of quotients of I. Then $u + -u = 0_{\text{the field of quotients of } I}$.

(45) Let I be a non degenerated integral domain-like ring and u be an element of the carrier of the field of quotients of I. Then $1_{\text{the field of quotients of } I} \cdot u = u$.

(46) Let I be a non degenerated integral domain-like ring and u, v be elements of the carrier of the field of quotients of I. Then $u \cdot v = v \cdot u$.

(47) Let I be a non degenerated integral domain-like ring and u, v, w be elements of the carrier of the field of quotients of I. Then $(u \cdot v) \cdot w = u \cdot (v \cdot w)$.

(48) Let I be a non degenerated integral domain-like ring and u be an element of the carrier of the field of quotients of I. Suppose $u \neq 0_{\text{the field of quotients of } I}$. Then there exists an element v of the carrier of the field of quotients of I such that $u \cdot v = 1_{\text{the field of quotients of } I}$.

(49) Let I be a non degenerated integral domain-like ring. Then the field of quotients of I is an add-associative right zeroed right complementable Abelian commutative associative left unital distributive field-like non degenerated non empty double loop structure.

Let I be a non degenerated integral domain-like ring. Note that the field of quotients of I is Abelian commutative associative left unital distributive field-like and non degenerated.

Next we state the proposition

(50) Let I be a non degenerated integral domain-like ring and x be an element of the carrier of the field of quotients of I. Suppose $x \neq 0_{\text{the field of quotients of } I}$. Let a be an element of the carrier of I. Suppose $a \neq 0_I$. Let u be an element of $\mathrm{Q}(I)$. Suppose $x = \mathrm{QClass}(u)$ and $u = \langle a, 1_I \rangle$. Let v be an element of $\mathrm{Q}(I)$. If $v = \langle 1_I, a \rangle$, then $x^{-1} = \mathrm{QClass}(v)$.

Let us observe that every add-associative right zeroed right complementable commutative associative left unital distributive field-like non degenerated non empty double loop structure is integral domain-like and right unital.

One can check that there exists a non empty double loop structure which is add-associative, right zeroed, right complementable, Abelian, commutative, associative, left unital, distributive, field-like, and non degenerated.

Let F be a commutative associative left unital distributive field-like non empty double loop structure and let x, y be elements of the carrier of F. The functor $\frac{x}{y}$ yields an element of the carrier of F and is defined as follows:

(Def. 17) $\frac{x}{y} = x \cdot y^{-1}$.

One can prove the following propositions:

(51) Let F be a non degenerated field-like ring and a, b, c, d be elements of the carrier of F. If $b \neq 0_F$ and $d \neq 0_F$, then $\frac{a}{b} \cdot \frac{c}{d} = \frac{a \cdot c}{b \cdot d}$.

(52) Let F be a non degenerated field-like ring and a, b, c, d be elements of the carrier of F. If $b \neq 0_F$ and $d \neq 0_F$, then $\frac{a}{b} + \frac{c}{d} = \frac{a \cdot d + c \cdot b}{b \cdot d}$.

4. Defining Ring Homomorphisms

Let R, S be non empty double loop structures and let f be a map from R into S. We say that f is a ring homomorphism if and only if:

(Def. 21)[1] f is additive, multiplicative, and unity-preserving.

Let R, S be non empty double loop structures. One can verify that every map from R into S which is ring homomorphism is also additive, multiplicative, and unity-preserving and every map from R into S which is additive, multiplicative, and unity-preserving is also a ring homomorphism.

Let R, S be non empty double loop structures and let f be a map from R into S. We say that f is a ring epimorphism if and only if:

(Def. 22) f is a ring homomorphism and $\operatorname{rng} f =$ the carrier of S.

We say that f is a ring monomorphism if and only if:

(Def. 23) f is a ring homomorphism and one-to-one.

We introduce f is an embedding as a synonym of f is a ring monomorphism.

Let R, S be non empty double loop structures and let f be a map from R into S. We say that f is a ring isomorphism if and only if:

(Def. 24) f is a ring monomorphism and a ring epimorphism.

Let R, S be non empty double loop structures. Note that every map from R into S which is ring isomorphism is also a ring monomorphism and a ring epimorphism and every map from R into S which is ring monomorphism and ring epimorphism is also a ring isomorphism.

We now state several propositions:

(53) For all rings R, S and for every map f from R into S such that f is a ring homomorphism holds $f(0_R) = 0_S$.

(54) Let R, S be rings and f be a map from R into S. Suppose f is a ring monomorphism. Let x be an element of the carrier of R. Then $f(x) = 0_S$ if and only if $x = 0_R$.

(55) Let R, S be non degenerated field-like rings and f be a map from R into S. Suppose f is a ring homomorphism. Let x be an element of the carrier of R. If $x \neq 0_R$, then $f(x^{-1}) = f(x)^{-1}$.

(56) Let R, S be non degenerated field-like rings and f be a map from R into S. Suppose f is a ring homomorphism. Let x, y be elements of the carrier of R. If $y \neq 0_R$, then $f(x \cdot y^{-1}) = f(x) \cdot f(y)^{-1}$.

(57) Let R, S, T be rings and f be a map from R into S. Suppose f is a ring homomorphism. Let g be a map from S into T. If g is a ring homomorphism, then $g \cdot f$ is a ring homomorphism.

[1]The definitions (Def. 18)–(Def. 20) have been removed.

(58) For every non empty double loop structure R holds id_R is a ring homomorphism.

Let R, S be non empty double loop structures. We say that R is embedded in S if and only if:

(Def. 25) There exists a map from R into S which is a ring monomorphism.

Let R, S be non empty double loop structures. We say that R is ring isomorphic to S if and only if:

(Def. 26) There exists a map from R into S which is a ring isomorphism.

Let us note that the predicate R is ring isomorphic to S is symmetric.

5. Some Further Properties

Let I be a non empty zero structure and let x, y be elements of the carrier of I. Let us assume that $y \neq 0_I$. The functor $\mathrm{quotient}(x, y)$ yielding an element of $\mathrm{Q}(I)$ is defined as follows:

(Def. 27) $\mathrm{quotient}(x, y) = \langle x,\, y \rangle$.

Let I be a non degenerated integral domain-like ring. The canonical homomorphism of I into quotient field is a map from I into the field of quotients of I and is defined by the condition (Def. 28).

(Def. 28) Let x be an element of the carrier of I. Then (the canonical homomorphism of I into quotient field)$(x) = \mathrm{QClass}(\mathrm{quotient}(x, 1_I))$.

Next we state four propositions:

(59) Let I be a non degenerated integral domain-like ring. Then the canonical homomorphism of I into quotient field is a ring homomorphism.

(60) Let I be a non degenerated integral domain-like ring. Then the canonical homomorphism of I into quotient field is an embedding.

(61) For every non degenerated integral domain-like ring I holds I is embedded in the field of quotients of I.

(62) Let F be a non degenerated field-like integral domain-like ring. Then F is ring isomorphic to the field of quotients of F.

Let I be a non degenerated integral domain-like ring. Note that the field of quotients of I is integral domain-like right unital and right-distributive.

One can prove the following proposition

(63) Let I be a non degenerated integral domain-like ring. Then the field of quotients of the field of quotients of I is ring isomorphic to the field of quotients of I.

Let I be a non empty double loop structure, let F be a non empty double loop structure, and let f be a map from I into F. We say that F is a field of quotients for I via f if and only if the conditions (Def. 29) are satisfied.

(Def. 29)(i) f is a ring monomorphism, and

(ii) for every add-associative right zeroed right complementable Abelian commutative associative left unital distributive field-like non degenerated non empty double loop structure F' and for every map f' from I into F' such that f' is a ring monomorphism there exists a map h from F into F' such that h is a ring homomorphism and $h \cdot f = f'$ and for every map h' from F into F' such that h' is a ring homomorphism and $h' \cdot f = f'$ holds $h' = h$.

Next we state two propositions:

(64) Let I be a non degenerated integral domain-like ring. Then there exists an add-associative right zeroed right complementable Abelian commutative associative left unital distributive field-like non degenerated non empty double loop structure F and there exists a map f from I into F such that F is a field of quotients for I via f.

(65) Let I be an integral domain-like ring, F, F' be add-associative right zeroed right complementable Abelian commutative associative left unital distributive field-like non degenerated non empty double loop structures, f be a map from I into F, and f' be a map from I into F'. Suppose F is a field of quotients for I via f and F' is a field of quotients for I via f'. Then F is ring isomorphic to F'.

References

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

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

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

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

[7] Michał Muzalewski. Categories of groups. *Formalized Mathematics*, 2(**4**):563–571, 1991.

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

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

[10] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.

[11] Wojciech A. Trybulec. Vectors in real linear space. *Formalized Mathematics*, 1(**2**):291–296, 1990.

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

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

Received May 4, 1998

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

First-countable, Sequential, and Frechet Spaces

Bartłomiej Skorulski
University of Białystok

Summary. This article contains a definition of three classes of topological spaces: first-countable, Frechet, and sequential. Next there are some facts about them, that every first-countable space is Frechet and every Frechet space is sequential. Next section constains a formalized construction of topological space which is Frechet but not first-countable. This article is based on [9, pp. 73–81].

MML Identifier: `FRECHET`.

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

1. Preliminaries

One can prove the following proposition

(1) For every non empty 1-sorted structure T and for every sequence S of T holds $\operatorname{rng} S$ is a subset of T.

Let T be a non empty 1-sorted structure and let S be a sequence of T. Then $\operatorname{rng} S$ is a subset of T.

The following propositions are true:

(2) Let T_1 be a non empty 1-sorted structure, T_2 be a 1-sorted structure, and S be a sequence of T_1. If $\operatorname{rng} S \subseteq$ the carrier of T_2, then S is a sequence of T_2.

(3) For every non empty topological space T and for every point x of T and for every basis B of x holds $B \neq \emptyset$.

ISSN 1426–2630

Let T be a non empty topological space and let x be a point of T. Note that every basis of x is non empty.

We now state a number of propositions:

(4) For every topological space T and for all subsets A, B of T such that A is open and B is closed holds $A \setminus B$ is open.

(5) Let T be a topological structure. Suppose that

(i) $\emptyset_T$ is closed,

(ii) Ω_T is closed,

(iii) for all subsets A, B of T such that A is closed and B is closed holds $A \cup B$ is closed, and

(iv) for every family F of subsets of T such that F is closed holds $\bigcap F$ is closed.

Then T is a topological space.

(6) Let T be a topological space, S be a non empty topological structure, and f be a map from T into S. Suppose that for every subset A of S holds A is closed iff $f^{-1}(A)$ is closed. Then S is a topological space.

(7) Let x be a point of the metric space of real numbers and x', r be real numbers. If $x' = x$ and $r > 0$, then $\mathrm{Ball}(x, r) =]x' - r, x' + r[$.

(8) Let A be a subset of $\mathbb{R}^{\mathbf{1}}$. Then A is open if and only if for every real number x such that $x \in A$ there exists a real number r such that $r > 0$ and $]x - r, x + r[\subseteq A$.

(9) For every sequence S of $\mathbb{R}^{\mathbf{1}}$ such that for every natural number n holds $S(n) \in]n - \frac{1}{4}, n + \frac{1}{4}[$ holds $\mathrm{rng}\, S$ is closed.

(10) For every subset B of $\mathbb{R}^{\mathbf{1}}$ such that $B = \mathbb{N}$ holds B is closed.

(11) Let M be a metric space, x be a point of M_{top}, and x' be a point of M. Suppose $x = x'$. Then there exists a basis B of x such that

(i) $B = \{\mathrm{Ball}(x', \frac{1}{n}); n$ ranges over natural numbers: $n \neq 0\}$,

(ii) B is countable, and

(iii) there exists a function f from $\mathbb{N}$ into B such that for every set n such that $n \in \mathbb{N}$ there exists a natural number n' such that $n = n'$ and $f(n) = \mathrm{Ball}(x', \frac{1}{n'+1})$.

(12) For all functions f, g holds $\mathrm{rng}(f{+\cdot}g) = f^{\circ}(\mathrm{dom}\, f \setminus \mathrm{dom}\, g) \cup \mathrm{rng}\, g$.

(13) For all sets A, B such that $B \subseteq A$ holds $(\mathrm{id}_A)^{\circ}B = B$.

(14) For all sets B, x holds $\mathrm{dom}(B \longmapsto x) = B$.

(15) For all sets A, B, x holds $\mathrm{dom}(\mathrm{id}_A{+\cdot}(B \longmapsto x)) = A \cup B$.

(16) For all sets A, B, x such that $B \neq \emptyset$ holds $\mathrm{rng}(\mathrm{id}_A{+\cdot}(B \longmapsto x)) = (A \setminus B) \cup \{x\}$.

(17) For all sets A, B, C, x such that $C \subseteq A$ holds $(\mathrm{id}_A{+\cdot}(B \longmapsto x))^{-1}(C \setminus \{x\}) = C \setminus B \setminus \{x\}$.

(18) For all sets A, B, x such that $x \notin A$ holds $(\mathrm{id}_A{+\cdot}(B \longmapsto x))^{-1}(\{x\}) = B$.

(19) For all sets A, B, C, x such that $C \subseteq A$ and $x \notin A$ holds $(\mathrm{id}_A +\cdot (B \longmapsto x))^{-1}(C \cup \{x\}) = C \cup B$.

(20) For all sets A, B, C, x such that $C \subseteq A$ and $x \notin A$ holds $(\mathrm{id}_A +\cdot (B \longmapsto x))^{-1}(C \setminus \{x\}) = C \setminus B$.

2. First-countable, Sequential, and Frechet Spaces

Let T be a non empty topological structure. We say that T is first-countable if and only if:

(Def. 1) For every point x of T holds there exists a basis of x which is countable.

The following two propositions are true:

(21) For every metric space M holds M_{top} is first-countable.

(22) $\mathbb{R}^{\mathbf{1}}$ is first-countable.

Let us note that $\mathbb{R}^{\mathbf{1}}$ is first-countable.

Let T be a topological structure, let S be a sequence of T, and let x be a point of T. We say that S is convergent to x if and only if the condition (Def. 2) is satisfied.

(Def. 2) Let U_1 be a subset of T. Suppose U_1 is open and $x \in U_1$. Then there exists a natural number n such that for every natural number m such that $n \leqslant m$ holds $S(m) \in U_1$.

The following proposition is true

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

Let T be a topological structure and let S be a sequence of T. We say that S is convergent if and only if:

(Def. 3) There exists a point x of T such that S is convergent to x.

Let T be a non empty topological structure and let S be a sequence of T. The functor $\mathrm{Lim}\, S$ yields a subset of T and is defined as follows:

(Def. 4) For every point x of T holds $x \in \mathrm{Lim}\, S$ iff S is convergent to x.

Let T be a non empty topological structure. We say that T is Frechet if and only if the condition (Def. 5) is satisfied.

(Def. 5) Let A be a subset of T and x be a point of T. If $x \in \overline{A}$, then there exists a sequence S of T such that $\mathrm{rng}\, S \subseteq A$ and $x \in \mathrm{Lim}\, S$.

Let T be a non empty topological structure. We say that T is sequential if and only if the condition (Def. 6) is satisfied.

(Def. 6) Let A be a subset of T. Then A is closed if and only if for every sequence S of T such that S is convergent and $\mathrm{rng}\, S \subseteq A$ holds $\mathrm{Lim}\, S \subseteq A$.

The following proposition is true

(24) For every non empty topological space T such that T is first-countable holds T is Frechet.

Let us observe that every non empty topological space which is first-countable is also Frechet.

We now state four propositions:

(25) $\mathbb{R}^{\mathbf{1}}$ is Frechet.

(26) Let T be a non empty topological space and A be a subset of T. Suppose A is closed. Let S be a sequence of T. If S is convergent and $\operatorname{rng} S \subseteq A$, then $\operatorname{Lim} S \subseteq A$.

(27) Let T be a non empty topological space. Suppose that for every subset A of T such that for every sequence S of T such that S is convergent and $\operatorname{rng} S \subseteq A$ holds $\operatorname{Lim} S \subseteq A$ holds A is closed. Then T is sequential.

(28) For every non empty topological space T such that T is Frechet holds T is sequential.

Let us mention that every non empty topological space which is Frechet is also sequential.

Next we state the proposition

(29) $\mathbb{R}^{\mathbf{1}}$ is sequential.

3. Counterexample of Frechet but Not First-countable Space

The strict non empty topological space $\mathbb{R}^1{}_{/\mathbb{N}}$ is defined by the conditions (Def. 7).

(Def. 7)(i) The carrier of $\mathbb{R}^1{}_{/\mathbb{N}} = (\mathbb{R} \setminus \mathbb{N}) \cup \{\mathbb{R}\}$, and

(ii) there exists a map f from $\mathbb{R}^{\mathbf{1}}$ into $\mathbb{R}^1{}_{/\mathbb{N}}$ such that $f = \mathrm{id}_{\mathbb{R}} + \cdot (\mathbb{N} \longmapsto \mathbb{R})$ and for every subset A of $\mathbb{R}^1{}_{/\mathbb{N}}$ holds A is closed iff $f^{-1}(A)$ is closed.

We now state several propositions:

(30) $\mathbb{R}$ is a point of $\mathbb{R}^1{}_{/\mathbb{N}}$.

(31) Let A be a subset of $\mathbb{R}^1{}_{/\mathbb{N}}$. Then A is open and $\mathbb{R} \in A$ if and only if there exists a subset O of $\mathbb{R}^{\mathbf{1}}$ such that O is open and $\mathbb{N} \subseteq O$ and $A = (O \setminus \mathbb{N}) \cup \{\mathbb{R}\}$.

(32) For every set A holds A is a subset of $\mathbb{R}^1{}_{/\mathbb{N}}$ and $\mathbb{R} \notin A$ iff A is a subset of $\mathbb{R}^{\mathbf{1}}$ and $\mathbb{N} \cap A = \emptyset$.

(33) Let A be a subset of $\mathbb{R}^{\mathbf{1}}$ and B be a subset of $\mathbb{R}^1{}_{/\mathbb{N}}$. If $A = B$, then $\mathbb{N} \cap A = \emptyset$ and A is open iff $\mathbb{R} \notin B$ and B is open.

(34) For every subset A of $\mathbb{R}^1{}_{/\mathbb{N}}$ such that $A = \{\mathbb{R}\}$ holds A is closed.

(35) $\mathbb{R}^1/_{\mathbb{N}}$ is not first-countable.

(36) $\mathbb{R}^1/_{\mathbb{N}}$ is Frechet.

(37) It is not true that for every non empty topological space T such that T is Frechet holds T is first-countable.

4. Auxiliary Theorems

Next we state three propositions:

(38) $\frac{1}{4} > 0$ and $\frac{1}{4} < \frac{1}{2}$.

(39) For every real number r there exists a natural number n such that $r < n$.

(40) For every real number r such that $r > 0$ there exists a natural number n such that $\frac{1}{n} < r$ and $n \neq 0$.

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

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

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

[9] Ryszard Engelking. *General Topology*, volume 60 of *Monografie Matematyczne*. PWN - Polish Scientific Publishers, Warsaw, 1977.

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

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

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

[13] Jan Popiołek. Real normed space. *Formalized Mathematics*, 2(**1**):111–115, 1991.

[14] Konrad Raczkowski and Paweł Sadowski. Topological properties of subsets in real numbers. *Formalized Mathematics*, 1(**4**):777–780, 1990.

[15] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.

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

[17] Andrzej Trybulec. Baire spaces, Sober spaces. *Formalized Mathematics*, 6(**2**):289–294, 1997.

[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.
[21] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received May 13, 1998

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

On the Composition of Non-parahalting Macro Instructions

Piotr Rudnicki[1]
University of Alberta
Edmonton

Summary. An attempt to use the `Times` macro, [2], was the origin of writing this article. First, the semantics of the macro composition as developed in [23, 3, 4] is extended to the case of macro instructions which are not always halting. Next, several functors extending the memory handling for $\mathbf{SCM}_{\mathrm{FSA}}$, [18], are defined; they are convenient when writing more complicated programs. After this preparatory work, we define a macro instruction computing the Fibonacci sequence (see the SCM program computing the same sequence in [10]) and prove its correctness. The semantics of the `Times` macro is given in [2] only for the case when the iterated instruction is parahalting; this is remedied in [17].

MML Identifier: `SFMASTR1`.

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

1. Good Instructions and Good Macro Instruction

Let i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. We say that i is good if and only if:

(Def. 1) Macro(i) is good.

Let a be a read-write integer location and let b be an integer location. One can check the following observations:

* a:=b is good,

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

ISSN 1426–2630

* AddTo(a, b) is good,
* SubFrom(a, b) is good, and
* MultBy(a, b) is good.

Let us note that there exists an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ which is good and parahalting.

Let a, b be read-write integer locations. Observe that Divide(a, b) is good.

Let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. One can verify that goto l is good.

Let a be an integer location and let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. Note that **if** $a = 0$ **goto** l is good and **if** $a > 0$ **goto** l is good.

Let a be an integer location, let f be a finite sequence location, and let b be a read-write integer location. One can check that $b{:=}f_a$ is good.

Let f be a finite sequence location and let b be a read-write integer location. One can verify that $b{:=}\mathrm{len} f$ is good.

Let f be a finite sequence location and let a be an integer location. One can check that $f{:=}\langle\underbrace{0, \ldots, 0}_{a}\rangle$ is good. Let b be an integer location. Note that $f_a{:=}b$ is good.

Let us note that there exists an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ which is good.

Let i be a good instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Note that Macro$(i)$ is good.

Let i, j be good instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Note that $i;j$ is good.

Let i be a good instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let I be a good macro instruction. Note that $i;I$ is good and $I;i$ is good.

Let a, b be read-write integer locations. Note that swap(a, b) is good.

Let I be a good macro instruction and let a be a read-write integer location. One can verify that Times(a, I) is good.

One can prove the following proposition

(1) For every integer location a and for every macro instruction I such that $a \notin \mathrm{UsedIntLoc}(I)$ holds I does not destroy a.

2. Composition of Non-parahalting Macro Instructions

For simplicity, we use the following convention: s, S denote states of $\mathbf{SCM}_{\mathrm{FSA}}$, I, J denote macro instructions, I_1 denotes a good macro instruction, i denotes a good parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, j denotes a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, a, b denote integer locations, and f denotes a finite sequence location.

We now state a number of propositions:

(2) $(I {+\cdot}\, \mathrm{Start\text{-}At}(\mathrm{insloc}(0))) {\restriction} D = \emptyset$, where $D = \mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations}$.

(3) If I is halting on Initialize(S) and closed on Initialize(S) and J is closed on IExec(I, S), then I;J is closed on Initialize(S).

(4) If I is halting on Initialize(S) and J is halting on IExec(I, S) and I is closed on Initialize(S) and J is closed on IExec(I, S), then I;J is halting on Initialize(S).

(5) Suppose I is closed on s and $I +\cdot$ Start-At(insloc(0)) $\subseteq s$ and s is halting. Let m be a natural number. Suppose $m \leqslant$ LifeSpan(s). Then (Computation(s))(m) and (Computation($s +\cdot (I;J)$))(m) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(6) Suppose I_1 is halting on Initialize(s) and J is halting on IExec(I_1, s) and I_1 is closed on Initialize(s) and J is closed on IExec(I_1, s). Then LifeSpan($s +\cdot$ Initialized(I_1;J)) $=$ LifeSpan($s +\cdot$ Initialized(I_1)) $+ 1 +$ LifeSpan(Result($s +\cdot$ Initialized(I_1))$+\cdot$ Initialized(J)).

(7) Suppose I_1 is halting on Initialize(s) and J is halting on IExec(I_1, s) and I_1 is closed on Initialize(s) and J is closed on IExec(I_1, s). Then IExec(I_1;J, s) $=$ IExec(J, IExec(I_1, s))$+\cdot$ Start-At($\mathbf{IC}_{\mathrm{IExec}(J,\mathrm{IExec}(I_1,s))} +$ card I_1).

(8) Suppose that

(i) I_1 is parahalting, or halting on Initialize(s), or closed on Initialize(s), and

(ii) J is parahalting, or halting on IExec(I_1, s), or closed on IExec(I_1, s).

Then (IExec(I_1;J, s))(a) $=$ (IExec(J, IExec(I_1, s)))(a).

(9) Suppose that

(i) I_1 is parahalting, or halting on Initialize(s), or closed on Initialize(s), and

(ii) J is parahalting, or halting on IExec(I_1, s), or closed on IExec(I_1, s).

Then (IExec(I_1;J, s))(f) $=$ (IExec(J, IExec(I_1, s)))(f).

(10) Suppose that

(i) I_1 is parahalting, or halting on Initialize(s), or closed on Initialize(s), and

(ii) J is parahalting, or halting on IExec(I_1, s), or closed on IExec(I_1, s).

Then IExec(I_1;J, s)$\restriction D$ $=$ IExec(J, IExec(I_1, s))$\restriction D$, where $D =$ Int-Locations $\cup$ FinSeq-Locations.

(11) If I_1 is parahalting, or closed on Initialize(s), or halting on Initialize(s), then Initialize(IExec(I_1, s))$\restriction D$ $=$ IExec(I_1, s)$\restriction D$, where $D =$ Int-Locations $\cup$ FinSeq-Locations.

