FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Adjacency Concept for Pairs of Natural Numbers

Yatsuka Nakamura
Shinshu University
Nagano

Andrzej Trybulec
Warsaw University
Białystok

Summary. First, we introduce the concept of adjacency for a pair of natural numbers. Second, we extend the concept for two pairs of natural numbers. The pairs represent points of a lattice in a plane. We show that if some property is infectious among adjacent points, and some points have the property, then all points have the property.

MML Identifier: GOBRD10.

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

In this paper i, j, k, k_1, k_2, n, m, i_1, i_2, j_1, j_2 are natural numbers.

Let us consider i_1, i_2. We say that i_1 and i_2 are adjacent if and only if:

(Def. 1) $i_2 = i_1 + 1$ or $i_1 = i_2 + 1$.

One can prove the following propositions:

(1) For all i_1, i_2 such that i_1 and i_2 are adjacent holds $i_1 + 1$ and $i_2 + 1$ are adjacent.

(2) For all i_1, i_2 such that i_1 and i_2 are adjacent and $1 \leq i_1$ and $1 \leq i_2$ holds $i_1 -' 1$ and $i_2 -' 1$ are adjacent.

Let us consider i_1, j_1, i_2, j_2. We say that i_1, j_1, i_2, and j_2 are adjacent if and only if:

(Def. 2) i_1 and i_2 are adjacent and $j_1 = j_2$ or $i_1 = i_2$ and j_1 and j_2 are adjacent.

The following propositions are true:

(3) For all i_1, i_2, j_1, j_2 such that i_1, j_1, i_2, and j_2 are adjacent holds $i_1 + 1$, $j_1 + 1$, $i_2 + 1$, and $j_2 + 1$ are adjacent.

(4) Given i_1, i_2, j_1, j_2. Suppose i_1, j_1, i_2, and j_2 are adjacent and $1 \leq i_1$ and $1 \leq i_2$ and $1 \leq j_1$ and $1 \leq j_2$. Then $i_1 -' 1$, $j_1 -' 1$, $i_2 -' 1$, and $j_2 -' 1$ are adjacent.

ISSN 1426–2630

Let us consider i, n. The functor $\operatorname{Repeat}(i,n)$ yields a finite sequence of elements of $\mathbb{N}$ and is defined as follows:

(Def. 3) $\operatorname{len}\operatorname{Repeat}(i,n) = n$ and for every j such that $1 \leq j$ and $j \leq n$ holds $(\operatorname{Repeat}(i,n))(j) = i$.

Next we state four propositions:

(5) For every i holds $\operatorname{Repeat}(i,0) = \varepsilon$.

(6) Given n, i, j. Suppose $i \leq n$ and $j \leq n$. Then there exists a finite sequence f_1 of elements of $\mathbb{N}$ such that

(i) $f_1(1) = i$,

(ii) $f_1(\operatorname{len} f_1) = j$,

(iii) $\operatorname{len} f_1 = i -' j + j -' i + 1$,

(iv) for all k, k_1 such that $1 \leq k$ and $k \leq \operatorname{len} f_1$ and $k_1 = f_1(k)$ holds $k_1 \leq n$, and

(v) for every i_1 such that $1 \leq i_1$ and $i_1 < \operatorname{len} f_1$ holds $f_1(i_1+1) = \pi_{i_1} f_1 + 1$ or $f_1(i_1) = \pi_{i_1+1} f_1 + 1$.

(7) Given n, i, j. Suppose $i \leq n$ and $j \leq n$. Then there exists a finite sequence f_1 of elements of $\mathbb{N}$ such that

(i) $f_1(1) = i$,

(ii) $f_1(\operatorname{len} f_1) = j$,

(iii) $\operatorname{len} f_1 = i -' j + j -' i + 1$,

(iv) for all k, k_1 such that $1 \leq k$ and $k \leq \operatorname{len} f_1$ and $k_1 = f_1(k)$ holds $k_1 \leq n$, and

(v) for every i_1 such that $1 \leq i_1$ and $i_1 < \operatorname{len} f_1$ holds $\pi_{i_1} f_1$ and $\pi_{i_1+1} f_1$ are adjacent.

(8) Given n, m, i_1, j_1, i_2, j_2. Suppose $i_1 \leq n$ and $j_1 \leq m$ and $i_2 \leq n$ and $j_2 \leq m$. Then there exist finite sequences f_1, f_2 of elements of $\mathbb{N}$ such that

(i) for all i, k_1, k_2 such that $i \in \operatorname{dom} f_1$ and $k_1 = f_1(i)$ and $k_2 = f_2(i)$ holds $k_1 \leq n$ and $k_2 \leq m$,

(ii) $f_1(1) = i_1$,

(iii) $f_1(\operatorname{len} f_1) = i_2$,

(iv) $f_2(1) = j_1$,

(v) $f_2(\operatorname{len} f_2) = j_2$,

(vi) $\operatorname{len} f_1 = \operatorname{len} f_2$,

(vii) $\operatorname{len} f_1 = i_1 -' i_2 + i_2 -' i_1 + j_1 -' j_2 + j_2 -' j_1 + 1$, and

(viii) for every i such that $1 \leq i$ and $i < \operatorname{len} f_1$ holds $\pi_i f_1$, $\pi_i f_2$, $\pi_{i+1} f_1$, and $\pi_{i+1} f_2$ are adjacent.

In the sequel S is a set.

Next we state the proposition

(9) Let Y be a subset of S and let F be a matrix over 2^S of dimension $n \times m$. Suppose that

(i) there exist i, j such that $i \in \operatorname{Seg} n$ and $j \in \operatorname{Seg} m$ and $F_{i,j} \subseteq Y$, and

(ii) for all i_1, j_1, i_2, j_2 such that $i_1 \in \operatorname{Seg} n$ and $i_2 \in \operatorname{Seg} n$ and $j_1 \in \operatorname{Seg} m$ and $j_2 \in \operatorname{Seg} m$ and i_1, j_1, i_2, and j_2 are adjacent holds $F_{i_1,j_1} \subseteq Y$ iff

$F_{i_2,j_2} \subseteq Y$.
Given i, j. If $i \in \operatorname{Seg} n$ and $j \in \operatorname{Seg} m$, then $F_{i,j} \subseteq Y$.

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. *Formalized Mathematics*, 1(**1**):175–180, 1990.
[4] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[5] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[6] Katarzyna Jankowska. Matrices. Abelian group of matrices. *Formalized Mathematics*, 2(**4**):475–480, 1991.
[7] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[8] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[9] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[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 June 10, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Inverse Limits of Many Sorted Algebras

Adam Grabowski
Warsaw University
Białystok

Summary. This article introduces the construction of an inverse limit of many sorted algebras. A few preliminary notions such as an ordered family of many sorted algebras and a binding of family are formulated. Definitions of a set of many sorted signatures and a set of signature morphisms are also given.

MML Identifier: `MSALIMIT`.

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

1. Inverse Limits of Many Sorted Algebras

We adopt the following rules: P denotes a non empty poset, i, j, k denote elements of P, and S denotes a non void non empty many sorted signature.

Let I be a non empty set, let us consider S, let A_1 be an algebra family of I over S, let i be an element of I, and let o be an operation symbol of S. One can verify that $(\mathrm{OPER}(A_1))(i)(o)$ is function-like and relation-like.

Let I be a non empty set, let us consider S, let A_1 be an algebra family of I over S, and let s be a sort symbol of S. Note that $(\mathrm{SORTS}(A_1))(s)$ is functional.

Let us consider P, S. An algebra family of the carrier of P over S is called a family of algebras over S ordered by P if it satisfies the condition (Def. 1).

(Def. 1) There exists a many sorted function F of the internal relation of P such that for all i, j, k if $i \geq j$ and $j \geq k$, then there exists a many sorted function f_1 from it(i) into it(j) and there exists a many sorted function f_2 from it(j) into it(k) such that $f_1 = F(j, i)$ and $f_2 = F(k, j)$ and $F(k, i) = f_2 \circ f_1$ and f_1 is a homomorphism of it(i) into it(j).

ISSN 1426–2630

In the sequel O_1 is a family of algebras over S ordered by P.

Let us consider P, S, O_1. A many sorted function of the internal relation of P is called a binding of O_1 if it satisfies the condition (Def. 2).

(Def. 2) Given i, j, k. Suppose $i \geq j$ and $j \geq k$. Then there exists a many sorted function f_1 from $O_1(i)$ into $O_1(j)$ and there exists a many sorted function f_2 from $O_1(j)$ into $O_1(k)$ such that $f_1 = \mathrm{it}(j, i)$ and $f_2 = \mathrm{it}(k, j)$ and $\mathrm{it}(k, i) = f_2 \circ f_1$ and f_1 is a homomorphism of $O_1(i)$ into $O_1(j)$.

Let us consider P, S, O_1, let B be a binding of O_1, and let us consider i, j. Let us assume that $i \geq j$. The functor $\mathrm{bind}(B, i, j)$ yielding a many sorted function from $O_1(i)$ into $O_1(j)$ is defined by:

(Def. 3) $\mathrm{bind}(B, i, j) = B(j, i)$.

In the sequel B will be a binding of O_1.

Next we state the proposition

(1) If $i \geq j$ and $j \geq k$, then $\mathrm{bind}(B, j, k) \circ \mathrm{bind}(B, i, j) = \mathrm{bind}(B, i, k)$.

Let us consider P, S, O_1 and let I_1 be a binding of O_1. We say that I_1 is normalized if and only if:

(Def. 4) For every i holds $I_1(i, i) = \mathrm{id}_{(\text{the sorts of } O_1(i))}$.

We now state the proposition

(2) Given P, S, O_1, B, i, j. Suppose $i \geq j$. Let f be a many sorted function from $O_1(i)$ into $O_1(j)$. If $f = \mathrm{bind}(B, i, j)$, then f is a homomorphism of $O_1(i)$ into $O_1(j)$.

Let us consider P, S, O_1, B. The functor $\mathrm{Normalized}(B)$ yields a binding of O_1 and is defined as follows:

(Def. 5) For all i, j such that $i \geq j$ holds $(\mathrm{Normalized}(B))(j, i) = (j = i \rightarrow \mathrm{id}_{(\text{the sorts of } O_1(i))}, \mathrm{bind}(B, i, j) \circ \mathrm{id}_{(\text{the sorts of } O_1(i))})$.

Next we state the proposition

(3) For all i, j such that $i \geq j$ and $i \neq j$ holds $B(j, i) = (\mathrm{Normalized}(B))(j, i)$.

Let us consider P, S, O_1, B. One can verify that $\mathrm{Normalized}(B)$ is normalized.

Let us consider P, S, O_1. Note that there exists a binding of O_1 which is normalized.

The following proposition is true

(4) For every normalized binding N_1 of O_1 and for all i, j such that $i \geq j$ holds $(\mathrm{Normalized}(N_1))(j, i) = N_1(j, i)$.

Let us consider P, S, O_1 and let B be a binding of O_1. The functor $\varprojlim B$ yields a strict subalgebra of $\prod O_1$ and is defined by the condition (Def. 6).

(Def. 6) Let s be a sort symbol of S and let f be an element of $(\mathrm{SORTS}(O_1))(s)$. Then $f \in (\text{the sorts of } \varprojlim B)(s)$ if and only if for all i, j such that $i \geq j$ holds $(\mathrm{bind}(B, i, j))(s)(f(i)) = f(j)$.

Next we state the proposition

(5) Let D_1 be a discrete non empty poset, and given S, and let O_1 be a family of algebras over S ordered by D_1, and let B be a normalized binding of O_1. Then $\varprojlim B = \prod O_1$.

2. Sets and Morphisms of Many Sorted Signatures

In the sequel x will be a set and A will be a non empty set.

Let X be a set. We say that X is MSS-membered if and only if:

(Def. 7) If $x \in X$, then x is a strict non empty non void many sorted signature.

One can verify that there exists a set which is non empty and MSS-membered.

The strict many sorted signature TrivialMSSign is defined by:

(Def. 8) TrivialMSSign is empty and void.

Let us note that TrivialMSSign is empty and void.

One can check that there exists a many sorted signature which is strict, empty, and void.

The following proposition is true

(6) Let S be a void many sorted signature. Then $\mathrm{id}_{(\text{the carrier of } S)}$ and $\mathrm{id}_{(\text{the operation symbols of } S)}$ form morphism between S and S.

Let us consider A. The functor MSS-set(A) is defined by the condition (Def. 9).

(Def. 9) $x \in$ MSS-set(A) if and only if there exists a strict non empty non void many sorted signature S such that $x = S$ and the carrier of $S \subseteq A$ and the operation symbols of $S \subseteq A$.

Let us consider A. One can check that MSS-set(A) is non empty and MSS-membered.

Let A be a non empty MSS-membered set. We see that the element of A is a strict non empty non void many sorted signature.

The following proposition is true

(7) Let x be an element of MSS-set(A). Then $\mathrm{id}_{(\text{the carrier of } x)}$ and $\mathrm{id}_{(\text{the operation symbols of } x)}$ form morphism between x and x.

Let S_1, S_2 be many sorted signatures. The functor MSS-morph(S_1, S_2) is defined by:

(Def. 10) $x \in$ MSS-morph(S_1, S_2) iff there exist functions f, g such that $x = \langle f, g\rangle$ and f and g form morphism between S_1 and S_2.

References

[1] Grzegorz Bancerek. Cartesian product of functions. *Formalized Mathematics*, 2(**4**):547–552, 1991.

[2] Grzegorz Bancerek. Curried and uncurried functions. *Formalized Mathematics*, 1(**3**):537–541, 1990.

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

[4] Grzegorz Bancerek. Minimal signature for partial algebra. *Formalized Mathematics*, 5(**3**):405–414, 1996.

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

[6] Ewa Burakowska. Subalgebras of many sorted algebra. Lattice of subalgebras. *Formalized Mathematics*, 5(**1**):47–54, 1996.

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

[8] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.

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

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

[11] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.

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

[13] Adam Grabowski. On the category of posets. *Formalized Mathematics*, 5(**4**):501–505, 1996.

[14] Małgorzata Korolkiewicz. Homomorphisms of many sorted algebras. *Formalized Mathematics*, 5(**1**):61–65, 1996.

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

[16] Beata Madras. Products of many sorted algebras. *Formalized Mathematics*, 5(**1**):55–60, 1996.

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

[18] Andrzej Trybulec. Function domains and Frænkel operator. *Formalized Mathematics*, 1(**3**):495–500, 1990.

[19] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.

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

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

[22] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.

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

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

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

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

[27] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

[28] Edmund Woronowicz and Anna Zalewska. Properties of binary relations. *Formalized Mathematics*, 1(**1**):85–89, 1990.

Received June 11, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

On the Trivial Many Sorted Algebras and Many Sorted Congruences

Artur Korniłowicz
Warsaw University
Białystok

Summary. This paper contains properties of many sorted functions between two many sorted sets. Other theorems describe trivial many sorted algebras. In the last section there are theorems about many sorted congruences, which are defined on many sorted algebras. I have also proved facts about natural epimorphism.

MML Identifier: `MSUALG_9`.

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

1. Preliminaries

In this paper a, I will be sets and S will be a non empty non void many sorted signature.

The scheme *MSSExD* deals with a non empty set $\mathcal{A}$ and a binary predicate $\mathcal{P}$, and states that:

> There exists a many sorted set f indexed by $\mathcal{A}$ such that for every element i of $\mathcal{A}$ holds $\mathcal{P}[i, f(i)]$

provided the parameters meet the following condition:

- For every element i of $\mathcal{A}$ there exists a set j such that $\mathcal{P}[i, j]$.

Let I be a set and let M be a many sorted set indexed by I. Note that there exists an element of $\mathrm{Bool}(M)$ which is locally-finite.

Let I be a set and let M be a non-empty many sorted set indexed by I. Note that there exists a many sorted subset of M which is non-empty and locally-finite.

ISSN 1426–2630

Let S be a non empty non void many sorted signature, let A be a non-empty algebra over S, and let o be an operation symbol of S. One can verify that every element of $\mathrm{Args}(o, A)$ is finite sequence-like.

Let S be a non void non empty many sorted signature, let I be a set, let s be a sort symbol of S, and let F be an algebra family of I over S. Note that every element of $\mathrm{SORTS}(F)(s)$ is function-like and relation-like.

Let S be a non void non empty many sorted signature and let X be a non-empty many sorted set indexed by the carrier of S. Note that $\mathrm{FreeGenerator}(X)$ is free and non-empty.

Let S be a non void non empty many sorted signature and let X be a non-empty many sorted set indexed by the carrier of S. One can verify that $\mathrm{Free}(X)$ is free.

Let S be a non empty non void many sorted signature and let A, B be non-empty algebras over S. One can check that $[\!:A, B:\!]$ is non-empty.

The following propositions are true:

(1) For all sets X, Y and for every function f such that $a \in \mathrm{dom}\, f$ and $f(a) \in [\!:X, Y:\!]$ holds $f(a) = \langle \mathrm{pr1}(f)(a), \mathrm{pr2}(f)(a) \rangle$.

(2) For every non empty set X and for every set Y and for every function f from X into $\{Y\}$ holds $\mathrm{rng}\, f = \{Y\}$.

(3) For every non empty finite set A there exists a function f from $\mathbb{N}$ into A such that $\mathrm{rng}\, f = A$.

(4) $\mathrm{Classes}(\nabla_I) \subseteq \{I\}$.

(5) For every non empty set I holds $\mathrm{Classes}(\nabla_I) = \{I\}$.

(6) There exists a many sorted set A indexed by I such that $\{A\} = I \longmapsto \{a\}$.

(7) For every many sorted set A indexed by I there exists a non-empty many sorted set B indexed by I such that $A \subseteq B$.

(8) Let M be a non-empty many sorted set indexed by I and let B be a locally-finite many sorted subset of M. Then there exists a non-empty locally-finite many sorted subset C of M such that $B \subseteq C$.

(9) For all many sorted sets A, B indexed by I and for all many sorted functions F, G from A into $\{B\}$ holds $F = G$.

(10) For every non-empty many sorted set A indexed by I and for every many sorted set B indexed by I holds every many sorted function from A into $\{B\}$ is onto.

(11) Let A be a many sorted set indexed by I and let B be a non-empty many sorted set indexed by I. Then every many sorted function from $\{A\}$ into B is "1-1".

(12) For every non-empty many sorted set X indexed by the carrier of S holds $\mathrm{Reverse}(X)$ is "1-1".

(13) For every non-empty locally-finite many sorted set A indexed by I holds there exists many sorted function from $I \longmapsto \mathbb{N}$ into A which is onto.

(14) Let S be a non empty many sorted signature, and let A be a non-empty algebra over S, and let f, g be elements of $\prod$(the sorts of A). Suppose that for every set i holds (proj(the sorts of A, i))(f) = (proj(the sorts of A, i))(g). Then $f = g$.

(15) Let I be a non empty set, and let s be an element of the carrier of S, and let A be an algebra family of I over S, and let f, g be elements of $\prod \mathrm{Carrier}(A, s)$. If for every element a of I holds $(\mathrm{proj}(\mathrm{Carrier}(A, s), a))(f) = (\mathrm{proj}(\mathrm{Carrier}(A, s), a))(g)$, then $f = g$.

(16) Let A, B be non-empty algebras over S, and let C be a strict non-empty subalgebra of A, and let h_1 be a many sorted function from B into C. Suppose h_1 is a homomorphism of B into C. Let h_2 be a many sorted function from B into A. If $h_1 = h_2$, then h_2 is a homomorphism of B into A.

(17) Let A, B be non-empty algebras over S and let F be a many sorted function from A into B. If F is a monomorphism of A into B, then A and $\mathrm{Im}\, F$ are isomorphic.

(18) Let A, B be non-empty algebras over S and let F be a many sorted function from A into B. Suppose F is onto. Let o be an operation symbol of S and let x be an element of $\mathrm{Args}(o, B)$. Then there exists an element y of $\mathrm{Args}(o, A)$ such that $F\#y = x$.

(19) Let A be a non-empty algebra over S, and let o be an operation symbol of S, and let x be an element of $\mathrm{Args}(o, A)$. Then $(\mathrm{Den}(o, A))(x) \in$ (the sorts of A)(the result sort of o).

(20) Let A, B, C be non-empty algebras over S, and let F_1 be a many sorted function from A into B, and let F_2 be a many sorted function from A into C. Suppose F_1 is an epimorphism of A onto B and F_2 is a homomorphism of A into C. Let G be a many sorted function from B into C. If $G \circ F_1 = F_2$, then G is a homomorphism of B into C.

In the sequel A, M will be many sorted sets indexed by I and B, C will be non-empty many sorted sets indexed by I.

Let I be a set, let A be a many sorted set indexed by I, let B, C be non-empty many sorted sets indexed by I, and let F be a many sorted function from A into $[\![B, C]\!]$. The functor $\mathrm{Mpr1}(F)$ yields a many sorted function from A into B and is defined as follows:

(Def. 1) For every set i such that $i \in I$ holds $(\mathrm{Mpr1}(F))(i) = \mathrm{pr1}(F(i))$.

The functor $\mathrm{Mpr2}(F)$ yielding a many sorted function from A into C is defined by:

(Def. 2) For every set i such that $i \in I$ holds $(\mathrm{Mpr2}(F))(i) = \mathrm{pr2}(F(i))$.

One can prove the following four propositions:

(21) For every many sorted function F from A into $[\![I \longmapsto \{a\}, I \longmapsto \{a\}]\!]$ holds $\mathrm{Mpr1}(F) = \mathrm{Mpr2}(F)$.

(22) For every many sorted function F from A into $[\![B, C]\!]$ such that F is onto holds $\mathrm{Mpr1}(F)$ is onto.

(23) For every many sorted function F from A into $[\![B, C]\!]$ such that F is onto holds $\mathrm{Mpr2}(F)$ is onto.

(24) Let F be a many sorted function from A into $[\![B, C]\!]$. If $M \in \mathrm{dom}_\kappa F(\kappa)$, then for every set i such that $i \in I$ holds $(F \upharpoonleft M)(i) = \langle ((\mathrm{Mpr1}(F)) \upharpoonleft M)(i), ((\mathrm{Mpr2}(F)) \upharpoonleft M)(i) \rangle$.

2. On the Trivial Many Sorted Algebras

Let S be a non empty many sorted signature. Note that the sorts of the trivial algebra of S is locally-finite and non-empty.

Let S be a non empty many sorted signature. Note that the trivial algebra of S is locally-finite and non-empty.

We now state three propositions:

(25) Let A be a non-empty algebra over S, and let F be a many sorted function from A into the trivial algebra of S, and let o be an operation symbol of S, and let x be an element of $\mathrm{Args}(o, A)$. Then F(the result sort of o)$((\mathrm{Den}(o, A))(x)) = 0$ and $(\mathrm{Den}(o,$ the trivial algebra of $S))(F\#x) = 0$.

(26) For every non-empty algebra A over S holds every many sorted function from A into the trivial algebra of S is an epimorphism of A onto the trivial algebra of S.

(27) Let A be an algebra over S. Given a many sorted set X indexed by the carrier of S such that the sorts of $A = \{X\}$. Then A and the trivial algebra of S are isomorphic.

3. On the Many Sorted Congruences

One can prove the following propositions:

(28) For every non-empty algebra A over S holds every congruence of A is a many sorted subset of $[\![$the sorts of A, the sorts of A $]\!]$.

(29) Let A be a non-empty algebra over S, and let R be a subset of the carrier of $\mathrm{CongrLatt}(A)$, and let F be a family of many sorted subsets of $[\![$the sorts of A, the sorts of A $]\!]$. If $R = F$, then $\bigcap |{:}F{:}|$ is a congruence of A.

(30) Let A be a non-empty algebra over S and let C be a congruence of A. Suppose $C = [\![$the sorts of A, the sorts of A $]\!]$. Then the sorts of $\mathrm{QuotMSAlg}(C) = \{$the sorts of A $\}$.

(31) Let A, B be non-empty algebras over S and let F be a many sorted function from A into B. If F is a homomorphism of A into B, then $\mathrm{MSHomQuot}(F) \circ \mathrm{MSNatHom}(A, \mathrm{Congruence}(F)) = F$.

(32) Let A be a non-empty algebra over S, and let C be a congruence of A, and let s be a sort symbol of S, and let a be an element of (the sorts of QuotMSAlg(C))(s). Then there exists an element x of (the sorts of A)(s) such that $a = [x]_C$.

(33) Let A be an algebra over S and let C_1, C_2 be equivalence many sorted relations of A. Suppose $C_1 \subseteq C_2$. Let i be an element of the carrier of S and let x, y be elements of (the sorts of A)(i). If $\langle x, y\rangle \in C_1(i)$, then $[x]_{(C_1)} \subseteq [y]_{(C_2)}$ and if A is non-empty, then $[y]_{(C_1)} \subseteq [x]_{(C_2)}$.

(34) Let A be a non-empty algebra over S, and let C be a congruence of A, and let s be a sort symbol of S, and let x, y be elements of (the sorts of A)(s). Then $(\mathrm{MSNatHom}(A, C))(s)(x) = (\mathrm{MSNatHom}(A, C))(s)(y)$ if and only if $\langle x, y\rangle \in C(s)$.

(35) Let A be a non-empty algebra over S, and let C_1, C_2 be congruences of A, and let G be a many sorted function from QuotMSAlg(C_1) into QuotMSAlg(C_2). Suppose that for every element i of the carrier of S and for every element x of (the sorts of QuotMSAlg(C_1))(i) and for every element x_1 of (the sorts of A)(i) such that $x = [x_1]_{(C_1)}$ holds $G(i)(x) = [x_1]_{(C_2)}$. Then $G \circ \mathrm{MSNatHom}(A, C_1) = \mathrm{MSNatHom}(A, C_2)$.

(36) Let A be a non-empty algebra over S, and let C_1, C_2 be congruences of A, and let G be a many sorted function from QuotMSAlg(C_1) into QuotMSAlg(C_2). Suppose that for every element i of the carrier of S and for every element x of (the sorts of QuotMSAlg(C_1))(i) and for every element x_1 of (the sorts of A)(i) such that $x = [x_1]_{(C_1)}$ holds $G(i)(x) = [x_1]_{(C_2)}$. Then G is an epimorphism of QuotMSAlg(C_1) onto QuotMSAlg(C_2).

(37) Let A, B be non-empty algebras over S and let F be a many sorted function from A into B. Suppose F is a homomorphism of A into B. Let C_1 be a congruence of A. Suppose $C_1 \subseteq \mathrm{Congruence}(F)$. Then there exists a many sorted function H from QuotMSAlg(C_1) into B such that H is a homomorphism of QuotMSAlg(C_1) into B and $F = H \circ \mathrm{MSNatHom}(A, C_1)$.

References

[1] Grzegorz Bancerek. Joining of decorated trees. *Formalized Mathematics*, 4(**1**):77–82, 1993.

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

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

[4] Grzegorz Bancerek and Piotr Rudnicki. On defining functions on trees. *Formalized Mathematics*, 4(**1**):91–101, 1993.

[5] Ewa Burakowska. Subalgebras of many sorted algebra. Lattice of subalgebras. *Formalized Mathematics*, 5(**1**):47–54, 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] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.

[9] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.

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

[11] Patricia L. Carlson and Grzegorz Bancerek. Context-free grammar - Part 1. *Formalized Mathematics*, 2(**5**):683–687, 1991.

[12] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.

[13] Mariusz Giero. More on products of many sorted algebras. *Formalized Mathematics*, 5(**4**):621–626, 1996.

[14] Artur Korniłowicz. Certain facts about families of subsets of many sorted sets. *Formalized Mathematics*, 5(**3**):451–456, 1996.

[15] Artur Korniłowicz. Extensions of mappings on generator set. *Formalized Mathematics*, 5(**2**):269–272, 1996.

[16] Artur Korniłowicz. On the closure operator and the closure system of many sorted sets. *Formalized Mathematics*, 5(**4**):543–551, 1996.

[17] Artur Korniłowicz. On the group of automorphisms of universal algebra & many sorted algebra. *Formalized Mathematics*, 5(**2**):221–226, 1996.

[18] Artur Korniłowicz. On the many sorted closure operator and the many sorted closure system. *Formalized Mathematics*, 5(**4**):529–536, 1996.

[19] Artur Korniłowicz. Some basic properties of many sorted sets. *Formalized Mathematics*, 5(**3**):395–399, 1996.

[20] Małgorzata Korolkiewicz. Homomorphisms of many sorted algebras. *Formalized Mathematics*, 5(**1**):61–65, 1996.

[21] Małgorzata Korolkiewicz. Many sorted quotient algebra. *Formalized Mathematics*, 5(**1**):79–84, 1996.

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

[23] Beata Madras. Products of many sorted algebras. *Formalized Mathematics*, 5(**1**):55–60, 1996.

[24] Robert Milewski. Lattice of congruences in many sorted algebra. *Formalized Mathematics*, 5(**4**):479–483, 1996.

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

[26] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, II. *Formalized Mathematics*, 5(**2**):215–220, 1996.

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

[28] Beata Perkowska. Free many sorted universal algebra. *Formalized Mathematics*, 5(**1**):67–74, 1996.

[29] Konrad Raczkowski and Paweł Sadowski. Equivalence relations and classes of abstraction. *Formalized Mathematics*, 1(**3**):441–444, 1990.

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

[31] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.

[32] Andrzej Trybulec. Function domains and Frænkel operator. *Formalized Mathematics*, 1(**3**):495–500, 1990.

[33] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.

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

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

[36] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.

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

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

[39] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[40] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.
[41] Edmund Woronowicz and Anna Zalewska. Properties of binary relations. *Formalized Mathematics*, 1(**1**):85–89, 1990.

Received June 11, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Examples of Category Structures

Adam Grabowski
Warsaw University
Białystok

Summary. This article contains definitions of two category structures: the category of many sorted signatures and the category of many sorted algebras. Some facts about these structures are proved.

MML Identifier: `MSINST_1`.

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

1. Category of Many Sorted Signatures

In this paper A denotes a non empty set, S denotes a non void non empty many sorted signature, and x denotes a set.

Let us consider A. The functor MSSCat(A) yields a strict non empty category structure and is defined by the conditions (Def. 1).

(Def. 1) (i) The carrier of MSSCat(A) = MSS-set(A),

(ii) for all elements i, j of MSS-set(A) holds (the arrows of MSSCat(A))(i, j) = MSS-morph(i, j), and

(iii) for all objects i, j, k of MSSCat(A) such that $i \in$ MSS-set(A) and $j \in$ MSS-set(A) and $k \in$ MSS-set(A) and for all functions f_1, f_2, g_1, g_2 such that $\langle f_1, f_2\rangle \in$ (the arrows of MSSCat(A))(i, j) and $\langle g_1, g_2\rangle \in$ (the arrows of MSSCat(A))(j, k) holds (the composition of MSSCat(A))$(i, j, k)(\langle g_1, g_2\rangle, \langle f_1, f_2\rangle) = \langle g_1 \cdot f_1, g_2 \cdot f_2\rangle$.

Let us consider A. Note that MSSCat(A) is transitive and associative and has units.

The following proposition is true

ISSN 1426–2630

(1) For every category C such that $C = \mathrm{MSSCat}(A)$ holds every object of C is a non empty non void many sorted signature.

Let us consider S. Note that there exists an algebra over S which is strict and feasible.

Let us consider S, A. The functor MSAlg_set(S, A) is defined by the condition (Def. 2).

(Def. 2) $x \in$ MSAlg_set(S, A) if and only if there exists a strict feasible algebra M over S such that $x = M$ and for every component C of the sorts of M holds $C \subseteq A$.

Let us consider S, A. Observe that MSAlg_set(S, A) is non empty.

2. Category of Many Sorted Algebras

In the sequel o is an operation symbol of S.

One can prove the following four propositions:

(2) Let x be an algebra over S. Suppose $x \in$ MSAlg_set(S, A). Then the sorts of $x \in (2^A)^{\text{the carrier of } S}$ and the characteristics of $x \in ((\mathbb{N} \dot{\rightarrow} A) \dot{\rightarrow} A)^{\text{the operation symbols of } S}$.

(3) Let U_1, U_2 be algebras over S. Suppose the sorts of U_1 is transformable to the sorts of U_2 and $\mathrm{Args}(o, U_1) \neq \emptyset$. Then $\mathrm{Args}(o, U_2) \neq \emptyset$.

(4) Let U_1, U_2, U_3 be feasible algebras over S, and let F be a many sorted function from U_1 into U_2, and let G be a many sorted function from U_2 into U_3, and let x be an element of $\mathrm{Args}(o, U_1)$. Suppose that

(i) $\mathrm{Args}(o, U_1) \neq \emptyset$,

(ii) the sorts of U_1 is transformable to the sorts of U_2, and

(iii) the sorts of U_2 is transformable to the sorts of U_3.

Then there exists a many sorted function G_1 from U_1 into U_3 such that $G_1 = G \circ F$ and $G_1 \# x = G \# (F \# x)$.

(5) Let U_1, U_2, U_3 be feasible algebras over S, and let F be a many sorted function from U_1 into U_2, and let G be a many sorted function from U_2 into U_3. Suppose that

(i) the sorts of U_1 is transformable to the sorts of U_2,

(ii) the sorts of U_2 is transformable to the sorts of U_3,

(iii) F is a homomorphism of U_1 into U_2, and

(iv) G is a homomorphism of U_2 into U_3.

Then there exists a many sorted function G_1 from U_1 into U_3 such that $G_1 = G \circ F$ and G_1 is a homomorphism of U_1 into U_3.

Let us consider S, A and let i, j be sets. Let us assume that $i \in$ MSAlg_set(S, A) and $j \in$ MSAlg_set(S, A). The functor MSAlg_morph(S, A, i, j) is defined by the condition (Def. 3).

(Def. 3) $x \in$ MSAlg_morph(S, A, i, j) if and only if there exist strict feasible algebras M, N over S and there exists a many sorted function f from M

into N such that $M = i$ and $N = j$ and $f = x$ and the sorts of M is transformable to the sorts of N and f is a homomorphism of M into N.

Let us consider S, A. The functor MSAlgCat(S, A) yields a strict non empty category structure and is defined by the conditions (Def. 4).

(Def. 4) (i) The carrier of MSAlgCat(S, A) = MSAlg_set(S, A),

(ii) for all elements i, j of MSAlg_set(S, A) holds (the arrows of MSAlgCat(S, A))(i, j) = MSAlg_morph(S, A, i, j), and

(iii) for all objects i, j, k of MSAlgCat(S, A) and for all function yielding functions f, g such that $f \in$ (the arrows of MSAlgCat(S, A))(i, j) and $g \in$ (the arrows of MSAlgCat(S, A))(j, k) holds (the composition of MSAlgCat(S, A))$(i, j, k)(g, f) = g \circ f$.

Let us consider S, A. One can verify that MSAlgCat(S, A) is transitive and associative and has units.

One can prove the following proposition

(6) For every category C such that C = MSAlgCat(S, A) holds every object of C is a strict feasible algebra over S.

References

[1] Grzegorz Bancerek. Cartesian product of functions. *Formalized Mathematics*, 2(**4**):547–552, 1991.
[2] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[3] Grzegorz Bancerek. Minimal signature for partial algebra. *Formalized Mathematics*, 5(**3**):405–414, 1996.
[4] Grzegorz Bancerek. Translations, endomorphisms, and stable equational theories. *Formalized Mathematics*, 5(**4**):553–564, 1996.
[5] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(**1**):175–180, 1990.
[6] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.
[7] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[8] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[9] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.
[10] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[11] Adam Grabowski. Inverse limits of many sorted algebras. *Formalized Mathematics*, 6(**1**):5–8, 1997.
[12] Artur Korniłowicz. Extensions of mappings on generator set. *Formalized Mathematics*, 5(**2**):269–272, 1996.
[13] Artur Korniłowicz. On the group of automorphisms of universal algebra & many sorted algebra. *Formalized Mathematics*, 5(**2**):221–226, 1996.
[14] Małgorzata Korolkiewicz. Homomorphisms of many sorted algebras. *Formalized Mathematics*, 5(**1**):61–65, 1996.
[15] Beata Madras. Product of family of universal algebras. *Formalized Mathematics*, 4(**1**):103–108, 1993.
[16] Beata Madras. Products of many sorted algebras. *Formalized Mathematics*, 5(**1**):55–60, 1996.
[17] Michał Muzalewski and Wojciech Skaba. N-tuples and Cartesian products for n=5. *Formalized Mathematics*, 2(**2**):255–258, 1991.
[18] Michał Muzalewski and Wojciech Skaba. Three-argument operations and four-argument operations. *Formalized Mathematics*, 2(**2**):221–224, 1991.

[19] Andrzej Trybulec. Categories without uniqueness of **cod** and **dom**. *Formalized Mathematics*, 5(**2**):259–267, 1996.
[20] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[21] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[22] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[23] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[24] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[25] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received June 11, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

On the Compositions of Macro Instructions. Part I

Andrzej Trybulec
Warsaw University
Białystok

Yatsuka Nakamura
Shinshu University
Nagano

Noriko Asamoto
Ochanomizu University
Tokyo

MML Identifier: `SCMFSA6A`.

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

1. Preliminaries

One can prove the following propositions:

(1) For all functions f, g and for all sets x, y such that $g \subseteq f$ and $x \notin \operatorname{dom} g$ holds $g \subseteq f +\cdot (x, y)$.

(2) For all functions f, g and for every set A such that $f \restriction A = g \restriction A$ and f and g are equal outside A holds $f = g$.