(12) If I_1 is parahalting, or halting on Initialize(s), or closed on Initialize(s), then (IExec(I_1;j, s))(a) $=$ (Exec(j, IExec(I_1, s)))(a).

(13) If I_1 is parahalting, or halting on Initialize(s), or closed on Initialize(s), then (IExec(I_1;j, s))(f) $=$ (Exec(j, IExec(I_1, s)))(f).

(14) If I_1 is parahalting, or halting on Initialize(s), or closed on Initialize(s), then IExec(I_1;j, s)$\restriction D$ = Exec(j, IExec(I_1, s))$\restriction D$, where D = Int-Locations $\cup$ FinSeq-Locations.

(15) If J is parahalting, or halting on Exec(i, Initialize(s)), or closed on Exec(i, Initialize(s)), then (IExec(i;J, s))(a) = (IExec(J, Exec(i, Initialize(s))))(a).

(16) If J is parahalting, or halting on Exec(i, Initialize(s)), or closed on Exec(i, Initialize(s)), then (IExec(i;J, s))(f) = (IExec(J, Exec(i, Initialize(s))))(f).

(17) If J is parahalting, or halting on Exec(i, Initialize(s)), or closed on Exec(i, Initialize(s)), then IExec(i;J, s)$\restriction D$ = IExec(J, Exec(i, Initialize(s))) $\restriction D$, where D = Int-Locations $\cup$ FinSeq-Locations.

3. Memory Allocation

In the sequel L is a finite subset of Int-Locations and m, n are natural numbers.

Let d be an integer location. Then $\{d\}$ is a subset of Int-Locations. Let e be an integer location. Then $\{d, e\}$ is a subset of Int-Locations. Let f be an integer location. Then $\{d, e, f\}$ is a subset of Int-Locations. Let g be an integer location. Then $\{d, e, f, g\}$ is a subset of Int-Locations.

Let L be a finite subset of Int-Locations. The functor RWNotIn-seq L yields a function from $\mathbb{N}$ into $2^{\mathbb{N}}$ and is defined by the conditions (Def. 2).

(Def. 2)(i) (RWNotIn-seq L)(0) = $\{k; k$ ranges over natural numbers: intloc(k) $\notin L \wedge k \neq 0\}$,

(ii) for every natural number i and for every non empty subset s_1 of $\mathbb{N}$ such that (RWNotIn-seq L)(i) = s_1 holds (RWNotIn-seq L)($i + 1$) = $s_1 \setminus \{\min s_1\}$, and

(iii) for every natural number i holds (RWNotIn-seq L)(i) is infinite.

Let L be a finite subset of Int-Locations and let n be a natural number. Note that (RWNotIn-seq L)(n) is non empty.

One can prove the following propositions:

(18) $0 \notin$ (RWNotIn-seq L)(n) and for every m such that $m \in$ (RWNotIn-seq L)(n) holds intloc(m) $\notin L$.

(19) min(RWNotIn-seq L)(n) < min(RWNotIn-seq L)($n + 1$).

(20) If $n < m$, then min(RWNotIn-seq L)(n) < min(RWNotIn-seq L)(m).

Let n be a natural number and let L be a finite subset of Int-Locations. The functor n^{th}-RWNotIn(L) yields an integer location and is defined as follows:

(Def. 3) n^{th}-RWNotIn(L) = intloc(min(RWNotIn-seq L)(n)).

We introduce 1^{st}-RWNotIn(L), 2^{nd}-RWNotIn(L), 3^{rd}-RWNotIn(L) as synonyms of n^{th}-RWNotIn(L).

Let n be a natural number and let L be a finite subset of Int-Locations. One can verify that n^{th}-RWNotIn(L) is read-write.

We now state two propositions:

(21) $n^{\mathrm{th}}\text{-RWNotIn}(L) \notin L$.

(22) If $n \neq m$, then $n^{\mathrm{th}}\text{-RWNotIn}(L) \neq m^{\mathrm{th}}\text{-RWNotIn}(L)$.

Let n be a natural number and let p be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. The functor n^{th}-NotUsed(p) yielding an integer location is defined by:

(Def. 4) $n^{\mathrm{th}}\text{-NotUsed}(p) = n^{\mathrm{th}}\text{-RWNotIn}(\mathrm{UsedIntLoc}(p))$.

We introduce 1^{st}-NotUsed(p), 2^{nd}-NotUsed(p), 3^{rd}-NotUsed(p) as synonyms of n^{th}-NotUsed(p).

Let n be a natural number and let p be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. Observe that n^{th}-NotUsed(p) is read-write.

4. A Macro for the Fibonacci Sequence

One can prove the following proposition

(23) $a \in \mathrm{UsedIntLoc}(\mathrm{swap}(a, b))$ and $b \in \mathrm{UsedIntLoc}(\mathrm{swap}(a, b))$.

Let N, r_1 be integer locations. The functor Fib_macro(N, r_1) yielding a macro instruction is defined by:

(Def. 5) $\mathrm{Fib\_macro}(N, r_1) =$
$(N_1{:=}N)$;
$\mathrm{SubFrom}(r_1, r_1)$;
$(n_1{:=}\,\mathrm{intloc}(0))$;
$(a_1{:=}N_1)$;
$\mathrm{Times}(a_1, \mathrm{AddTo}(r_1, n_1); \mathrm{swap}(r_1, n_1))$;
$(N{:=}N_1)$,
where $N_1 = 2^{\mathrm{nd}}\text{-RWNotIn}(\mathrm{UsedIntLoc}(\mathrm{swap}(r_1, n_1)))$, $n_1 = 1^{\mathrm{st}}\text{-RWNotIn}(\{N, r_1\})$, and $a_1 = 1^{\mathrm{st}}\text{-RWNotIn}(\mathrm{UsedIntLoc}(\mathrm{swap}(r_1, n_1)))$.

Next we state the proposition

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

References

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

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

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

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

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

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

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

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

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

[10] Grzegorz Bancerek and Piotr Rudnicki. Two programs for **scm**. Part II - programs. *Formalized Mathematics*, 4(**1**):73–75, 1993.

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

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

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

[14] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.

[15] Agata Darmochwał and Andrzej Trybulec. Similarity of formulae. *Formalized Mathematics*, 2(**5**):635–642, 1991.

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

[17] Piotr Rudnicki. Another `times` macro instruction. *Formalized Mathematics*, 7(**1**):101–105, 1998.

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

[19] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.

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

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

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

[23] Andrzej Trybulec, Yatsuka Nakamura, and Noriko Asamoto. On the compositions of macro instructions. Part I. *Formalized Mathematics*, 6(**1**):21–27, 1997.

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

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

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

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

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

Received June 3, 1998

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

The `while` Macro Instructions of $\mathbf{SCM_{FSA}}$. Part II

Piotr Rudnicki[1]
University of Alberta
Edmonton

Summary. An attempt to use the `while` macro, [14], was the origin of writing this article. The `while` semantics, as given by J.-C. Chen, is slightly extended by weakening its correctness conditions and this forced a quite straightforward remake of a number of theorems from [14]. Numerous additional properties of the `while` macro are then proven. In the last section, we define a macro instruction computing the `fusc` function (see the SCM program computing the same function in [10]) and prove its correctness.

MML Identifier: `SCMFSA9A`.

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

1. Arithmetic Preliminaries

We follow the rules: k, m, n are natural numbers, i, j are integers, and r is a real number.

The scheme *MinPred* deals with a unary functor $\mathcal{F}$ yielding a natural number and a unary predicate $\mathcal{P}$, and states that:

There exists k such that $\mathcal{P}[k]$ and for every n such that $\mathcal{P}[n]$ holds $k \leqslant n$

provided the parameters meet the following condition:

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

ISSN 1426–2630

- For every k holds $\mathcal{F}(k+1) < \mathcal{F}(k)$ or $\mathcal{P}[k]$.

We now state several propositions:

(1) n is odd iff there exists a natural number k such that $n = 2 \cdot k + 1$.

(2) If $0 \leqslant r$, then $0 \leqslant \lfloor r \rfloor$.

(3) If $0 < n$, then $0 \leqslant (m$ **qua** integer$)$ $\div n$.

(4) If $0 < i$ and $1 < j$, then $i \div j < i$.

(5) If $0 < n$, then $(m$ **qua** integer$)$ $\div n = m \div n$ and $(m$ **qua** integer$)$ $\bmod n = m \bmod n$.

2. $\mathbf{SCM}_{\rm FSA}$ Preliminaries

In the sequel l is an instruction-location of $\mathbf{SCM}_{\rm FSA}$ and i is an instruction of $\mathbf{SCM}_{\rm FSA}$.

Next we state several propositions:

(6) Let N be a non empty set with non empty elements, S be a halting von Neumann definite AMI over N, s be a state of S, and k be a natural number. If $\mathrm{CurInstr}((\mathrm{Computation}(s))(k)) = \mathbf{halt}_S$, then $(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s)) = (\mathrm{Computation}(s))(k)$.

(7) $\mathrm{UsedIntLoc}(l \dot{\longmapsto} i) = \mathrm{UsedIntLoc}(i)$.

(8) $\mathrm{UsedInt}^* \mathrm{Loc}(l \dot{\longmapsto} i) = \mathrm{UsedInt}^* \mathrm{Loc}(i)$.

(9) $\mathrm{UsedIntLoc}(\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}}) = \emptyset$.

(10) $\mathrm{UsedInt}^* \mathrm{Loc}(\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}}) = \emptyset$.

(11) $\mathrm{UsedIntLoc}(\mathrm{Goto}(l)) = \emptyset$.

(12) $\mathrm{UsedInt}^* \mathrm{Loc}(\mathrm{Goto}(l)) = \emptyset$.

For simplicity, we use the following convention: s, s_1, s_2 are states of $\mathbf{SCM}_{\rm FSA}$, a is a read-write integer location, b is an integer location, f is a finite sequence location, I, J are macro instructions, I_1 is a good macro instruction, and i, j, k are natural numbers.

The following four propositions are true:

(13) $\mathrm{UsedIntLoc}(\mathbf{if}\ b = 0\ \mathbf{then}\ I\ \mathbf{else}\ J) = \{b\} \cup \mathrm{UsedIntLoc}(I) \cup \mathrm{UsedIntLoc}(J)$.

(14) For every integer location a holds $\mathrm{UsedInt}^* \mathrm{Loc}(\mathbf{if}\ a = 0\ \mathbf{then}\ I\ \mathbf{else}\ J) = \mathrm{UsedInt}^* \mathrm{Loc}(I) \cup \mathrm{UsedInt}^* \mathrm{Loc}(J)$.

(15) $\mathrm{UsedIntLoc}(\mathbf{if}\ b > 0\ \mathbf{then}\ I\ \mathbf{else}\ J) = \{b\} \cup \mathrm{UsedIntLoc}(I) \cup \mathrm{UsedIntLoc}(J)$.

(16) $\mathrm{UsedInt}^* \mathrm{Loc}(\mathbf{if}\ b > 0\ \mathbf{then}\ I\ \mathbf{else}\ J) = \mathrm{UsedInt}^* \mathrm{Loc}(I) \cup \mathrm{UsedInt}^* \mathrm{Loc}(J)$.

3. The `while=0` Macro Instruction

Next we state two propositions:

(17) UsedIntLoc(**while** $b = 0$ **do** I) $= \{b\} \cup$ UsedIntLoc(I).

(18) UsedInt* Loc(**while** $b = 0$ **do** I) $=$ UsedInt* Loc(I).

Let s be a state of $\mathbf{SCM}_{\rm FSA}$, let a be a read-write integer location, and let I be a macro instruction. The predicate ProperBodyWhile=0(a, I, s) is defined as follows:

(Def. 1) For every natural number k such that $(\mathit{StepWhile{=}0}(a, I, s))(k)(a) = 0$ holds I is closed on $(\mathit{StepWhile{=}0}(a, I, s))(k)$ and halting on $(\mathit{StepWhile{=}0}(a, I, s))(k)$.

The predicate WithVariantWhile=0(a, I, s) is defined by the condition (Def. 2).

(Def. 2) There exists a function f from $\prod$(the object kind of $\mathbf{SCM}_{\rm FSA}$) into $\mathbb{N}$ such that for every natural number k holds $f((\mathit{StepWhile{=}0}(a, I, s))(k + 1)) < f((\mathit{StepWhile{=}0}(a, I, s))(k))$ or $(\mathit{StepWhile{=}0}(a, I, s))(k)(a) \neq 0$.

We now state several propositions:

(19) For every parahalting macro instruction I holds ProperBodyWhile=0(a, I, s).

(20) If ProperBodyWhile=0(a, I, s) and WithVariantWhile=0(a, I, s), then **while** $a = 0$ **do** I is halting on s and **while** $a = 0$ **do** I is closed on s.

(21) For every parahalting macro instruction I such that WithVariantWhile=0(a, I, s) holds **while** $a = 0$ **do** I is halting on s and **while** $a = 0$ **do** I is closed on s.

(22) If (**while** $a = 0$ **do** I)$+\cdot S_1 \subseteq s$ and $s(a) \neq 0$, then LifeSpan$(s) = 4$ and for every natural number k holds (Computation(s))$(k){\restriction}D = s{\restriction}D$, where $S_1 =$ Start-At(insloc(0)) and $D =$ Int-Locations $\cup$ FinSeq-Locations.

(23) If I is closed on s and halting on s and $s(a) = 0$, then (Computation$(s{+\cdot}((\textbf{while}\ a = 0\ \textbf{do}\ I){+\cdot}S_1))$)(LifeSpan$(s{+\cdot}(I{+\cdot}S_1)) + 3){\restriction}D =$ (Computation$(s{+\cdot}(I{+\cdot}S_1))$)(LifeSpan$(s{+\cdot}(I{+\cdot}S_1))$)${\restriction}D$, where $S_1 =$ Start-At(insloc(0)) and $D =$ Int-Locations $\cup$ FinSeq-Locations.

(24) If $(\mathit{StepWhile{=}0}(a, I, s))(k)(a) \neq 0$, then $(\mathit{StepWhile{=}0}(a, I, s))(k + 1){\restriction}D = (\mathit{StepWhile{=}0}(a, I, s))(k){\restriction}D$, where $D =$ Int-Locations $\cup$ FinSeq-Locations.

(25) Suppose I is halting on Initialize$((\mathit{StepWhile{=}0}(a, I, s))(k))$, closed on Initialize$((\mathit{StepWhile{=}0}(a, I, s))(k))$, and parahalting and $(\mathit{StepWhile{=}0}(a, I, s))(k)(a) = 0$ and $(\mathit{StepWhile{=}0}(a, I, s))(k)(\mathrm{intloc}(0)) = 1$. Then $(\mathit{StepWhile{=}0}(a, I, s))(k{+}1){\restriction}D =$ IExec$(I, (\mathit{StepWhile{=}0}(a, I, s))(k)){\restriction}D$, where $D =$ Int-Locations $\cup$ FinSeq-Locations.

(26) If ProperBodyWhile=0(a, I_1, s) or I_1 is parahalting and if $s(\mathrm{intloc}(0)) = 1$, then for every k holds $(\mathit{StepWhile{=}0}(a, I_1, s))(k)(\mathrm{intloc}(0)) = 1$.

(27) If ProperBodyWhile=0(a, I, s_1) and $s_1 \restriction D = s_2 \restriction D$, then for every k holds $(\mathit{StepWhile{=}0}(a, I, s_1))(k) \restriction D = (\mathit{StepWhile{=}0}(a, I, s_2))(k) \restriction D$, where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, let a be a read-write integer location, and let I be a macro instruction. Let us assume that ProperBodyWhile=0(a, I, s) or I is parahalting and WithVariantWhile=0(a, I, s). The functor $\mathit{ExitsAtWhile{=}0}(a, I, s)$ yielding a natural number is defined by the condition (Def. 3).

(Def. 3) There exists a natural number k such that

(i) $\mathit{ExitsAtWhile{=}0}(a, I, s) = k$,

(ii) $(\mathit{StepWhile{=}0}(a, I, s))(k)(a) \neq 0$,

(iii) for every natural number i such that $(\mathit{StepWhile{=}0}(a, I, s))(i)(a) \neq 0$ holds $k \leqslant i$, and

(iv) $(\mathrm{Computation}(s + \cdot ((\textbf{while } a = 0 \textbf{ do } I) + \cdot S_1)))(\mathrm{LifeSpan}(s + \cdot ((\textbf{while } a = 0 \textbf{ do } I) + \cdot S_1))) \restriction D = (\mathit{StepWhile{=}0}(a, I, s))(k) \restriction D$,

where $S_1 = \text{Start-At}(\mathrm{insloc}(0))$ and $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

One can prove the following two propositions:

(28) If $s(\mathrm{intloc}(0)) = 1$ and $s(a) \neq 0$, then $\mathrm{IExec}(\textbf{while } a = 0 \textbf{ do } I, s) \restriction D = s \restriction D$, where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

(29) If ProperBodyWhile=0$(a, I, \mathrm{Initialize}(s))$ or I is parahalting and if WithVariantWhile=0$(a, I, \mathrm{Initialize}(s))$, then $\mathrm{IExec}(\textbf{while } a = 0 \textbf{ do } I, s) \restriction D = (\mathit{StepWhile{=}0}(a, I, \mathrm{Initialize}(s)))(\mathit{ExitsAtWhile{=}0}(a, I, \mathrm{Initialize}(s))) \restriction D$, where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

4. The `while>0` Macro Instruction

The following propositions are true:

(30) $\mathrm{UsedIntLoc}(\textbf{while } b > 0 \textbf{ do } I) = \{b\} \cup \mathrm{UsedIntLoc}(I)$.

(31) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(\textbf{while } b > 0 \textbf{ do } I) = \mathrm{UsedInt}^{*}\,\mathrm{Loc}(I)$.

Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, let a be a read-write integer location, and let I be a macro instruction. The predicate ProperBodyWhile>0(a, I, s) is defined as follows:

(Def. 4) For every natural number k such that $(\mathit{StepWhile{>}0}(a, I, s))(k)(a) > 0$ holds I is closed on $(\mathit{StepWhile{>}0}(a, I, s))(k)$ and halting on $(\mathit{StepWhile{>}0}(a, I, s))(k)$.

The predicate WithVariantWhile>0(a, I, s) is defined by the condition (Def. 5).

(Def. 5) There exists a function f from $\prod$(the object kind of $\mathbf{SCM}_{\mathrm{FSA}}$) into $\mathbb{N}$ such that for every natural number k holds $f((\mathit{StepWhile{>}0}(a, I, s))(k + 1)) < f((\mathit{StepWhile{>}0}(a, I, s))(k))$ or $(\mathit{StepWhile{>}0}(a, I, s))(k)(a) \leqslant 0$.

Next we state several propositions:

(32) For every parahalting macro instruction I holds ProperBodyWhile>0(a, I, s).

(33) If ProperBodyWhile>0(a, I, s) and WithVariantWhile>0(a, I, s), then **while** $a > 0$ **do** I is halting on s and **while** $a > 0$ **do** I is closed on s.

(34) For every parahalting macro instruction I such that WithVariantWhile>0(a, I, s) holds **while** $a > 0$ **do** I is halting on s and **while** $a > 0$ **do** I is closed on s.

(35) If (**while** $a > 0$ **do** I)$+\cdot S_1 \subseteq s$ and $s(a) \leqslant 0$, then LifeSpan$(s) = 4$ and for every natural number k holds (Computation$(s))(k){\restriction}D = s{\restriction}D$, where $S_1 =$ Start-At(insloc(0)) and $D =$ Int-Locations $\cup$ FinSeq-Locations.

(36) If I is closed on s and halting on s and $s(a) > 0$, then (Computation($s+\cdot$((**while** $a > 0$ **do** I)$+\cdot S_1$)))(LifeSpan($s+\cdot(I+\cdot S_1)$) $+ 3){\restriction}D =$ (Computation($s+\cdot(I+\cdot S_1)$))(LifeSpan($s+\cdot(I+\cdot S_1)$))$\restriction D$, where $S_1 =$ Start-At(insloc(0)) and $D =$ Int-Locations $\cup$ FinSeq-Locations.

(37) If $(\mathit{StepWhile{>}0}(a, I, s))(k)(a) \leqslant 0$, then $(\mathit{StepWhile{>}0}(a, I, s))(k + 1){\restriction}D = (\mathit{StepWhile{>}0}(a, I, s))(k){\restriction}D$, where $D =$ Int-Locations $\cup$ FinSeq-Locations.

(38) Suppose I is halting on Initialize$((\mathit{StepWhile{>}0}(a, I, s))(k))$, closed on Initialize$((\mathit{StepWhile{>}0}(a, I, s))(k))$, and parahalting and $(\mathit{StepWhile{>}0}(a, I, s))(k)(a) > 0$ and $(\mathit{StepWhile{>}0}(a, I, s))(k)(\mathrm{intloc}(0)) = 1$. Then $(\mathit{StepWhile{>}0}(a, I, s))(k + 1){\restriction}D = \mathrm{IExec}(I, (\mathit{StepWhile{>}0}(a, I, s))(k)){\restriction}D$, where $D =$ Int-Locations $\cup$ FinSeq-Locations.

(39) If ProperBodyWhile>0(a, I_1, s) or I_1 is parahalting and if $s(\mathrm{intloc}(0)) = 1$, then for every k holds $(\mathit{StepWhile{>}0}(a, I_1, s))(k)(\mathrm{intloc}(0)) = 1$.

(40) If ProperBodyWhile>0(a, I, s_1) and $s_1{\restriction}D = s_2{\restriction}D$, then for every k holds $(\mathit{StepWhile{>}0}(a, I, s_1))(k){\restriction}D = (\mathit{StepWhile{>}0}(a, I, s_2))(k){\restriction}D$, where $D =$ Int-Locations $\cup$ FinSeq-Locations.

Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, let a be a read-write integer location, and let I be a macro instruction. Let us assume that ProperBodyWhile>0(a, I, s) or I is parahalting and WithVariantWhile>0(a, I, s).

The functor $\mathit{ExitsAtWhile{>}0}(a, I, s)$ yields a natural number and is defined by the condition (Def. 6).

(Def. 6) There exists a natural number k such that

(i) $\mathit{ExitsAtWhile{>}0}(a, I, s) = k$,

(ii) $(\mathit{StepWhile{>}0}(a, I, s))(k)(a) \leqslant 0$,

(iii) for every natural number i such that $(\mathit{StepWhile{>}0}(a, I, s))(i)(a) \leqslant 0$ holds $k \leqslant i$, and

(iv) $(\mathrm{Computation}(s{+}{\cdot}((\mathbf{while}\ a > 0\ \mathbf{do}\ I){+}{\cdot}S_1)))(\mathrm{LifeSpan}(s{+}{\cdot}((\mathbf{while}\ a > 0\ \mathbf{do}\ I){+}{\cdot}S_1)))\restriction D = (\mathit{StepWhile{>}0}(a, I, s))(k)\restriction D$,

where $S_1 = \text{Start-At}(\mathrm{insloc}(0))$ and $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

Next we state several propositions:

(41) If $s(\mathrm{intloc}(0)) = 1$ and $s(a) \leqslant 0$, then $\mathrm{IExec}(\mathbf{while}\ a > 0\ \mathbf{do}\ I, s)\restriction D = s\restriction D$, where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

(42) If ProperBodyWhile>0$(a, I, \mathrm{Initialize}(s))$ or I is parahalting and if WithVariantWhile>0$(a, I, \mathrm{Initialize}(s))$, then $\mathrm{IExec}(\mathbf{while}\ a > 0\ \mathbf{do}\ I, s)\restriction D = (\mathit{StepWhile{>}0}(a, I, \mathrm{Initialize}(s)))(\mathit{ExitsAtWhile{>}0}(a, I, \mathrm{Initialize}(s)))\restriction D$,

where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

(43) If $(\mathit{StepWhile{>}0}(a, I, s))(k)(a) \leqslant 0$, then for every natural number n such that $k \leqslant n$ holds $(\mathit{StepWhile{>}0}(a, I, s))(n)\restriction D = (\mathit{StepWhile{>}0}(a, I, s))(k)\restriction D$, where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

(44) If $s_1\restriction D = s_2\restriction D$ and ProperBodyWhile>0(a, I, s_1), then

ProperBodyWhile>0(a, I, s_2), where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

(45) Suppose $s(\mathrm{intloc}(0)) = 1$ and ProperBodyWhile>0(a, I_1, s) and WithVariantWhile>0(a, I_1, s). Let given i, j. Suppose $i \neq j$ and $i \leqslant \mathit{ExitsAtWhile{>}0}(a, I_1, s)$ and $j \leqslant \mathit{ExitsAtWhile{>}0}(a, I_1, s)$. Then $(\mathit{StepWhile{>}0}(a, I_1, s))(i) \neq (\mathit{StepWhile{>}0}(a, I_1, s))(j)$ and $(\mathit{StepWhile{>}0}(a, I_1, s))(i)\restriction D \neq (\mathit{StepWhile{>}0}(a, I_1, s))(j)\restriction D$, where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

Let f be a function from $\prod$(the object kind of $\mathbf{SCM}_{\mathrm{FSA}}$) into $\mathbb{N}$. We say that f is on data only if and only if:

(Def. 7) For all s_1, s_2 such that $s_1\restriction D = s_2\restriction D$ holds $f(s_1) = f(s_2)$, where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

We now state two propositions:

(46) Suppose $s(\mathrm{intloc}(0)) = 1$ and ProperBodyWhile>0(a, I_1, s) and WithVariantWhile>0(a, I_1, s). Then there exists a function f from $\prod$(the object kind of $\mathbf{SCM}_{\mathrm{FSA}}$) into $\mathbb{N}$ such that f is on data only and for every natural number k holds $f((\mathit{StepWhile{>}0}(a, I_1, s))(k+1)) < f((\mathit{StepWhile{>}0}(a, I_1, s))(k))$ or $(\mathit{StepWhile{>}0}(a, I_1, s))(k)(a) \leqslant 0$.

(47) If $s_1(\mathrm{intloc}(0)) = 1$ and $s_1\restriction D = s_2\restriction D$ and ProperBodyWhile>0(a, I_1, s_1) and WithVariantWhile>0(a, I_1, s_1), then WithVariantWhile>0(a, I_1, s_2), where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

5. A Macro for the fusc Function

Let N, r_1 be integer locations. The functor Fusc_macro(N, r_1) yields a macro instruction and is defined as follows:

(Def. 8) Fusc_macro$(N, r_1) =$
SubFrom(r_1, r_1);
$(n_1{:=}\,\mathrm{intloc}(0))$;
$(a_1{:=}N)$;
(**while** $a_1 > 0$ **do**
$((r_2{:=}2)$;
Divide(a_1, r_2);
(**if** $r_2 = 0$ **then**
Macro(AddTo(n_1, r_1)) **else**
Macro(AddTo(r_1, n_1)))))),
where $n_1 = 1^{\mathrm{st}}$-RWNotIn$(\{N, r_1\})$, $a_1 = 2^{\mathrm{nd}}$-RWNotIn$(\{N, r_1\})$, and $r_2 = 3^{\mathrm{rd}}$-RWNotIn$(\{N, r_1\})$.

One can prove the following proposition

(48) Let N, r_1 be read-write integer locations. Suppose $N \neq r_1$. Let n be a natural number. If $n = s(N)$, then (IExec(Fusc_macro$(N, r_1), s$))$(r_1) =$ Fusc(n) and (IExec(Fusc_macro$(N, r_1), s$))$(N) = n$.

References

[1] Noriko Asamoto. Conditional branch macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Part I. *Formalized Mathematics*, 6(**1**):65–72, 1997.
[2] Noriko Asamoto. Conditional branch macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Part II. *Formalized Mathematics*, 6(**1**):73–80, 1997.
[3] Noriko Asamoto. Constant assignment macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Part II. *Formalized Mathematics*, 6(**1**):59–63, 1997.
[4] Noriko Asamoto, Yatsuka Nakamura, Piotr Rudnicki, and Andrzej Trybulec. On the composition of macro instructions. Part II. *Formalized Mathematics*, 6(**1**):41–47, 1997.
[5] Noriko Asamoto, Yatsuka Nakamura, Piotr Rudnicki, and Andrzej Trybulec. On the composition of macro instructions. Part III. *Formalized Mathematics*, 6(**1**):53–57, 1997.
[6] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[7] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[8] Grzegorz Bancerek and Piotr Rudnicki. Development of terminology for **scm**. *Formalized Mathematics*, 4(**1**):61–67, 1993.
[9] Grzegorz Bancerek and Piotr Rudnicki. Two programs for **scm**. Part I - preliminaries. *Formalized Mathematics*, 4(**1**):69–72, 1993.
[10] Grzegorz Bancerek and Piotr Rudnicki. Two programs for **scm**. Part II - programs. *Formalized Mathematics*, 4(**1**):73–75, 1993.
[11] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.
[12] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[13] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.

[14] Jing-Chao Chen. While macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 6(**4**):553–561, 1997.
[15] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[16] Piotr Rudnicki. On the composition of non-parahalting macro instructions. *Formalized Mathematics*, 7(**1**):87–92, 1998.
[17] Piotr Rudnicki and Andrzej Trybulec. Abian's fixed point theorem. *Formalized Mathematics*, 6(**3**):335–338, 1997.
[18] Piotr Rudnicki and Andrzej Trybulec. Memory handling for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 6(**1**):29–36, 1997.
[19] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[20] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[21] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[22] Andrzej Trybulec and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):571–576, 1996.
[23] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\mathrm{FSA}}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[24] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[25] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[26] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[27] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[28] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.
[29] Wojciech Zielonka. Preliminaries to the Lambek calculus. *Formalized Mathematics*, 2(**3**):413–418, 1991.

Received June 3, 1998

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

Another times Macro Instruction

Piotr Rudnicki[1]
University of Alberta
Edmonton

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

MML Identifier: `SFMASTR2`.

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

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

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

One can prove the following propositions:

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

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

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

ISSN 1426–2630

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

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

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

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

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

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

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

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

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

2. Another `times` Macro Instruction

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

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

We introduce a times I as a synonym of times(a, I).

Next we state two propositions:

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

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

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

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

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

Next we state several propositions:

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

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

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

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

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

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

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

One can prove the following propositions:

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

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

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

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

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

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

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

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

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

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

3. A Trivial Example

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

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

One can prove the following propositions:

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

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

4. A Macro for the Fibonacci Sequence

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

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

One can prove the following proposition

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

References

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

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

Received June 4, 1998

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

The for (going up) Macro Instruction

Piotr Rudnicki[1]
University of Alberta
Edmonton

Summary. We define a `for` type (going up) macro instruction in terms of the `while` macro. This gives an iterative macro with an explicit control variable. The `for` macro is used to define a macro for the selection sort acting on a finite sequence location of $\mathbf{SCM}_{\mathrm{FSA}}$. On the way, a macro for finding a minimum in a section of an array is defined.

MML Identifier: `SFMASTR3`.

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

1. General Preliminaries

The following propositions are true:

(1) Let X be a set, p be a permutation of X, and x, y be elements of X. Then $p \mathbin{+\cdot} (x, p(y)) \mathbin{+\cdot} (y, p(x))$ is a permutation of X.

(2) Let f be a function and x, y be sets. Suppose $x \in \operatorname{dom} f$ and $y \in \operatorname{dom} f$. Then there exists a permutation p of $\operatorname{dom} f$ such that $f \mathbin{+\cdot} (x, f(y)) \mathbin{+\cdot} (y, f(x)) = f \cdot p$.

Let X be a finite non empty subset of $\mathbb{R}$. The functor $\min X$ yielding a real number is defined by:

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

ISSN 1426–2630

(Def. 1) $\min X \in X$ and for every real number k such that $k \in X$ holds $\min X \leqslant k$.

Let X be a finite non empty subset of $\mathbb{Z}$. The functor $\min X$ yielding an integer is defined by:

(Def. 2) There exists a finite non empty subset Y of $\mathbb{R}$ such that $Y = X$ and $\min X = \min Y$.

Let F be a finite sequence of elements of $\mathbb{Z}$ and let m, n be natural numbers. Let us assume that $1 \leqslant m$ and $m \leqslant n$ and $n \leqslant \operatorname{len} F$. The functor $\min_m^n F$ yields a natural number and is defined as follows:

(Def. 3) There exists a finite non empty subset X of $\mathbb{Z}$ such that $X = \operatorname{rng}\langle F(m), \ldots, F(n)\rangle$ and $(\min_m^n F) + 1 = (\min X) \leftarrow\!\!\!\!\!\!\!\circ\ \langle F(m), \ldots, F(n)\rangle + m$.

We use the following convention: F, F_1 denote finite sequences of elements of $\mathbb{Z}$ and k, m, n, m_1 denote natural numbers.

The following propositions are true:

(3) Suppose $1 \leqslant m$ and $m \leqslant n$ and $n \leqslant \operatorname{len} F$. Then $m_1 = \min_m^n F$ if and only if the following conditions are satisfied:

(i) $m \leqslant m_1$,

(ii) $m_1 \leqslant n$,

(iii) for every natural number i such that $m \leqslant i$ and $i \leqslant n$ holds $F(m_1) \leqslant F(i)$, and

(iv) for every natural number i such that $m \leqslant i$ and $i < m_1$ holds $F(m_1) < F(i)$.

(4) If $1 \leqslant m$ and $m \leqslant \operatorname{len} F$, then $\min_m^m F = m$.

Let F be a finite sequence of elements of $\mathbb{Z}$ and let m, n be natural numbers. We say that F is non decreasing on m, n if and only if:

(Def. 4) For all natural numbers i, j such that $m \leqslant i$ and $i \leqslant j$ and $j \leqslant n$ holds $F(i) \leqslant F(j)$.

Let F be a finite sequence of elements of $\mathbb{Z}$ and let n be a natural number. We say that F is split at n if and only if:

(Def. 5) For all natural numbers i, j such that $1 \leqslant i$ and $i \leqslant n$ and $n < j$ and $j \leqslant \operatorname{len} F$ holds $F(i) \leqslant F(j)$.

We now state two propositions:

(5) Suppose $k + 1 \leqslant \operatorname{len} F$ and $m_1 = \min_{(k+1)}^{(\operatorname{len} F)} F$ and F is split at k and F is non decreasing on 1, k and $F_1 = F +\cdot (k+1, F(m_1)) +\cdot (m_1, F(k+1))$. Then F_1 is non decreasing on 1, $k+1$.

(6) If $k + 1 \leqslant \operatorname{len} F$ and $m_1 = \min_{(k+1)}^{(\operatorname{len} F)} F$ and F is split at k and $F_1 = F +\cdot (k+1, F(m_1)) +\cdot (m_1, F(k+1))$, then F_1 is split at $k+1$.

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

For simplicity, we use the following convention: s is a state of $\mathbf{SCM}_{\mathrm{FSA}}$, a, c are read-write integer locations, a_1, b_1, c_1, d_1, x are integer locations, f is a finite sequence location, I, J are macro instructions, I_1 is a good macro instruction, and k is a natural number.

The following propositions are true:

(7) If I is closed on Initialize(s) and halting on Initialize(s) and I does not destroy a_1, then $(\mathrm{IExec}(I,s))(a_1) = (\mathrm{Initialize}(s))(a_1)$.

(8) If $s(\mathrm{intloc}(0)) = 1$, then $\mathrm{IExec}(\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}}, s){\restriction}D = s{\restriction}D$, where $D =$ Int-Locations $\cup$ FinSeq-Locations.

(9) $\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}}$ does not refer a_1.

(10) If $a_1 \neq b_1$, then $c_1{:=}b_1$ does not refer a_1.

(11) $(\mathrm{Exec}(a{:=}f_{b_1}, s))(a) = \pi_{|s(b_1)|}s(f)$.

(12) $(\mathrm{Exec}(f_{a_1}{:=}b_1, s))(f) = s(f) +\cdot (|s(a_1)|, s(b_1))$.

Let a be a read-write integer location, let b be an integer location, and let I, J be good macro instructions. Observe that **if** $a > b$ **then** I **else** J is good.

One can prove the following propositions:

(13) UsedIntLoc(**if** $a_1 > b_1$ **then** I **else** J) $= \{a_1, b_1\} \cup \mathrm{UsedIntLoc}(I) \cup \mathrm{UsedIntLoc}(J)$.

(14) If I does not destroy a_1, then **while** $b_1 > 0$ **do** I does not destroy a_1.

(15) If $c_1 \neq a_1$ and I does not destroy c_1 and J does not destroy c_1, then **if** $a_1 > b_1$ **then** I **else** J does not destroy c_1.

3. The for-up Macro Instruction

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

(Def. 6) $\mathrm{StepForUp}(a,b,c,I,s) = \mathit{StepWhile{>}0}$
$(a_2, I;$
$\mathrm{AddTo}(a, \mathrm{intloc}(0));$
$\mathrm{SubFrom}(a_2, \mathrm{intloc}(0)), s +\cdot (a_2, (s(c) - s(b)) + 1) +\cdot (a, s(b)))$,
where $a_2 = 1^{\mathrm{st}}$-RWNotIn($\{a, b, c\} \cup \mathrm{UsedIntLoc}(I)$).

Next we state several propositions:

(16) If $s(\mathrm{intloc}(0)) = 1$, then $(\mathrm{StepForUp}(a, b_1, c_1, I, s))(0)(\mathrm{intloc}(0)) = 1$.

(17) $(\mathrm{StepForUp}(a, b_1, c_1, I, s))(0)(a) = s(b_1)$.

(18) If $a \neq b_1$, then $(\mathrm{StepForUp}(a, b_1, c_1, I, s))(0)(b_1) = s(b_1)$.

(19) If $a \neq c_1$, then $(\mathrm{StepForUp}(a, b_1, c_1, I, s))(0)(c_1) = s(c_1)$.

(20) If $a \neq d_1$ and $d_1 \in \mathrm{UsedIntLoc}(I)$, then $(\mathrm{StepForUp}(a, b_1, c_1, I, s))(0)(d_1) = s(d_1)$.

(21) $(\mathrm{StepForUp}(a, b_1, c_1, I, s))(0)(f) = s(f)$.

(22) Suppose $s(\mathrm{intloc}(0)) = 1$. Let a_2 be a read-write integer location. If $a_2 = 1^{\mathrm{st}}\text{-}\mathrm{RWNotIn}(\{a, b_1, c_1\} \cup \mathrm{UsedIntLoc}(I))$, then $\mathrm{IExec}((a_2{:=}c_1);\mathrm{SubFrom}(a_2, b_1);\mathrm{AddTo}(a_2, \mathrm{intloc}(0));(a{:=}b_1), s){\restriction}D = (s{+\cdot}(a_2, (s(c_1){-}s(b_1)){+}1){+\cdot}(a, s(b_1))){\restriction}D$, where $a_2 = 1^{\mathrm{st}}\text{-}\mathrm{RWNotIn}(\{a, b, c\}\cup \mathrm{UsedIntLoc}(I))$ and $D = \mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations}$.

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

(Def. 7) For every natural number i such that $i < (s(c){-}s(b)){+}1$ holds I is closed on $(\mathrm{StepForUp}(a, b, c, I, s))(i)$ and halting on $(\mathrm{StepForUp}(a, b, c, I, s))(i)$.

Next we state several propositions:

(23) For every parahalting macro instruction I holds ProperForUpBody a_1, b_1, c_1, I, s.

(24) If $(\mathrm{StepForUp}(a, b_1, c_1, I_1, s))(k)(\mathrm{intloc}(0)) = 1$ and I_1 is closed on $(\mathrm{StepForUp}(a, b_1, c_1, I_1, s))(k)$ and halting on $(\mathrm{StepForUp}(a, b_1, c_1, I_1, s))(k)$, then $(\mathrm{StepForUp}(a, b_1, c_1, I_1, s))(k + 1)(\mathrm{intloc}(0)) = 1$.

(25) Suppose $s(\mathrm{intloc}(0)) = 1$ and ProperForUpBody a, b_1, c_1, I_1, s. Let given k. Suppose $k \leqslant (s(c_1) - s(b_1)) + 1$. Then

(i) $(\mathrm{StepForUp}(a, b_1, c_1, I_1, s))(k)(\mathrm{intloc}(0)) = 1$,

(ii) if I_1 does not destroy a, then $(\mathrm{StepForUp}(a, b_1, c_1, I_1, s))(k)(a) = k + s(b_1)$ and $(\mathrm{StepForUp}(a, b_1, c_1, I_1, s))(k)(a) \leqslant s(c_1) + 1$, and

(iii) $(\mathrm{StepForUp}(a, b_1, c_1, I_1, s))(k)(1^{\mathrm{st}}\text{-}\mathrm{RWNotIn}(\{a, b_1, c_1\}\cup\mathrm{UsedIntLoc}(I_1))) + k = (s(c_1) - s(b_1)) + 1$.

(26) Suppose $s(\mathrm{intloc}(0)) = 1$ and ProperForUpBody a, b_1, c_1, I_1, s. Let given k. Then $(\mathrm{StepForUp}(a, b_1, c_1, I_1, s))(k)(1^{\mathrm{st}}\text{-}\mathrm{RWNotIn}(\{a, b_1, c_1\} \cup \mathrm{UsedIntLoc}(I_1))) > 0$ if and only if $k < (s(c_1) - s(b_1)) + 1$.

(27) Suppose $s(\mathrm{intloc}(0)) = 1$ and ProperForUpBody a, b_1, c_1, I_1, s and $k < (s(c_1){-}s(b_1)){+}1$. Then $(\mathrm{StepForUp}(a, b_1, c_1, I_1, s))(k{+}1){\restriction}(\{a, b_1, c_1\}\cup \mathrm{UsedIntLoc}(I_1)\cup F_2) = \mathrm{IExec}(I_1;\mathrm{AddTo}(a, \mathrm{intloc}(0)), (\mathrm{StepForUp}(a, b_1, c_1, I_1, s))(k)){\restriction}(\{a, b_1, c_1\}\cup\mathrm{UsedIntLoc}(I_1)\cup F_2)$, where $F_2 = \mathrm{FinSeq\text{-}Locations}$.

Let a, b, c be integer locations and let I be a macro instruction. The functor for-up(a, b, c, I) yields a macro instruction and is defined by:

(Def. 8) $\text{for-up}(a, b, c, I) =$
$(a_2{:=}c)$;
$\mathrm{SubFrom}(a_2, b)$;
$\mathrm{AddTo}(a_2, \mathrm{intloc}(0))$;

$(a{:=}b)$;(**while** $a_2 > 0$ **do** $(I$;
$\mathrm{AddTo}(a, \mathrm{intloc}(0))$;$\mathrm{SubFrom}(a_2, \mathrm{intloc}(0))))$,
where $a_2 = 1^{\mathrm{st}}\text{-RWNotIn}(\{a, b, c\} \cup \mathrm{UsedIntLoc}(I))$.

The following proposition is true

(28) $\{a_1, b_1, c_1\} \cup \mathrm{UsedIntLoc}(I) \subseteq \mathrm{UsedIntLoc}(\text{for-up}(a_1, b_1, c_1, I))$.

Let a be a read-write integer location, let b, c be integer locations, and let I be a good macro instruction. Note that for-up(a, b, c, I) is good.

Next we state four propositions:

(29) If $a \neq a_1$ and $a_1 \neq 1^{\mathrm{st}}\text{-RWNotIn}(\{a, b_1, c_1\} \cup \mathrm{UsedIntLoc}(I))$ and I does not destroy a_1, then for-up(a, b_1, c_1, I) does not destroy a_1.

(30) Suppose $s(\mathrm{intloc}(0)) = 1$ and $s(b_1) > s(c_1)$. Then for every x such that $x \neq a$ and $x \in \{b_1, c_1\} \cup \mathrm{UsedIntLoc}(I)$ holds $(\mathrm{IExec}(\text{for-up}(a, b_1, c_1, I), s))(x) = s(x)$ and for every f holds $(\mathrm{IExec}(\text{for-up}(a, b_1, c_1, I), s))(f) = s(f)$.

(31) Suppose $s(\mathrm{intloc}(0)) = 1$ but $k = (s(c_1) - s(b_1)) + 1$ but ProperForUpBody a, b_1, c_1, I_1, s or I_1 is parahalting. Then $\mathrm{IExec}(\text{for-up}(a, b_1, c_1, I_1), s) \restriction D = (\mathrm{StepForUp}(a, b_1, c_1, I_1, s))(k) \restriction D$, where $D = \text{Int-Locations} \cup \text{FinSeq-Locations}$.

(32) Suppose $s(\mathrm{intloc}(0)) = 1$ but ProperForUpBody a, b_1, c_1, I_1, s or I_1 is parahalting. Then for-up(a, b_1, c_1, I_1) is closed on s and for-up(a, b_1, c_1, I_1) is halting on s.

4. Finding Minimum in a Section of an Array

Let s_1, f_1, m_2 be integer locations and let f be a finite sequence location. The functor $\mathrm{FinSeqMin}(f, s_1, f_1, m_2)$ yielding a macro instruction is defined by:

(Def. 9) $\mathrm{FinSeqMin}(f, s_1, f_1, m_2) =$
$(m_2{:=}s_1)$;
for-up$(c_2, s_1, f_1,$
$(a_3{:=}f_{c_2})$;
$(a_4{:=}f_{m_2})$;
(**if** $a_4 > a_3$ **then** $\mathrm{Macro}(m_2{:=}c_2)$ **else** $(\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}})))$,
where $c_2 = 3^{\mathrm{rd}}\text{-RWNotIn}(\{s_1, f_1, m_2\})$,
$a_3 = 1^{\mathrm{st}}\text{-RWNotIn}(\{s_1, f_1, m_2\})$, and
$a_4 = 2^{\mathrm{nd}}\text{-RWNotIn}(\{s_1, f_1, m_2\})$.

Let s_1, f_1 be integer locations, let m_2 be a read-write integer location, and let f be a finite sequence location. Note that $\mathrm{FinSeqMin}(f, s_1, f_1, m_2)$ is good.

The following propositions are true:

(33) If $c \neq a_1$, then $\mathrm{FinSeqMin}(f, a_1, b_1, c)$ does not destroy a_1.

(34) $\{a_1, b_1, c\} \subseteq \mathrm{UsedIntLoc}(\mathrm{FinSeqMin}(f, a_1, b_1, c))$.

(35) If $s(\mathrm{intloc}(0)) = 1$, then $\mathrm{FinSeqMin}(f, a_1, b_1, c)$ is closed on s and $\mathrm{FinSeqMin}(f, a_1, b_1, c)$ is halting on s.

(36) If $a_1 \neq c$ and $b_1 \neq c$ and $s(\mathrm{intloc}(0)) = 1$, then $(\mathrm{IExec}(\mathrm{FinSeqMin}(f, a_1, b_1, c), s))(f) = s(f)$ and $(\mathrm{IExec}(\mathrm{FinSeqMin}(f, a_1, b_1, c), s))(a_1) = s(a_1)$ and $(\mathrm{IExec}(\mathrm{FinSeqMin}(f, a_1, b_1, c), s))(b_1) = s(b_1)$.

(37) If $1 \leqslant s(a_1)$ and $s(a_1) \leqslant s(b_1)$ and $s(b_1) \leqslant \mathrm{len}\, s(f)$ and $a_1 \neq c$ and $b_1 \neq c$ and $s(\mathrm{intloc}(0)) = 1$, then $(\mathrm{IExec}(\mathrm{FinSeqMin}(f, a_1, b_1, c), s))(c) = \min_{|s(a_1)|}^{|s(b_1)|} s(f)$.

5. A Swap Macro Instruction

Let f be a finite sequence location and let a, b be integer locations. The functor $\mathrm{swap}(f, a, b)$ yields a macro instruction and is defined as follows:

(Def. 10) $\mathrm{swap}(f, a, b) = (a_3{:=}f_a);(a_4{:=}f_b);(f_a{:=}a_4);(f_b{:=}a_3)$, where $a_3 = 1^{\mathrm{st}}\text{-}\mathrm{RWNotIn}(\{s_1, f_1, m_2\})$ and $a_4 = 2^{\mathrm{nd}}\text{-}\mathrm{RWNotIn}(\{s_1, f_1, m_2\})$.

Let f be a finite sequence location and let a, b be integer locations. Note that $\mathrm{swap}(f, a, b)$ is good and parahalting.

The following propositions are true:

(38) If $c_1 \neq 1^{\mathrm{st}}\text{-}\mathrm{RWNotIn}(\{a_1, b_1\})$ and $c_1 \neq 2^{\mathrm{nd}}\text{-}\mathrm{RWNotIn}(\{a_1, b_1\})$, then $\mathrm{swap}(f, a_1, b_1)$ does not destroy c_1.

(39) If $1 \leqslant s(a_1)$ and $s(a_1) \leqslant \mathrm{len}\, s(f)$ and $1 \leqslant s(b_1)$ and $s(b_1) \leqslant \mathrm{len}\, s(f)$ and $s(\mathrm{intloc}(0)) = 1$, then $(\mathrm{IExec}(\mathrm{swap}(f, a_1, b_1), s))(f) = s(f) +\cdot (s(a_1), s(f)(s(b_1))) +\cdot (s(b_1), s(f)(s(a_1)))$.

(40) Suppose $1 \leqslant s(a_1)$ and $s(a_1) \leqslant \mathrm{len}\, s(f)$ and $1 \leqslant s(b_1)$ and $s(b_1) \leqslant \mathrm{len}\, s(f)$ and $s(\mathrm{intloc}(0)) = 1$. Then $(\mathrm{IExec}(\mathrm{swap}(f, a_1, b_1), s))(f)(s(a_1)) = s(f)(s(b_1))$ and $(\mathrm{IExec}(\mathrm{swap}(f, a_1, b_1), s))(f)(s(b_1)) = s(f)(s(a_1))$.

(41) $\{a_1, b_1\} \subseteq \mathrm{UsedIntLoc}(\mathrm{swap}(f, a_1, b_1))$.

(42) $\mathrm{UsedInt}^*\,\mathrm{Loc}(\mathrm{swap}(f, a_1, b_1)) = \{f\}$.

6. Selection Sort

Let f be a finite sequence location. The functor Selection-sort f yielding a macro instruction is defined as follows:

(Def. 11) $\text{Selection-sort}\, f = (f_1{:=}\mathrm{len} f); \text{for-up}(c_2, \mathrm{intloc}(0), f'_1, \mathrm{FinSeqMin}(f, c_2, f'_1, m'_1); \mathrm{swap}(f, c_2, m'_1))$, where $c_2 = 3^{\mathrm{rd}}\text{-}\mathrm{RWNotIn}(\{s_1, f_1, m_2\})$, $f'_1 = 1^{\mathrm{st}}\text{-}\mathrm{NotUsed}(\mathrm{swap}(f, c_2, m'_1))$, and $m'_1 = 2^{\mathrm{nd}}\text{-}\mathrm{RWNotIn}(\emptyset_{\text{Int-Locations}})$.