(3) For every function f and for all sets a, b, A such that $a \in A$ holds f and $f +\cdot (a, b)$ are equal outside A.

(4) For every function f and for all sets a, b, A holds $a \in A$ or $(f +\cdot (a, b)) \restriction A = f \restriction A$.

(5) For all functions f, g and for all sets a, b, A such that $f \restriction A = g \restriction A$ holds $(f +\cdot (a, b)) \restriction A = (g +\cdot (a, b)) \restriction A$.

(6) For all functions f, g, h such that $f \subseteq h$ and $g \subseteq h$ holds $f +\cdot g \subseteq h$.

(7) For arbitrary a, b and for every function f holds $a \dot{\longmapsto} b \subseteq f$ iff $a \in \operatorname{dom} f$ and $f(a) = b$.

(8) For every function f and for every set A holds $\operatorname{dom}(f \restriction (\operatorname{dom} f \setminus A)) = \operatorname{dom} f \setminus A$.

ISSN 1426–2630

(9) Let f, g be functions and let D be a set. Suppose $D \subseteq \operatorname{dom} f$ and $D \subseteq \operatorname{dom} g$. Then $f \restriction D = g \restriction D$ if and only if for arbitrary x such that $x \in D$ holds $f(x) = g(x)$.

(10) For every function f and for every set D holds $f \restriction D = f \restriction (\operatorname{dom} f \cap D)$.

(11) Let f, g, h be functions and let A be a set. Suppose f and g are equal outside A. Then $f{+}\cdot h$ and $g{+}\cdot h$ are equal outside A.

(12) Let f, g, h be functions and let A be a set. Suppose f and g are equal outside A. Then $h{+}\cdot f$ and $h{+}\cdot g$ are equal outside A.

(13) For all functions f, g, h holds $f{+}\cdot h = g{+}\cdot h$ iff f and g are equal outside $\operatorname{dom} h$.

2. Macroinstructions

A macro instruction is an initial programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$.

We follow a convention: m, n denote natural numbers, i, j, k denote instructions of $\mathbf{SCM}_{\mathrm{FSA}}$, and I, J, K denote macro instructions.

Let I be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. The functor $\mathrm{Directed}(I)$ yields a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and is defined by:

(Def. 1) $\mathrm{Directed}(I) = (\mathrm{id}_{(\text{the instructions of } \mathbf{SCM}_{\mathrm{FSA}})}{+}\cdot(\mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}} \dot{\longmapsto} \mathrm{goto}\ \mathrm{insloc}(\operatorname{card} I))) \cdot I$.

The following proposition is true

(14) $\operatorname{dom} \mathrm{Directed}(I) = \operatorname{dom} I$.

Let I be a macro instruction. Note that $\mathrm{Directed}(I)$ is initial.

Let us consider i. The functor $\mathrm{Macro}(i)$ yields a macro instruction and is defined by:

(Def. 2) $\mathrm{Macro}(i) = [\mathrm{insloc}(0) \longmapsto i, \mathrm{insloc}(1) \longmapsto \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}]$.

Let us consider i. One can check that $\mathrm{Macro}(i)$ is non empty.

We now state the proposition

(15) For every macro instruction P and for every n holds $n < \operatorname{card} P$ iff $\mathrm{insloc}(n) \in \operatorname{dom} P$.

Let I be an initial finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. Observe that $\mathrm{ProgramPart}(I)$ is initial.

One can prove the following propositions:

(16) $\operatorname{dom} I$ misses $\operatorname{dom} \mathrm{ProgramPart}(\mathrm{Relocated}(J, \operatorname{card} I))$.

(17) For every programmed finite partial state I of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\operatorname{card} \mathrm{ProgramPart}(\mathrm{Relocated}(I, m)) = \operatorname{card} I$.

(18) $\mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}} \notin \operatorname{rng} \mathrm{Directed}(I)$.

(19) $\mathrm{ProgramPart}(\mathrm{Relocated}(\mathrm{Directed}(I), m)) = (\mathrm{id}_{(\text{the instructions of } \mathbf{SCM}_{\mathrm{FSA}})} {+}\cdot(\mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}} \dot{\longmapsto} \mathrm{goto}\ \ \mathrm{insloc}(m{+}\operatorname{card} I))) \cdot \mathrm{ProgramPart}(\mathrm{Relocated}(I, m))$.

(20) For all finite partial states I, J of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{ProgramPart}(I{+}\cdot J) = \mathrm{ProgramPart}(I){+}\cdot \mathrm{ProgramPart}(J)$.

(21) For all finite partial states I, J of $\mathbf{SCM}_{\mathrm{FSA}}$ holds ProgramPart (Relocated($I+\cdot J,n$)) = ProgramPart(Relocated(I,n)) $+\cdot$ ProgramPart(Relocated(J,n)).

(22) ProgramPart(Relocated(ProgramPart(Relocated(I,m)), n)) = ProgramPart(Relocated($I,m+n$)).

In the sequel s, s_1, s_2 denote states of $\mathbf{SCM}_{\mathrm{FSA}}$.

Let us consider I. The functor Initialized(I) yields a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and is defined by:

(Def. 3) Initialized(I) = $I+\cdot$(intloc(0)$\longmapsto$1)$+\cdot$ Start-At(insloc(0)).

Next we state a number of propositions:

(23) InsCode(i) $\in \{0,6,7,8\}$ or (Exec(i,s))($\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}$) = Next($\mathbf{IC}_s$).

(24) $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in$ dom Initialized(I).

(25) $\mathbf{IC}_{\mathrm{Initialized}(I)}$ = insloc(0).

(26) $I \subseteq$ Initialized(I).

(27) s and $s+\cdot I$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(28) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $\mathbf{IC}_{(s_1)} = \mathbf{IC}_{(s_2)}$ and for every integer location a holds $s_1(a) = s_2(a)$ and for every finite sequence location f holds $s_1(f) = s_2(f)$. Then s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(29) If s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$, then $\mathbf{IC}_{(s_1)} = \mathbf{IC}_{(s_2)}$.

(30) Suppose s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$. Let a be an integer location. Then $s_1(a) = s_2(a)$.

(31) Suppose s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$. Let f be a finite sequence location. Then $s_1(f) = s_2(f)$.

(32) Suppose s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$. Then Exec($i,s_1$) and Exec($i,s_2$) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(33) Initialized(I) $\restriction$ (the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$) = I.

The scheme *SCMFSAEx* deals with a unary functor $\mathcal{F}$ yielding an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, a unary functor $\mathcal{G}$ yielding an integer, a unary functor $\mathcal{H}$ yielding a finite sequence of elements of $\mathbb{Z}$, and an instruction-location $\mathcal{A}$ of $\mathbf{SCM}_{\mathrm{FSA}}$, and states that:

There exists a state S of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $\mathbf{IC}_S = \mathcal{A}$ and for every natural number i holds S(insloc(i)) = $\mathcal{F}(i)$ and S(intloc(i)) = $\mathcal{G}(i)$ and S(fsloc(i)) = $\mathcal{H}(i)$

for all values of the parameters.

One can prove the following propositions:

(34) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ holds dom s = Int-Locations $\cup$ FinSeq-Locations $\cup \{\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\} \cup$ the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(35) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let x be arbitrary. Suppose $x \in$ dom s. Then

(i) x is an integer location or a finite sequence location, or
(ii) $x = \mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}$, or
(iii) x is an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$.

(36) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Then for every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $s_1(l) = s_2(l)$ if and only if $s_1 \restriction$ (the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$) $= s_2 \restriction$ (the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$).

(37) For every instruction-location i of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $i \notin \text{Int-Locations} \cup \text{FinSeq-Locations}$ and $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \notin \text{Int-Locations} \cup \text{FinSeq-Locations}$.

(38) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Then for every integer location a holds $s_1(a) = s_2(a)$ and for every finite sequence location f holds $s_1(f) = s_2(f)$ if and only if $s_1 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = s_2 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(39) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.
Then $s_1 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = s_2 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(40) For all states s, s_3 of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every set A holds $(s_3 {+\cdot}\, s \restriction A) \restriction A = s \restriction A$.

(41) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$, and let n be a natural number, and let i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $\mathbf{IC}_{(s_1)} + n = \mathbf{IC}_{(s_2)}$ and $s_1 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = s_2 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$. Then $\mathbf{IC}_{\mathrm{Exec}(i,s_1)} + n = \mathbf{IC}_{\mathrm{Exec}(\mathrm{IncAddr}(i,n),s_2)}$ and $\mathrm{Exec}(i, s_1) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = \mathrm{Exec}(\mathrm{IncAddr}(i, n), s_2) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(42) For all macro instructions I, J holds I and J are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(43) For every macro instruction I holds $\mathrm{dom}\,\mathrm{Initialized}(I) = \mathrm{dom}\, I \cup \{\mathrm{intloc}(0)\} \cup \{\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\}$.

(44) For every macro instruction I and for arbitrary x such that $x \in \mathrm{dom}\,\mathrm{Initialized}(I)$ holds $x \in \mathrm{dom}\, I$ or $x = \mathrm{intloc}(0)$ or $x = \mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(45) For every macro instruction I holds $\mathrm{intloc}(0) \in \mathrm{dom}\,\mathrm{Initialized}(I)$.

(46) For every macro instruction I holds $(\mathrm{Initialized}(I))(\mathrm{intloc}(0)) = 1$ and $(\mathrm{Initialized}(I))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{insloc}(0)$.

(47) For every macro instruction I holds $\mathrm{intloc}(0) \notin \mathrm{dom}\, I$ and $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \notin \mathrm{dom}\, I$.

(48) For every macro instruction I and for every integer location a such that $a \neq \mathrm{intloc}(0)$ holds $a \notin \mathrm{dom}\,\mathrm{Initialized}(I)$.

(49) For every macro instruction I and for every finite sequence location f holds $f \notin \mathrm{dom}\,\mathrm{Initialized}(I)$.

(50) For every macro instruction I and for arbitrary x such that $x \in \mathrm{dom}\, I$ holds $I(x) = (\mathrm{Initialized}(I))(x)$.

(51) For all macro instructions I, J and for every state s of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{Initialized}(J) \subseteq s$ holds $s{+}\cdot \mathrm{Initialized}(I) = s{+}\cdot I$.

(52) For all macro instructions I, J and for every state s of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{Initialized}(J) \subseteq s$ holds $\mathrm{Initialized}(I) \subseteq s{+}\cdot I$.

(53) Let I, J be macro instructions and let s be a state of $\mathbf{SCM}_{\rm FSA}$. Then $s{+}\cdot \mathrm{Initialized}(I)$ and $s{+}\cdot \mathrm{Initialized}(J)$ are equal outside the instruction locations of $\mathbf{SCM}_{\rm FSA}$.

3. The composition of macroinstructions

Let I, J be macro instructions. The functor $I;J$ yields a macro instruction and is defined by:

(Def. 4) $I;J = \mathrm{Directed}(I){+}\cdot \mathrm{ProgramPart}(\mathrm{Relocated}(J, \mathrm{card}\, I))$.

Let I, J be macro instructions. Note that $I;J$ is initial.

Next we state several propositions:

(54) Let I, J be macro instructions and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. If $l \in \mathrm{dom}\, I$ and $I(l) \neq \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}$, then $(I;J)(l) = I(l)$.

(55) For all macro instructions I, J holds $\mathrm{Directed}(I) \subseteq I;J$.

(56) For all macro instructions I, J holds $\mathrm{dom}\, I \subseteq \mathrm{dom}(I;J)$.

(57) For all macro instructions I, J holds $I{+}\cdot(I;J) = I;J$.

(58) For all macro instructions I, J holds $\mathrm{Initialized}(I){+}\cdot(I;J) = \mathrm{Initialized}(I;J)$.

4. The compostion of instruction and macroinstructions

Let us consider i, J. The functor $i;J$ yielding a macro instruction is defined as follows:

(Def. 5) $i;J = \mathrm{Macro}(i);J$.

Let us consider I, j. The functor $I;j$ yields a macro instruction and is defined by:

(Def. 6) $I;j = I; \mathrm{Macro}(j)$.

Let us consider i, j. The functor $i;j$ yields a macro instruction and is defined by:

(Def. 7) $i;j = \mathrm{Macro}(i); \mathrm{Macro}(j)$.

The following propositions are true:

(59) $i;j = \mathrm{Macro}(i);j$.

(60) $i;j = i; \mathrm{Macro}(j)$.

(61) $\mathrm{card}(I;J) = \mathrm{card}\, I + \mathrm{card}\, J$.

(62) $(I;J);K = I;(J;K)$.

(63) $(I;J);k = I;(J;k)$.

(64) $(I;j);K = I;(j;K)$.

(65) $(I;j);k = I;(j;k)$.

(66) $(i;J);K = i;(J;K)$.

(67) $(i;J);k = i;(J;k)$.

(68) $(i;j);K = i;(j;K)$.

(69) $(i;j);k = i;(j;k)$.

References

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

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

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

[4] Grzegorz Bancerek and Piotr Rudnicki. Development of terminology for **SCM**. *Formalized Mathematics*, 4(**1**):61–67, 1993.

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

[6] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.

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

[10] 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.

[11] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.

[12] Czesław Byliński. Products and coproducts in categories. *Formalized Mathematics*, 2(**5**):701–709, 1991.

[13] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.

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

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

[16] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.

[17] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.

[18] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.

[19] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.

[20] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.

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

[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 and Yatsuka Nakamura. Relocability for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):583–586, 1996.

[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, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\mathrm{FSA}}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[26] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[27] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 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.

Received June 20, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Memory Handling for $\mathbf{SCM}_{\mathrm{FSA}}$ [1]

Piotr Rudnicki
University of Alberta
Edmonton

Andrzej Trybulec
Warsaw University
Białystok

Summary. We introduce some terminology for reasoning about memory used in programs in general and in macro instructions (introduced in [26]) in particular. The usage of integer locations and of finite sequence locations by a program is treated separately. We define some functors for selecting memory locations needed for local (temporary) variables in macro instructions. Some semantic properties of the introduced notions are given in terms of executions of macro instructions.

MML Identifier: SF_MASTR.

The articles [21], [31], [19], [12], [30], [22], [14], [2], [28], [15], [20], [6], [13], [1], [3], [17], [11], [4], [7], [29], [32], [8], [9], [10], [5], [16], [25], [18], [27], [23], [24], and [26] provide the terminology and notation for this paper.

1. Preliminaries

One can prove the following three propositions:

(1) For all sets x, y, a and for every function f such that $f(x) = f(y)$ holds $f(a) = (f \cdot (\mathrm{id}_{\mathrm{dom}\, f} +\cdot (x, y)))(a)$.

(2) For all sets x, y and for every function f such that if $x \in \mathrm{dom}\, f$, then $y \in \mathrm{dom}\, f$ and $f(x) = f(y)$ holds $f = f \cdot (\mathrm{id}_{\mathrm{dom}\, f} +\cdot (x, y))$.

(3) For all sets A, B and for every function f from A into B holds $\mathrm{dom}\, f \subseteq A$.

Let A be a finite set and let B be a set. Note that every function from A into B is finite.

Let A be a finite set, let B be a set, and let f be a function from A into $\mathrm{Fin}\, B$. Observe that $\mathrm{Union}\, f$ is finite.

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

ISSN 1426–2630

In the sequel N will be a non empty set with non empty elements.

The following proposition is true

(4) Let S be a definite AMI over N and let p be a programmed finite partial state of S. Then $\operatorname{rng} p \subseteq$ the instructions of S.

Let us mention that Int-Locations is non empty.

Let us mention that FinSeq-Locations is non empty.

2. Uniqueness of instruction components

For simplicity we adopt the following rules: a, b, c, a_1, a_2, b_1, b_2 will be integer locations, l, l_1, l_2 will be instructions-locations of $\mathbf{SCM}_{\mathrm{FSA}}$, f, f_1, f_2 will be finite sequence locations, and i, j will be instructions of $\mathbf{SCM}_{\mathrm{FSA}}$.

The following propositions are true:

(5) If $a_1{:=}b_1 = a_2{:=}b_2$, then $a_1 = a_2$ and $b_1 = b_2$.

(6) If $\mathrm{AddTo}(a_1, b_1) = \mathrm{AddTo}(a_2, b_2)$, then $a_1 = a_2$ and $b_1 = b_2$.

(7) If $\mathrm{SubFrom}(a_1, b_1) = \mathrm{SubFrom}(a_2, b_2)$, then $a_1 = a_2$ and $b_1 = b_2$.

(8) If $\mathrm{MultBy}(a_1, b_1) = \mathrm{MultBy}(a_2, b_2)$, then $a_1 = a_2$ and $b_1 = b_2$.

(9) If $\mathrm{Divide}(a_1, b_1) = \mathrm{Divide}(a_2, b_2)$, then $a_1 = a_2$ and $b_1 = b_2$.

(10) If $\mathrm{goto}\ l_1 = \mathrm{goto}\ l_2$, then $l_1 = l_2$.

(11) If $\mathbf{if}\ a_1 = 0\ \mathbf{goto}\ l_1 = \mathbf{if}\ a_2 = 0\ \mathbf{goto}\ l_2$, then $a_1 = a_2$ and $l_1 = l_2$.

(12) If $\mathbf{if}\ a_1 > 0\ \mathbf{goto}\ l_1 = \mathbf{if}\ a_2 > 0\ \mathbf{goto}\ l_2$, then $a_1 = a_2$ and $l_1 = l_2$.

(13) If $b_1{:=}{f_1}_{a_1} = b_2{:=}{f_2}_{a_2}$, then $a_1 = a_2$ and $b_1 = b_2$ and $f_1 = f_2$.

(14) If ${f_1}_{a_1}{:=}b_1 = {f_2}_{a_2}{:=}b_2$, then $a_1 = a_2$ and $b_1 = b_2$ and $f_1 = f_2$.

(15) If $a_1{:=}\mathrm{len} f_1 = a_2{:=}\mathrm{len} f_2$, then $a_1 = a_2$ and $f_1 = f_2$.

(16) If $f_1{:=}\langle\underbrace{0, \ldots, 0}_{a_1}\rangle = f_2{:=}\langle\underbrace{0, \ldots, 0}_{a_2}\rangle$, then $a_1 = a_2$ and $f_1 = f_2$.

3. Integer locations used in macros

Let i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. The functor $\mathrm{UsedIntLoc}(i)$ yields an element of Fin Int-Locations and is defined as follows:

(Def. 1) (i) There exist integer locations a, b such that $i = a{:=}b$ or $i = \mathrm{AddTo}(a, b)$ or $i = \mathrm{SubFrom}(a, b)$ or $i = \mathrm{MultBy}(a, b)$ or $i = \mathrm{Divide}(a, b)$ but $\mathrm{UsedIntLoc}(i) = \{a, b\}$ if $\mathrm{InsCode}(i) \in \{1, 2, 3, 4, 5\}$,

(ii) there exists an integer location a and there exists an instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $i = \mathbf{if}\ a = 0\ \mathbf{goto}\ l$ or $i = \mathbf{if}\ a > 0\ \mathbf{goto}\ l$ but $\mathrm{UsedIntLoc}(i) = \{a\}$ if $\mathrm{InsCode}(i) = 7$ or $\mathrm{InsCode}(i) = 8$,

(iii) there exist integer locations a, b and there exists a finite sequence location f such that $i = b{:=}f_a$ or $i = f_a{:=}b$ but $\mathrm{UsedIntLoc}(i) = \{a, b\}$ if $\mathrm{InsCode}(i) = 9$ or $\mathrm{InsCode}(i) = 10$,

(iv) there exists an integer location a and there exists a finite sequence location f such that $i = a$:=len$f$ or $i = f{:=}\langle\underbrace{0,\dots,0}_{a}\rangle$ but UsedIntLoc$(i) = \{a\}$ if InsCode$(i) = 11$ or InsCode$(i) = 12$,

(v) UsedIntLoc$(i) = \emptyset$, otherwise.

Next we state several propositions:

(17) UsedIntLoc$(\mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \emptyset$.

(18) If $i = a{:=}b$ or $i = \mathrm{AddTo}(a, b)$ or $i = \mathrm{SubFrom}(a, b)$ or $i = \mathrm{MultBy}(a, b)$ or $i = \mathrm{Divide}(a, b)$, then UsedIntLoc$(i) = \{a, b\}$.

(19) UsedIntLoc(goto l) $= \emptyset$.

(20) If $i = \mathbf{if}\ a = 0\ \mathbf{goto}\ l$ or $i = \mathbf{if}\ a > 0\ \mathbf{goto}\ l$, then UsedIntLoc$(i) = \{a\}$.

(21) If $i = b{:=}f_a$ or $i = f_a{:=}b$, then UsedIntLoc$(i) = \{a, b\}$.

(22) If $i = a$:=len$f$ or $i = f{:=}\langle\underbrace{0,\dots,0}_{a}\rangle$, then UsedIntLoc$(i) = \{a\}$.

Let p be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. The functor UsedIntLoc$(p)$ yields a subset of Int-Locations and is defined by the condition (Def. 2).

(Def. 2) There exists a function U_1 from the instructions of $\mathbf{SCM}_{\mathrm{FSA}}$ into Fin Int-Locations such that for every instruction i of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $U_1(i) = $ UsedIntLoc(i) and UsedIntLoc$(p) = \mathrm{Union}(U_1 \cdot p)$.

Let p be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. Note that UsedIntLoc$(p)$ is finite.

We follow the rules: p, r denote programmed finite partial states of $\mathbf{SCM}_{\mathrm{FSA}}$, I, J denote macro instructions, and k, m, n denote natural numbers.

Next we state a number of propositions:

(23) If $i \in \mathrm{rng}\, p$, then UsedIntLoc$(i) \subseteq$ UsedIntLoc(p).

(24) UsedIntLoc$(p{+}{\cdot}r) \subseteq$ UsedIntLoc$(p) \cup$ UsedIntLoc(r).

(25) If $\mathrm{dom}\, p$ misses $\mathrm{dom}\, r$, then UsedIntLoc$(p{+}{\cdot}r) =$ UsedIntLoc$(p) \cup$ UsedIntLoc(r).

(26) UsedIntLoc$(p) =$ UsedIntLoc(Shift(p, k)).

(27) UsedIntLoc$(i) =$ UsedIntLoc(IncAddr(i, k)).

(28) UsedIntLoc$(p) =$ UsedIntLoc(IncAddr(p, k)).

(29) UsedIntLoc$(I) =$ UsedIntLoc(ProgramPart(Relocated(I, k))).

(30) UsedIntLoc$(I) =$ UsedIntLoc(Directed(I)).

(31) UsedIntLoc$(I;J) =$ UsedIntLoc$(I) \cup$ UsedIntLoc(J).

(32) UsedIntLoc(Macro(i)) $=$ UsedIntLoc(i).

(33) UsedIntLoc$(i;J) =$ UsedIntLoc$(i) \cup$ UsedIntLoc(J).

(34) UsedIntLoc$(I;j) =$ UsedIntLoc$(I) \cup$ UsedIntLoc(j).

(35) UsedIntLoc$(i;j) =$ UsedIntLoc$(i) \cup$ UsedIntLoc(j).

4. Finite sequence locations used in macros

Let i be an instruction of $\mathbf{SCM}_{\rm FSA}$. The functor $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(i)$ yielding an element of Fin FinSeq-Locations is defined by:

(Def. 3) (i) There exist integer locations a, b and there exists a finite sequence location f such that $i = b{:=}f_a$ or $i = f_a{:=}b$ but $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(i) = \{f\}$ if $\mathrm{InsCode}(i) = 9$ or $\mathrm{InsCode}(i) = 10$,

(ii) there exists an integer location a and there exists a finite sequence location f such that $i = a{:=}\mathrm{len} f$ or $i = f{:=}\langle\underbrace{0, \ldots, 0}_{a}\rangle$ but $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(i) = \{f\}$ if $\mathrm{InsCode}(i) = 11$ or $\mathrm{InsCode}(i) = 12$,

(iii) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(i) = \emptyset$, otherwise.

One can prove the following propositions:

(36) If $i = \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}$ or $i = a{:=}b$ or $i = \mathrm{AddTo}(a, b)$ or $i = \mathrm{SubFrom}(a, b)$ or $i = \mathrm{MultBy}(a, b)$ or $i = \mathrm{Divide}(a, b)$ or $i = \mathrm{goto}\ l$ or $i = \mathbf{if}\ a = 0\ \mathbf{goto}\ l$ or $i = \mathbf{if}\ a > 0\ \mathbf{goto}\ l$, then $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(i) = \emptyset$.

(37) If $i = b{:=}f_a$ or $i = f_a{:=}b$, then $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(i) = \{f\}$.

(38) If $i = a{:=}\mathrm{len} f$ or $i = f{:=}\langle\underbrace{0, \ldots, 0}_{a}\rangle$, then $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(i) = \{f\}$.

Let p be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$. The functor $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(p)$ yields a subset of FinSeq-Locations and is defined by the condition (Def. 4).

(Def. 4) There exists a function U_1 from the instructions of $\mathbf{SCM}_{\rm FSA}$ into Fin FinSeq-Locations such that for every instruction i of $\mathbf{SCM}_{\rm FSA}$ holds $U_1(i) = \mathrm{UsedInt}^{*}\,\mathrm{Loc}(i)$ and $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(p) = \mathrm{Union}(U_1 \cdot p)$.

Let p be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$. Note that $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(p)$ is finite.

The following propositions are true:

(39) If $i \in \mathrm{rng}\, p$, then $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(i) \subseteq \mathrm{UsedInt}^{*}\,\mathrm{Loc}(p)$.

(40) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(p{+}{\cdot}r) \subseteq \mathrm{UsedInt}^{*}\,\mathrm{Loc}(p) \cup \mathrm{UsedInt}^{*}\,\mathrm{Loc}(r)$.

(41) If $\mathrm{dom}\, p$ misses $\mathrm{dom}\, r$, then $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(p{+}{\cdot}r) = \mathrm{UsedInt}^{*}\,\mathrm{Loc}(p) \cup \mathrm{UsedInt}^{*}\,\mathrm{Loc}(r)$.

(42) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(p) = \mathrm{UsedInt}^{*}\,\mathrm{Loc}(\mathrm{Shift}(p, k))$.

(43) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(i) = \mathrm{UsedInt}^{*}\,\mathrm{Loc}(\mathrm{IncAddr}(i, k))$.

(44) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(p) = \mathrm{UsedInt}^{*}\,\mathrm{Loc}(\mathrm{IncAddr}(p, k))$.

(45) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(I) = \mathrm{UsedInt}^{*}\,\mathrm{Loc}(\mathrm{ProgramPart}(\mathrm{Relocated}(I, k)))$.

(46) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(I) = \mathrm{UsedInt}^{*}\,\mathrm{Loc}(\mathrm{Directed}(I))$.

(47) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(I;J) = \mathrm{UsedInt}^{*}\,\mathrm{Loc}(I) \cup \mathrm{UsedInt}^{*}\,\mathrm{Loc}(J)$.

(48) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(\mathrm{Macro}(i)) = \mathrm{UsedInt}^{*}\,\mathrm{Loc}(i)$.

(49) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(i;J) = \mathrm{UsedInt}^{*}\,\mathrm{Loc}(i) \cup \mathrm{UsedInt}^{*}\,\mathrm{Loc}(J)$.

(50) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(I;j) = \mathrm{UsedInt}^{*}\,\mathrm{Loc}(I) \cup \mathrm{UsedInt}^{*}\,\mathrm{Loc}(j)$.

(51) $\mathrm{UsedInt}^* \mathrm{Loc}(i;j) = \mathrm{UsedInt}^* \mathrm{Loc}(i) \cup \mathrm{UsedInt}^* \mathrm{Loc}(j)$.

5. Choosing an integer location not used in a program

Let I_1 be an integer location. We say that I_1 is read-only if and only if:

(Def. 5) $I_1 = \mathrm{intloc}(0)$.

We introduce I_1 is read-write as an antonym of I_1 is read-only.

Let us observe that intloc(0) is read-only.

One can check that there exists an integer location which is read-write.

In the sequel L will be a finite subset of Int-Locations.

Let L be a finite subset of Int-Locations. The functor FirstNotIn(L) yields an integer location and is defined by:

(Def. 6) There exists a non empty subset s_1 of $\mathbb{N}$ such that $\mathrm{FirstNotIn}(L) = \mathrm{intloc}(\min s_1)$ and $s_1 = \{k : k \text{ ranges over natural numbers}, \mathrm{intloc}(k) \notin L\}$.

Next we state two propositions:

(52) $\mathrm{FirstNotIn}(L) \notin L$.

(53) If $\mathrm{FirstNotIn}(L) = \mathrm{intloc}(m)$ and $\mathrm{intloc}(n) \notin L$, then $m \leq n$.

Let p be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. The functor FirstNotUsed(p) yields an integer location and is defined by:

(Def. 7) There exists a finite subset s_2 of Int-Locations such that $s_2 = \mathrm{UsedIntLoc}(p) \cup \{\mathrm{intloc}(0)\}$ and $\mathrm{FirstNotUsed}(p) = \mathrm{FirstNotIn}(s_2)$.

Let p be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. Observe that FirstNotUsed(p) is read-write.

We now state several propositions:

(54) $\mathrm{FirstNotUsed}(p) \notin \mathrm{UsedIntLoc}(p)$.

(55) If $a{:=}b \in \mathrm{rng}\, p$ or $\mathrm{AddTo}(a,b) \in \mathrm{rng}\, p$ or $\mathrm{SubFrom}(a,b) \in \mathrm{rng}\, p$ or $\mathrm{MultBy}(a,b) \in \mathrm{rng}\, p$ or $\mathrm{Divide}(a,b) \in \mathrm{rng}\, p$, then $\mathrm{FirstNotUsed}(p) \neq a$ and $\mathrm{FirstNotUsed}(p) \neq b$.

(56) If $\mathbf{if}\ a = 0\ \mathbf{goto}\ l \in \mathrm{rng}\, p$ or $\mathbf{if}\ a > 0\ \mathbf{goto}\ l \in \mathrm{rng}\, p$, then $\mathrm{FirstNotUsed}(p) \neq a$.

(57) If $b{:=}f_a \in \mathrm{rng}\, p$ or $f_a{:=}b \in \mathrm{rng}\, p$, then $\mathrm{FirstNotUsed}(p) \neq a$ and $\mathrm{FirstNotUsed}(p) \neq b$.

(58) If $a{:=}\mathrm{len} f \in \mathrm{rng}\, p$ or $f{:=}\langle \underbrace{0, \ldots, 0}_{a} \rangle \in \mathrm{rng}\, p$, then $\mathrm{FirstNotUsed}(p) \neq a$.

6. Choosing a finite sequence location not used in a program

In the sequel L is a finite subset of FinSeq-Locations.

Let L be a finite subset of FinSeq-Locations. The functor $\mathrm{First}^*\,\mathrm{NotIn}(L)$ yielding a finite sequence location is defined by:

(Def. 8) There exists a non empty subset s_1 of $\mathbb{N}$ such that $\mathrm{First}^*\,\mathrm{NotIn}(L) = \mathrm{fsloc}(\min s_1)$ and $s_1 = \{k : k$ ranges over natural numbers, $\mathrm{fsloc}(k) \notin L\}$.

We now state two propositions:

(59) $\mathrm{First}^*\,\mathrm{NotIn}(L) \notin L$.

(60) If $\mathrm{First}^*\,\mathrm{NotIn}(L) = \mathrm{fsloc}(m)$ and $\mathrm{fsloc}(n) \notin L$, then $m \leq n$.

Let p be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. The functor $\mathrm{First}^*\,\mathrm{NotUsed}(p)$ yields a finite sequence location and is defined by:

(Def. 9) There exists a finite subset s_2 of FinSeq-Locations such that $s_2 = \mathrm{UsedInt}^*\,\mathrm{Loc}(p)$ and $\mathrm{First}^*\,\mathrm{NotUsed}(p) = \mathrm{First}^*\,\mathrm{NotIn}(s_2)$.

One can prove the following propositions:

(61) $\mathrm{First}^*\,\mathrm{NotUsed}(p) \notin \mathrm{UsedInt}^*\,\mathrm{Loc}(p)$.

(62) If $b{:=}f_a \in \mathrm{rng}\, p$ or $f_a{:=}b \in \mathrm{rng}\, p$, then $\mathrm{First}^*\,\mathrm{NotUsed}(p) \neq f$.

(63) If $a{:=}\mathrm{len} f \in \mathrm{rng}\, p$ or $f{:=}\langle \underbrace{0, \ldots, 0}_{a} \rangle \in \mathrm{rng}\, p$, then $\mathrm{First}^*\,\mathrm{NotUsed}(p) \neq f$.

7. Semantics

In the sequel s, t will be states of $\mathbf{SCM}_{\mathrm{FSA}}$.

We now state a number of propositions:

(64) $\mathrm{dom}\, I \cap \mathrm{dom}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(n)) = \emptyset$.

(65) $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}(I +\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(n)))$.

(66) $(I +\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(n)))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{insloc}(n)$.

(67) If $I +\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(n)) \subseteq s$, then $\mathbf{IC}_s = \mathrm{insloc}(n)$.

(68) If $c \notin \mathrm{UsedIntLoc}(i)$, then $(\mathrm{Exec}(i, s))(c) = s(c)$.

(69) If $I +\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)) \subseteq s$ and for every m such that $m < n$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(m)} \in \mathrm{dom}\, I$ and $a \notin \mathrm{UsedIntLoc}(I)$, then $(\mathrm{Computation}(s))(n)(a) = s(a)$.

(70) If $f \notin \mathrm{UsedInt}^*\,\mathrm{Loc}(i)$, then $(\mathrm{Exec}(i, s))(f) = s(f)$.

(71) If $I +\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)) \subseteq s$ and for every m such that $m < n$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(m)} \in \mathrm{dom}\, I$ and $f \notin \mathrm{UsedInt}^*\,\mathrm{Loc}(I)$, then $(\mathrm{Computation}(s))(n)(f) = s(f)$.

(72) If $s \restriction \mathrm{UsedIntLoc}(i) = t \restriction \mathrm{UsedIntLoc}(i)$ and $s \restriction \mathrm{UsedInt}^*\,\mathrm{Loc}(i) = t \restriction \mathrm{UsedInt}^*\,\mathrm{Loc}(i)$ and $\mathbf{IC}_s = \mathbf{IC}_t$, then $\mathbf{IC}_{\mathrm{Exec}(i,s)} = \mathbf{IC}_{\mathrm{Exec}(i,t)}$ and $\mathrm{Exec}(i, s) \restriction \mathrm{UsedIntLoc}(i) = \mathrm{Exec}(i, t) \restriction \mathrm{UsedIntLoc}(i)$ and $\mathrm{Exec}(i, s) \restriction \mathrm{UsedInt}^*\,\mathrm{Loc}(i) = \mathrm{Exec}(i, t) \restriction \mathrm{UsedInt}^*\,\mathrm{Loc}(i)$.

(73) Suppose $I +\cdot \text{Start-At}(\text{insloc}(0)) \subseteq s$ and $I +\cdot \text{Start-At}(\text{insloc}(0)) \subseteq t$ and $s \restriction \text{UsedIntLoc}(I) = t \restriction \text{UsedIntLoc}(I)$ and $s \restriction \text{UsedInt}^* \text{Loc}(I) = t \restriction \text{UsedInt}^* \text{Loc}(I)$ and for every m such that $m < n$ holds $\mathbf{IC}_{(\text{Computation}(s))(m)} \in \text{dom}\, I$. Then

(i) for every m such that $m < n$ holds $\mathbf{IC}_{(\text{Computation}(t))(m)} \in \text{dom}\, I$, and

(ii) for every m such that $m \leq n$ holds $\mathbf{IC}_{(\text{Computation}(s))(m)} = \mathbf{IC}_{(\text{Computation}(t))(m)}$ and for every a such that $a \in \text{UsedIntLoc}(I)$ holds $(\text{Computation}(s))(m)(a) = (\text{Computation}(t))(m)(a)$ and for every f such that $f \in \text{UsedInt}^* \text{Loc}(I)$ holds $(\text{Computation}(s))(m)(f) = (\text{Computation}(t))(m)(f)$.

References

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

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

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

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

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

[6] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.

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

[10] 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.

[11] Czesław Byliński. Products and coproducts in categories. *Formalized Mathematics*, 2(**5**):701–709, 1991.

[12] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.

[13] Agata Darmochwał and Andrzej Trybulec. Similarity of formulae. *Formalized Mathematics*, 2(**5**):635–642, 1991.

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

[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] Andrzej Nędzusiak. σ-fields and probability. *Formalized Mathematics*, 1(**2**):401–407, 1990.

[18] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.

[19] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.

[20] Andrzej Trybulec. Semilattice operations on finite subsets. *Formalized Mathematics*, 1(**2**):369–376, 1990.

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

[22] Andrzej Trybulec and Agata Darmochwał. Boolean domains. *Formalized Mathematics*, 1(**1**):187–190, 1990.

[23] Andrzej Trybulec and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):571–576, 1996.
[24] Andrzej Trybulec and Yatsuka Nakamura. Relocability for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):583–586, 1996.
[25] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[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 July 18, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Some Topological Properties of Cells in R^2

Yatsuka Nakamura
Shinshu University
Nagano

Andrzej Trybulec
Warsaw University
Białystok

Summary. We examine the topological property of cells (rectangles) in a plane. First, some Fraenkel expressions of cells are shown. Second, it is proved that cells are closed. The last theorem asserts that the closure of the interior of a cell is the same as itself.