The following proposition is true

(43) Let S be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $S = \mathrm{IExec}(\text{Selection-sort}\, f, s)$. Then $S(f)$ is non decreasing on 1, $\mathrm{len}\, S(f)$ and there exists a permutation p of $\mathrm{Seg}\,\mathrm{len}\, s(f)$ such that $S(f) = s(f) \cdot p$.

References

[1] Noriko Asamoto. Conditional branch macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Part II. *Formalized Mathematics*, 6(**1**):73–80, 1997.

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

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

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

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

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

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

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

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

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

[14] Andrzej Kondracki. The chinese remainder theorem. *Formalized Mathematics*, 6(**4**):573–577, 1997.

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

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

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

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

[19] Andrzej Trybulec. Semilattice operations on finite subsets. *Formalized Mathematics*, 1(**2**):369–376, 1990.

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

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

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

[23] Andrzej Trybulec, Yatsuka Nakamura, and Noriko Asamoto. On the compositions of macro instructions. Part I. *Formalized Mathematics*, 6(**1**):21–27, 1997.

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

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

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

[27] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[28] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[29] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[30] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received June 4, 1998

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

Bounding Boxes for Special Sequences in $\mathcal{E}^2$

Yatsuka Nakamura
Shinshu University
Nagano

Adam Grabowski[1]
University of Białystok

Summary. This is the continuation of the proof of the Jordan Theorem according to [18].

MML Identifier: `JORDAN5D`.

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

1. Preliminaries

For simplicity, we use the following convention: p, q denote points of $\mathcal{E}^2_{\rm T}$, s, r denote real numbers, h denotes a non constant standard special circular sequence, g denotes a finite sequence of elements of $\mathcal{E}^2_{\rm T}$, f denotes a non empty finite sequence of elements of $\mathcal{E}^2_{\rm T}$, and I, i_1, i, j, k denote natural numbers.

We now state a number of propositions:

(1) Let B be a subset of $\mathbb{R}$. Suppose there exists a real number r_1 such that $r_1 \in B$ and B is lower bounded and for every r such that $r \in B$ holds $s \leqslant r$. Then $s \leqslant \inf B$.

(2) Let B be a subset of $\mathbb{R}$. Suppose there exists a real number r_1 such that $r_1 \in B$ and B is upper bounded and for every r such that $r \in B$ holds $s \geqslant r$. Then $s \geqslant \sup B$.

(3) $\pi_{\operatorname{len} h} h \in \mathcal{L}(h, \operatorname{len} h -' 1)$.

[1]A part of this paper was written while the author visited the Shinshu University in the winter of 1997.

ISSN 1426–2630

(4) If $3 \leqslant i$, then $i \bmod (i -' 1) = 1$.

(5) If $p \in \operatorname{rng} h$, then there exists a natural number i such that $1 \leqslant i$ and $i + 1 \leqslant \operatorname{len} h$ and $h(i) = p$.

(6) For every finite sequence g of elements of $\mathbb{R}$ such that $r \in \operatorname{rng} g$ holds $(\operatorname{Inc}(g))(1) \leqslant r$ and $r \leqslant (\operatorname{Inc}(g))(\operatorname{len} \operatorname{Inc}(g))$.

(7) Suppose $1 \leqslant i$ and $i \leqslant \operatorname{len} h$ and $1 \leqslant I$ and $I \leqslant$ width the Go-board of h. Then $((\text{the Go-board of } h)_{1,I})_{\mathbf{1}} \leqslant (\pi_i h)_{\mathbf{1}}$ and $(\pi_i h)_{\mathbf{1}} \leqslant ((\text{the Go-board of } h)_{\text{len the Go-board of } h,\, I})_{\mathbf{1}}$.

(8) Suppose $1 \leqslant i$ and $i \leqslant \operatorname{len} h$ and $1 \leqslant I$ and $I \leqslant$ len the Go-board of h. Then $((\text{the Go-board of } h)_{I,1})_{\mathbf{2}} \leqslant (\pi_i h)_{\mathbf{2}}$ and $(\pi_i h)_{\mathbf{2}} \leqslant ((\text{the Go-board of } h)_{I,\text{width the Go-board of } h})_{\mathbf{2}}$.

(9) Suppose $1 \leqslant i$ and $i \leqslant$ len the Go-board of f. Then there exist k, j such that $k \in \operatorname{dom} f$ and $\langle i, j \rangle \in$ the indices of the Go-board of f and $\pi_k f = (\text{the Go-board of } f)_{i,j}$.

(10) Suppose $1 \leqslant j$ and $j \leqslant$ width the Go-board of f. Then there exist k, i such that $k \in \operatorname{dom} f$ and $\langle i, j \rangle \in$ the indices of the Go-board of f and $\pi_k f = (\text{the Go-board of } f)_{i,j}$.

(11) Suppose $1 \leqslant i$ and $i \leqslant$ len the Go-board of f and $1 \leqslant j$ and $j \leqslant$ width the Go-board of f. Then there exists k such that $k \in \operatorname{dom} f$ and $\langle i, j \rangle \in$ the indices of the Go-board of f and $(\pi_k f)_{\mathbf{1}} = ((\text{the Go-board of } f)_{i,j})_{\mathbf{1}}$.

(12) Suppose $1 \leqslant i$ and $i \leqslant$ len the Go-board of f and $1 \leqslant j$ and $j \leqslant$ width the Go-board of f. Then there exists k such that $k \in \operatorname{dom} f$ and $\langle i, j \rangle \in$ the indices of the Go-board of f and $(\pi_k f)_{\mathbf{2}} = ((\text{the Go-board of } f)_{i,j})_{\mathbf{2}}$.

2. Extrema of Projections

One can prove the following propositions:

(13) If $1 \leqslant i$ and $i \leqslant \operatorname{len} h$, then S-bound $\widetilde{\mathcal{L}}(h) \leqslant (\pi_i h)_{\mathbf{2}}$ and $(\pi_i h)_{\mathbf{2}} \leqslant$ N-bound $\widetilde{\mathcal{L}}(h)$.

(14) If $1 \leqslant i$ and $i \leqslant \operatorname{len} h$, then W-bound $\widetilde{\mathcal{L}}(h) \leqslant (\pi_i h)_{\mathbf{1}}$ and $(\pi_i h)_{\mathbf{1}} \leqslant$ E-bound $\widetilde{\mathcal{L}}(h)$.

(15) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{2}} : q_{\mathbf{1}} = \text{W-bound}\,\widetilde{\mathcal{L}}(h) \wedge q \in \widetilde{\mathcal{L}}(h)\}$ holds $X = (\text{proj2} \restriction \text{W-most}\,\widetilde{\mathcal{L}}(h))^{\circ}$(the carrier of $(\mathcal{E}^2_{\mathrm{T}}) \restriction \text{W-most}\,\widetilde{\mathcal{L}}(h)$).

(16) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{2}} : q_{\mathbf{1}} = \text{E-bound}\,\widetilde{\mathcal{L}}(h) \wedge q \in \widetilde{\mathcal{L}}(h)\}$ holds $X = (\text{proj2} \restriction \text{E-most}\,\widetilde{\mathcal{L}}(h))^{\circ}$(the carrier of $(\mathcal{E}^2_{\mathrm{T}}) \restriction \text{E-most}\,\widetilde{\mathcal{L}}(h)$).

(17) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{1}} : q_{\mathbf{2}} = \text{N-bound}\,\widetilde{\mathcal{L}}(h) \wedge q \in \widetilde{\mathcal{L}}(h)\}$ holds $X = (\text{proj1} \restriction \text{N-most}\,\widetilde{\mathcal{L}}(h))^{\circ}$(the carrier of $(\mathcal{E}^2_{\mathrm{T}}) \restriction \text{N-most}\,\widetilde{\mathcal{L}}(h)$).

(18) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{1}} : q_{\mathbf{2}} = \text{S-bound}\,\widetilde{\mathcal{L}}(h) \wedge q \in \widetilde{\mathcal{L}}(h)\}$ holds $X = (\text{proj1} \restriction \text{S-most}\,\widetilde{\mathcal{L}}(h))^{\circ}(\text{the carrier of } (\mathcal{E}^2_{\mathrm{T}}) \restriction \text{S-most}\,\widetilde{\mathcal{L}}(h))$.

(19) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{1}} : q \in \widetilde{\mathcal{L}}(g)\}$ holds $X = (\text{proj1} \restriction \widetilde{\mathcal{L}}(g))^{\circ}(\text{the carrier of } (\mathcal{E}^2_{\mathrm{T}}) \restriction \widetilde{\mathcal{L}}(g))$.

(20) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{2}} : q \in \widetilde{\mathcal{L}}(g)\}$ holds $X = (\text{proj2} \restriction \widetilde{\mathcal{L}}(g))^{\circ}(\text{the carrier of } (\mathcal{E}^2_{\mathrm{T}}) \restriction \widetilde{\mathcal{L}}(g))$.

(21) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{2}} : q_{\mathbf{1}} = \text{W-bound}\,\widetilde{\mathcal{L}}(h) \wedge q \in \widetilde{\mathcal{L}}(h)\}$ holds $\inf X = \inf(\text{proj2} \restriction \text{W-most}\,\widetilde{\mathcal{L}}(h))$.

(22) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{2}} : q_{\mathbf{1}} = \text{W-bound}\,\widetilde{\mathcal{L}}(h) \wedge q \in \widetilde{\mathcal{L}}(h)\}$ holds $\sup X = \sup(\text{proj2} \restriction \text{W-most}\,\widetilde{\mathcal{L}}(h))$.

(23) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{2}} : q_{\mathbf{1}} = \text{E-bound}\,\widetilde{\mathcal{L}}(h) \wedge q \in \widetilde{\mathcal{L}}(h)\}$ holds $\inf X = \inf(\text{proj2} \restriction \text{E-most}\,\widetilde{\mathcal{L}}(h))$.

(24) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{2}} : q_{\mathbf{1}} = \text{E-bound}\,\widetilde{\mathcal{L}}(h) \wedge q \in \widetilde{\mathcal{L}}(h)\}$ holds $\sup X = \sup(\text{proj2} \restriction \text{E-most}\,\widetilde{\mathcal{L}}(h))$.

(25) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{1}} : q \in \widetilde{\mathcal{L}}(g)\}$ holds $\inf X = \inf(\text{proj1} \restriction \widetilde{\mathcal{L}}(g))$.

(26) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{1}} : q_{\mathbf{2}} = \text{S-bound}\,\widetilde{\mathcal{L}}(h) \wedge q \in \widetilde{\mathcal{L}}(h)\}$ holds $\inf X = \inf(\text{proj1} \restriction \text{S-most}\,\widetilde{\mathcal{L}}(h))$.

(27) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{1}} : q_{\mathbf{2}} = \text{S-bound}\,\widetilde{\mathcal{L}}(h) \wedge q \in \widetilde{\mathcal{L}}(h)\}$ holds $\sup X = \sup(\text{proj1} \restriction \text{S-most}\,\widetilde{\mathcal{L}}(h))$.

(28) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{1}} : q_{\mathbf{2}} = \text{N-bound}\,\widetilde{\mathcal{L}}(h) \wedge q \in \widetilde{\mathcal{L}}(h)\}$ holds $\inf X = \inf(\text{proj1} \restriction \text{N-most}\,\widetilde{\mathcal{L}}(h))$.

(29) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{1}} : q_{\mathbf{2}} = \text{N-bound}\,\widetilde{\mathcal{L}}(h) \wedge q \in \widetilde{\mathcal{L}}(h)\}$ holds $\sup X = \sup(\text{proj1} \restriction \text{N-most}\,\widetilde{\mathcal{L}}(h))$.

(30) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{2}} : q \in \widetilde{\mathcal{L}}(g)\}$ holds $\inf X = \inf(\text{proj2} \restriction \widetilde{\mathcal{L}}(g))$.

(31) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{1}} : q \in \widetilde{\mathcal{L}}(g)\}$ holds $\sup X = \sup(\text{proj1} \restriction \widetilde{\mathcal{L}}(g))$.

(32) For every subset X of $\mathbb{R}$ such that $X = \{q_{\mathbf{2}} : q \in \widetilde{\mathcal{L}}(g)\}$ holds $\sup X = \sup(\text{proj2} \restriction \widetilde{\mathcal{L}}(g))$.

(33) If $p \in \widetilde{\mathcal{L}}(h)$ and $1 \leqslant I$ and $I \leqslant \text{width}\,\text{the Go-board of } h$, then $((\text{the Go-board of } h)_{1,I})_{\mathbf{1}} \leqslant p_{\mathbf{1}}$.

(34) If $p \in \widetilde{\mathcal{L}}(h)$ and $1 \leqslant I$ and $I \leqslant \text{width}\,\text{the Go-board of } h$, then $p_{\mathbf{1}} \leqslant ((\text{the Go-board of } h)_{\text{len the Go-board of } h,\, I})_{\mathbf{1}}$.

(35) If $p \in \widetilde{\mathcal{L}}(h)$ and $1 \leqslant I$ and $I \leqslant \text{len}\,\text{the Go-board of } h$, then $((\text{the Go-board of } h)_{I,1})_{\mathbf{2}} \leqslant p_{\mathbf{2}}$.

(36) If $p \in \widetilde{\mathcal{L}}(h)$ and $1 \leqslant I$ and $I \leqslant \text{len}\,\text{the Go-board of } h$, then $p_{\mathbf{2}} \leqslant ((\text{the Go-board of } h)_{I,\text{width the Go-board of } h})_{\mathbf{2}}$.

(37) Suppose $1 \leqslant i$ and $i \leqslant \operatorname{len}$ the Go-board of h and $1 \leqslant j$ and $j \leqslant \operatorname{width}$ the Go-board of h. Then there exists q such that $q_{\mathbf{1}} = ((\text{the Go-board of } h)_{i,j})_{\mathbf{1}}$ and $q \in \widetilde{\mathcal{L}}(h)$.

(38) Suppose $1 \leqslant i$ and $i \leqslant \operatorname{len}$ the Go-board of h and $1 \leqslant j$ and $j \leqslant \operatorname{width}$ the Go-board of h. Then there exists q such that $q_{\mathbf{2}} = ((\text{the Go-board of } h)_{i,j})_{\mathbf{2}}$ and $q \in \widetilde{\mathcal{L}}(h)$.

(39) W-bound $\widetilde{\mathcal{L}}(h) = ((\text{the Go-board of } h)_{1,1})_{\mathbf{1}}$.

(40) S-bound $\widetilde{\mathcal{L}}(h) = ((\text{the Go-board of } h)_{1,1})_{\mathbf{2}}$.

(41) E-bound $\widetilde{\mathcal{L}}(h) = ((\text{the Go-board of } h)_{\text{len the Go-board of } h,\, 1})_{\mathbf{1}}$.

(42) N-bound $\widetilde{\mathcal{L}}(h) = ((\text{the Go-board of } h)_{1,\text{width the Go-board of } h})_{\mathbf{2}}$.

(43) Let Y be a non empty finite subset of $\mathbb{N}$. Suppose that

(i) $1 \leqslant i$,
(ii) $i \leqslant \operatorname{len} f$,
(iii) $1 \leqslant I$,
(iv) $I \leqslant \operatorname{len}$ the Go-board of f,
(v) $Y = \{j : \langle I, j\rangle \in \text{the indices of the Go-board of } f \ \wedge\ \bigvee_k (k \in \operatorname{dom} f \ \wedge\ \pi_k f = (\text{the Go-board of } f)_{I,j})\}$,
(vi) $(\pi_i f)_{\mathbf{1}} = ((\text{the Go-board of } f)_{I,1})_{\mathbf{1}}$, and
(vii) $i_1 = \min Y$.

Then $((\text{the Go-board of } f)_{I,i_1})_{\mathbf{2}} \leqslant (\pi_i f)_{\mathbf{2}}$.

(44) Let Y be a non empty finite subset of $\mathbb{N}$. Suppose that

(i) $1 \leqslant i$,
(ii) $i \leqslant \operatorname{len} h$,
(iii) $1 \leqslant I$,
(iv) $I \leqslant \operatorname{width}$ the Go-board of h,
(v) $Y = \{j : \langle j, I\rangle \in \text{the indices of the Go-board of } h \ \wedge\ \bigvee_k (k \in \operatorname{dom} h \ \wedge\ \pi_k h = (\text{the Go-board of } h)_{j,I})\}$,
(vi) $(\pi_i h)_{\mathbf{2}} = ((\text{the Go-board of } h)_{1,I})_{\mathbf{2}}$, and
(vii) $i_1 = \min Y$.

Then $((\text{the Go-board of } h)_{i_1,I})_{\mathbf{1}} \leqslant (\pi_i h)_{\mathbf{1}}$.

(45) Let Y be a non empty finite subset of $\mathbb{N}$. Suppose that

(i) $1 \leqslant i$,
(ii) $i \leqslant \operatorname{len} h$,
(iii) $1 \leqslant I$,
(iv) $I \leqslant \operatorname{width}$ the Go-board of h,
(v) $Y = \{j : \langle j, I\rangle \in \text{the indices of the Go-board of } h \ \wedge\ \bigvee_k (k \in \operatorname{dom} h \ \wedge\ \pi_k h = (\text{the Go-board of } h)_{j,I})\}$,
(vi) $(\pi_i h)_{\mathbf{2}} = ((\text{the Go-board of } h)_{1,I})_{\mathbf{2}}$, and
(vii) $i_1 = \max Y$.

Then $((\text{the Go-board of } h)_{i_1,I})_{\mathbf{1}} \geqslant (\pi_i h)_{\mathbf{1}}$.

(46) Let Y be a non empty finite subset of $\mathbb{N}$. Suppose that
(i) $1 \leqslant i$,
(ii) $i \leqslant \operatorname{len} f$,
(iii) $1 \leqslant I$,
(iv) $I \leqslant \operatorname{len}$ the Go-board of f,
(v) $Y = \{j : \langle I, j\rangle \in$ the indices of the Go-board of $f \ \wedge\ \bigvee_k (k \in \operatorname{dom} f \ \wedge\ \pi_k f = (\text{the Go-board of } f)_{I,j})\}$,
(vi) $(\pi_i f)_{\mathbf{1}} = ((\text{the Go-board of } f)_{I,1})_{\mathbf{1}}$, and
(vii) $i_1 = \max Y$.
Then $((\text{the Go-board of } f)_{I,i_1})_{\mathbf{2}} \geqslant (\pi_i f)_{\mathbf{2}}$.

3. Coordinates of the Special Circular Sequences Bounding Boxes

Let g be a non constant standard special circular sequence. The functor $\mathrm{i}_{\mathrm{SW}}\, g$ yields a natural number and is defined as follows:

(Def. 1) $\langle 1, \mathrm{i}_{\mathrm{SW}}\, g\rangle \in$ the indices of the Go-board of g and $(\text{the Go-board of } g)_{1,\mathrm{i}_{\mathrm{SW}}\, g} = \text{W-min}\, \widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{i}_{\mathrm{NW}}\, g$ yields a natural number and is defined by:

(Def. 2) $\langle 1, \mathrm{i}_{\mathrm{NW}}\, g\rangle \in$ the indices of the Go-board of g and $(\text{the Go-board of } g)_{1,\mathrm{i}_{\mathrm{NW}}\, g} = \text{W-max}\, \widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{i}_{\mathrm{SE}}\, g$ yielding a natural number is defined by the conditions (Def. 3).

(Def. 3)(i) $\langle \operatorname{len}$ the Go-board of $g,\ \mathrm{i}_{\mathrm{SE}}\, g\rangle \in$ the indices of the Go-board of g, and
(ii) $(\text{the Go-board of } g)_{\operatorname{len} \text{the Go-board of } g,\ \mathrm{i}_{\mathrm{SE}}\, g} = \text{E-min}\, \widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{i}_{\mathrm{NE}}\, g$ yielding a natural number is defined by the conditions (Def. 4).

(Def. 4)(i) $\langle \operatorname{len}$ the Go-board of $g,\ \mathrm{i}_{\mathrm{NE}}\, g\rangle \in$ the indices of the Go-board of g, and
(ii) $(\text{the Go-board of } g)_{\operatorname{len} \text{the Go-board of } g,\ \mathrm{i}_{\mathrm{NE}}\, g} = \text{E-max}\, \widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{i}_{\mathrm{WS}}\, g$ yields a natural number and is defined by:

(Def. 5) $\langle \mathrm{i}_{\mathrm{WS}}\, g, 1\rangle \in$ the indices of the Go-board of g and $(\text{the Go-board of } g)_{\mathrm{i}_{\mathrm{WS}}\, g,1} = \text{S-min}\, \widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{i}_{\mathrm{ES}}\, g$ yields a natural number and is defined by:

(Def. 6) $\langle \mathrm{i}_{\mathrm{ES}}\, g, 1\rangle \in$ the indices of the Go-board of g and $(\text{the Go-board of } g)_{\mathrm{i}_{\mathrm{ES}}\, g,1} = \text{S-max}\, \widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{i}_{\mathrm{WN}}\, g$ yields a natural number and is defined by the conditions (Def. 7).

(Def. 7)(i) $\langle \mathrm{i}_{\mathrm{WN}}\, g$, width the Go-board of $g\rangle \in$ the indices of the Go-board of g, and
(ii) $(\text{the Go-board of } g)_{\mathrm{i}_{\mathrm{WN}}\, g,\text{width the Go-board of } g} = \text{N-min}\, \widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{i}_{\mathrm{EN}}\, g$ yields a natural number and is defined by the conditions (Def. 8).

(Def. 8)(i) $\langle \mathrm{i}_{\mathrm{EN}}\, g, \mathrm{width}\,\text{the Go-board of } g\rangle \in$ the indices of the Go-board of g, and

(ii) $(\text{the Go-board of } g)_{\mathrm{i}_{\mathrm{EN}}\, g, \mathrm{width}\,\text{the Go-board of } g} = \text{N-max}\,\widetilde{\mathcal{L}}(g)$.

Next we state two propositions:

(47)(i) $1 \leqslant \mathrm{i}_{\mathrm{WN}}\, h$,

(ii) $\mathrm{i}_{\mathrm{WN}}\, h \leqslant \mathrm{len}\,\text{the Go-board of } h$,

(iii) $1 \leqslant \mathrm{i}_{\mathrm{EN}}\, h$,

(iv) $\mathrm{i}_{\mathrm{EN}}\, h \leqslant \mathrm{len}\,\text{the Go-board of } h$,

(v) $1 \leqslant \mathrm{i}_{\mathrm{WS}}\, h$,

(vi) $\mathrm{i}_{\mathrm{WS}}\, h \leqslant \mathrm{len}\,\text{the Go-board of } h$,

(vii) $1 \leqslant \mathrm{i}_{\mathrm{ES}}\, h$, and

(viii) $\mathrm{i}_{\mathrm{ES}}\, h \leqslant \mathrm{len}\,\text{the Go-board of } h$.

(48)(i) $1 \leqslant \mathrm{i}_{\mathrm{NE}}\, h$,

(ii) $\mathrm{i}_{\mathrm{NE}}\, h \leqslant \mathrm{width}\,\text{the Go-board of } h$,

(iii) $1 \leqslant \mathrm{i}_{\mathrm{SE}}\, h$,

(iv) $\mathrm{i}_{\mathrm{SE}}\, h \leqslant \mathrm{width}\,\text{the Go-board of } h$,

(v) $1 \leqslant \mathrm{i}_{\mathrm{NW}}\, h$,

(vi) $\mathrm{i}_{\mathrm{NW}}\, h \leqslant \mathrm{width}\,\text{the Go-board of } h$,

(vii) $1 \leqslant \mathrm{i}_{\mathrm{SW}}\, h$, and

(viii) $\mathrm{i}_{\mathrm{SW}}\, h \leqslant \mathrm{width}\,\text{the Go-board of } h$.