MML Identifier: GOBRD11.

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

We adopt the following convention: i, j, j_1, j_2 will be natural numbers, r, s, r_2, s_1, s_2 will be real numbers, and G_1 will be a non empty topological space.

Next we state two propositions:

(1) For every subset A of the carrier of G_1 and for every point p of G_1 such that $p \in A$ and A is connected holds $A \subseteq \mathrm{Component}(p)$.

(2) Let A, B, C be subsets of the carrier of G_1. Suppose C is a component of G_1 and $A \subseteq C$ and B is connected and $\overline{A} \cap \overline{B} \neq \emptyset$. Then $B \subseteq C$.

In the sequel G_2 denotes a non empty topological space.

Next we state three propositions:

(3) Let A, B be subsets of the carrier of G_2. Suppose A is a component of G_2 and B is a component of G_2. Then $A \cup B$ is a union of components of G_2.

(4) For all subsets B_1, B_2, V of the carrier of G_1 such that $V \neq \emptyset$ holds $\mathrm{Down}(B_1 \cup B_2, V) = \mathrm{Down}(B_1, V) \cup \mathrm{Down}(B_2, V)$.

(5) For all subsets B_1, B_2, V of the carrier of G_1 such that $V \neq \emptyset$ holds $\mathrm{Down}(B_1 \cap B_2, V) = \mathrm{Down}(B_1, V) \cap \mathrm{Down}(B_2, V)$.

In the sequel f will denote a non constant standard special circular sequence and G will denote a Go-board.

We now state a number of propositions:

ISSN 1426–2630

(6) $(\widetilde{\mathcal{L}}(f))^{\rm c} \neq \emptyset$.

(7) Given j_1, j_2. Suppose $j_1 = \mathrm{len}$ the Go-board of f and $j_2 = \mathrm{width}$ the Go-board of f. Then the carrier of $\mathcal{E}^2_{\rm T} = \bigcup\{\mathrm{cell}($the Go-board of f, $i,j) : i \leq j_1 \ \wedge \ j \leq j_2\}$.

(8) For all subsets P_1, P_2 of the carrier of $\mathcal{E}^2_{\rm T}$ such that $P_1 = \{[r,s] : s \leq s_1\}$ and $P_2 = \{[r_2,s_2] : s_2 > s_1\}$ holds $P_1 = -P_2$.

(9) For all subsets P_1, P_2 of the carrier of $\mathcal{E}^2_{\rm T}$ such that $P_1 = \{[r,s] : s \geq s_1\}$ and $P_2 = \{[r_2,s_2] : s_2 < s_1\}$ holds $P_1 = -P_2$.

(10) For all subsets P_1, P_2 of the carrier of $\mathcal{E}^2_{\rm T}$ such that $P_1 = \{[s,r] : s \geq s_1\}$ and $P_2 = \{[s_2,r_2] : s_2 < s_1\}$ holds $P_1 = -P_2$.

(11) For all subsets P_1, P_2 of the carrier of $\mathcal{E}^2_{\rm T}$ such that $P_1 = \{[s,r] : s \leq s_1\}$ and $P_2 = \{[s_2,r_2] : s_2 > s_1\}$ holds $P_1 = -P_2$.

(12) For every subset P of the carrier of $\mathcal{E}^2_{\rm T}$ and for every s_1 such that $P = \{[r,s] : s \leq s_1\}$ holds P is closed.

(13) For every subset P of the carrier of $\mathcal{E}^2_{\rm T}$ and for every s_1 such that $P = \{[r,s] : s_1 \leq s\}$ holds P is closed.

(14) For every subset P of the carrier of $\mathcal{E}^2_{\rm T}$ and for every s_1 such that $P = \{[s,r] : s \leq s_1\}$ holds P is closed.

(15) For every subset P of the carrier of $\mathcal{E}^2_{\rm T}$ and for every s_1 such that $P = \{[s,r] : s_1 \leq s\}$ holds P is closed.

(16) For every j holds $\mathrm{hstrip}(G,j)$ is closed.

(17) For every i holds $\mathrm{vstrip}(G,i)$ is closed.

(18) $\mathrm{vstrip}(G,0) = \{[r,s] : r \leq (G_{1,1})_{\mathbf{1}}\}$.

(19) $\mathrm{vstrip}(G,\mathrm{len}\,G) = \{[r,s] : (G_{\mathrm{len}\,G,1})_{\mathbf{1}} \leq r\}$.

(20) If $1 \leq i$ and $i < \mathrm{len}\,G$, then $\mathrm{vstrip}(G,i) = \{[r,s] : (G_{i,1})_{\mathbf{1}} \leq r \ \wedge \ r \leq (G_{i+1,1})_{\mathbf{1}}\}$.

(21) $\mathrm{hstrip}(G,0) = \{[r,s] : s \leq (G_{1,1})_{\mathbf{2}}\}$.

(22) $\mathrm{hstrip}(G,\mathrm{width}\,G) = \{[r,s] : (G_{1,\mathrm{width}\,G})_{\mathbf{2}} \leq s\}$.

(23) If $1 \leq j$ and $j < \mathrm{width}\,G$, then $\mathrm{hstrip}(G,j) = \{[r,s] : (G_{1,j})_{\mathbf{2}} \leq s \wedge s \leq (G_{1,j+1})_{\mathbf{2}}\}$.

(24) $\mathrm{cell}(G,0,0) = \{[r,s] : r \leq (G_{1,1})_{\mathbf{1}} \ \wedge \ s \leq (G_{1,1})_{\mathbf{2}}\}$.

(25) $\mathrm{cell}(G,0,\mathrm{width}\,G) = \{[r,s] : r \leq (G_{1,1})_{\mathbf{1}} \ \wedge \ (G_{1,\mathrm{width}\,G})_{\mathbf{2}} \leq s\}$.

(26) If $1 \leq j$ and $j < \mathrm{width}\,G$, then $\mathrm{cell}(G,0,j) = \{[r,s] : r \leq (G_{1,1})_{\mathbf{1}} \ \wedge \ (G_{1,j})_{\mathbf{2}} \leq s \ \wedge \ s \leq (G_{1,j+1})_{\mathbf{2}}\}$.

(27) $\mathrm{cell}(G,\mathrm{len}\,G,0) = \{[r,s] : (G_{\mathrm{len}\,G,1})_{\mathbf{1}} \leq r \ \wedge \ s \leq (G_{1,1})_{\mathbf{2}}\}$.

(28) $\mathrm{cell}(G,\mathrm{len}\,G,\mathrm{width}\,G) = \{[r,s] : (G_{\mathrm{len}\,G,1})_{\mathbf{1}} \leq r \ \wedge \ (G_{1,\mathrm{width}\,G})_{\mathbf{2}} \leq s\}$.

(29) If $1 \leq j$ and $j < \mathrm{width}\,G$, then $\mathrm{cell}(G,\mathrm{len}\,G,j) = \{[r,s] : (G_{\mathrm{len}\,G,1})_{\mathbf{1}} \leq r \ \wedge \ (G_{1,j})_{\mathbf{2}} \leq s \ \wedge \ s \leq (G_{1,j+1})_{\mathbf{2}}\}$.

(30) If $1 \leq i$ and $i < \mathrm{len}\,G$, then $\mathrm{cell}(G,i,0) = \{[r,s] : (G_{i,1})_{\mathbf{1}} \leq r \ \wedge \ r \leq (G_{i+1,1})_{\mathbf{1}} \ \wedge \ s \leq (G_{1,1})_{\mathbf{2}}\}$.

(31) If $1 \leq i$ and $i < \operatorname{len} G$, then $\operatorname{cell}(G, i, \operatorname{width} G) = \{[r, s] : (G_{i,1})_{\mathbf{1}} \leq r \ \wedge \ r \leq (G_{i+1,1})_{\mathbf{1}} \ \wedge \ (G_{1,\operatorname{width} G})_{\mathbf{2}} \leq s\}$.

(32) If $1 \leq i$ and $i < \operatorname{len} G$ and $1 \leq j$ and $j < \operatorname{width} G$, then $\operatorname{cell}(G, i, j) = \{[r, s] : (G_{i,1})_{\mathbf{1}} \leq r \ \wedge \ r \leq (G_{i+1,1})_{\mathbf{1}} \ \wedge \ (G_{1,j})_{\mathbf{2}} \leq s \ \wedge \ s \leq (G_{1,j+1})_{\mathbf{2}}\}$.

(33) For all i, j holds $\operatorname{cell}(G, i, j)$ is closed.

(34) $1 \leq \operatorname{len} G$ and $1 \leq \operatorname{width} G$.

(35) For all i, j such that $i \leq \operatorname{len} G$ and $j \leq \operatorname{width} G$ holds $\operatorname{cell}(G, i, j) = \overline{\operatorname{Int} \operatorname{cell}(G, i, j)}$.

References

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

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

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

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

[5] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 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] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.

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

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

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

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

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

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

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

[16] Beata Padlewska. Connected spaces. *Formalized Mathematics*, 1(**1**):239–244, 1990.

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

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

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

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

[21] Andrzej Trybulec and Czesław Byliński. Some properties of real numbers. *Formalized Mathematics*, 1(**3**):445–449, 1990.

[22] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 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] Mirosław Wysocki and Agata Darmochwał. Subsets of topological spaces. *Formalized Mathematics*, 1(**1**):231–237, 1990.

Received July 22, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

On the Composition of Macro Instructions. Part II [1]

Noriko Asamoto
Ochanomizu University
Tokyo

Yatsuka Nakamura
Shinshu University
Nagano

Piotr Rudnicki
University of Alberta
Edmonton

Andrzej Trybulec
Warsaw University
Białystok

Summary. We define the semantics of macro instructions (introduced in [26]) in terms of executions of $\mathbf{SCM}_{\rm FSA}$. In a similar way, we define the semantics of macro composition. Several attributes of macro instructions are introduced (paraclosed, parahalting, keeping 0) and their usage enables a systematic treatment of the composition of macro intructions. This article is continued in [1].

MML Identifier: `SCMFSA6B`.

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

1. Preliminaries

The following propositions are true:

(1) For all functions f, g and for all sets x, y such that $x \notin \mathrm{dom}\, f$ and $f \subseteq g$ holds $f \subseteq g +\cdot (x, y)$.

(2) For every function f and for all sets x, y, A such that $x \notin A$ holds $f \restriction A = (f +\cdot (x, y)) \restriction A$.

(3) For all functions f, g and for every set A such that $A \cap \mathrm{dom}\, f \subseteq A \cap \mathrm{dom}\, g$ holds $(f +\cdot g \restriction A) \restriction A = g \restriction A$.

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

ISSN 1426–2630

2. Properties of Start-At

For simplicity we follow a convention: m, n will denote natural numbers, x will denote a set, i will denote an instruction of $\mathbf{SCM}_{\rm FSA}$, I, J will denote macro instructions, a will denote an integer location, f will denote a finite sequence location, l, l_1 will denote instructions-locations of $\mathbf{SCM}_{\rm FSA}$, and s, s_1, s_2 will denote states of $\mathbf{SCM}_{\rm FSA}$.

We now state a number of propositions:

(4) $\text{Start-At}(\text{insloc}(0)) \subseteq \text{Initialized}(I)$.

(5) If $I {+\cdot}\, \text{Start-At}(\text{insloc}(n)) \subseteq s$, then $I \subseteq s$.

(6) $(I {+\cdot}\, \text{Start-At}(\text{insloc}(n))) \upharpoonright (\text{the instruction locations of } \mathbf{SCM}_{\rm FSA}) = I$.

(7) If $x \in \text{dom}\, I$, then $I(x) = (I {+\cdot}\, \text{Start-At}(\text{insloc}(n)))(x)$.

(8) If $\text{Initialized}(I) \subseteq s$, then $I {+\cdot}\, \text{Start-At}(\text{insloc}(0)) \subseteq s$.

(9) $a \notin \text{dom}\, \text{Start-At}(l)$.

(10) $f \notin \text{dom}\, \text{Start-At}(l)$.

(11) $l_1 \notin \text{dom}\, \text{Start-At}(l)$.

(12) $a \notin \text{dom}(I {+\cdot}\, \text{Start-At}(l))$.

(13) $f \notin \text{dom}(I {+\cdot}\, \text{Start-At}(l))$.

(14) $s {+\cdot} I {+\cdot}\, \text{Start-At}(\text{insloc}(0)) = s {+\cdot}\, \text{Start-At}(\text{insloc}(0)) {+\cdot} I$.

3. Properties of AMI structures

In the sequel N will denote a non empty set with non empty elements.

Next we state two propositions:

(15) If $s = \text{Following}(s)$, then for every n holds $(\text{Computation}(s))(n) = s$.

(16) Let S be a halting von Neumann definite AMI over N and let s be a state of S. If s is halting, then $\text{Result}(s) = (\text{Computation}(s))(\text{LifeSpan}(s))$.

Let us consider N, let S be a von Neumann definite AMI over N, let s be a state of S, let l be an instruction-location of S, and let i be an instruction of S. Then $s +\cdot (l, i)$ is a state of S.

Let s be a state of $\mathbf{SCM}_{\rm FSA}$, let l_2 be an integer location, and let k be an integer. Then $s +\cdot (l_2, k)$ is a state of $\mathbf{SCM}_{\rm FSA}$.

We now state the proposition

(17) Let S be a steady-programmed von Neumann definite AMI over N, and let s be a state of S, and given n. Then $s \upharpoonright (\text{the instruction locations of } S) = (\text{Computation}(s))(n) \upharpoonright (\text{the instruction locations of } S)$.

4. Execution of macro instructions

Let I be a macro instruction and let s be a state of $\mathbf{SCM}_{\rm FSA}$. The functor IExec(I, s) yielding a state of $\mathbf{SCM}_{\rm FSA}$ is defined as follows:

(Def. 1) IExec(I, s) = Result$(s +\cdot$ Initialized$(I)) +\cdot s \restriction$ (the instruction locations of $\mathbf{SCM}_{\rm FSA}$).

Let I be a macro instruction. We say that I is paraclosed if and only if:

(Def. 2) For every state s of $\mathbf{SCM}_{\rm FSA}$ and for every natural number n such that $I +\cdot$ Start-At(insloc(0)) $\subseteq s$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(n)} \in \mathrm{dom}\, I$.

We say that I is parahalting if and only if:

(Def. 3) $I +\cdot$ Start-At(insloc(0)) is halting.

We say that I is keeping 0 if and only if:

(Def. 4) For every state s of $\mathbf{SCM}_{\rm FSA}$ such that $I +\cdot$ Start-At(insloc(0)) $\subseteq s$ and for every natural number k holds (Computation$(s))(k)$(intloc(0)) = s(intloc(0)).

Let us note that there exists a macro instruction which is parahalting.

Next we state two propositions:

(18) For every parahalting macro instruction I such that $I +\cdot$ Start-At(insloc (0)) $\subseteq s$ holds s is halting.

(19) For every parahalting macro instruction I such that Initialized$(I) \subseteq s$ holds s is halting.

Let I be a parahalting macro instruction. One can verify that Initialized(I) is halting.

We now state two propositions:

(20) $s_2 +\cdot (\mathbf{IC}_{(s_2)}, \text{goto } (\mathbf{IC}_{(s_2)}))$ is not halting.

(21) Suppose that
 (i) s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\rm FSA}$,
 (ii) $I \subseteq s_1$,
 (iii) $I \subseteq s_2$, and
 (iv) for every m such that $m < n$ holds $\mathbf{IC}_{(\mathrm{Computation}(s_2))(m)} \in \mathrm{dom}\, I$.

 Given m. Suppose $m \leq n$. Then (Computation$(s_1))(m)$ and (Computation$(s_2))(m)$ are equal outside the instruction locations of $\mathbf{SCM}_{\rm FSA}$.

One can check that every macro instruction which is parahalting is also paraclosed and every macro instruction which is keeping 0 is also paraclosed.

The following propositions are true:

(22) Let I be a parahalting macro instruction and let a be a read-write integer location. If $a \notin$ UsedIntLoc(I), then (IExec$(I, s))(a) = s(a)$.

(23) For every parahalting macro instruction I such that $f \notin$ UsedInt* Loc(I) holds (IExec$(I, s))(f) = s(f)$.

(24) If $\mathbf{IC}_s = l$ and $s(l) =$ goto l, then s is not halting.

One can verify that every macro instruction which is parahalting is also non empty.

One can prove the following propositions:

(25) For every parahalting macro instruction I holds $\mathrm{dom}\, I \neq \emptyset$.

(26) For every parahalting macro instruction I holds $\mathrm{insloc}(0) \in \mathrm{dom}\, I$.

(27) Let J be a parahalting macro instruction. Suppose $J +\cdot$ Start-At(insloc(0)) $\subseteq s_1$. Let n be a natural number. Suppose ProgramPart(Relocated(J, n)) $\subseteq s_2$ and $\mathbf{IC}_{(s_2)} = \mathrm{insloc}(n)$ and $s_1 \restriction$ (Int-Locations $\cup$ FinSeq-Locations) $= s_2 \restriction$ (Int-Locations $\cup$ FinSeq-Locations). Let i be a natural number. Then $\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)} + n = \mathbf{IC}_{(\mathrm{Computation}(s_2))(i)}$ and IncAddr(CurInstr((Computation(s_1))(i))$, n$) $=$ CurInstr((Computation(s_2))(i)) and (Computation(s_1))$(i) \restriction$ (Int-Locations $\cup$ FinSeq-Locations) $=$ (Computation(s_2))$(i) \restriction$ (Int-Locations $\cup$ FinSeq-Locations).

(28) Let I be a parahalting macro instruction. Suppose $I +\cdot$ Start-At(insloc(0)) $\subseteq s_1$ and $I +\cdot$ Start-At(insloc(0)) $\subseteq s_2$ and s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$. Let k be a natural number. Then (Computation(s_1))(k) and (Computation(s_2))(k) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$ and CurInstr((Computation(s_1))(k)) $=$ CurInstr((Computation(s_2))(k)).

(29) Let I be a parahalting macro instruction. Suppose $I +\cdot$ Start-At(insloc(0)) $\subseteq s_1$ and $I +\cdot$ Start-At(insloc(0)) $\subseteq s_2$ and s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$. Then LifeSpan$(s_1)$ $=$ LifeSpan(s_2) and Result(s_1) and Result(s_2) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(30) For every parahalting macro instruction I holds $\mathbf{IC}_{\mathrm{IExec}(I,s)} = \mathbf{IC}_{\mathrm{Result}(s +\cdot \mathrm{Initialized}(I))}$.

(31) For every non empty macro instruction I holds $\mathrm{insloc}(0) \in \mathrm{dom}\, I$ and $\mathrm{insloc}(0) \in \mathrm{dom}\,\mathrm{Initialized}(I)$ and $\mathrm{insloc}(0) \in \mathrm{dom}(I +\cdot$ Start-At(insloc(0))).

(32) $x \in \mathrm{dom}\,\mathrm{Macro}(i)$ iff $x = \mathrm{insloc}(0)$ or $x = \mathrm{insloc}(1)$.

(33) $(\mathrm{Macro}(i))(\mathrm{insloc}(0)) = i$ and $(\mathrm{Macro}(i))(\mathrm{insloc}(1)) = \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$ and $(\mathrm{Initialized}(\mathrm{Macro}(i)))(\mathrm{insloc}(0)) = i$ and $(\mathrm{Initialized}(\mathrm{Macro}(i)))(\mathrm{insloc}(1)) = \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$ and (Macro$(i) +\cdot$ Start-At(insloc(0)))(insloc(0)) $= i$.

(34) If $\mathrm{Initialized}(I) \subseteq s$, then $\mathbf{IC}_s = \mathrm{insloc}(0)$.

Let us observe that there exists a macro instruction which is keeping 0 and parahalting.

One can prove the following proposition

(35) For every keeping 0 parahalting macro instruction I holds $(\mathrm{IExec}(I, s))(\mathrm{intloc}(0)) = 1$.

5. The composition of macro instructions

We now state several propositions:

(36) Let I be a paraclosed macro instruction and let J be a macro instruction. Suppose $I+\cdot$ Start-At(insloc(0)) $\subseteq s$ and s is halting. Given m. Suppose $m \leq$ LifeSpan(s). Then (Computation(s))(m) and (Computation($s+\cdot(I;J)$))(m) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(37) For every paraclosed macro instruction I such that $s+\cdot I$ is halting and Directed(I) $\subseteq s$ and Start-At(insloc(0)) $\subseteq s$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s+\cdot I)+1)} = \mathrm{insloc}(\mathrm{card}\, I)$.

(38) Let I be a paraclosed macro instruction. If $s+\cdot I$ is halting and Directed(I) $\subseteq s$ and Start-At(insloc(0)) $\subseteq s$, then (Computation(s))(LifeSpan($s+\cdot I$))$\restriction$(Int-Locations $\cup$ FinSeq-Locations) = (Computation(s))(LifeSpan($s+\cdot I$)+1)$\restriction$(Int-Locations $\cup$ FinSeq-Locations).

(39) Let I be a parahalting macro instruction. Suppose Initialized(I) $\subseteq s$. Let k be a natural number. If $k \leq$ LifeSpan(s), then CurInstr((Computation($s+\cdot$ Directed(I)))(k)) $\neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(40) Let I be a paraclosed macro instruction. Suppose $s+\cdot(I+\cdot$ Start-At (insloc(0))) is halting. Let J be a macro instruction and let k be a natural number. Suppose $k \leq$ LifeSpan($s+\cdot(I+\cdot$ Start-At(insloc(0)))). Then (Computation($s+\cdot(I+\cdot$ Start-At(insloc(0)))))(k) and (Computation($s+\cdot$ $((I;J)+\cdot$ Start-At(insloc(0)))))(k) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

Let I, J be parahalting macro instructions. Note that $I;J$ is parahalting.

Next we state two propositions:

(41) Let I be a keeping 0 macro instruction. Suppose $s+\cdot(I+\cdot$ Start-At(insloc (0))) is not halting. Let J be a macro instruction and let k be a natural number. Then (Computation($s+\cdot(I+\cdot$ Start-At(insloc(0)))))(k) and (Computation($s+\cdot((I;J)+\cdot$ Start-At(insloc(0)))))(k) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(42) Let I be a keeping 0 macro instruction. Suppose $s+\cdot I$ is halting. Let J be a paraclosed macro instruction. Suppose $(I;J)+\cdot$ Start-At(insloc(0)) $\subseteq s$. Let k be a natural number. Then (Computation(Result($s+\cdot I$)$+\cdot(J+\cdot$ Start-At(insloc(0)))))(k)$+\cdot$ Start-At ($\mathbf{IC}_{(\mathrm{Computation}(\mathrm{Result}(s+\cdot I)+\cdot(J+\cdot \mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)} + \mathrm{card}\, I$) and (Computation($s+\cdot(I;J)$))(LifeSpan($s+\cdot I$) $+ 1 + k$) are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

Let I, J be keeping 0 macro instructions. Note that $I;J$ is keeping 0.

The following two propositions are true:

(43) Let I be a keeping 0 parahalting macro instruction and let J be a parahalting macro instruction. Then LifeSpan($s+\cdot$ Initialized($I;J$)) =

LifeSpan($s+\cdot$ Initialized(I)) + 1 + LifeSpan(Result($s+\cdot$ Initialized(I))$+\cdot$ Initialized(J)).

(44) Let I be a keeping 0 parahalting macro instruction and let J be a parahalting macro instruction. Then $\mathrm{IExec}(I;J,s) = \mathrm{IExec}(J,\mathrm{IExec}(I,s)) +\cdot \text{Start-At}(\mathbf{IC}_{\mathrm{IExec}(J,\mathrm{IExec}(I,s))} + \mathrm{card}\, I)$.

References

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

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

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

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

[5] Grzegorz Bancerek and Piotr Rudnicki. Development of terminology for **SCM**. *Formalized Mathematics*, 4(**1**):61–67, 1993.

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

[7] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.

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

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

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

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

[12] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.

[13] Czesław Byliński. Products and coproducts in categories. *Formalized Mathematics*, 2(**5**):701–709, 1991.

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

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

[16] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.

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

[18] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.

[19] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.

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

[21] Andrzej Trybulec and Agata Darmochwał. Boolean domains. *Formalized Mathematics*, 1(**1**):187–190, 1990.

[22] Andrzej Trybulec and Yatsuka Nakamura. Computation in $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):537–542, 1996.

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

[24] Andrzej Trybulec and Yatsuka Nakamura. Relocability for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):583–586, 1996.

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

[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 $\mathbf{SCM}_{\rm FSA}$ 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 and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[31] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received July 22, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

The First Part of Jordan's Theorem for Special Polygons

Yatsuka Nakamura
Shinshu University
Nagano

Andrzej Trybulec
Warsaw University
Białystok

Summary. We prove here the first part of Jordan's theorem for special polygons, i.e., the complement of a special polygon is the union of two components (a left component and a right component). At this stage, we do not know if the two components are different from each other.

MML Identifier: `GOBRD12`.

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

We adopt the following convention: i, j, k_1, k_2, i_1, i_2, j_1, j_2 will be natural numbers and f will be a non constant standard special circular sequence.

The following propositions are true:

(1) $(\widetilde{\mathcal{L}}(f))^{\rm c} \neq \emptyset$.

(2) For all i, j such that $i \leq \mathrm{len}$ the Go-board of f and $j \leq \mathrm{width}$ the Go-board of f holds $\mathrm{Int}\,\mathrm{cell}(\text{the Go-board of } f, i, j) \subseteq (\widetilde{\mathcal{L}}(f))^{\rm c}$.

(3) Given i, j. Suppose $i \leq \mathrm{len}$ the Go-board of f and $j \leq \mathrm{width}$ the Go-board of f. Then $\overline{\mathrm{Down}(\mathrm{Int}\,\mathrm{cell}(\text{the Go-board of } f, i, j), (\widetilde{\mathcal{L}}(f))^{\rm c})} = \mathrm{cell}(\text{the Go-board of } f, i, j) \cap (\widetilde{\mathcal{L}}(f))^{\rm c}$.

(4) Given i, j. Suppose $i \leq \mathrm{len}$ the Go-board of f and $j \leq \mathrm{width}$ the Go-board of f. Then $\overline{\mathrm{Down}(\mathrm{Int}\,\mathrm{cell}(\text{the Go-board of } f, i, j), (\widetilde{\mathcal{L}}(f))^{\rm c})}$ is connected and $\mathrm{Down}(\mathrm{Int}\,\mathrm{cell}(\text{the Go-board of } f, i, j), (\widetilde{\mathcal{L}}(f))^{\rm c}) = \mathrm{Int}\,\mathrm{cell}(\text{the Go-board of } f, i, j)$.

(5) $(\widetilde{\mathcal{L}}(f))^{\rm c} = \bigcup\{\overline{\mathrm{Down}(\mathrm{Int}\,\mathrm{cell}(\text{the Go-board of } f, i, j), (\widetilde{\mathcal{L}}(f))^{\rm c})} : i \leq \mathrm{len}\text{ the Go-board of } f \wedge j \leq \mathrm{width}\text{ the Go-board of } f\}$.

ISSN 1426–2630

(6) $\mathrm{Down}(\mathrm{LeftComp}(f), (\widetilde{\mathcal{L}}(f))^{\mathrm{c}}) \cup \mathrm{Down}(\mathrm{RightComp}(f), (\widetilde{\mathcal{L}}(f))^{\mathrm{c}})$ is a union of components of $(\mathcal{E}^2_{\mathrm{T}}) \restriction (\widetilde{\mathcal{L}}(f))^{\mathrm{c}}$ and $\mathrm{Down}(\mathrm{LeftComp}(f), (\widetilde{\mathcal{L}}(f))^{\mathrm{c}}) = \mathrm{LeftComp}(f)$ and $\mathrm{Down}(\mathrm{RightComp}(f), (\widetilde{\mathcal{L}}(f))^{\mathrm{c}}) = \mathrm{RightComp}(f)$.

(7) Given i_1, j_1, i_2, j_2. Suppose that
- (i) $i_1 \leq \mathrm{len}$ the Go-board of f,
- (ii) $j_1 \leq \mathrm{width}$ the Go-board of f,
- (iii) $i_2 \leq \mathrm{len}$ the Go-board of f,
- (iv) $j_2 \leq \mathrm{width}$ the Go-board of f, and
- (v) i_1, j_1, i_2, and j_2 are adjacent.

Then Int cell(the Go-board of f, $i_1, j_1) \subseteq \mathrm{LeftComp}(f) \cup \mathrm{RightComp}(f)$ if and only if Int cell(the Go-board of f, $i_2, j_2) \subseteq \mathrm{LeftComp}(f) \cup \mathrm{RightComp}(f)$.

(8) Let F_1, F_2 be finite sequences of elements of $\mathbb{N}$. Suppose that
- (i) $\mathrm{len}\, F_1 = \mathrm{len}\, F_2$,
- (ii) there exists i such that $i \in \mathrm{dom}\, F_1$ and Int cell(the Go-board of f, $\pi_i F_1, \pi_i F_2) \subseteq \mathrm{LeftComp}(f) \cup \mathrm{RightComp}(f)$,
- (iii) for every i such that $1 \leq i$ and $i < \mathrm{len}\, F_1$ holds $\pi_i F_1$, $\pi_i F_2$, $\pi_{i+1} F_1$, and $\pi_{i+1} F_2$ are adjacent, and
- (iv) for all i, k_1, k_2 such that $i \in \mathrm{dom}\, F_1$ and $k_1 = F_1(i)$ and $k_2 = F_2(i)$ holds $k_1 \leq \mathrm{len}$ the Go-board of f and $k_2 \leq \mathrm{width}$ the Go-board of f.

Given i. If $i \in \mathrm{dom}\, F_1$, then Int cell(the Go-board of f, $\pi_i F_1, \pi_i F_2) \subseteq \mathrm{LeftComp}(f) \cup \mathrm{RightComp}(f)$.

(9) There exist i, j such that $i \leq \mathrm{len}$ the Go-board of f and $j \leq \mathrm{width}$ the Go-board of f and Int cell(the Go-board of f, $i, j) \subseteq \mathrm{LeftComp}(f) \cup \mathrm{RightComp}(f)$.

(10) For all i, j such that $i \leq \mathrm{len}$ the Go-board of f and $j \leq \mathrm{width}$ the Go-board of f holds Int cell(the Go-board of f, $i, j) \subseteq \mathrm{LeftComp}(f) \cup \mathrm{RightComp}(f)$.

(11) $(\widetilde{\mathcal{L}}(f))^{\mathrm{c}} = \mathrm{LeftComp}(f) \cup \mathrm{RightComp}(f)$.

References

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

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

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

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

[5] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 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] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.

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

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

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

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

[13] Yatsuka Nakamura and Andrzej Trybulec. Adjacency concept for pairs of natural numbers. *Formalized Mathematics*, 6(**1**):1–3, 1997.

[14] Yatsuka Nakamura and Andrzej Trybulec. Components and unions of components. *Formalized Mathematics*, 5(**4**):513–517, 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. Connected spaces. *Formalized Mathematics*, 1(**1**):239–244, 1990.

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

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

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

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

[22] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 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] Mirosław Wysocki and Agata Darmochwał. Subsets of topological spaces. *Formalized Mathematics*, 1(**1**):231–237, 1990.

Received July 22, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

On the Composition of Macro Instructions. Part III [1]

Noriko Asamoto
Ochanomizu University
Tokyo

Yatsuka Nakamura
Shinshu University
Nagano

Piotr Rudnicki
University of Alberta
Edmonton

Andrzej Trybulec
Warsaw University
Białystok

Summary. This article is a continuation of [27] and [2]. First, we recast the semantics of the macro composition in more convenient terms. Then, we introduce terminology and basic properties of macros constructed out of single instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. We give the complete semantics of composing a macro instruction with an instruction and for composing two machine instructions (this is also done in terms of macros). The introduced terminology is tested on the simple example of a macro for swapping two integer locations.

MML Identifier: `SCMFSA6C`.

The papers [23], [31], [15], [4], [29], [18], [32], [10], [11], [5], [24], [9], [30], [13], [3], [21], [8], [14], [12], [22], [16], [17], [26], [6], [20], [7], [28], [25], [27], [19], and [1] provide the notation and terminology for this paper.

1. Preliminaries

For simplicity we adopt the following rules: i will denote an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, a, b will denote integer locations, f will denote a finite sequence location, l will denote an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$, and s, s_1, s_2 will denote states of $\mathbf{SCM}_{\mathrm{FSA}}$.

The following propositions are true:

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

ISSN 1426–2630

(1) Let I be a keeping 0 parahalting macro instruction and let J be a parahalting macro instruction. Then $(\mathrm{IExec}(I;J,s))(a) = (\mathrm{IExec}(J,\mathrm{IExec}(I,s)))(a)$.

(2) Let I be a keeping 0 parahalting macro instruction and let J be a parahalting macro instruction. Then $(\mathrm{IExec}(I;J,s))(f) = (\mathrm{IExec}(J,\mathrm{IExec}(I,s)))(f)$.

2. Parahalting and keeping 0 macro instructions

Let i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. We say that i is parahalting if and only if:

(Def. 1) Macro(i) is parahalting.

We say that i is keeping 0 if and only if:

(Def. 2) Macro(i) is keeping 0.

Let us observe that $\mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$ is keeping 0 and parahalting.

Let us note that there exists an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ which is keeping 0 and parahalting.

Let i be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Observe that Macro($i$) is parahalting.

Let i be a keeping 0 instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Observe that Macro($i$) is keeping 0.

Let a, b be integer locations. One can check the following observations:

* $a{:=}b$ is parahalting,
* AddTo(a, b) is parahalting,
* SubFrom(a, b) is parahalting,
* MultBy(a, b) is parahalting, and
* Divide(a, b) is parahalting.

Let f be a finite sequence location. Note that $b{:=}f_a$ is parahalting and $f_a{:=}b$ is parahalting and keeping 0.

Let a be an integer location and let f be a finite sequence location. Note that $a{:=}\mathrm{len}f$ is parahalting and $f{:=}\langle\underbrace{0,\ldots,0}_{a}\rangle$ is parahalting and keeping 0.

Let a be a read-write integer location and let b be an integer location. One can verify the following observations:

* $a{:=}b$ is keeping 0,
* AddTo(a, b) is keeping 0,
* SubFrom(a, b) is keeping 0, and
* MultBy(a, b) is keeping 0.

Let a, b be read-write integer locations. Note that Divide(a, b) is keeping 0.

Let a be an integer location, let f be a finite sequence location, and let b be a read-write integer location. Observe that $b{:=}f_a$ is keeping 0.

Let f be a finite sequence location and let b be a read-write integer location. Observe that b:=len$f$ is keeping 0.

Let i be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let J be a parahalting macro instruction. One can verify that i;J is parahalting.

Let I be a parahalting macro instruction and let j be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Note that I;j is parahalting.

Let i be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let j be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Note that i;j is parahalting.

Let i be a keeping 0 instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let J be a keeping 0 macro instruction. Observe that i;J is keeping 0.

Let I be a keeping 0 macro instruction and let j be a keeping 0 instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. One can check that I;j is keeping 0.

Let i, j be keeping 0 instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. One can check that i;j is keeping 0.

3. Semantics of compositions

Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. The functor Initialize(s) yielding a state of $\mathbf{SCM}_{\mathrm{FSA}}$ is defined as follows:

(Def. 3) $\mathrm{Initialize}(s) = s {+\cdot} (\mathrm{intloc}(0) \mathrel{\dot{\longmapsto}} 1) {+\cdot} \mathrm{Start\text{-}At}(\mathrm{insloc}(0))$.

The following propositions are true:

(3) (i) $\mathbf{IC}_{\mathrm{Initialize}(s)} = \mathrm{insloc}(0)$,

(ii) $(\mathrm{Initialize}(s))(\mathrm{intloc}(0)) = 1$,

(iii) for every read-write integer location a holds $(\mathrm{Initialize}(s))(a) = s(a)$,

(iv) for every f holds $(\mathrm{Initialize}(s))(f) = s(f)$, and

(v) for every l holds $(\mathrm{Initialize}(s))(l) = s(l)$.

(4) s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$ iff $s_1 \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations} \cup \{\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\}) = s_2 \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations} \cup \{\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\})$.

(5) If $s_1 \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations}) = s_2 \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations})$, then $\mathrm{Exec}(i, s_1) \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations}) = \mathrm{Exec}(i, s_2) \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations})$.

(6) For every parahalting instruction i of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{Exec}(i, \mathrm{Initialize}(s)) = \mathrm{IExec}(\mathrm{Macro}(i), s)$.

(7) Let I be a keeping 0 parahalting macro instruction and let j be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $(\mathrm{IExec}(I;j, s))(a) = (\mathrm{Exec}(j, \mathrm{IExec}(I, s)))(a)$.

(8) Let I be a keeping 0 parahalting macro instruction and let j be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $(\mathrm{IExec}(I;j, s))(f) = (\mathrm{Exec}(j, \mathrm{IExec}(I, s)))(f)$.

(9) Let i be a keeping 0 parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let j be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $(\mathrm{IExec}(i;j,s))(a) = (\mathrm{Exec}(j,\mathrm{Exec}(i,\mathrm{Initialize}(s))))(a)$.

(10) Let i be a keeping 0 parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let j be a parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $(\mathrm{IExec}(i;j,s))(f) = (\mathrm{Exec}(j,\mathrm{Exec}(i,\mathrm{Initialize}(s))))(f)$.