Let g be a non constant standard special circular sequence. The functor $\mathrm{n}_{\mathrm{SW}}\, g$ yields a natural number and is defined as follows:

(Def. 9) $1 \leqslant \mathrm{n}_{\mathrm{SW}}\, g$ and $\mathrm{n}_{\mathrm{SW}}\, g + 1 \leqslant \mathrm{len}\, g$ and $g(\mathrm{n}_{\mathrm{SW}}\, g) = \text{W-min}\,\widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{n}_{\mathrm{NW}}\, g$ yielding a natural number is defined as follows:

(Def. 10) $1 \leqslant \mathrm{n}_{\mathrm{NW}}\, g$ and $\mathrm{n}_{\mathrm{NW}}\, g + 1 \leqslant \mathrm{len}\, g$ and $g(\mathrm{n}_{\mathrm{NW}}\, g) = \text{W-max}\,\widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{n}_{\mathrm{SE}}\, g$ yielding a natural number is defined by:

(Def. 11) $1 \leqslant \mathrm{n}_{\mathrm{SE}}\, g$ and $\mathrm{n}_{\mathrm{SE}}\, g + 1 \leqslant \mathrm{len}\, g$ and $g(\mathrm{n}_{\mathrm{SE}}\, g) = \text{E-min}\,\widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{n}_{\mathrm{NE}}\, g$ yielding a natural number is defined by:

(Def. 12) $1 \leqslant \mathrm{n}_{\mathrm{NE}}\, g$ and $\mathrm{n}_{\mathrm{NE}}\, g + 1 \leqslant \mathrm{len}\, g$ and $g(\mathrm{n}_{\mathrm{NE}}\, g) = \text{E-max}\,\widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{n}_{\mathrm{WS}}\, g$ yielding a natural number is defined by:

(Def. 13) $1 \leqslant \mathrm{n}_{\mathrm{WS}}\, g$ and $\mathrm{n}_{\mathrm{WS}}\, g + 1 \leqslant \mathrm{len}\, g$ and $g(\mathrm{n}_{\mathrm{WS}}\, g) = \text{S-min}\,\widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{n}_{\mathrm{ES}}\, g$ yields a natural number and is defined as follows:

(Def. 14) $1 \leqslant \mathrm{n}_{\mathrm{ES}}\, g$ and $\mathrm{n}_{\mathrm{ES}}\, g + 1 \leqslant \mathrm{len}\, g$ and $g(\mathrm{n}_{\mathrm{ES}}\, g) = \text{S-max}\,\widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{n}_{\mathrm{WN}}\, g$ yielding a natural number is defined by:

(Def. 15) $1 \leqslant \mathrm{n}_{\mathrm{WN}}\, g$ and $\mathrm{n}_{\mathrm{WN}}\, g + 1 \leqslant \mathrm{len}\, g$ and $g(\mathrm{n}_{\mathrm{WN}}\, g) = \text{N-min}\,\widetilde{\mathcal{L}}(g)$.

The functor $\mathrm{n}_{\mathrm{EN}}\, g$ yielding a natural number is defined by:

(Def. 16) $1 \leqslant \mathrm{n}_{\mathrm{EN}}\, g$ and $\mathrm{n}_{\mathrm{EN}}\, g + 1 \leqslant \mathrm{len}\, g$ and $g(\mathrm{n}_{\mathrm{EN}}\, g) = \text{N-max}\,\widetilde{\mathcal{L}}(g)$.

Next we state four propositions:

(49) $\mathrm{n}_{\mathrm{WN}}\,h \neq \mathrm{n}_{\mathrm{WS}}\,h$.

(50) $\mathrm{n}_{\mathrm{SW}}\,h \neq \mathrm{n}_{\mathrm{SE}}\,h$.

(51) $\mathrm{n}_{\mathrm{EN}}\,h \neq \mathrm{n}_{\mathrm{ES}}\,h$.

(52) $\mathrm{n}_{\mathrm{NW}}\,h \neq \mathrm{n}_{\mathrm{NE}}\,h$.

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 and Piotr Rudnicki. Bounding boxes for compact sets in $\mathcal{E}^2$. *Formalized Mathematics*, 6(**3**):427–440, 1997.
[5] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[6] Agata Darmochwał. The Euclidean space. *Formalized Mathematics*, 2(**4**):599–603, 1991.
[7] 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.
[8] Agata Darmochwał and Andrzej Trybulec. Similarity of formulae. *Formalized Mathematics*, 2(**5**):635–642, 1991.
[9] Katarzyna Jankowska. Matrices. Abelian group of matrices. *Formalized Mathematics*, 2(**4**):475–480, 1991.
[10] Jarosław Kotowicz. Convergent real sequences. Upper and lower bound of sets of real numbers. *Formalized Mathematics*, 1(**3**):477–481, 1990.
[11] Jarosław Kotowicz. Monotone real sequences. Subsequences. *Formalized Mathematics*, 1(**3**):471–475, 1990.
[12] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part I. *Formalized Mathematics*, 3(**1**):107–115, 1992.
[13] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part II. *Formalized Mathematics*, 3(**1**):117–121, 1992.
[14] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, I. *Formalized Mathematics*, 5(**2**):167–172, 1996.
[15] Yatsuka Nakamura and Andrzej Trybulec. Decomposing a Go-board into cells. *Formalized Mathematics*, 5(**3**):323–328, 1996.
[16] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[17] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[18] 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.
[19] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[20] Andrzej Trybulec. On the decomposition of finite sequences. *Formalized Mathematics*, 5(**3**):317–322, 1996.
[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] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received June 8, 1998

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

Euler's Theorem and Small Fermat's Theorem

Yoshinori Fujisawa
Shinshu University
Nagano

Yasushi Fuwa
Shinshu University
Nagano

Hidetaka Shimizu
Information Technology Research Institute
of Nagano Prefecture

Summary. This article is concerned with Euler's theorem and small Fermat's theorem that play important roles in public-key cryptograms. In the first section, we present some selected theorems on integers. In the following section, we remake definitions about the finite sequence of natural, the function of natural times finite sequence of natural and π of the finite sequence of natural. We also prove some basic theorems that concern these redefinitions. Next, we define the function of modulus for finite sequence of natural and some fundamental theorems about this function are proved. Finally, Euler's theorem and small Fermat's theorem are proved.

MML Identifier: `EULER_2`.

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

1. Preliminary

We use the following convention: a, b, m, n, k, l, i, j, n_1, n_2, n_3 are natural numbers, t is an integer, and f, F are finite sequences of elements of $\mathbb{N}$.

We now state a number of propositions:

(1) a and b **qua** integer are relative prime iff a and b are relative prime.

ISSN 1426–2630

(2) If $m > 1$ and $m \cdot t \geqslant 1$, then $t \geqslant 1$.

(3) If $m > 1$ and $m \cdot t \geqslant 0$, then $t \geqslant 0$.

(4) If $m \neq 0$, then $n \bmod m = (n$ **qua** integer$)\ \bmod m$.

(5) Suppose $a \neq 0$ and $b \neq 0$ and $m \neq 0$ and a and m are relative prime and b and m are relative prime. Then m and $a \cdot b \bmod m$ are relative prime.

(6) Suppose $m > 1$ and $b \neq 0$ and m and n are relative prime and a and m are relative prime and $n = a \cdot b \bmod m$. Then m and b are relative prime.

(7) For every n such that $n \neq 0$ holds $m \bmod n \bmod n = m \bmod n$.

(8) For every n such that $n \neq 0$ holds $(l + m) \bmod n = ((l \bmod n) + (m \bmod n)) \bmod n$.

(9) For every n such that $n \neq 0$ holds $l \cdot m \bmod n = l \cdot (m \bmod n) \bmod n$.

(10) For every n such that $n \neq 0$ holds $l \cdot m \bmod n = (l \bmod n) \cdot m \bmod n$.

(11) For every n such that $n \neq 0$ holds $l \cdot m \bmod n = (l \bmod n) \cdot (m \bmod n) \bmod n$.

2. Finite Sequence of Naturals

We now state two propositions:

(12) For every finite sequence f of elements of $\mathbb{N}$ such that $n \neq 0$ and $n \leqslant m$ holds $(f{\restriction}m)(n) = f(n)$.

(13) For every finite sequence f of elements of $\mathbb{N}$ such that $n \leqslant m$ holds $f{\restriction}m{\restriction}n = f{\restriction}n$.

Let us consider a, f. Then $a \cdot f$ is a finite sequence of elements of $\mathbb{N}$.

One can prove the following propositions:

(14) For every finite sequence f of elements of $\mathbb{N}$ and for every natural number r holds $\prod(f \frown \langle r \rangle) = \prod f \cdot r$.

(15) For all finite sequences f_1, f_2 of elements of $\mathbb{N}$ holds $\prod(f_1 \frown f_2) = \prod f_1 \cdot \prod f_2$.

(16) $\prod(\varepsilon_{\mathbb{N}}) = 1$.

(17) $\prod\langle a \rangle = a$.

(18) $\prod(\langle a \rangle \frown F) = a \cdot \prod F$.

(19) $\prod\langle n_1, n_2 \rangle = n_1 \cdot n_2$.

(20) $\prod\langle n_1, n_2, n_3 \rangle = n_1 \cdot n_2 \cdot n_3$.

(21) $\prod(i \mapsto (1$ **qua** real number$)) = 1$.

(22) $\prod((i + j) \mapsto m) = \prod(i \mapsto m) \cdot \prod(j \mapsto m)$.

(23) $\prod((i \cdot j) \mapsto m) = \prod(j \mapsto \prod(i \mapsto m))$.

(24) $\prod(i \mapsto (n_1 \cdot n_2)) = \prod(i \mapsto n_1) \cdot \prod(i \mapsto n_2)$.

(25) For all finite sequences R_1, R_2 of elements of $\mathbb{N}$ such that R_1 and R_2 are fiberwise equipotent holds $\prod R_1 = \prod R_2$.

3. Modulus for Finite Sequence of Naturals

Let f be a finite sequence of elements of $\mathbb{N}$ and let m be a natural number. The functor $f \bmod m$ yielding a finite sequence of elements of $\mathbb{N}$ is defined by:

(Def. 1) $\operatorname{len}(f \bmod m) = \operatorname{len} f$ and for every natural number i such that $i \in \operatorname{dom} f$ holds $(f \bmod m)(i) = f(i) \bmod m$.

We now state several propositions:

(26) For every finite sequence f of elements of $\mathbb{N}$ such that $m \neq 0$ holds $\prod(f \bmod m) \bmod m = \prod f \bmod m$.

(27) If $a \neq 0$ and $m > 1$ and $n \neq 0$ and $a \cdot n \bmod m = n \bmod m$ and m and n are relative prime, then $a \bmod m = 1$.

(28) For every F such that $m \neq 0$ holds $F \bmod m \bmod m = F \bmod m$.

(29) For every F such that $m \neq 0$ holds $a \cdot (F \bmod m) \bmod m = a \cdot F \bmod m$.

(30) For all finite sequences F, G of elements of $\mathbb{N}$ such that $m \neq 0$ holds $F \frown G \bmod m = (F \bmod m) \frown (G \bmod m)$.

(31) For all finite sequences F, G of elements of $\mathbb{N}$ such that $m \neq 0$ holds $a \cdot (F \frown G) \bmod m = (a \cdot F \bmod m) \frown (a \cdot G \bmod m)$.

Let us consider n, k. Then $n_{\mathbb{N}}^{k}$ is a natural number.

We now state the proposition

(32) If $a \neq 0$ and $m \neq 0$ and a and m are relative prime, then for every b holds $a_{\mathbb{N}}^{b}$ and m are relative prime.

4. Euler's Theorem and Small Fermat's Theorem

The following propositions are true:

(33) If $a \neq 0$ and $m > 1$ and a and m are relative prime, then $(a_{\mathbb{N}}^{\mathrm{Euler}\, m}) \bmod m = 1$.

(34) If $a \neq 0$ and m is prime and a and m are relative prime, then $(a_{\mathbb{N}}^{m}) \bmod m = a \bmod m$.

Acknowledgments

The authors wish to thank Professor A. Trybulec for all of his advice on this 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] Czesław Byliński. Binary operations applied to finite sequences. *Formalized Mathematics*, 1(**4**):643–649, 1990.
[4] Czesław Byliński. The sum and product of finite sequences of real numbers. *Formalized Mathematics*, 1(**4**):661–668, 1990.
[5] 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.
[6] Yoshinori Fujisawa and Yasushi Fuwa. The Euler's function. *Formalized Mathematics*, 6(**4**):549–551, 1997.
[7] Andrzej Kondracki. The chinese remainder theorem. *Formalized Mathematics*, 6(**4**):573–577, 1997.
[8] Jarosław Kotowicz. Functions and finite sequences of real numbers. *Formalized Mathematics*, 3(**2**):275–278, 1992.
[9] Rafał Kwiatek and Grzegorz Zwara. The divisibility of integers and integer relative primes. *Formalized Mathematics*, 1(**5**):829–832, 1990.
[10] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 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.

Received June 10, 1998

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

The Product of the Families of the Groups

Artur Korniłowicz
University of Białystok

MML Identifier: GROUP_7.

The terminology and notation used here are introduced in the following articles: [6], [1], [4], [2], [3], [9], [10], [8], [12], [13], [11], [7], and [5].

1. Preliminaries

In this paper a, b, c, d, e, f are sets.

Next we state three propositions:

(1) If $\langle a \rangle = \langle b \rangle$, then $a = b$.

(2) If $\langle a, b \rangle = \langle c, d \rangle$, then $a = c$ and $b = d$.

(3) If $\langle a, b, c \rangle = \langle d, e, f \rangle$, then $a = d$ and $b = e$ and $c = f$.

2. The Product of the Families of the Groups

We use the following convention: i, I denote sets, f, g, h denote functions, and s denotes a many sorted set indexed by I.

Let R be a binary relation. We say that R is semigroup yielding if and only if:

(Def. 1) For every set y such that $y \in \operatorname{rng} R$ holds y is a non empty semigroup.

Let us note that every function which is semigroup yielding is also 1-sorted yielding.

Let I be a set. One can verify that there exists a many sorted set indexed by I which is semigroup yielding.

Let us observe that there exists a function which is semigroup yielding.

ISSN 1426–2630

Let I be a set. A family of semigroups indexed by I is a semigroup yielding many sorted set indexed by I.

Let I be a non empty set, let F be a family of semigroups indexed by I, and let i be an element of I. Then $F(i)$ is a non empty semigroup.

Let I be a set and let F be a family of semigroups indexed by I. One can verify that the support of F is non-empty.

Let I be a set and let F be a family of semigroups indexed by I. The functor $\prod F$ yielding a strict semigroup is defined by the conditions (Def. 2).

(Def. 2)(i) The carrier of $\prod F = \prod$ (the support of F), and

(ii) for all elements f, g of $\prod$ (the support of F) and for every set i such that $i \in I$ there exists a non empty semigroup F_1 and there exists a function h such that $F_1 = F(i)$ and $h =$ (the multiplication of $\prod F)(f, g)$ and $h(i) =$ (the multiplication of $F_1)(f(i), g(i))$.

Let I be a set and let F be a family of semigroups indexed by I. Note that $\prod F$ is non empty.

Let I be a set and let F be a family of semigroups indexed by I. Observe that every element of the carrier of $\prod F$ is function-like and relation-like.

Let I be a set, let F be a family of semigroups indexed by I, and let f, g be elements of $\prod$ (the support of F). Observe that (the multiplication of $\prod F)(f, g)$ is function-like and relation-like.

One can prove the following proposition

(4) Let F be a family of semigroups indexed by I, G be a non empty semigroup, p, q be elements of the carrier of $\prod F$, and x, y be elements of the carrier of G. Suppose $i \in I$ and $G = F(i)$ and $f = p$ and $g = q$ and $h = p \cdot q$ and $f(i) = x$ and $g(i) = y$. Then $x \cdot y = h(i)$.

Let I be a set and let F be a family of semigroups indexed by I. We say that F is group-like if and only if:

(Def. 3) For every set i such that $i \in I$ there exists a group-like non empty semigroup F_1 such that $F_1 = F(i)$.

We say that F is associative if and only if:

(Def. 4) For every set i such that $i \in I$ there exists an associative non empty semigroup F_1 such that $F_1 = F(i)$.

We say that F is commutative if and only if:

(Def. 5) For every set i such that $i \in I$ there exists a commutative non empty semigroup F_1 such that $F_1 = F(i)$.

Let I be a non empty set and let F be a family of semigroups indexed by I. Let us observe that F is group-like if and only if:

(Def. 6) For every element i of I holds $F(i)$ is group-like.

Let us observe that F is associative if and only if:

(Def. 7) For every element i of I holds $F(i)$ is associative.

Let us observe that F is commutative if and only if:

(Def. 8) For every element i of I holds $F(i)$ is commutative.

Let I be a set. Note that there exists a family of semigroups indexed by I which is group-like, associative, and commutative.

Let I be a set and let F be a group-like family of semigroups indexed by I. Note that $\prod F$ is group-like.

Let I be a set and let F be an associative family of semigroups indexed by I. One can check that $\prod F$ is associative.

Let I be a set and let F be a commutative family of semigroups indexed by I. One can verify that $\prod F$ is commutative.

We now state several propositions:

(5) Let F be a family of semigroups indexed by I and G be a non empty semigroup. If $i \in I$ and $G = F(i)$ and $\prod F$ is group-like, then G is group-like.

(6) Let F be a family of semigroups indexed by I and G be a non empty semigroup. If $i \in I$ and $G = F(i)$ and $\prod F$ is associative, then G is associative.

(7) Let F be a family of semigroups indexed by I and G be a non empty semigroup. If $i \in I$ and $G = F(i)$ and $\prod F$ is commutative, then G is commutative.

(8) Let F be a group-like family of semigroups indexed by I. Suppose that for every set i such that $i \in I$ there exists a group-like non empty semigroup G such that $G = F(i)$ and $s(i) = 1_G$. Then $s = 1_{\prod F}$.

(9) Let F be a group-like family of semigroups indexed by I and G be a group-like non empty semigroup. If $i \in I$ and $G = F(i)$ and $f = 1_{\prod F}$, then $f(i) = 1_G$.

(10) Let F be an associative group-like family of semigroups indexed by I and x be an element of the carrier of $\prod F$. Suppose that

(i) $x = g$, and

(ii) for every set i such that $i \in I$ there exists a group G and there exists an element y of the carrier of G such that $G = F(i)$ and $s(i) = y^{-1}$ and $y = g(i)$.

Then $s = x^{-1}$.

(11) Let F be an associative group-like family of semigroups indexed by I, x be an element of the carrier of $\prod F$, G be a group, and y be an element of the carrier of G. If $i \in I$ and $G = F(i)$ and $f = x$ and $g = x^{-1}$ and $f(i) = y$, then $g(i) = y^{-1}$.

Let I be a set and let F be an associative group-like family of semigroups indexed by I. The functor $\operatorname{sum} F$ yielding a strict subgroup of $\prod F$ is defined by the condition (Def. 9).

(Def. 9) Let x be a set. Then $x \in$ the carrier of $\operatorname{sum} F$ if and only if there exists an element g of $\prod$ (the support of F) and there exists a finite subset J of I and there exists a many sorted set f indexed by J such that $g = 1_{\prod F}$ and $x = g{+}{\cdot}f$ and for every set j such that $j \in J$ there exists a group-like non empty semigroup G such that $G = F(j)$ and $f(j) \in$ the carrier of G and $f(j) \neq 1_G$.

Let I be a set, let F be an associative group-like family of semigroups indexed by I, and let f, g be elements of the carrier of $\operatorname{sum} F$. One can check that (the multiplication of $\operatorname{sum} F)(f, g)$ is function-like and relation-like.

The following proposition is true

(12) For every finite set I and for every associative group-like family F of semigroups indexed by I holds $\prod F = \operatorname{sum} F$.

3. The Product of One, Two and Three Groups

One can prove the following proposition

(13) For every non empty semigroup G_1 holds $\langle G_1 \rangle$ is a family of semigroups indexed by $\{1\}$.

Let G_1 be a non empty semigroup. Then $\langle G_1 \rangle$ is a family of semigroups indexed by $\{1\}$.

We now state the proposition

(14) For every group-like non empty semigroup G_1 holds $\langle G_1 \rangle$ is a group-like family of semigroups indexed by $\{1\}$.

Let G_1 be a group-like non empty semigroup. Then $\langle G_1 \rangle$ is a group-like family of semigroups indexed by $\{1\}$.

Next we state the proposition

(15) For every associative non empty semigroup G_1 holds $\langle G_1 \rangle$ is an associative family of semigroups indexed by $\{1\}$.

Let G_1 be an associative non empty semigroup. Then $\langle G_1 \rangle$ is an associative family of semigroups indexed by $\{1\}$.

The following proposition is true

(16) For every commutative non empty semigroup G_1 holds $\langle G_1 \rangle$ is a commutative family of semigroups indexed by $\{1\}$.

Let G_1 be a commutative non empty semigroup. Then $\langle G_1 \rangle$ is a commutative family of semigroups indexed by $\{1\}$.

We now state the proposition

(17) For every group G_1 holds $\langle G_1 \rangle$ is a group-like associative family of semigroups indexed by $\{1\}$.

Let G_1 be a group. Then $\langle G_1 \rangle$ is a group-like associative family of semigroups indexed by $\{1\}$.

Next we state the proposition

(18) Let G_1 be a commutative group. Then $\langle G_1 \rangle$ is a commutative group-like associative family of semigroups indexed by $\{1\}$.

Let G_1 be a commutative group. Then $\langle G_1 \rangle$ is a group-like associative commutative family of semigroups indexed by $\{1\}$.

Let G_1 be a non empty semigroup. Note that every element of $\prod$ the support of $\langle G_1 \rangle$ is finite sequence-like.

Let G_1 be a non empty semigroup. Note that every element of the carrier of $\prod \langle G_1 \rangle$ is finite sequence-like.

Let G_1 be a non empty semigroup and let x be an element of the carrier of G_1. Then $\langle x \rangle$ is an element of $\prod \langle G_1 \rangle$.

One can prove the following proposition

(19) For all non empty semigroups G_1, G_2 holds $\langle G_1, G_2 \rangle$ is a family of semigroups indexed by $\{1, 2\}$.

Let G_1, G_2 be non empty semigroups. Then $\langle G_1, G_2 \rangle$ is a family of semigroups indexed by $\{1, 2\}$.

One can prove the following proposition

(20) For all group-like non empty semigroups G_1, G_2 holds $\langle G_1, G_2 \rangle$ is a group-like family of semigroups indexed by $\{1, 2\}$.

Let G_1, G_2 be group-like non empty semigroups. Then $\langle G_1, G_2 \rangle$ is a group-like family of semigroups indexed by $\{1, 2\}$.

Next we state the proposition

(21) For all associative non empty semigroups G_1, G_2 holds $\langle G_1, G_2 \rangle$ is an associative family of semigroups indexed by $\{1, 2\}$.

Let G_1, G_2 be associative non empty semigroups. Then $\langle G_1, G_2 \rangle$ is an associative family of semigroups indexed by $\{1, 2\}$.

One can prove the following proposition

(22) For all commutative non empty semigroups G_1, G_2 holds $\langle G_1, G_2 \rangle$ is a commutative family of semigroups indexed by $\{1, 2\}$.

Let G_1, G_2 be commutative non empty semigroups. Then $\langle G_1, G_2 \rangle$ is a commutative family of semigroups indexed by $\{1, 2\}$.

The following proposition is true

(23) For all groups G_1, G_2 holds $\langle G_1, G_2 \rangle$ is a group-like associative family of semigroups indexed by $\{1, 2\}$.

Let G_1, G_2 be groups. Then $\langle G_1, G_2 \rangle$ is a group-like associative family of semigroups indexed by $\{1, 2\}$.

Next we state the proposition

(24) Let G_1, G_2 be commutative groups. Then $\langle G_1, G_2\rangle$ is a group-like associative commutative family of semigroups indexed by $\{1,2\}$.

Let G_1, G_2 be commutative groups. Then $\langle G_1, G_2\rangle$ is a group-like associative commutative family of semigroups indexed by $\{1,2\}$.

Let G_1, G_2 be non empty semigroups. Note that every element of $\prod$ the support of $\langle G_1, G_2\rangle$ is finite sequence-like.

Let G_1, G_2 be non empty semigroups. Note that every element of the carrier of $\prod\langle G_1, G_2\rangle$ is finite sequence-like.

Let G_1, G_2 be non empty semigroups, let x be an element of the carrier of G_1, and let y be an element of the carrier of G_2. Then $\langle x, y\rangle$ is an element of $\prod\langle G_1, G_2\rangle$.

One can prove the following proposition

(25) For all non empty semigroups G_1, G_2, G_3 holds $\langle G_1, G_2, G_3\rangle$ is a family of semigroups indexed by $\{1,2,3\}$.

Let G_1, G_2, G_3 be non empty semigroups. Then $\langle G_1, G_2, G_3\rangle$ is a family of semigroups indexed by $\{1,2,3\}$.

Next we state the proposition

(26) For all group-like non empty semigroups G_1, G_2, G_3 holds $\langle G_1, G_2, G_3\rangle$ is a group-like family of semigroups indexed by $\{1,2,3\}$.

Let G_1, G_2, G_3 be group-like non empty semigroups. Then $\langle G_1, G_2, G_3\rangle$ is a group-like family of semigroups indexed by $\{1,2,3\}$.

Next we state the proposition

(27) Let G_1, G_2, G_3 be associative non empty semigroups. Then $\langle G_1, G_2, G_3\rangle$ is an associative family of semigroups indexed by $\{1,2,3\}$.

Let G_1, G_2, G_3 be associative non empty semigroups. Then $\langle G_1, G_2, G_3\rangle$ is an associative family of semigroups indexed by $\{1,2,3\}$.

One can prove the following proposition

(28) Let G_1, G_2, G_3 be commutative non empty semigroups. Then $\langle G_1, G_2, G_3\rangle$ is a commutative family of semigroups indexed by $\{1,2,3\}$.

Let G_1, G_2, G_3 be commutative non empty semigroups. Then $\langle G_1, G_2, G_3\rangle$ is a commutative family of semigroups indexed by $\{1,2,3\}$.

Next we state the proposition

(29) For all groups G_1, G_2, G_3 holds $\langle G_1, G_2, G_3\rangle$ is a group-like associative family of semigroups indexed by $\{1,2,3\}$.