4. An example: swap

Let a, b be integer locations. The functor $\mathrm{swap}(a,b)$ yields a macro instruction and is defined as follows:

(Def. 4) $\mathrm{swap}(a,b) = (\mathrm{FirstNotUsed}(\mathrm{Macro}(a{:=}b)){:=}a);(a{:=}b);(b{:=}\,\mathrm{FirstNotUsed}(\mathrm{Macro}(a{:=}b)))$.

Let a, b be integer locations. Observe that $\mathrm{swap}(a,b)$ is parahalting.

Let a, b be read-write integer locations. Note that $\mathrm{swap}(a,b)$ is keeping 0.

We now state two propositions:

(11) For all read-write integer locations a, b holds $(\mathrm{IExec}(\mathrm{swap}(a,b),s))(a) = s(b)$ and $(\mathrm{IExec}(\mathrm{swap}(a,b),s))(b) = s(a)$.

(12) $\mathrm{UsedInt}^{*}\,\mathrm{Loc}(\mathrm{swap}(a,b)) = \emptyset$.

References

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

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

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

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

[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. Development of terminology for **SCM**. *Formalized Mathematics*, 4(**1**):61–67, 1993.

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

[8] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.

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

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

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

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

[13] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.

[14] Czesław Byliński. Products and coproducts in categories. *Formalized Mathematics*, 2(**5**):701–709, 1991.

[15] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[16] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[17] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.
[18] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.
[19] Piotr Rudnicki and Andrzej Trybulec. Memory handling for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 6(**1**):29–36, 1997.
[20] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[21] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[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 Agata Darmochwał. Boolean domains. *Formalized Mathematics*, 1(**1**):187–190, 1990.
[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 and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[27] Andrzej Trybulec, Yatsuka Nakamura, and Noriko Asamoto. On the compositions of macro instructions. Part I. *Formalized Mathematics*, 6(**1**):21–27, 1997.
[28] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\mathrm{FSA}}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[29] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[30] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 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 July 22, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Constant Assignment Macro Instructions of $\mathbf{SCM}_{\rm FSA}$. Part II

Noriko Asamoto
Ochanomizu University
Tokyo

MML Identifier: SCMFSA7B.

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

In this paper m is a natural number.

Next we state two propositions:

(1) For every finite sequence p of elements of the instructions of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{dom}\,\mathrm{Load}(p) = \{\mathrm{insloc}(m) : m < \mathrm{len}\, p\}$.

(2) For every finite sequence p of elements of the instructions of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{rng}\,\mathrm{Load}(p) = \mathrm{rng}\, p$.

Let p be a finite sequence of elements of the instructions of $\mathbf{SCM}_{\rm FSA}$. Observe that $\mathrm{Load}(p)$ is initial and programmed.

We now state several propositions:

(3) For every instruction i of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{Load}(\langle i\rangle) = \mathrm{insloc}(0) \mapsto i$.

(4) For every instruction i of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{dom}\,\mathrm{Macro}(i) = \{\mathrm{insloc}(0), \mathrm{insloc}(1)\}$.

(5) For every instruction i of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{Macro}(i) = \mathrm{Load}(\langle i, \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}\rangle)$.

(6) For every instruction i of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{card}\,\mathrm{Macro}(i) = 2$.

(7) For every instruction i of $\mathbf{SCM}_{\rm FSA}$ holds if $i = \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}$, then $(\mathrm{Directed}(\mathrm{Macro}(i)))(\mathrm{insloc}(0)) = \mathrm{goto}\ \mathrm{insloc}(2)$ and if $i \neq \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}$, then $(\mathrm{Directed}(\mathrm{Macro}(i)))(\mathrm{insloc}(0)) = i$.

(8) For every instruction i of $\mathbf{SCM}_{\rm FSA}$ holds $(\mathrm{Directed}(\mathrm{Macro}(i)))(\mathrm{insloc}(1)) = \mathrm{goto}\ \mathrm{insloc}(2)$.

Let a be an integer location and let k be an integer. Observe that $a{:=}k$ is initial and programmed.

ISSN 1426–2630

Let a be an integer location and let k be an integer. Observe that $a{:=}k$ is parahalting.

We now state the proposition

(9) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let a be a read-write integer location, and let k be an integer. Then

(i) $(\mathrm{IExec}(a{:=}k, s))(a) = k$,

(ii) for every read-write integer location b such that $b \neq a$ holds $(\mathrm{IExec}(a{:=}k, s))(b) = s(b)$, and

(iii) for every finite sequence location f holds $(\mathrm{IExec}(a{:=}k, s))(f) = s(f)$.

Let f be a finite sequence location and let p be a finite sequence of elements of $\mathbb{Z}$. One can check that $f{:=}p$ is initial and programmed.

Let f be a finite sequence location and let p be a finite sequence of elements of $\mathbb{Z}$. Observe that $f{:=}p$ is parahalting.

The following proposition is true

(10) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let f be a finite sequence location, and let p be a finite sequence of elements of $\mathbb{Z}$. Then

(i) $(\mathrm{IExec}(f{:=}p, s))(f) = p$,

(ii) for every read-write integer location a such that $a \neq \mathrm{intloc}(1)$ and $a \neq \mathrm{intloc}(2)$ holds $(\mathrm{IExec}(f{:=}p, s))(a) = s(a)$, and

(iii) for every finite sequence location g such that $g \neq f$ holds $(\mathrm{IExec}(f{:=}p, s))(g) = s(g)$.

Let i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let a be an integer location. We say that i does not refer a if and only if the condition (Def. 1) is satisfied.

(Def. 1) Let b be an integer location, and let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$, and let f be a finite sequence location. Then

(i) $b{:=}a \neq i$,

(ii) $\mathrm{AddTo}(b, a) \neq i$,

(iii) $\mathrm{SubFrom}(b, a) \neq i$,

(iv) $\mathrm{MultBy}(b, a) \neq i$,

(v) $\mathrm{Divide}(b, a) \neq i$,

(vi) $\mathrm{Divide}(a, b) \neq i$,

(vii) $\mathbf{if}\ a = 0\ \mathbf{goto}\ l \neq i$,

(viii) $\mathbf{if}\ a > 0\ \mathbf{goto}\ l \neq i$,

(ix) $b{:=}f_a \neq i$,

(x) $f_b{:=}a \neq i$,

(xi) $f_a{:=}b \neq i$, and

(xii) $f{:=}\langle\underbrace{0, \ldots, 0}_{a}\rangle \neq i$.

Let I be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let a be an integer location. We say that I does not refer a if and only if:

(Def. 2) For every instruction i of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $i \in \operatorname{rng} I$ holds i does not refer a.

Let i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let a be an integer location. We say that i does not destroy a if and only if the condition (Def. 3) is satisfied.

(Def. 3) Let b be an integer location and let f be a finite sequence location. Then $a{:=}b \neq i$ and $\mathrm{AddTo}(a,b) \neq i$ and $\mathrm{SubFrom}(a,b) \neq i$ and $\mathrm{MultBy}(a,b) \neq i$ and $\mathrm{Divide}(a,b) \neq i$ and $\mathrm{Divide}(b,a) \neq i$ and $a{:=}f_b \neq i$ and $a{:=}\mathrm{len} f \neq i$.

Let I be a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let a be an integer location. We say that I does not destroy a if and only if:

(Def. 4) For every instruction i of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $i \in \mathrm{rng}\, I$ holds i does not destroy a.

Let I be a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. We say that I is good if and only if:

(Def. 5) I does not destroy intloc(0).

Let I be a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. We say that I is halt-free if and only if:

(Def. 6) $\mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}} \notin \mathrm{rng}\, I$.

Let us observe that there exists a macro instruction which is halt-free and good.

The following propositions are true:

(11) For every integer location a holds $\mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$ does not destroy a.

(12) For all integer locations a, b, c such that $a \neq b$ holds $b{:=}c$ does not destroy a.

(13) For all integer locations a, b, c such that $a \neq b$ holds $\mathrm{AddTo}(b,c)$ does not destroy a.

(14) For all integer locations a, b, c such that $a \neq b$ holds $\mathrm{SubFrom}(b,c)$ does not destroy a.

(15) For all integer locations a, b, c such that $a \neq b$ holds $\mathrm{MultBy}(b,c)$ does not destroy a.

(16) For all integer locations a, b, c such that $a \neq b$ and $a \neq c$ holds $\mathrm{Divide}(b,c)$ does not destroy a.

(17) For every integer location a and for every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds goto l does not destroy a.

(18) For all integer locations a, b and for every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds **if** $b = 0$ **goto** l does not destroy a.

(19) For all integer locations a, b and for every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds **if** $b > 0$ **goto** l does not destroy a.

(20) Let a, b, c be integer locations and let f be a finite sequence location. If $a \neq b$, then $b{:=}f_c$ does not destroy a.

(21) For all integer locations a, b, c and for every finite sequence location f holds $f_c{:=}b$ does not destroy a.

(22) Let a, b be integer locations and let f be a finite sequence location. If $a \neq b$, then $b{:=}\mathrm{len} f$ does not destroy a.

(23) For all integer locations a, b and for every finite sequence location f holds $f{:=}\langle\underbrace{0,\ldots,0}_{b}\rangle$ does not destroy a.

Let I be a finite partial state of $\mathbf{SCM}_{\rm FSA}$ and let s be a state of $\mathbf{SCM}_{\rm FSA}$. We say that I is closed on s if and only if:

(Def. 7) For every natural number k holds $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot(I+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)} \in \mathrm{dom}\, I$.

We say that I is halting on s if and only if:

(Def. 8) $s+\cdot(I+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))$ is halting.

We now state several propositions:

(24) For every macro instruction I holds I is paraclosed iff for every state s of $\mathbf{SCM}_{\rm FSA}$ holds I is closed on s.

(25) For every macro instruction I holds I is parahalting iff for every state s of $\mathbf{SCM}_{\rm FSA}$ holds I is halting on s.

(26) Let i be an instruction of $\mathbf{SCM}_{\rm FSA}$, and let a be an integer location, and let s be a state of $\mathbf{SCM}_{\rm FSA}$. If i does not destroy a then $(\mathrm{Exec}(i, s))(a) = s(a)$.

(27) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let I be a macro instruction, and let a be an integer location. Suppose I does not destroy a and I is closed on s. Let k be a natural number. Then $(\mathrm{Computation}(s+\cdot(I+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)(a) = s(a)$.

(28) $\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}}$ does not destroy $\mathrm{intloc}(0)$.

One can verify that there exists a macro instruction which is parahalting and good.

One can check that $\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}}$ is parahalting and good.

One can check that every macro instruction which is paraclosed and good is also keeping 0.

One can prove the following two propositions:

(29) For every integer location a and for every integer k holds $\mathrm{rng}\, \mathrm{aSeq}(a, k) \subseteq \{a{:=}\,\mathrm{intloc}(0), \mathrm{AddTo}(a, \mathrm{intloc}(0)), \mathrm{SubFrom}(a, \mathrm{intloc}(0))\}$.

(30) For every integer location a and for every integer k holds $\mathrm{rng}(a{:=}k) \subseteq \{\mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}, a{:=}\,\mathrm{intloc}(0), \mathrm{AddTo}(a, \mathrm{intloc}(0)), \mathrm{SubFrom}(a, \mathrm{intloc}(0))\}$.

Let a be a read-write integer location and let k be an integer. One can check that $a{:=}k$ is good.

Let a be a read-write integer location and let k be an integer. Observe that $a{:=}k$ is keeping 0.

References

[1] Noriko Asamoto. Some multi instructions defined by sequence of instructions of $\mathbf{SCM}_{\rm FSA}$. *Formalized Mathematics*, 5(**4**):615–619, 1996.

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

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

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

[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 trees. *Formalized Mathematics*, 4(**1**):91–101, 1993.
[7] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.
[8] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.
[9] Czesław Byliński. Finite sequences and tuples of elements of a non-empty sets. *Formalized Mathematics*, 1(**3**):529–536, 1990.
[10] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[11] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[12] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[13] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[14] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.
[15] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[16] Piotr Rudnicki and Andrzej Trybulec. Memory handling for $\mathbf{SCM}_{\rm FSA}$. *Formalized Mathematics*, 6(**1**):29–36, 1997.
[17] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[18] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[19] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[20] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[21] Andrzej Trybulec and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\rm FSA}$. *Formalized Mathematics*, 5(**4**):571–576, 1996.
[22] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[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 $\mathbf{SCM}_{\rm FSA}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[25] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[26] Wojciech A. Trybulec. Binary operations on finite sequences. *Formalized Mathematics*, 1(**5**):979–981, 1990.
[27] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 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.

Received August 27, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Conditional Branch Macro Instructions of $\mathbf{SCM}_{\rm FSA}$. Part I

Noriko Asamoto
Ochanomizu University
Tokyo

MML Identifier: `SCMFSA8A`.

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

One can prove the following propositions:

(1) For all functions f, g such that $\mathrm{dom}\, f$ misses $\mathrm{dom}\, g$ holds $f{+}\cdot g = g{+}\cdot f$.

(2) For all functions f, g and for every set D such that $\mathrm{dom}\, g$ misses D holds $(f{+}\cdot g) \restriction D = f \restriction D$.

(3) For every state s of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{dom}(s \restriction$ (the instruction locations of $\mathbf{SCM}_{\rm FSA}$)) = the instruction locations of $\mathbf{SCM}_{\rm FSA}$.

(4) For every state s of $\mathbf{SCM}_{\rm FSA}$ such that s is halting and for every natural number k such that $\mathrm{LifeSpan}(s) \leq k$ holds $\mathrm{CurInstr}((\mathrm{Computation}(s))(k)) = \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}$.

(5) For every state s of $\mathbf{SCM}_{\rm FSA}$ such that s is halting and for every natural number k such that $\mathrm{LifeSpan}(s) \leq k$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(k)} = \mathbf{IC}_{(\mathrm{Computation}(s))(\mathrm{LifeSpan}(s))}$.

(6) Let s_1, s_2 be states of $\mathbf{SCM}_{\rm FSA}$. Then s_1 and s_2 are equal outside the instruction locations of $\mathbf{SCM}_{\rm FSA}$ if and only if $\mathbf{IC}_{(s_1)} = \mathbf{IC}_{(s_2)}$ and $s_1 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = s_2 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(7) For every state s of $\mathbf{SCM}_{\rm FSA}$ and for every macro instruction I holds $\mathbf{IC}_{\mathrm{IExec}(I,s)} = \mathbf{IC}_{\mathrm{Result}(s+\cdot\, \mathrm{Initialized}(I))}$.

(8) For every state s of $\mathbf{SCM}_{\rm FSA}$ and for every macro instruction I holds $\mathrm{Initialize}(s){+}\cdot \mathrm{Initialized}(I) = s{+}\cdot \mathrm{Initialized}(I)$.

(9) For every macro instruction I and for every instruction-location l of $\mathbf{SCM}_{\rm FSA}$ holds $I \subseteq I{+}\cdot \text{Start-At}(l)$.

ISSN 1426–2630

(10) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $s \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = (s {+\cdot}\, \text{Start-At}(l)) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(11) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I be a macro instruction, and let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $s \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = (s {+\cdot} (I {+\cdot}\, \text{Start-At}(l))) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(12) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $\mathrm{dom}(s \restriction (\text{the instruction locations of } \mathbf{SCM}_{\mathrm{FSA}}))$ misses $\mathrm{dom}\, \text{Start-At}(l)$.

(13) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every macro instruction I holds $s {+\cdot}\, \text{Initialized}(I) = \text{Initialize}(s) {+\cdot} (I {+\cdot}\, \text{Start-At}(\text{insloc}(0)))$.

(14) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I_1, I_2 be macro instructions, and let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $s {+\cdot} (I_1 {+\cdot}\, \text{Start-At}(l))$ and $s {+\cdot} (I_2 {+\cdot}\, \text{Start-At}(l))$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(15) $\mathrm{dom}(\text{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}}) = \{\text{insloc}(0)\}$.

(16) $\text{insloc}(0) \in \mathrm{dom}(\text{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}})$ and $\text{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}}(\text{insloc}(0)) = \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(17) $\mathrm{card}(\text{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}}) = 1$.

Let P be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. The functor $\text{Directed}(P, l)$ yields a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and is defined as follows:

(Def. 1) $\text{Directed}(P, l) = (\mathrm{id}_{(\text{the instructions of } \mathbf{SCM}_{\mathrm{FSA}})} {+\cdot} (\mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}} \dot{\longmapsto} \text{goto } l)) \cdot P$.

One can prove the following proposition

(18) For every programmed finite partial state I of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\text{Directed}(I) = \text{Directed}(I, \text{insloc}(\mathrm{card}\, I))$.

Let P be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. One can check that $\text{Directed}(P, l)$ is halt-free.

Let P be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. Note that $\text{Directed}(P)$ is halt-free.

Next we state several propositions:

(19) For every programmed finite partial state P of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{dom}\, \text{Directed}(P, l) = \mathrm{dom}\, P$.

(20) Let P be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $\text{Directed}(P, l) = P {+\cdot} (\mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}} \dot{\longmapsto} \text{goto } l) \cdot P$.

(21) Let P be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$, and let x be arbitrary. Suppose $x \in \mathrm{dom}\, P$. Then if $P(x) = \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$, then $(\text{Directed}(P, l))(x) = \text{goto } l$ and if $P(x) \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$, then $(\text{Directed}(P, l))(x) = P(x)$.

(22) Let i be an instruction of $\mathbf{SCM}_{\rm FSA}$, and let a be an integer location, and let n be a natural number. If i does not destroy a, then $\mathrm{IncAddr}(i, n)$ does not destroy a.

(23) Let P be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$, and let n be a natural number, and let a be an integer location. If P does not destroy a, then $\mathrm{ProgramPart}(\mathrm{Relocated}(P, n))$ does not destroy a.

(24) For every good programmed finite partial state P of $\mathbf{SCM}_{\rm FSA}$ and for every natural number n holds $\mathrm{ProgramPart}(\mathrm{Relocated}(P, n))$ is good.

(25) Let I, J be programmed finite partial states of $\mathbf{SCM}_{\rm FSA}$ and let a be an integer location. Suppose I does not destroy a and J does not destroy a. Then $I{+\cdot}J$ does not destroy a.

(26) For all good programmed finite partial states I, J of $\mathbf{SCM}_{\rm FSA}$ holds $I{+\cdot}J$ is good.

(27) Let I be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$, and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$, and let a be an integer location. If I does not destroy a, then $\mathrm{Directed}(I, l)$ does not destroy a.

Let I be a good programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. Note that $\mathrm{Directed}(I, l)$ is good.

Let I be a good macro instruction. Note that $\mathrm{Directed}(I)$ is good.

Let I be a macro instruction and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. One can verify that $\mathrm{Directed}(I, l)$ is initial.

Let I, J be good macro instructions. Observe that $I;J$ is good.