Let G_1, G_2, G_3 be groups. Then $\langle G_1, G_2, G_3\rangle$ is a group-like associative family of semigroups indexed by $\{1,2,3\}$.

One can prove the following proposition

(30) Let G_1, G_2, G_3 be commutative groups. Then $\langle G_1, G_2, G_3\rangle$ is a group-like associative commutative family of semigroups indexed by $\{1,2,3\}$.

Let G_1, G_2, G_3 be commutative groups. Then $\langle G_1, G_2, G_3\rangle$ is a group-like associative commutative family of semigroups indexed by $\{1, 2, 3\}$.

Let G_1, G_2, G_3 be non empty semigroups. Observe that every element of $\prod$ the support of $\langle G_1, G_2, G_3\rangle$ is finite sequence-like.

Let G_1, G_2, G_3 be non empty semigroups. Note that every element of the carrier of $\prod\langle G_1, G_2, G_3\rangle$ is finite sequence-like.

Let G_1, G_2, G_3 be non empty semigroups, let x be an element of the carrier of G_1, let y be an element of the carrier of G_2, and let z be an element of the carrier of G_3. Then $\langle x, y, z\rangle$ is an element of $\prod\langle G_1, G_2, G_3\rangle$.

For simplicity, we adopt the following rules: G_1, G_2, G_3 denote non empty semigroups, x_1, x_2 denote elements of the carrier of G_1, y_1, y_2 denote elements of the carrier of G_2, and z_1, z_2 denote elements of the carrier of G_3.

One can prove the following propositions:

(31) $\langle x_1\rangle \cdot \langle x_2\rangle = \langle x_1 \cdot x_2\rangle$.

(32) $\langle x_1, y_1\rangle \cdot \langle x_2, y_2\rangle = \langle x_1 \cdot x_2, y_1 \cdot y_2\rangle$.

(33) $\langle x_1, y_1, z_1\rangle \cdot \langle x_2, y_2, z_2\rangle = \langle x_1 \cdot x_2, y_1 \cdot y_2, z_1 \cdot z_2\rangle$.

In the sequel G_1, G_2, G_3 denote group-like non empty semigroups.

We now state three propositions:

(34) $1_{\prod\langle G_1\rangle} = \langle 1_{(G_1)}\rangle$.

(35) $1_{\prod\langle G_1, G_2\rangle} = \langle 1_{(G_1)}, 1_{(G_2)}\rangle$.

(36) $1_{\prod\langle G_1, G_2, G_3\rangle} = \langle 1_{(G_1)}, 1_{(G_2)}, 1_{(G_3)}\rangle$.

For simplicity, we adopt the following rules: G_1, G_2, G_3 are groups, x is an element of the carrier of G_1, y is an element of the carrier of G_2, and z is an element of the carrier of G_3.

The following propositions are true:

(37) $(\langle x\rangle$ **qua** element of the carrier of $\prod\langle G_1\rangle)^{-1} = \langle x^{-1}\rangle$.

(38) $(\langle x, y\rangle$ **qua** element of the carrier of $\prod\langle G_1, G_2\rangle)^{-1} = \langle x^{-1}, y^{-1}\rangle$.

(39) $(\langle x, y, z\rangle$ **qua** element of the carrier of $\prod\langle G_1, G_2, G_3\rangle)^{-1} = \langle x^{-1}, y^{-1}, z^{-1}\rangle$.

(40) Let f be a function from the carrier of G_1 into the carrier of $\prod\langle G_1\rangle$. Suppose that for every element x of the carrier of G_1 holds $f(x) = \langle x\rangle$. Then f is a homomorphism from G_1 to $\prod\langle G_1\rangle$.

(41) Let f be a homomorphism from G_1 to $\prod\langle G_1\rangle$. Suppose that for every element x of the carrier of G_1 holds $f(x) = \langle x\rangle$. Then f is an isomorphism.

(42) G_1 and $\prod\langle G_1\rangle$ are isomorphic.

References

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

[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] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[5] Eugeniusz Kusak, Wojciech Leończuk, and Michał Muzalewski. Abelian groups, fields and vector spaces. *Formalized Mathematics*, 1(**2**):335–342, 1990.
[6] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[7] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[8] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[9] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.
[10] Wojciech A. Trybulec and Michał J. Trybulec. Homomorphisms and isomorphisms of groups. Quotient group. *Formalized Mathematics*, 2(**4**):573–578, 1991.
[11] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[12] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[13] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received June 10, 1998

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

On the Dividing Function of the Simple Closed Curve into Segments

Yatsuka Nakamura
Shinshu University
Nagano

Summary. At the beginning, the concept of the segment of the simple closed curve in 2-dimensional Euclidean space is defined. Some properties of segments are shown in the succeeding theorems. At the end, the existence of the function which can divide the simple closed curve into segments is shown. We can make the diameter of segments as small as we want.

MML Identifier: `JORDAN7`.

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

1. Definition of the Segment and Its Property

In this paper p, p_1, q are points of $\mathcal{E}^2_{\mathrm{T}}$.

The following three propositions are true:

(1) Let P be a compact non empty subset of $\mathcal{E}^2_{\mathrm{T}}$. Suppose P is a simple closed curve. Then $\text{W-min}\,P \in \text{LowerArc}\,P$ and $\text{E-max}\,P \in \text{LowerArc}\,P$ and $\text{W-min}\,P \in \text{UpperArc}\,P$ and $\text{E-max}\,P \in \text{UpperArc}\,P$.

(2) For every compact non empty subset P of $\mathcal{E}^2_{\mathrm{T}}$ and for every q such that P is a simple closed curve and $\mathrm{LE}(q, \text{W-min}\,P, P)$ holds $q = \text{W-min}\,P$.

(3) For every compact non empty subset P of $\mathcal{E}^2_{\mathrm{T}}$ and for every q such that P is a simple closed curve and $q \in P$ holds $\mathrm{LE}(\text{W-min}\,P, q, P)$.

ISSN 1426–2630

Let P be a compact non empty subset of $\mathcal{E}^2_{\mathrm{T}}$ and let q_1, q_2 be points of $\mathcal{E}^2_{\mathrm{T}}$. The functor $\mathrm{Segment}(q_1, q_2, P)$ yields a subset of $\mathcal{E}^2_{\mathrm{T}}$ and is defined by:

(Def. 1) $$\mathrm{Segment}(q_1, q_2, P) = \begin{cases} \{p : \mathrm{LE}(q_1, p, P) \ \wedge \ \mathrm{LE}(p, q_2, P)\}, \\ \text{ if } q_2 \neq \text{W-min}\, P, \\ \{p_1 : \mathrm{LE}(q_1, p_1, P) \ \vee \ q_1 \in P \ \wedge \ p_1 = \text{W-min}\, P\}, \\ \text{ otherwise.} \end{cases}$$

One can prove the following propositions:

(4) For every compact non empty subset P of $\mathcal{E}^2_{\mathrm{T}}$ such that P is a simple closed curve holds $\mathrm{Segment}(\text{W-min}\, P, \text{E-max}\, P, P) = \mathrm{UpperArc}\, P$ and $\mathrm{Segment}(\text{E-max}\, P, \text{W-min}\, P, P) = \mathrm{LowerArc}\, P$.

(5) Let P be a compact non empty subset of $\mathcal{E}^2_{\mathrm{T}}$ and q_1, q_2 be points of $\mathcal{E}^2_{\mathrm{T}}$. If P is a simple closed curve and $\mathrm{LE}(q_1, q_2, P)$, then $q_1 \in P$ and $q_2 \in P$.

(6) Let P be a compact non empty subset of $\mathcal{E}^2_{\mathrm{T}}$ and q_1, q_2 be points of $\mathcal{E}^2_{\mathrm{T}}$. If P is a simple closed curve and $\mathrm{LE}(q_1, q_2, P)$, then $q_1 \in \mathrm{Segment}(q_1, q_2, P)$ and $q_2 \in \mathrm{Segment}(q_1, q_2, P)$.

(7) Let P be a compact non empty subset of $\mathcal{E}^2_{\mathrm{T}}$ and q be a point of $\mathcal{E}^2_{\mathrm{T}}$. If P is a simple closed curve and $q \in P$ and $q \neq \text{W-min}\, P$, then $\mathrm{Segment}(q, q, P) = \{q\}$.

(8) Let P be a compact non empty subset of $\mathcal{E}^2_{\mathrm{T}}$ and q_1, q_2 be points of $\mathcal{E}^2_{\mathrm{T}}$. If P is a simple closed curve and $q_1 \neq \text{W-min}\, P$ and $q_2 \neq \text{W-min}\, P$, then $\text{W-min}\, P \notin \mathrm{Segment}(q_1, q_2, P)$.

(9) Let P be a compact non empty subset of $\mathcal{E}^2_{\mathrm{T}}$ and q_1, q_2, q_3 be points of $\mathcal{E}^2_{\mathrm{T}}$. Suppose P is a simple closed curve and $\mathrm{LE}(q_1, q_2, P)$ and $\mathrm{LE}(q_2, q_3, P)$ and $q_1 = q_2$ and $q_1 = \text{W-min}\, P$ and $q_1 \neq q_3$ and $q_2 = q_3$ and $q_2 = \text{W-min}\, P$. Then $\mathrm{Segment}(q_1, q_2, P) \cap \mathrm{Segment}(q_2, q_3, P) = \{q_2\}$.

(10) Let P be a compact non empty subset of $\mathcal{E}^2_{\mathrm{T}}$ and q_1, q_2 be points of $\mathcal{E}^2_{\mathrm{T}}$. Suppose P is a simple closed curve and $\mathrm{LE}(q_1, q_2, P)$ and $q_1 \neq q_2$ and $q_1 \neq \text{W-min}\, P$. Then $\mathrm{Segment}(q_2, \text{W-min}\, P, P) \cap \mathrm{Segment}(\text{W-min}\, P, q_1, P) = \{\text{W-min}\, P\}$.

(11) Let P be a compact non empty subset of $\mathcal{E}^2_{\mathrm{T}}$ and q_1, q_2, q_3, q_4 be points of $\mathcal{E}^2_{\mathrm{T}}$. Suppose P is a simple closed curve and $\mathrm{LE}(q_1, q_2, P)$ and $\mathrm{LE}(q_2, q_3, P)$ and $\mathrm{LE}(q_3, q_4, P)$ and $q_1 \neq q_2$ and $q_2 \neq q_3$. Then $\mathrm{Segment}(q_1, q_2, P) \cap \mathrm{Segment}(q_3, q_4, P) = \emptyset$.

2. A Function to Divide the Simple Closed Curve

In the sequel n is a natural number.

We now state three propositions:

(12) Let P be a non empty subset of the carrier of $\mathcal{E}^n_{\mathrm{T}}$ and f be a map from $\mathbb{I}$ into $(\mathcal{E}^n_{\mathrm{T}})\restriction P$. Suppose $P \neq \emptyset$ and f is a homeomorphism. Then there exists a map g from $\mathbb{I}$ into $\mathcal{E}^n_{\mathrm{T}}$ such that $f = g$ and g is continuous and one-to-one.

(13) For every finite sequence f of elements of $\mathbb{R}$ such that f is increasing holds f is one-to-one.

(14) Let P be a compact non empty subset of $\mathcal{E}^2_{\mathrm{T}}$ and e be a real number. Suppose P is a simple closed curve and $e > 0$. Then there exists a finite sequence h of elements of the carrier of $\mathcal{E}^2_{\mathrm{T}}$ such that

(i) $h(1) = \text{W-min}\, P$,

(ii) h is one-to-one,

(iii) $8 \leqslant \operatorname{len} h$,

(iv) $\operatorname{rng} h \subseteq P$,

(v) for every natural number i such that $1 \leqslant i$ and $i < \operatorname{len} h$ holds $\mathrm{LE}(\pi_i h, \pi_{i+1} h, P)$,

(vi) for every natural number i and for every subset W of the carrier of $\mathcal{E}^2$ such that $1 \leqslant i$ and $i < \operatorname{len} h$ and $W = \mathrm{Segment}(\pi_i h, \pi_{i+1} h, P)$ holds $\varnothing W < e$,

(vii) for every subset W of the carrier of $\mathcal{E}^2$ such that $W = \mathrm{Segment}(\pi_{\operatorname{len} h} h, \pi_1 h, P)$ holds $\varnothing W < e$,

(viii) for every natural number i such that $1 \leqslant i$ and $i + 1 < \operatorname{len} h$ holds $\mathrm{Segment}(\pi_i h, \pi_{i+1} h, P) \cap \mathrm{Segment}(\pi_{i+1} h, \pi_{i+2} h, P) = \{\pi_{i+1} h\}$,

(ix) $\mathrm{Segment}(\pi_{\operatorname{len} h} h, \pi_1 h, P) \cap \mathrm{Segment}(\pi_1 h, \pi_2 h, P) = \{\pi_1 h\}$, and

(x) for all natural numbers i, j such that $1 \leqslant i$ and $i < \operatorname{len} h$ and $1 \leqslant j$ and $j < \operatorname{len} h$ and $i \neq j$ and i and j are not adjacent holds $\mathrm{Segment}(\pi_i h, \pi_{i+1} h, P) \cap \mathrm{Segment}(\pi_j h, \pi_{j+1} h, P) = \emptyset$.

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

[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. The Euclidean space. *Formalized Mathematics*, 2(**4**):599–603, 1991.
[8] Agata Darmochwał and Yatsuka Nakamura. Metric spaces as topological spaces - fundamental concepts. *Formalized Mathematics*, 2(**4**):605–608, 1991.
[9] Agata Darmochwał and Yatsuka Nakamura. The topological space $\mathcal{E}^2_{\rm T}$. Simple closed curves. *Formalized Mathematics*, 2(**5**):663–664, 1991.
[10] Alicia de la Cruz. Totally bounded metric spaces. *Formalized Mathematics*, 2(**4**):559–562, 1991.
[11] Jarosław Kotowicz and Yatsuka Nakamura. Introduction to Go-board - part I. *Formalized Mathematics*, 3(**1**):107–115, 1992.
[12] Yatsuka Nakamura and Andrzej Trybulec. Adjacency concept for pairs of natural numbers. *Formalized Mathematics*, 6(**1**):1–3, 1997.
[13] Yatsuka Nakamura and Andrzej Trybulec. A decomposition of a simple closed curves and the order of their points. *Formalized Mathematics*, 6(**4**):563–572, 1997.
[14] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[15] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[16] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[17] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[18] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received June 16, 1998

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

Initialization Halting Concepts and Their Basic Properties of $\mathbf{SCM}_{\mathrm{FSA}}$

Jing-Chao Chen
Shanghai Jiaotong University

Yatsuka Nakamura
Shinshu University
Nagano

Summary. Up to now, many properties of macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$ are described by the parahalting concepts. However, many practical programs are not always halting while they are halting for initialization states. For this reason, we propose initialization halting concepts. That a program is initialization halting (called "InitHalting" for short) means it is halting for initialization states.In order to make the halting proof of more complicated programs easy, we present "InitHalting" basic properties of the compositions of the macro instructions, if-Macro (conditional branch macro instructions) and Times-Macro (for-loop macro instructions) etc.

MML Identifier: `SCM_HALT`.

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

1. The Definition of Several Notions Related to Initialization

For simplicity, we adopt the following rules: m is a natural number, I is a macro instruction, s, s_1, s_2 are states of $\mathbf{SCM}_{\mathrm{FSA}}$, a is an integer location, and f is a finite sequence location.

Let I be a macro instruction. We say that I is InitClosed if and only if:

(Def. 1) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every natural number n such that $\mathrm{Initialized}(I) \subseteq s$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(n)} \in \mathrm{dom}\, I$.

We say that I is InitHalting if and only if:

ISSN 1426–2630

(Def. 2) Initialized(I) is halting.

We say that I is keepInt0 1 if and only if:

(Def. 3) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ such that Initialized(I) $\subseteq s$ and for every natural number k holds (Computation(s))(k)(intloc(0)) $= 1$.

2. The Relationship Between Initialization Halting and Unconditional Halting

The following four propositions are true:

(1) For every set x and for all natural numbers i, m, n such that $x \in$ dom((intloc(i)$\dot{\longmapsto}m$)$+\cdot$ Start-At(insloc(n))) holds $x =$ intloc(i) or $x = \mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(2) For every macro instruction I and for all natural numbers i, m, n holds dom $I \cap$ dom((intloc(i)$\dot{\longmapsto}m$)$+\cdot$ Start-At(insloc(n))) $= \emptyset$.

(3) Initialized(I) $= I+\cdot$((intloc(0)$\dot{\longmapsto}$1)$+\cdot$ Start-At(insloc(0))).

(4) Macro($\mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$) is InitHalting.

Let us mention that there exists a macro instruction which is InitHalting.

One can prove the following three propositions:

(5) For every InitHalting macro instruction I such that Initialized(I) $\subseteq s$ holds s is halting.

(6) $I+\cdot$ Start-At(insloc(0)) $\subseteq$ Initialized(I).

(7) For every macro instruction I and for every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ such that Initialized(I) $\subseteq s$ holds s(intloc(0)) $= 1$.

Let us mention that every macro instruction which is paraclosed is also InitClosed.

Let us note that every macro instruction which is parahalting is also InitHalting.

One can check the following observations:

* every macro instruction which is InitHalting is also InitClosed,
* every macro instruction which is keepInt0 1 is also InitClosed, and
* every macro instruction which is keeping 0 is also keepInt0 1.

3. The Other Properties of Initialization Halting

One can prove the following two propositions:

(8) Let I be a InitHalting macro instruction and a be a read-write integer location. If $a \notin$ UsedIntLoc(I), then (IExec(I, s))(a) $= s(a)$.

(9) Let I be a InitHalting macro instruction and f be a finite sequence location. If $f \notin \text{UsedInt}^* \text{Loc}(I)$, then $(\text{IExec}(I, s))(f) = s(f)$.

Let I be a InitHalting macro instruction. Note that Initialized(I) is halting.

Let us observe that every macro instruction which is InitHalting is also non empty.

The following propositions are true:

(10) For every InitHalting macro instruction I holds $\text{dom}\, I \neq \emptyset$.

(11) For every InitHalting macro instruction I holds $\text{insloc}(0) \in \text{dom}\, I$.

(12) Let J be a InitHalting macro instruction. Suppose $\text{Initialized}(J) \subseteq s_1$. Let n be a natural number. Suppose $\text{ProgramPart}(\text{Relocated}(J, n)) \subseteq s_2$ and $\mathbf{IC}_{(s_2)} = \text{insloc}(n)$ and $s_1 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = s_2 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$. Let i be a natural number. Then $\mathbf{IC}_{(\text{Computation}(s_1))(i)} + n = \mathbf{IC}_{(\text{Computation}(s_2))(i)}$ and IncAddr(CurInstr $((\text{Computation}(s_1))(i)), n) = \text{CurInstr}((\text{Computation}(s_2))(i))$ and $(\text{Computation}(s_1))(i) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = (\text{Computation}(s_2))(i) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(13) If $\text{Initialized}(I) \subseteq s$, then $I \subseteq s$.

(14) Let I be a InitHalting macro instruction. Suppose $\text{Initialized}(I) \subseteq s_1$ and $\text{Initialized}(I) \subseteq s_2$ and s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\text{FSA}}$. Let k be a natural number. Then $(\text{Computation}(s_1))(k)$ and $(\text{Computation}(s_2))(k)$ are equal outside the instruction locations of $\mathbf{SCM}_{\text{FSA}}$ and $\text{CurInstr}((\text{Computation}(s_1))(k)) = \text{CurInstr}((\text{Computation}(s_2))(k))$.

(15) Let I be a InitHalting macro instruction. Suppose $\text{Initialized}(I) \subseteq s_1$ and $\text{Initialized}(I) \subseteq s_2$ and s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\text{FSA}}$. Then $\text{LifeSpan}(s_1) = \text{LifeSpan}(s_2)$ and $\text{Result}(s_1)$ and $\text{Result}(s_2)$ are equal outside the instruction locations of $\mathbf{SCM}_{\text{FSA}}$.

(16) $\text{Macro}(\mathbf{halt}_{\mathbf{SCM}_{\text{FSA}}})$ is keeping 0 and InitHalting.

Let us observe that there exists a macro instruction which is keeping 0 and InitHalting.

One can verify that there exists a macro instruction which is keepInt0 1 and InitHalting.

Next we state several propositions:

(17) For every keepInt0 1 InitHalting macro instruction I holds $(\text{IExec}(I, s))(\text{intloc}(0)) = 1$.

(18) Let I be a InitClosed macro instruction and J be a macro instruction. Suppose $\text{Initialized}(I) \subseteq s$ and s is halting. Let given m. Suppose $m \leqslant \text{LifeSpan}(s)$. Then $(\text{Computation}(s))(m)$ and $(\text{Computation}(s {+\cdot} (I;J)))(m)$ are equal outside the instruction locations of $\mathbf{SCM}_{\text{FSA}}$.

(19) For all natural numbers i, m, n holds $s+\cdot I+\cdot((\mathrm{intloc}(i)\dot{\longmapsto}m)+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(n))) = (s+\cdot((\mathrm{intloc}(i)\dot{\longmapsto}m)+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(n))))+\cdot I$.

(20) If $(\mathrm{intloc}(0)\dot{\longmapsto}1)+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)) \subseteq s$, then $\mathrm{Initialized}(I) \subseteq s+\cdot(I+\cdot((\mathrm{intloc}(0)\dot{\longmapsto}1)+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0))))$ and $s+\cdot(I+\cdot((\mathrm{intloc}(0)\dot{\longmapsto}1)+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))) = s+\cdot I$ and $s+\cdot(I+\cdot((\mathrm{intloc}(0)\dot{\longmapsto}1)+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0))))+\cdot\,\mathrm{Directed}(I) = s+\cdot\,\mathrm{Directed}(I)$.

(21) For every InitClosed macro instruction I such that $s+\cdot I$ is halting and $\mathrm{Directed}(I) \subseteq s$ and $(\mathrm{intloc}(0)\dot{\longmapsto}1)+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)) \subseteq s$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s+\cdot I)+1)} = \mathrm{insloc}(\mathrm{card}\,I)$.

(22) Let I be a InitClosed macro instruction. Suppose $s+\cdot I$ is halting and $\mathrm{Directed}(I) \subseteq s$ and $(\mathrm{intloc}(0)\dot{\longmapsto}1)+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)) \subseteq s$. Then $(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s+\cdot I))\restriction(\mathrm{Int\text{-}Locations}\cup\mathrm{FinSeq\text{-}Locations}) = (\mathrm{Computation}(s))(\mathrm{LifeSpan}(s+\cdot I)+1)\restriction(\mathrm{Int\text{-}Locations}\cup\mathrm{FinSeq\text{-}Locations})$.

(23) Let I be a InitHalting macro instruction. Suppose $\mathrm{Initialized}(I) \subseteq s$. Let k be a natural number. If $k \leqslant \mathrm{LifeSpan}(s)$, then $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot\,\mathrm{Directed}(I)))(k)) \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(24) Let I be a InitClosed macro instruction. Suppose $s+\cdot\,\mathrm{Initialized}(I)$ is halting. Let J be a macro instruction and k be a natural number. Suppose $k \leqslant \mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(I))$. Then $(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(I)))(k)$ and $(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(I;J)))(k)$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

4. The Initialization Halting for Two Continuous Macro-Instructions

One can prove the following proposition

(25) Let I be a keepInt0 1 InitHalting macro instruction, J be a InitHalting macro instruction, and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $\mathrm{Initialized}(I;J) \subseteq s$. Then

(i) $\mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s+\cdot I)+1)} = \mathrm{insloc}(\mathrm{card}\,I)$,

(ii) $(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s+\cdot I)+1)\restriction(\mathrm{Int\text{-}Locations}\cup\mathrm{FinSeq\text{-}Locations}) = ((\mathrm{Computation}(s+\cdot I))(\mathrm{LifeSpan}(s+\cdot I))+\cdot\,\mathrm{Initialized}(J))\restriction(\mathrm{Int\text{-}Locations}\cup\mathrm{FinSeq\text{-}Locations})$,

(iii) $\mathrm{ProgramPart}(\mathrm{Relocated}(J,\mathrm{card}\,I)) \subseteq (\mathrm{Computation}(s))(\mathrm{LifeSpan}(s+\cdot I)+1)$,

(iv) $(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s+\cdot I)+1)(\mathrm{intloc}(0)) = 1$,

(v) s is halting,

(vi) $\mathrm{LifeSpan}(s) = \mathrm{LifeSpan}(s+\cdot I)+1+\mathrm{LifeSpan}(\mathrm{Result}(s+\cdot I)+\cdot\,\mathrm{Initialized}(J))$, and

(vii) if J is keeping 0, then $(\mathrm{Result}(s))(\mathrm{intloc}(0)) = 1$.

Let I be a keepInt0 1 InitHalting macro instruction and let J be a InitHalting macro instruction. Note that I;J is InitHalting.

Next we state four propositions:

(26) Let I be a keepInt0 1 macro instruction. Suppose $s+\cdot I$ is halting. Let J be a InitClosed macro instruction. Suppose Initialized(I;J) $\subseteq s$. Let k be a natural number. Then (Computation(Result($s+\cdot I$)$+\cdot$ Initialized(J)))(k) $+\cdot$ Start-At($\mathbf{IC}_{(\mathrm{Computation}(\mathrm{Result}(s+\cdot I)+\cdot\,\mathrm{Initialized}(J)))(k)} + \mathrm{card}\, I$) and (Computation($s+\cdot(I;J)$))(LifeSpan($s+\cdot I$) $+ 1 + k$) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(27) Let I be a keepInt0 1 macro instruction. Suppose $s+\cdot$ Initialized(I) is not halting. Let J be a macro instruction and k be a natural number. Then (Computation($s+\cdot$ Initialized(I)))(k) and (Computation($s+\cdot$ Initialized (I;J)))(k) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(28) Let I be a keepInt0 1 InitHalting macro instruction and J be a InitHalting macro instruction. Then LifeSpan($s+\cdot$ Initialized(I;J)) $=$ LifeSpan($s+\cdot$ Initialized(I)) $+ 1 +$ LifeSpan(Result($s+\cdot$ Initialized(I)) $+\cdot$ Initialized(J)).

(29) Let I be a keepInt0 1 InitHalting macro instruction and J be a InitHalting macro instruction. Then IExec(I;J, s) $=$ IExec(J, IExec(I, s)) $+\cdot$ Start-At($\mathbf{IC}_{\mathrm{IExec}(J,\mathrm{IExec}(I,s))} + \mathrm{card}\, I$).

Let i be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Observe that Macro($i$) is InitHalting.

Let i be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let J be a parahalting macro instruction. Observe that i;J is InitHalting.

Let i be a keeping 0 parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let J be a InitHalting macro instruction. Note that i;J is InitHalting.

Let I, J be keepInt0 1 macro instructions. One can verify that I;J is keepInt0 1.

Let j be a keeping 0 parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let I be a keepInt0 1 InitHalting macro instruction. One can check that I;j is InitHalting and keepInt0 1.

Let i be a keeping 0 parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let J be a keepInt0 1 InitHalting macro instruction. Observe that i;J is InitHalting and keepInt0 1.

Let j be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let I be a parahalting macro instruction. One can check that I;j is InitHalting.

Let i, j be parahalting instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. One can check that i;j is InitHalting.

Next we state several propositions:

(30) Let I be a keepInt0 1 InitHalting macro instruction and J be a InitHalting macro instruction. Then (IExec(I;J, s))(a) $=$

(IExec(J, IExec(I, s)))(a).

(31) Let I be a keepInt0 1 InitHalting macro instruction and J be a InitHalting macro instruction. Then (IExec(I;J, s))(f) = (IExec(J, IExec(I, s)))(f).

(32) For every keepInt0 1 InitHalting macro instruction I and for every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ holds Initialize(IExec(I, s))$\restriction$(Int-Locations $\cup$ FinSeq-Locations) = IExec(I, s)$\restriction$(Int-Locations $\cup$ FinSeq-Locations).

(33) Let I be a keepInt0 1 InitHalting macro instruction and j be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Then (IExec($I$;$j$, s))(a) = (Exec(j, IExec(I, s)))(a).

(34) Let I be a keepInt0 1 InitHalting macro instruction and j be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Then (IExec($I$;$j$, s))(f) = (Exec(j, IExec(I, s)))(f).

Let I be a macro instruction and let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. We say that I is closed onInit s if and only if:

(Def. 4) For every natural number k holds $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(I)))(k)} \in \mathrm{dom}\, I$.

We say that I is halting onInit s if and only if:

(Def. 5) $s+\cdot$ Initialized(I) is halting.

We now state three propositions:

(35) Let I be a macro instruction. Then I is InitClosed if and only if for every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ holds I is closed onInit s.

(36) Let I be a macro instruction. Then I is InitHalting if and only if for every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ holds I is halting onInit s.

(37) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I be a macro instruction, and a be an integer location. Suppose I does not destroy a and I is closed onInit s and Initialized(I) $\subseteq$ s. Let k be a natural number. Then (Computation(s))(k)(a) = s(a).

Let us observe that there exists a macro instruction which is InitHalting and good.

Let us observe that every macro instruction which is InitClosed and good is also keepInt0 1.

Let us mention that $\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}}$ is InitHalting and good.

We now state several propositions:

(38) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, i be a keeping 0 parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, J be a InitHalting macro instruction, and a be an integer location. Then (IExec(i;J, s))(a) = (IExec(J, Exec(i, Initialize(s))))(a).

(39) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, i be a keeping 0 parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, J be a InitHalting macro instruction, and f be a finite sequence location. Then (IExec(i;J, s))(f) = (IExec(J, Exec(i, Initialize(s))))(f).

(40) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. Then I is closed onInit s if and only if I is closed on Initialize(s).

(41) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. Then I is halting onInit s if and only if I is halting on Initialize(s).

(42) For every macro instruction I and for every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{IExec}(I, s) = \mathrm{IExec}(I, \mathrm{Initialize}(s))$.

5. IF-Programs with Initialization Halting

The following propositions are true:

(43) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I, J be macro instructions, and a be a read-write integer location. Suppose $s(a) = 0$ and I is closed onInit s and I is halting onInit s. Then **if** $a = 0$ **then** I **else** J is closed onInit s and **if** $a = 0$ **then** I **else** J is halting onInit s.

(44) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I, J be macro instructions, and a be a read-write integer location. Suppose $s(a) = 0$ and I is closed onInit s and I is halting onInit s. Then $\mathrm{IExec}(\mathbf{if}\ a = 0\ \mathbf{then}\ I\ \mathbf{else}\ J, s) = \mathrm{IExec}(I, s) +\cdot \text{Start-At}(\mathrm{insloc}(\mathrm{card}\, I + \mathrm{card}\, J + 3))$.

(45) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I, J be macro instructions, and a be a read-write integer location. Suppose $s(a) \neq 0$ and J is closed onInit s and J is halting onInit s. Then **if** $a = 0$ **then** I **else** J is closed onInit s and **if** $a = 0$ **then** I **else** J is halting onInit s.

(46) Let I, J be macro instructions, a be a read-write integer location, and s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $s(a) \neq 0$ and J is closed onInit s and J is halting onInit s. Then $\mathrm{IExec}(\mathbf{if}\ a = 0\ \mathbf{then}\ I\ \mathbf{else}\ J, s) = \mathrm{IExec}(J, s) +\cdot \text{Start-At}(\mathrm{insloc}(\mathrm{card}\, I + \mathrm{card}\, J + 3))$.

(47) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I, J be InitHalting macro instructions, and a be a read-write integer location. Then **if** $a = 0$ **then** I **else** J is InitHalting and if $s(a) = 0$, then $\mathrm{IExec}(\mathbf{if}\ a = 0\ \mathbf{then}\ I\ \mathbf{else}\ J, s) = \mathrm{IExec}(I, s) +\cdot \text{Start-At}(\mathrm{insloc}(\mathrm{card}\, I + \mathrm{card}\, J + 3))$ and if $s(a) \neq 0$, then $\mathrm{IExec}(\mathbf{if}\ a = 0\ \mathbf{then}\ I\ \mathbf{else}\ J, s) = \mathrm{IExec}(J, s) +\cdot \text{Start-At}(\mathrm{insloc}(\mathrm{card}\, I + \mathrm{card}\, J + 3))$.

(48) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, I, J be InitHalting macro instructions, and a be a read-write integer location. Then

(i) $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a=0\ \mathbf{then}\ I\ \mathbf{else}\ J, s)} = \mathrm{insloc}(\mathrm{card}\, I + \mathrm{card}\, J + 3)$,

(ii) if $s(a) = 0$, then for every integer location d holds $(\mathrm{IExec}(\mathbf{if}\ a = 0\ \mathbf{then}\ I\ \mathbf{else}\ J, s))(d) = (\mathrm{IExec}(I, s))(d)$ and for every finite sequence location f holds $(\mathrm{IExec}(\mathbf{if}\ a = 0\ \mathbf{then}\ I\ \mathbf{else}\ J, s))(f) = (\mathrm{IExec}(I, s))(f)$, and

(iii) if $s(a) \neq 0$, then for every integer location d holds (IExec(**if** $a = 0$ **then** I **else** J, s))(d) = (IExec(J, s))(d) and for every finite sequence location f holds (IExec(**if** $a = 0$ **then** I **else** J, s))(f) = (IExec(J, s))(f).

(49) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I, J be macro instructions, and a be a read-write integer location. Suppose $s(a) > 0$ and I is closed onInit s and I is halting onInit s. Then **if** $a > 0$ **then** I **else** J is closed onInit s and **if** $a > 0$ **then** I **else** J is halting onInit s.

(50) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I, J be macro instructions, and a be a read-write integer location. Suppose $s(a) > 0$ and I is closed onInit s and I is halting onInit s. Then IExec(**if** $a > 0$ **then** I **else** J, s) = IExec(I, s)+· Start-At(insloc(card I + card J + 3)).

(51) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I, J be macro instructions, and a be a read-write integer location. Suppose $s(a) \leqslant 0$ and J is closed onInit s and J is halting onInit s. Then **if** $a > 0$ **then** I **else** J is closed onInit s and **if** $a > 0$ **then** I **else** J is halting onInit s.

(52) Let I, J be macro instructions, a be a read-write integer location, and s be a state of $\mathbf{SCM}_{\rm FSA}$. Suppose $s(a) \leqslant 0$ and J is closed onInit s and J is halting onInit s. Then IExec(**if** $a > 0$ **then** I **else** J, s) = IExec(J, s)+· Start-At(insloc(card I + card J + 3)).

(53) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I, J be InitHalting macro instructions, and a be a read-write integer location. Then **if** $a > 0$ **then** I **else** J is InitHalting and if $s(a) > 0$, then IExec(**if** $a > 0$ **then** I **else** J, s) = IExec(I, s)+· Start-At(insloc(card I + card J + 3)) and if $s(a) \leqslant 0$, then IExec(**if** $a > 0$ **then** I **else** J, s) = IExec(J, s)+· Start-At(insloc(card I + card J + 3)).

(54) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I, J be InitHalting macro instructions, and a be a read-write integer location. Then

(i) $\mathbf{IC}_{\mathrm{IExec}(\mathbf{if}\ a>0\ \mathbf{then}\ I\ \mathbf{else}\ J,s)}$ = insloc(card I + card J + 3),

(ii) if $s(a) > 0$, then for every integer location d holds (IExec(**if** $a > 0$ **then** I **else** J, s))(d) = (IExec(I, s))(d) and for every finite sequence location f holds (IExec(**if** $a > 0$ **then** I **else** J, s))(f) = (IExec(I, s))(f), and

(iii) if $s(a) \leqslant 0$, then for every integer location d holds (IExec(**if** $a > 0$ **then** I **else** J, s))(d) = (IExec(J, s))(d) and for every finite sequence location f holds (IExec(**if** $a > 0$ **then** I **else** J, s))(f) = (IExec(J, s))(f).

(55) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I, J be macro instructions, and a be a read-write integer location. Suppose $s(a) < 0$ and I is closed onInit s and I is halting onInit s. Then IExec(**if** $a < 0$ **then** I **else** J, s) = IExec(I, s)+· Start-At(insloc(card I + card J + card J + 7)).

(56) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I, J be macro instructions, and a be a read-write integer location. Suppose $s(a) = 0$ and J is closed onInit

s and J is halting onInit s. Then IExec(**if** $a < 0$ **then** I **else** J, s) = IExec(J, s)+· Start-At(insloc(card I + card J + card J + 7)).

(57) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I, J be macro instructions, and a be a read-write integer location. Suppose $s(a) > 0$ and J is closed onInit s and J is halting onInit s. Then IExec(**if** $a < 0$ **then** I **else** J, s) = IExec(J, s)+· Start-At(insloc(card I + card J + card J + 7)).

(58) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I, J be InitHalting macro instructions, and a be a read-write integer location. Then

(i) **if** $a < 0$ **then** I **else** J is InitHalting,

(ii) if $s(a) < 0$, then IExec(**if** $a < 0$ **then** I **else** J, s) = IExec(I, s)+· Start-At(insloc(card I + card J + card J + 7)), and

(iii) if $s(a) \geqslant 0$, then IExec(**if** $a < 0$ **then** I **else** J, s) = IExec(J, s)+· Start-At(insloc(card I + card J + card J + 7)).

Let I, J be InitHalting macro instructions and let a be a read-write integer location. One can verify the following observations:

* **if** $a = 0$ **then** I **else** J is InitHalting,
* **if** $a > 0$ **then** I **else** J is InitHalting, and
* **if** $a < 0$ **then** I **else** J is InitHalting.

Next we state a number of propositions:

(59) For every macro instruction I holds I is InitHalting iff for every state s of $\mathbf{SCM}_{\rm FSA}$ holds I is halting on Initialize(s).

(60) For every macro instruction I holds I is InitClosed iff for every state s of $\mathbf{SCM}_{\rm FSA}$ holds I is closed on Initialize(s).

(61) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I be a InitHalting macro instruction, and a be a read-write integer location. Then (IExec(I, s))(a) = (Computation(Initialize(s)+·(I+· Start-At(insloc(0)))))
(LifeSpan(Initialize(s)+·(I+· Start-At(insloc(0)))))(a).

(62) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I be a InitHalting macro instruction, a be an integer location, and k be a natural number. If I does not destroy a, then (IExec(I, s))(a) =
(Computation(Initialize(s)+·(I+· Start-At(insloc(0)))))(k)(a).

(63) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I be a InitHalting macro instruction, and a be an integer location. If I does not destroy a, then (IExec(I, s))(a) = (Initialize(s))(a).

(64) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I be a keepInt0 1 InitHalting macro instruction, and a be a read-write integer location. Suppose I does not destroy a. Then (Computation(Initialize(s)+·((I;SubFrom(a, intloc(0)))
+· Start-At(insloc(0)))))(LifeSpan(Initialize(s)+·((I;SubFrom(a, intloc(0)))
+· Start-At(insloc(0)))))(a) = $s(a) - 1$.

(65) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a InitClosed macro instruction. Suppose $\mathrm{Initialized}(I) \subseteq s$ and s is halting. Let m be a natural number. Suppose $m \leqslant \mathrm{LifeSpan}(s)$. Then $(\mathrm{Computation}(s))(m)$ and $(\mathrm{Computation}(s+\cdot\, \mathrm{loop}\, I))(m)$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(66) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a InitHalting macro instruction. Suppose $\mathrm{Initialized}(I) \subseteq s$. Let k be a natural number. If $k \leqslant \mathrm{LifeSpan}(s)$, then $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot\, \mathrm{loop}\, I))(k)) \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(67) $I \subseteq s+\cdot\, \mathrm{Initialized}(I)$.

(68) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. Suppose I is closed onInit s and I is halting onInit s. Let m be a natural number. Suppose $m \leqslant \mathrm{LifeSpan}(s+\cdot\, \mathrm{Initialized}(I))$. Then $(\mathrm{Computation}(s+\cdot\, \mathrm{Initialized}(I)))(m)$ and $(\mathrm{Computation}(s+\cdot\, \mathrm{Initialized}$ $(\mathrm{loop}\, I)))(m)$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(69) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. Suppose I is closed onInit s and I is halting onInit s. Let m be a natural number. If $m < \mathrm{LifeSpan}(s+\cdot\, \mathrm{Initialized}(I))$, then $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot\, \mathrm{Initialized}(I)))(m)) =$ $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot\, \mathrm{Initialized}(\mathrm{loop}\, I)))(m))$.

(70) For every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $l \notin \mathrm{dom}((\mathrm{intloc}(0) \longmapsto 1)$ $+\cdot\, \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))$.

(71) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. Suppose I is closed onInit s and I is halting onInit s. Then $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot\, \mathrm{Initialized}(\mathrm{loop}\, I)))$ $(\mathrm{LifeSpan}(s+\cdot\, \mathrm{Initialized}(I)))) = \mathrm{goto}\ \mathrm{insloc}(0)$ and for every natural number m such that $m \leqslant \mathrm{LifeSpan}(s+\cdot\, \mathrm{Initialized}(I))$ holds $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot\, \mathrm{Initialized}(\mathrm{loop}\, I)))(m)) \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(72) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and I be a macro instruction. Suppose I is closed onInit s and I is halting onInit s. Then $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot\, \mathrm{Initialized}(\mathrm{loop}\, I)))(\mathrm{LifeSpan}$ $(s+\cdot\, \mathrm{Initialized}(I)))) = \mathrm{goto}\ \mathrm{insloc}(0)$.

(73) 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. Suppose I does not destroy a and $s(\mathrm{intloc}(0)) = 1$ and $s(a) > 0$. Then $\mathrm{loop}\, \mathbf{if}\ a = 0\ \mathbf{then}\ \mathrm{Goto}(\mathrm{insloc}(2))\ \mathbf{else}\ (I;\mathrm{SubFrom}(a, \mathrm{intloc}(0)))$ is pseudo-closed on s.

(74) 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. Suppose I does not destroy a and $s(a) > 0$. Then $\mathrm{Initialized}(\mathrm{loop}\, \mathbf{if}\ a = 0\ \mathbf{then}\ \mathrm{Goto}(\mathrm{insloc}(2))\ \mathbf{else}\ (I;\mathrm{SubFrom}(a, \mathrm{intloc}(0))))$ is pseudo-closed

on s.

6. LOOP-Programs with Initialization Halting

We now state two propositions:

(75) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I be a good InitHalting macro instruction, and a be a read-write integer location. Suppose I does not destroy a and $s(\mathrm{intloc}(0)) = 1$. Then $\mathrm{Times}(a, I)$ is closed on s and $\mathrm{Times}(a, I)$ is halting on s.

(76) Let I be a good InitHalting macro instruction and a be a read-write integer location. If I does not destroy a, then $\mathrm{Initialized}(\mathrm{Times}(a, I))$ is halting.

Let a be a read-write integer location and let I be a good macro instruction. Observe that $\mathrm{Times}(a, I)$ is good.

Next we state several propositions:

(77) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I be a good InitHalting macro instruction, and a be a read-write integer location. Suppose I does not destroy a and $s(\mathrm{intloc}(0)) = 1$ and $s(a) > 0$. Then there exists a state s_2 of $\mathbf{SCM}_{\rm FSA}$ and there exists a natural number k such that

(i) $s_2 = s {+\cdot}\, \mathrm{Initialized}(\mathrm{loop}\,\mathbf{if}\ a = 0\ \mathbf{then}\ \mathrm{Goto}(\mathrm{insloc}(2))$ $\mathbf{else}\ (I;\mathrm{SubFrom}(a, \mathrm{intloc}(0))))$,

(ii) $k = \mathrm{LifeSpan}(s {+\cdot}\, \mathrm{Initialized}(\mathbf{if}\ a = 0\ \mathbf{then}\ \mathrm{Goto}(\mathrm{insloc}(2))$ $\mathbf{else}\ (I;\mathrm{SubFrom}(a, \mathrm{intloc}(0))))) + 1$,

(iii) $(\mathrm{Computation}(s_2))(k)(a) = s(a) - 1$,

(iv) $(\mathrm{Computation}(s_2))(k)(\mathrm{intloc}(0)) = 1$,

(v) for every read-write integer location b such that $b \neq a$ holds $(\mathrm{Computation}(s_2))(k)(b) = (\mathrm{IExec}(I, s))(b)$,

(vi) for every finite sequence location f holds $(\mathrm{Computation}(s_2))(k)(f) = (\mathrm{IExec}(I, s))(f)$,

(vii) $\mathbf{IC}_{(\mathrm{Computation}(s_2))(k)} = \mathrm{insloc}(0)$, and

(viii) for every natural number n such that $n \leqslant k$ holds $\mathbf{IC}_{(\mathrm{Computation}(s_2))(n)} \in \mathrm{dom}\, \mathrm{loop}\,\mathbf{if}\ a = 0\ \mathbf{then}\ \mathrm{Goto}(\mathrm{insloc}(2))$ $\mathbf{else}\ (I;\mathrm{SubFrom}(a, \mathrm{intloc}(0)))$.

(78) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I be a good InitHalting macro instruction, and a be a read-write integer location. If $s(\mathrm{intloc}(0)) = 1$ and $s(a) \leqslant 0$, then $\mathrm{IExec}(\mathrm{Times}(a, I), s){\restriction}(\text{Int-Locations} \cup \text{FinSeq-Locations}) = s{\restriction}(\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(79) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I be a good InitHalting macro instruction, and a be a read-write integer location. Suppose I does not destroy a and $s(a) > 0$. Then $(\mathrm{IExec}(I;\mathrm{SubFrom}(a, \mathrm{intloc}(0)), s))(a) =$

$s(a) - 1$ and $\mathrm{IExec}(\mathrm{Times}(a, I), s) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = \mathrm{IExec}(\mathrm{Times}(a, I), \mathrm{IExec}(I;\mathrm{SubFrom}(a, \mathrm{intloc}(0)), s)) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(80) 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 $s(a) \leqslant 0$, then $(\mathrm{IExec}(\mathrm{Times}(a, I), s))(f) = s(f)$.

(81) 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 $s(a) \leqslant 0$, then $(\mathrm{IExec}(\mathrm{Times}(a, I), s))(b) = (\mathrm{Initialize}(s))(b)$.

(82) 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 I does not destroy a and $s(a) > 0$, then $(\mathrm{IExec}(\mathrm{Times}(a, I), s))(f) = (\mathrm{IExec}(\mathrm{Times}(a, I), \mathrm{IExec}(I;\mathrm{SubFrom}(a, \mathrm{intloc}(0)), s)))(f)$.

(83) 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 I does not destroy a and $s(a) > 0$, then $(\mathrm{IExec}(\mathrm{Times}(a, I), s))(b) = (\mathrm{IExec}(\mathrm{Times}(a, I), \mathrm{IExec}(I;\mathrm{SubFrom}(a, \mathrm{intloc}(0)), s)))(b)$.

Let i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. We say that i is good if and only if:

(Def. 6) i does not destroy $\mathrm{intloc}(0)$.

Let us observe that there exists an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ which is parahalting and good.

Let i be a good instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let J be a good macro instruction. Observe that $i;J$ is good and $J;i$ is good.

Let i, j be good instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Note that $i;j$ is good.

Let a be a read-write integer location and let b be an integer location. Observe that $a{:=}b$ is good and $\mathrm{SubFrom}(a, b)$ is good.

Let a be a read-write integer location, let b be an integer location, and let f be a finite sequence location. Observe that $a{:=}f_b$ is good.

Let a, b be integer locations and let f be a finite sequence location. One can check that $f_a{:=}b$ is good.

Let a be a read-write integer location and let f be a finite sequence location. One can verify that $a{:=}\mathrm{len} f$ is good.

Let n be a natural number. One can check that $\mathrm{intloc}(n + 1)$ is read-write.

References

[1] Noriko Asamoto. Conditional branch macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Part I. *Formalized Mathematics*, 6(**1**):65–72, 1997.

[2] Noriko Asamoto. Conditional branch macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Part II. *Formalized Mathematics*, 6(**1**):73–80, 1997.

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

[4] Noriko Asamoto. The `loop` and `Times` macroinstruction for $\mathbf{SCM}_{\mathrm{FSA}}$. *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 Piotr Rudnicki. Development of terminology for **scm**. *Formalized Mathematics*, 4(**1**):61–67, 1993.
[11] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.
[12] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.
[13] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[14] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[15] Piotr Rudnicki and Andrzej Trybulec. Memory handling for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 6(**1**):29–36, 1997.
[16] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[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. Computation in $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):537–542, 1996.
[20] Andrzej Trybulec and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):571–576, 1996.
[21] Andrzej Trybulec and Yatsuka Nakamura. Relocability for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):583–586, 1996.
[22] Andrzej Trybulec, Yatsuka Nakamura, and Noriko Asamoto. On the compositions of macro instructions. Part I. *Formalized Mathematics*, 6(**1**):21–27, 1997.
[23] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\mathrm{FSA}}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[24] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[25] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[26] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[27] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received June 17, 1998

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

Bubble Sort on $\mathbf{SCM}_{\mathrm{FSA}}$

Jing-Chao Chen
Shanghai Jiaotong University

Yatsuka Nakamura
Shinshu University
Nagano

Summary. We present the bubble sorting algorithm using macro instructions such as the if-Macro (conditional branch macro instructions) and the Times-Macro (for-loop macro instructions) etc. The correctness proof of the program should include the proof of autonomic, halting and the correctness of the program result. In the three terms, we justify rigorously the correctness of the bubble sorting algorithm. In order to prove it is autonomic, we use the following theorem: if all variables used by the program are initialized, it is autonomic. This justification method probably reveals that autonomic concept is not important.

MML Identifier: `SCMBSORT`.

The articles [18], [24], [21], [19], [31], [7], [9], [12], [22], [10], [13], [29], [14], [15], [11], [28], [8], [32], [17], [26], [5], [6], [3], [1], [2], [4], [27], [25], [16], [20], [30], and [23] provide the terminology and notation for this paper.

1. Preliminaries

For simplicity, we adopt the following rules: p is a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, i_1 is an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, i, j, k are natural numbers, f_1, f are finite sequence locations, a, b, d_1, d_2 are integer locations, l, l_1 are instructions-locations of $\mathbf{SCM}_{\mathrm{FSA}}$, and s_1 is a state of $\mathbf{SCM}_{\mathrm{FSA}}$.

We now state a number of propositions:

(1) Let I, J be macro instructions and a, b be integer locations. Suppose I does not destroy b and J does not destroy b. Then **if** $a > 0$ **then** I **else** J does not destroy b.

ISSN 1426–2630

(2) Let I, J be macro instructions and a, b be integer locations. Suppose I does not destroy b and J does not destroy b. Then **if** $a = 0$ **then** I **else** J does not destroy b.

(3) Let I be a macro instruction and a, b be integer locations. If I does not destroy b and $a \neq b$, then $\mathrm{Times}(a, I)$ does not destroy b.

(4) For every function f and for all sets n, m holds $(f{+}\cdot(n \dot{\longmapsto} m){+}\cdot(m \dot{\longmapsto} n))(m) = n$.

(5) For every function f and for all sets n, m holds $(f{+}\cdot(n \dot{\longmapsto} m){+}\cdot(m \dot{\longmapsto} n))(n) = m$.

(6) For every function f and for all sets n, m, x such that $x \in \mathrm{dom}\, f$ and $x \neq m$ and $x \neq n$ holds $(f{+}\cdot(n \dot{\longmapsto} m){+}\cdot(m \dot{\longmapsto} n))(x) = f(x)$.

(7) Let f, g be functions and m, n be sets. Suppose that

(i) $f(m) = g(n)$,

(ii) $f(n) = g(m)$,

(iii) $m \in \mathrm{dom}\, f$,

(iv) $n \in \mathrm{dom}\, f$,

(v) $\mathrm{dom}\, f = \mathrm{dom}\, g$, and

(vi) for every set k such that $k \neq m$ and $k \neq n$ and $k \in \mathrm{dom}\, f$ holds $f(k) = g(k)$.

Then f and g are fiberwise equipotent.

(8) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, f be a finite sequence location, and a, b be integer locations. Then $(\mathrm{Exec}(b{:=}f_a, s))(b) = \pi_{|s(a)|} s(f)$.

(9) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, f be a finite sequence location, and a, b be integer locations. Then $(\mathrm{Exec}(f_a{:=}b, s))(f) = s(f) +\cdot (|s(a)|, s(b))$.

(10) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, f be a finite sequence location, m, n be natural numbers, and a be an integer location. If $m \neq n+1$, then $(\mathrm{Exec}(\mathrm{intloc}(m){:=}f_a, \mathrm{Initialize}(s)))(\mathrm{intloc}(n+1)) = s(\mathrm{intloc}(n+1))$.

(11) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, m, n be natural numbers, and a be an integer location. If $m \neq n{+}1$, then $(\mathrm{Exec}(\mathrm{intloc}(m){:=}a, \mathrm{Initialize}(s)))(\mathrm{intloc}(n{+}1)) = s(\mathrm{intloc}(n+1))$.

(12) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, f be a finite sequence location, and a be a read-write integer location. Then $(\mathrm{IExec}(\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}}, s))(a) = s(a)$ and $(\mathrm{IExec}(\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}}, s))(f) = s(f)$.

In the sequel n denotes a natural number.

One can prove the following propositions:

(13) If $n \leqslant 10$, 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 $n = 10$.

(14) Suppose $n \leqslant 12$. 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 $n = 12$.

(15) Let f, g be functions and X be a set. If $\mathrm{dom}\, f = \mathrm{dom}\, g$ and for every set x such that $x \in X$ holds $f(x) = g(x)$, then $f \restriction X = g \restriction X$.

(16) If $i_1 \in \mathrm{rng}\, p$ and if $i_1 = a{:=}b$ or $i_1 = \mathrm{AddTo}(a, b)$ or $i_1 = \mathrm{SubFrom}(a, b)$ or $i_1 = \mathrm{MultBy}(a, b)$ or $i_1 = \mathrm{Divide}(a, b)$, then $a \in \mathrm{UsedIntLoc}(p)$ and $b \in \mathrm{UsedIntLoc}(p)$.

(17) If $i_1 \in \mathrm{rng}\, p$ and if $i_1 = \mathbf{if}\ a = 0\ \mathbf{goto}\ l_1$ or $i_1 = \mathbf{if}\ a > 0\ \mathbf{goto}\ l_1$, then $a \in \mathrm{UsedIntLoc}(p)$.

(18) If $i_1 \in \mathrm{rng}\, p$ and if $i_1 = b{:=}{f_1}_a$ or $i_1 = {f_1}_a{:=}b$, then $a \in \mathrm{UsedIntLoc}(p)$ and $b \in \mathrm{UsedIntLoc}(p)$.

(19) If $i_1 \in \mathrm{rng}\, p$ and if $i_1 = b{:=}{f_1}_a$ or $i_1 = {f_1}_a{:=}b$, then $f_1 \in \mathrm{UsedInt}^* \,\mathrm{Loc}(p)$.

(20) If $i_1 \in \mathrm{rng}\, p$ and if $i_1 = a{:=}\mathrm{len} f_1$ or $i_1 = f_1{:=}\langle\underbrace{0, \ldots, 0}_{a}\rangle$, then $a \in \mathrm{UsedIntLoc}(p)$.

(21) If $i_1 \in \mathrm{rng}\, p$ and if $i_1 = a{:=}\mathrm{len} f_1$ or $i_1 = f_1{:=}\langle\underbrace{0, \ldots, 0}_{a}\rangle$, then $f_1 \in \mathrm{UsedInt}^* \,\mathrm{Loc}(p)$.

(22) Let p be a macro instruction, s_2, s_3 be states of $\mathbf{SCM}_{\mathrm{FSA}}$, and given i. If $p \subseteq s_2$ and $p \subseteq s_3$, then $(\mathrm{Computation}(s_2))(i) \restriction \mathrm{dom}\, p = (\mathrm{Computation}(s_3))(i) \restriction \mathrm{dom}\, p$.

(23) Let t be a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, p be a macro instruction, and x be a set. Suppose $\mathrm{dom}\, t \subseteq \text{Int-Locations} \cup \text{FinSeq-Locations}$ and $x \in \mathrm{dom}\, t \cup \mathrm{UsedInt}^* \,\mathrm{Loc}(p) \cup \mathrm{UsedIntLoc}(p)$. Then x is an integer location or a finite sequence location.

(24) For every f_1 holds $(\mathrm{Exec}(\mathrm{Divide}(d_1, d_2), s_1))(f_1) = s_1(f_1)$ and $(\mathrm{Exec}(\mathrm{Divide}(d_1, d_2), s_1))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{Next}(\mathbf{IC}_{(s_1)})$.

(25) Let i, k be natural numbers, t be a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, p be a macro instruction, and s_2, s_3 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose that

(i) $k \leqslant i$,

(ii) $p \subseteq s_2$,

(iii) $p \subseteq s_3$,

(iv) $\mathrm{dom}\, t \subseteq \text{Int-Locations} \cup \text{FinSeq-Locations}$,

(v) for every j holds $\mathbf{IC}_{(\mathrm{Computation}(s_2))(j)} \in \mathrm{dom}\, p$ and $\mathbf{IC}_{(\mathrm{Computation}(s_3))(j)} \in \mathrm{dom}\, p$,

(vi) $(\mathrm{Computation}(s_2))(k)(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = (\mathrm{Computation}(s_3))(k)(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}})$, and

(vii) $(\mathrm{Computation}(s_2))(k) \restriction (\mathrm{dom}\, t \cup \mathrm{UsedInt}^* \,\mathrm{Loc}(p) \cup \mathrm{UsedIntLoc}(p)) = (\mathrm{Computation}(s_3))(k) \restriction (\mathrm{dom}\, t \cup \mathrm{UsedInt}^* \,\mathrm{Loc}(p) \cup \mathrm{UsedIntLoc}(p))$.

Then $(\mathrm{Computation}(s_2))(i)(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = (\mathrm{Computation}(s_3))(i)(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}})$ and $(\mathrm{Computation}(s_2))(i) \restriction (\mathrm{dom}\, t \cup \mathrm{UsedInt}^* \,\mathrm{Loc}(p) \cup \mathrm{UsedIntLoc}(p)) = (\mathrm{Computation}(s_3))(i) \restriction (\mathrm{dom}\, t \cup \mathrm{UsedInt}^* \,\mathrm{Loc}(p) \cup \mathrm{UsedIntLoc}(p))$.

(26) Let i, k be natural numbers, p be a macro instruction, and s_2, s_3 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $k \leqslant i$ and $p \subseteq s_2$ and $p \subseteq s_3$ and for every j holds $\mathbf{IC}_{(\mathrm{Computation}(s_2))(j)} \in \operatorname{dom} p$ and $\mathbf{IC}_{(\mathrm{Computation}(s_3))(j)} \in \operatorname{dom} p$ and $(\mathrm{Computation}(s_2))(k)(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = (\mathrm{Computation}(s_3))(k)(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}})$ and $(\mathrm{Computation}(s_2))(k) \restriction (\mathrm{UsedInt}^* \mathrm{Loc}(p) \cup \mathrm{UsedIntLoc}(p)) = (\mathrm{Computation}(s_3))(k) \restriction (\mathrm{UsedInt}^* \mathrm{Loc}(p) \cup \mathrm{UsedIntLoc}(p))$. Then $(\mathrm{Computation}(s_2))(i)(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = (\mathrm{Computation}(s_3))(i)(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}})$ and $(\mathrm{Computation}(s_2))(i) \restriction (\mathrm{UsedInt}^* \mathrm{Loc}(p) \cup \mathrm{UsedIntLoc}(p)) = (\mathrm{Computation}(s_3))(i) \restriction (\mathrm{UsedInt}^* \mathrm{Loc}(p) \cup \mathrm{UsedIntLoc}(p))$.

(27) $\mathrm{UsedIntLoc}(\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}}) = \emptyset$.

(28) $\mathrm{UsedIntLoc}(\mathrm{Goto}(l)) = \emptyset$.

(29) For all macro instructions I, J and for every integer location a holds $\mathrm{UsedIntLoc}(\mathbf{if}\ a = 0\ \mathbf{then}\ I\ \mathbf{else}\ J) = \{a\} \cup \mathrm{UsedIntLoc}(I) \cup \mathrm{UsedIntLoc}(J)$ and $\mathrm{UsedIntLoc}(\mathbf{if}\ a > 0\ \mathbf{then}\ I\ \mathbf{else}\ J) = \{a\} \cup \mathrm{UsedIntLoc}(I) \cup \mathrm{UsedIntLoc}(J)$.

(30) For every macro instruction I and for every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{UsedIntLoc}(\mathrm{Directed}(I, l)) = \mathrm{UsedIntLoc}(I)$.

(31) For every integer location a and for every macro instruction I holds $\mathrm{UsedIntLoc}(\mathrm{Times}(a, I)) = \mathrm{UsedIntLoc}(I) \cup \{a, \mathrm{intloc}(0)\}$.

(32) For all sets x_1, x_2, x_3 holds $\{x_2, x_1\} \cup \{x_3, x_1\} = \{x_1, x_2, x_3\}$.

(33) $\mathrm{UsedInt}^* \mathrm{Loc}(\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}}) = \emptyset$.

(34) $\mathrm{UsedInt}^* \mathrm{Loc}(\mathrm{Goto}(l)) = \emptyset$.

(35) For all macro instructions I, J and for every integer location a holds $\mathrm{UsedInt}^* \mathrm{Loc}(\mathbf{if}\ a = 0\ \mathbf{then}\ I\ \mathbf{else}\ J) = \mathrm{UsedInt}^* \mathrm{Loc}(I) \cup \mathrm{UsedInt}^* \mathrm{Loc}(J)$ and $\mathrm{UsedInt}^* \mathrm{Loc}(\mathbf{if}\ a > 0\ \mathbf{then}\ I\ \mathbf{else}\ J) = \mathrm{UsedInt}^* \mathrm{Loc}(I) \cup \mathrm{UsedInt}^* \mathrm{Loc}(J)$.

(36) For every macro instruction I and for every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{UsedInt}^* \mathrm{Loc}(\mathrm{Directed}(I, l)) = \mathrm{UsedInt}^* \mathrm{Loc}(I)$.

(37) For every integer location a and for every macro instruction I holds $\mathrm{UsedInt}^* \mathrm{Loc}(\mathrm{Times}(a, I)) = \mathrm{UsedInt}^* \mathrm{Loc}(I)$.

Let f be a finite sequence location and let t be a finite sequence of elements of $\mathbb{Z}$. Then $f \longmapsto t$ is a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$.

One can prove the following propositions:

(38) Every finite sequence of elements of $\mathbb{Z}$ is a finite sequence of elements of $\mathbb{R}$.

(39) Let t be a finite sequence of elements of $\mathbb{Z}$. Then there exists a finite sequence u of elements of $\mathbb{R}$ such that t and u are fiberwise equipotent and u is a finite sequence of elements of $\mathbb{Z}$ and non-increasing.

(40) $\operatorname{dom}((\mathrm{intloc}(0) \longmapsto 1) +\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0))) = \{\mathrm{intloc}(0), \mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\}$.

(41) For every macro instruction I holds $\mathrm{dom}\,\mathrm{Initialized}(I) = \mathrm{dom}\,I \cup \{\mathrm{intloc}(0), \mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\}$.

(42) Let w be a finite sequence of elements of $\mathbb{Z}$, f be a finite sequence location, and I be a macro instruction. Then $\mathrm{dom}(\mathrm{Initialized}(I)+\cdot(f \mathrel{\dot{\longmapsto}} w)) = \mathrm{dom}\,I \cup \{\mathrm{intloc}(0), \mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}, f\}$.

(43) For every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \neq l$.

(44) For every integer location a and for every macro instruction I holds $\mathrm{card}\,\mathrm{Times}(a, I) = \mathrm{card}\,I + 12$.

(45) For all instructions i_2, i_3, i_4 of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{card}(i_2;i_3;i_4) = 6$, where $i_2 = b_4{:=}b_3$, $b_4 = \mathrm{intloc}(3+1)$, $b_3 = \mathrm{intloc}(2+1)$, $i_3 = \mathrm{SubFrom}(b_3, a_0)$, $a_0 = \mathrm{intloc}(0)$, $i_4 = b_5{:=}{f_0}_{b_3}$, $b_5 = \mathrm{intloc}(4+1)$, and $f_0 = \mathrm{fsloc}(0)$.

(46) Let t be a finite sequence of elements of $\mathbb{Z}$, f be a finite sequence location, and I be a macro instruction. Then $\mathrm{dom}\,\mathrm{Initialized}(I) \cap \mathrm{dom}(f \mathrel{\dot{\longmapsto}} t) = \emptyset$.

(47) Let w be a finite sequence of elements of $\mathbb{Z}$, f be a finite sequence location, and I be a macro instruction. Then $\mathrm{Initialized}(I)+\cdot(f \mathrel{\dot{\longmapsto}} w)$ starts at $\mathrm{insloc}(0)$.

(48) Let I, J be macro instructions, k be a natural number, and i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. If $k < \mathrm{card}\,J$ and $i = J(\mathrm{insloc}(k))$, then $(I;J)(\mathrm{insloc}(\mathrm{card}\,I + k)) = \mathrm{IncAddr}(i, \mathrm{card}\,I)$.

(49) Suppose that

(i) $i_1 = a{:=}b$, or
(ii) $i_1 = \mathrm{AddTo}(a, b)$, or
(iii) $i_1 = \mathrm{SubFrom}(a, b)$, or
(iv) $i_1 = \mathrm{MultBy}(a, b)$, or
(v) $i_1 = \mathrm{Divide}(a, b)$, or
(vi) $i_1 = \mathrm{goto}\ l_1$, or
(vii) $i_1 = \mathbf{if}\ a = 0\ \mathbf{goto}\ l_1$, or
(viii) $i_1 = \mathbf{if}\ a > 0\ \mathbf{goto}\ l_1$, or
(ix) $i_1 = b{:=}f_a$, or
(x) $i_1 = f_a{:=}b$, or
(xi) $i_1 = a{:=}\mathrm{len}f$, or
(xii) $i_1 = f{:=}\langle\underbrace{0, \ldots, 0}_{a}\rangle$.

Then $i_1 \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(50) Let I, J be macro instructions, k be a natural number, and i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose for every natural number n holds $\mathrm{IncAddr}(i, n) = i$ and $i \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$ and $k = \mathrm{card}\,I$. Then $(I;i;J)(\mathrm{insloc}(k)) = i$ and $(I;i;J)(\mathrm{insloc}(k+1)) = \mathrm{goto}\ \mathrm{insloc}(\mathrm{card}\,I + 2)$.

(51) Let I, J be macro instructions and k be a natural number. If $k = \mathrm{card}\,I$, then $(I;(a{:=}b);J)(\mathrm{insloc}(k)) = a{:=}b$ and $(I;(a{:=}b);J)(\mathrm{insloc}(k+1)) =$

goto insloc(card $I+2$).

(52) Let I, J be macro instructions and k be a natural number. If $k = \operatorname{card} I$, then $(I;(a{:=}\operatorname{len} f);J)(\operatorname{insloc}(k)) = a{:=}\operatorname{len} f$ and $(I;(a{:=}\operatorname{len} f);J)(\operatorname{insloc}(k+1)) = \text{goto } \operatorname{insloc}(\operatorname{card} I + 2)$.

(53) Let w be a finite sequence of elements of $\mathbb{Z}$, f be a finite sequence location, s be a state of $\mathbf{SCM}_{\rm FSA}$, and I be a macro instruction. If $\operatorname{Initialized}(I) {+\cdot} (f \longmapsto w) \subseteq s$, then $I \subseteq s$.

(54) Let w be a finite sequence of elements of $\mathbb{Z}$, f be a finite sequence location, s be a state of $\mathbf{SCM}_{\rm FSA}$, and I be a macro instruction. If $\operatorname{Initialized}(I) {+\cdot} (f \longmapsto w) \subseteq s$, then $s(f) = w$ and $s(\operatorname{intloc}(0)) = 1$.

(55) For every finite sequence location f and for every integer location a and for every state s of $\mathbf{SCM}_{\rm FSA}$ holds $\{a, \mathbf{IC}_{\mathbf{SCM}_{\rm FSA}}, f\} \subseteq \operatorname{dom} s$.

(56) For every macro instruction p and for every state s of $\mathbf{SCM}_{\rm FSA}$ holds $\operatorname{UsedInt}^* \operatorname{Loc}(p) \cup \operatorname{UsedIntLoc}(p) \subseteq \operatorname{dom} s$.

(57) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, I be a macro instruction, and f be a finite sequence location. Then $(\operatorname{Result}(s {+\cdot} \operatorname{Initialized}(I)))(f) = (\operatorname{IExec}(I, s))(f)$.

2. The Program Code for Buble Sort

Let f be a finite sequence location. The functor bubble-sort(f) yields a macro instruction and is defined as follows:

(Def. 1) bubble-sort$(f) = i_5$;
$(a_1{:=}\operatorname{len} f)$;
$\operatorname{Times}(a_1,$
$(a_2{:=}a_1)$;
$\operatorname{SubFrom}(a_2, a_0)$;
$(a_3{:=}\operatorname{len} f)$;
$\operatorname{Times}(a_2,$
$(a_4{:=}a_3)$;
$\operatorname{SubFrom}(a_3, a_0)$;
$(a_5{:=}f_{a_3})$;
$(a_6{:=}f_{a_4})$;
$\operatorname{SubFrom}(a_6, a_5)$;
$(\mathbf{if}\ a_6 > 0\ \mathbf{then}\ (a_6{:=}f_{a_4});(f_{a_3}{:=}a_6);(f_{a_4}{:=}a_5)\ \mathbf{else}\ (\operatorname{Stop}_{\rm SCM_{FSA}}))))$,
where $i_5 = (a_2{:=}a_0);(a_3{:=}a_0);(a_4{:=}a_0);(a_5{:=}a_0);(a_6{:=}a_0)$,
$a_2 = \operatorname{intloc}(2)$, $a_0 = \operatorname{intloc}(0)$, $a_3 = \operatorname{intloc}(3)$, $a_4 = \operatorname{intloc}(4)$, $a_5 = \operatorname{intloc}(5)$, $a_6 = \operatorname{intloc}(6)$, and $a_1 = \operatorname{intloc}(1)$.

The macro instruction the bubble sort algorithm is defined by:

(Def. 2) The bubble sort algorithm = bubble-sort(fsloc(0)).

The following propositions are true:

(58) For every finite sequence location f holds UsedIntLoc(bubble-sort(f)) $= \{a_0, a_1, a_2, a_3, a_4, a_5, a_6\}$, where $a_0 = \mathrm{intloc}(0)$, $a_1 = \mathrm{intloc}(1)$, $a_2 = \mathrm{intloc}(2)$, $a_3 = \mathrm{intloc}(3)$, $a_4 = \mathrm{intloc}(4)$, $a_5 = \mathrm{intloc}(5)$, and $a_6 = \mathrm{intloc}(6)$.

(59) For every finite sequence location f holds UsedInt* Loc(bubble-sort(f)) $= \{f\}$.

3. Defining Relationship Between the Input and Output of Sorting Algorithms

The partial function Sorting-Function from FinPartSt($\mathbf{SCM}_{\mathrm{FSA}}$) to FinPartSt($\mathbf{SCM}_{\mathrm{FSA}}$) is defined by the condition (Def. 3).

(Def. 3) Let p, q be finite partial states of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $\langle p, q\rangle \in$ Sorting-Function if and only if there exists a finite sequence t of elements of $\mathbb{Z}$ and there exists a finite sequence u of elements of $\mathbb{R}$ such that t and u are fiberwise equipotent and u is a finite sequence of elements of $\mathbb{Z}$ and non-increasing and $p = \mathrm{fsloc}(0) \mapsto t$ and $q = \mathrm{fsloc}(0) \mapsto u$.

We now state two propositions:

(60) For every set p holds $p \in \mathrm{dom}$ Sorting-Function iff there exists a finite sequence t of elements of $\mathbb{Z}$ such that $p = \mathrm{fsloc}(0) \mapsto t$.

(61) Let t be a finite sequence of elements of $\mathbb{Z}$. 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) (Sorting-Function)($\mathrm{fsloc}(0) \mapsto t$) $= \mathrm{fsloc}(0) \mapsto u$.

4. The Basic Property of Buble Sort

Next we state several propositions:

(62) For every finite sequence location f holds card bubble-sort(f) $= 63$.

(63) For every finite sequence location f and for every natural number k such that $k < 63$ holds insloc(k) $\in$ dom bubble-sort(f).

(64) bubble-sort(fsloc(0)) is keepInt0 1 and InitHalting.

(65) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Then

(i) $s(f_0)$ and (IExec(bubble-sort(f_0), s))(f_0) are fiberwise equipotent, and

(ii) for all natural numbers i, j such that $i \geqslant 1$ and $j \leqslant \mathrm{len}\, s(f_0)$ and $i < j$ and for all integers x_1, x_2 such that $x_1 =$ (IExec(bubble-sort(f_0), s))(f_0)(i) and $x_2 =$ (IExec(bubble-sort(f_0), s))(f_0)(j) holds $x_1 \geqslant x_2$, where $f_0 = \mathrm{fsloc}(0)$.

(66) Let i be a natural number, s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and w be a finite sequence of elements of $\mathbb{Z}$. Suppose Initialized(the bubble sort algorithm) $+\cdot(\mathrm{fsloc}(0) \mapsto w) \subseteq s$. Then $\mathbf{IC}_{(\mathrm{Computation}(s))(i)} \in \mathrm{dom}$ (the bubble sort algorithm).

(67) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and t be a finite sequence of elements of $\mathbb{Z}$. Suppose Initialized(the bubble sort algorithm) $+\cdot(\mathrm{fsloc}(0) \mapsto t) \subseteq s$. 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) $(\mathrm{Result}(s))(\mathrm{fsloc}(0)) = u$.

5. The Correctness and Autonomousness of Buble Sort Algorithm

We now state two propositions:

(68) For every finite sequence w of elements of $\mathbb{Z}$ holds Initialized(the bubble sort algorithm) $+\cdot(\mathrm{fsloc}(0) \mapsto w)$ is autonomic.

(69) Initialized(the bubble sort algorithm) computes Sorting-Function.

References

[1] Noriko Asamoto. Conditional branch macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Part I. *Formalized Mathematics*, 6(**1**):65–72, 1997.

[2] Noriko Asamoto. Conditional branch macro instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. Part II. *Formalized Mathematics*, 6(**1**):73–80, 1997.

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

[4] Noriko Asamoto. The `loop` and `Times` macroinstruction for $\mathbf{SCM}_{\mathrm{FSA}}$. *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 Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.

[12] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.

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

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

[15] 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.
[16] Jing-Chao Chen and Yatsuka Nakamura. Initialization halting concepts and their basic properties of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 7(**1**):139–151, 1998.
[17] Jarosław Kotowicz. Functions and finite sequences of real numbers. *Formalized Mathematics*, 3(**2**):275–278, 1992.
[18] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[19] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.
[20] Piotr Rudnicki and Andrzej Trybulec. Memory handling for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 6(**1**):29–36, 1997.
[21] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[22] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[23] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[24] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[25] Andrzej Trybulec and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):571–576, 1996.
[26] Andrzej Trybulec, Yatsuka Nakamura, and Noriko Asamoto. On the compositions of macro instructions. Part I. *Formalized Mathematics*, 6(**1**):21–27, 1997.
[27] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\mathrm{FSA}}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[28] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 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 and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received June 17, 1998

Index of MML Identifiers

Contents

Formaliz. Math. 7 (**1**)

Continued on inside back cover