Let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. The functor $\mathrm{Goto}(l)$ yields a halt-free good macro instruction and is defined by:

(Def. 2) $\mathrm{Goto}(l) = \mathrm{insloc}(0) \dot{\longmapsto} \mathrm{goto}\ l$.

Let s be a state of $\mathbf{SCM}_{\rm FSA}$ and let I be a finite partial state of $\mathbf{SCM}_{\rm FSA}$. We say that I is psuedo-closed on s if and only if the condition (Def. 3) is satisfied.

(Def. 3) There exists a natural number k such that $\mathbf{IC}_{(\mathrm{Computation}(s{+\cdot}(I{+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)} = \mathrm{insloc}(\mathrm{card}\, I)$ and for every natural number n such that $n < k$ holds $\mathbf{IC}_{(\mathrm{Computation}(s{+\cdot}(I{+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(n)} \in \mathrm{dom}\, I$.

Let I be a finite partial state of $\mathbf{SCM}_{\rm FSA}$. We say that I is psuedo-paraclosed if and only if:

(Def. 4) For every state s of $\mathbf{SCM}_{\rm FSA}$ holds I is psuedo-closed on s.

Let us observe that there exists a macro instruction which is psuedo-paraclosed.

Let s be a state of $\mathbf{SCM}_{\rm FSA}$ and let I be a macro instruction. Let us assume that I is psuedo-closed on s. The functor $\mathrm{psuedo} - \mathrm{LifeSpan}(s, I)$ yielding a natural number is defined by:

(Def. 5) $\mathbf{IC}_{(\mathrm{Computation}(s{+\cdot}(I{+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{psuedo-LifeSpan}(s,I))} = \mathrm{insloc}(\mathrm{card}\, I)$ and for every natural number n such that $\mathbf{IC}_{(\mathrm{Computation}(s{+\cdot}(I{+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(n)} \notin \mathrm{dom}\, I$ holds $\mathrm{psuedo} - \mathrm{LifeSpan}(s, I) \leq n$.

We now state a number of propositions:

(28) For all macro instructions I, J and for arbitrary x such that $x \in \operatorname{dom} I$ holds $(I;J)(x) = (\mathrm{Directed}(I))(x)$.

(29) For every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\operatorname{card} \mathrm{Goto}(l) = 1$.

(30) Let P be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let x be arbitrary. Suppose $x \in \operatorname{dom} P$. Then if $P(x) = \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$, then $(\mathrm{Directed}(P))(x) = \mathrm{goto}\ \mathrm{insloc}(\operatorname{card} P)$ and if $P(x) \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$, then $(\mathrm{Directed}(P))(x) = P(x)$.

(31) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let I be a macro instruction. Suppose I is psuedo-closed on s. Let n be a natural number. If $n <$ psuedo $-$ LifeSpan(s, I), then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot(I+\cdot\,\text{Start-At}(\mathrm{insloc}(0)))))(n)} \in \operatorname{dom} I$ and $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot(I+\cdot\,\text{Start-At}(\mathrm{insloc}(0)))))(n)) \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(32) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let I, J be macro instructions. Suppose I is psuedo-closed on s. Let k be a natural number. Suppose $k \leq$ psuedo $-$ LifeSpan(s, I). Then $(\mathrm{Computation}(s+\cdot(I+\cdot\,\text{Start-At}(\mathrm{insloc}(0)))))(k)$ and $(\mathrm{Computation}(s+\cdot((I;J)+\cdot\,\text{Start-At}(\mathrm{insloc}(0)))))(k)$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(33) For every programmed finite partial state I of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\operatorname{card} \mathrm{Directed}(I, l) = \operatorname{card} I$.

(34) For every macro instruction I holds $\operatorname{card} \mathrm{Directed}(I) = \operatorname{card} I$.

(35) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let I be a macro instruction. Suppose I is closed on s and halting on s. Let k be a natural number. Suppose $k \leq \mathrm{LifeSpan}(s+\cdot(I+\cdot\,\text{Start-At}(\mathrm{insloc}(0))))$. Then $(\mathrm{Computation}(s+\cdot(I+\cdot\,\text{Start-At}(\mathrm{insloc}(0)))))(k)$ and $(\mathrm{Computation}(s+\cdot(\mathrm{Directed}(I)+\cdot\,\text{Start-At}(\mathrm{insloc}(0)))))(k)$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$ and $\mathrm{CurInstr}((\mathrm{Computation}(s+\cdot(\mathrm{Directed}(I)+\cdot\,\text{Start-At}(\mathrm{insloc}(0)))))(k)) \neq \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(36) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let I be a macro instruction. Suppose I is closed on s and halting on s.
Then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot(\mathrm{Directed}(I)+\cdot\,\text{Start-At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s+\cdot(I+\cdot\,\text{Start-At}(\mathrm{insloc}(0))))+1)} = \mathrm{insloc}(\operatorname{card} I)$ and $(\mathrm{Computation}(s+\cdot(I+\cdot\,\text{Start-At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s+\cdot(I+\cdot\,\text{Start-At}(\mathrm{insloc}(0))))) \upharpoonright (\text{Int-Locations} \cup \text{FinSeq-Locations}) = (\mathrm{Computation}(s+\cdot(\mathrm{Directed}(I)+\cdot\,\text{Start-At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s+\cdot(I+\cdot\,\text{Start-At}(\mathrm{insloc}(0)))) + 1) \upharpoonright (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(37) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let I be a macro instruction. If I is closed on s and halting on s, then $\mathrm{Directed}(I)$ is psuedo-closed on s.

(38) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let I be a macro instruction. If I is closed on s and halting on s, then psuedo $-$ LifeSpan$(s, \mathrm{Directed}(I)) = \mathrm{LifeSpan}(s+\cdot(I+\cdot\,\text{Start-At}(\mathrm{insloc}(0)))) + 1$.

(39) Let I be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. If I is halt-free, then $\mathrm{Directed}(I, l) = I$.

(40) For every macro instruction I such that I is halt-free holds $\mathrm{Directed}(I) = I$.

(41) For all macro instructions I, J holds $\mathrm{Directed}(I);J = I;J$.

(42) Let s be a state of $\mathbf{SCM}_{\rm FSA}$ and let I, J be macro instructions. Suppose I is closed on s and halting on s. Then

(i) for every natural number k such that $k \leq \mathrm{LifeSpan}(s{+\cdot}(I{+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0))))$ holds $\mathbf{IC}_{(\mathrm{Computation}(s{+\cdot}(\mathrm{Directed}(I){+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)} = \mathbf{IC}_{(\mathrm{Computation}(s{+\cdot}((I;J){+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)}$ and $\mathrm{CurInstr}((\mathrm{Computation}(s{+\cdot}(\mathrm{Directed}(I){+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k)) = \mathrm{CurInstr}((\mathrm{Computation}(s{+\cdot}((I;J){+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(k))$,

(ii) $(\mathrm{Computation}(s{+\cdot}(\mathrm{Directed}(I){+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s{+\cdot}(I{+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))) + 1) \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations}) = (\mathrm{Computation}(s{+\cdot}((I;J){+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s{+\cdot}(I{+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))) + 1) \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations})$, and

(iii) $\mathbf{IC}_{(\mathrm{Computation}(s{+\cdot}(\mathrm{Directed}(I){+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s{+\cdot}(I{+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0))))+1)} = \mathbf{IC}_{(\mathrm{Computation}(s{+\cdot}((I;J){+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))))(\mathrm{LifeSpan}(s{+\cdot}(I{+\cdot}\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0))))+1)}$.

(43) Let s be a state of $\mathbf{SCM}_{\rm FSA}$ and let I, J be macro instructions. Suppose I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$. Then

(i) for every natural number k such that $k \leq \mathrm{LifeSpan}(s{+\cdot}\,\mathrm{Initialized}(I))$ holds $\mathbf{IC}_{(\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(\mathrm{Directed}(I))))(k)} = \mathbf{IC}_{(\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(I;J)))(k)}$ and $\mathrm{CurInstr}((\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(\mathrm{Directed}(I))))(k)) = \mathrm{CurInstr}((\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(I;J)))(k))$,

(ii) $(\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(\mathrm{Directed}(I))))(\mathrm{LifeSpan}(s{+\cdot}\,\mathrm{Initialized}(I))+1) \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations}) = (\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(I;J)))(\mathrm{LifeSpan}(s{+\cdot}\,\mathrm{Initialized}(I))+1) \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations})$, and

(iii) $\mathbf{IC}_{(\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(\mathrm{Directed}(I))))(\mathrm{LifeSpan}(s{+\cdot}\,\mathrm{Initialized}(I))+1)} = \mathbf{IC}_{(\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(I;J)))(\mathrm{LifeSpan}(s{+\cdot}\,\mathrm{Initialized}(I))+1)}$.

(44) Let s be a state of $\mathbf{SCM}_{\rm FSA}$ and let I be a macro instruction. Suppose I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$. Let k be a natural number. Suppose $k \leq \mathrm{LifeSpan}(s{+\cdot}\,\mathrm{Initialized}(I))$. Then $(\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(I)))(k)$ and $(\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(\mathrm{Directed}(I))))(k)$ are equal outside the instruction locations of $\mathbf{SCM}_{\rm FSA}$ and $\mathrm{CurInstr}((\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(\mathrm{Directed}(I))))(k)) \neq \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}$.

(45) Let s be a state of $\mathbf{SCM}_{\rm FSA}$ and let I be a macro instruction. Suppose I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$. Then $\mathbf{IC}_{(\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(\mathrm{Directed}(I))))(\mathrm{LifeSpan}(s{+\cdot}\,\mathrm{Initialized}(I))+1)} = \mathrm{insloc}(\mathrm{card}\, I)$ and $(\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(I)))(\mathrm{LifeSpan}(s{+\cdot}\,\mathrm{Initialized}(I))) \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations}) = (\mathrm{Computation}(s{+\cdot}\,\mathrm{Initialized}(\mathrm{Directed}(I))))(\mathrm{LifeSpan}(s{+\cdot}\,\mathrm{Initialized}(I))+1) \restriction (\mathrm{Int\text{-}Locations}$

$\cup$ FinSeq-Locations).

(46) Let I be a macro instruction and let s be a state of $\mathbf{SCM}_{\rm FSA}$. Suppose I is closed on s and halting on s. Then I;$\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}}$ is closed on s and I;$\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}}$ is halting on s.

(47) For every instruction-location l of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{insloc}(0) \in \mathrm{dom}\,\mathrm{Goto}(l)$ and $(\mathrm{Goto}(l))(\mathrm{insloc}(0)) = \mathrm{goto}\ l$.

(48) Let I be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ and let x be arbitrary. If $x \in \mathrm{dom}\, I$, then $I(x)$ is an instruction of $\mathbf{SCM}_{\rm FSA}$.

(49) Let I be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$, and let x be arbitrary, and let k be a natural number. If $x \in \mathrm{dom}\,\mathrm{ProgramPart}(\mathrm{Relocated}(I,k))$, then $(\mathrm{ProgramPart}(\mathrm{Relocated}(I,k)))(x) = (\mathrm{Relocated}(I,k))(x)$.

(50) For every programmed finite partial state I of $\mathbf{SCM}_{\rm FSA}$ and for every natural number k holds $\mathrm{ProgramPart}(\mathrm{Relocated}(\mathrm{Directed}(I),k)) = \mathrm{Directed}(\mathrm{ProgramPart}(\mathrm{Relocated}(I,k)),\mathrm{insloc}(\mathrm{card}\, I + k))$.

(51) Let I, J be programmed finite partial states of $\mathbf{SCM}_{\rm FSA}$ and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. Then $\mathrm{Directed}(I{+}{\cdot}J, l) = \mathrm{Directed}(I,l){+}{\cdot}\,\mathrm{Directed}(J,l)$.

(52) For all macro instructions I, J holds $\mathrm{Directed}(I;J) = I;\mathrm{Directed}(J)$.

(53) Let I be a macro instruction and let s be a state of $\mathbf{SCM}_{\rm FSA}$. If I is closed on Initialize(s) and halting on Initialize(s), then $\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Initialized}(I;\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}})))(\mathrm{LifeSpan}(s+\cdot\,\mathrm{Initialized}(I))+1)} = \mathrm{insloc}(\mathrm{card}\, I)$.

(54) Let I be a macro instruction and let s be a state of $\mathbf{SCM}_{\rm FSA}$. Suppose I is closed on Initialize(s) and halting on Initialize(s). Then $(\mathrm{Computation}(s{+}{\cdot}\,\mathrm{Initialized}(I)))(\mathrm{LifeSpan}(s{+}{\cdot}\,\mathrm{Initialized}(I))) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = (\mathrm{Computation}(s{+}{\cdot}\,\mathrm{Initialized}(I;\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}})))(\mathrm{LifeSpan}(s{+}{\cdot}\,\mathrm{Initialized}(I)) + 1) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(55) Let I be a macro instruction and let s be a state of $\mathbf{SCM}_{\rm FSA}$. If I is closed on Initialize(s) and halting on Initialize(s), then $s{+}{\cdot}\,\mathrm{Initialized}(I;\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}})$ is halting.

(56) Let I be a macro instruction and let s be a state of $\mathbf{SCM}_{\rm FSA}$. If I is closed on Initialize(s) and halting on Initialize(s), then $\mathrm{LifeSpan}(s{+}{\cdot}\,\mathrm{Initialized}(I;\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}})) = \mathrm{LifeSpan}(s{+}{\cdot}\,\mathrm{Initialized}(I)) + 1$.

(57) Let s be a state of $\mathbf{SCM}_{\rm FSA}$ and let I be a macro instruction. If I is closed on Initialize(s) and halting on Initialize(s), then $\mathrm{IExec}(I;\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}}, s) = \mathrm{IExec}(I,s){+}{\cdot}\,\text{Start-At}(\mathrm{insloc}(\mathrm{card}\, I))$.

(58) Let I, J be macro instructions and let s be a state of $\mathbf{SCM}_{\rm FSA}$. Suppose I is closed on s and halting on s. Then $I;\mathrm{Goto}(\mathrm{insloc}(\mathrm{card}\, J + 1));J;\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}}$ is closed on s and $I;\mathrm{Goto}(\mathrm{insloc}(\mathrm{card}\, J + 1));J;\mathrm{Stop}_{\mathrm{SCM}_{\rm FSA}}$ is halting on s.

(59) Let I, J be macro instructions and let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. If I is closed on s and halting on s, then $s+\cdot((I;\mathrm{Goto}(\mathrm{insloc}(\mathrm{card}\,J+1));J;\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}})+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(0)))$ is halting.

(60) Let I, J be macro instructions and let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. If I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$, then $s+\cdot\,\mathrm{Initialized}(I;\mathrm{Goto}(\mathrm{insloc}(\mathrm{card}\,J+1));J;\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}})$ is halting.

(61) Let I, J be macro instructions and let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. If I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$, then $\mathbf{IC}_{\mathrm{IExec}(I;\,\mathrm{Goto}(\mathrm{insloc}(\mathrm{card}\,J+1));J;\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}},s)}=\mathrm{insloc}(\mathrm{card}\,I+\mathrm{card}\,J+1)$.

(62) Let I, J be macro instructions and let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose I is closed on $\mathrm{Initialize}(s)$ and halting on $\mathrm{Initialize}(s)$. Then $\mathrm{IExec}(I;\mathrm{Goto}(\mathrm{insloc}(\mathrm{card}\,J+1));J;\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}},s)=\mathrm{IExec}(I,s)+\cdot\,\mathrm{Start\text{-}At}(\mathrm{insloc}(\mathrm{card}\,I+\mathrm{card}\,J+1))$.

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. Some multi instructions defined by sequence of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):615–619, 1996.
[3] Noriko Asamoto, Yatsuka Nakamura, Piotr Rudnicki, and Andrzej Trybulec. On the composition of macro instructions. Part III. *Formalized Mathematics*, 6(**1**):53–57, 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] 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 and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[8] Grzegorz Bancerek and Piotr Rudnicki. Development of terminology for **SCM**. *Formalized Mathematics*, 4(**1**):61–67, 1993.
[9] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.
[10] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.
[11] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[12] 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.
[13] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[14] Piotr Rudnicki and Andrzej Trybulec. Memory handling for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 6(**1**):29–36, 1997.
[15] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[16] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[17] Andrzej Trybulec and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):571–576, 1996.
[18] Andrzej Trybulec and Yatsuka Nakamura. Relocability for $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):583–586, 1996.
[19] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.

[20] Andrzej Trybulec, Yatsuka Nakamura, and Noriko Asamoto. On the compositions of macro instructions. Part I. *Formalized Mathematics*, 6(**1**):21–27, 1997.
[21] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\mathrm{FSA}}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[22] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[23] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received August 27, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Conditional Branch Macro Instructions of $\mathbf{SCM}_{\rm FSA}$. Part II

Noriko Asamoto
Ochanomizu University
Tokyo

MML Identifier: `SCMFSA8B`.

The papers [22], [31], [16], [7], [29], [11], [32], [13], [14], [10], [6], [8], [12], [30], [15], [21], [17], [18], [25], [20], [27], [28], [23], [24], [3], [9], [26], [19], [5], [4], [2], and [1] provide the terminology and notation for this paper.

One can prove the following propositions:

(1) For every state s of $\mathbf{SCM}_{\rm FSA}$ holds $\mathbf{IC}_{\mathbf{SCM}_{\rm FSA}} \in \operatorname{dom} s$.

(2) For every state s of $\mathbf{SCM}_{\rm FSA}$ and for every instruction-location l of $\mathbf{SCM}_{\rm FSA}$ holds $l \in \operatorname{dom} s$.

(3) For every macro instruction I and for every state s of $\mathbf{SCM}_{\rm FSA}$ such that I is closed on s holds $\operatorname{insloc}(0) \in \operatorname{dom} I$.

(4) For every state s of $\mathbf{SCM}_{\rm FSA}$ and for all instructions-locations l_1, l_2 of $\mathbf{SCM}_{\rm FSA}$ holds $s+\cdot \text{Start-At}(l_1)+\cdot \text{Start-At}(l_2) = s+\cdot \text{Start-At}(l_2)$.

(5) For every state s of $\mathbf{SCM}_{\rm FSA}$ and for every macro instruction I holds $\text{Initialize}(s) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = (s+\cdot \text{Initialized}(I)) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(6) Let s_1, s_2 be states of $\mathbf{SCM}_{\rm FSA}$ and let I be a macro instruction. If $s_1 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = s_2 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$, then if I is closed on s_1, then I is closed on s_2.

(7) Let s_1, s_2 be states of $\mathbf{SCM}_{\rm FSA}$ and let I, J be macro instructions. Suppose $s_1 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = s_2 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$. Then $s_1+\cdot(I+\cdot \text{Start-At}(\operatorname{insloc}(0)))$ and $s_2+\cdot(J+\cdot \text{Start-At}(\operatorname{insloc}(0)))$ are equal outside the instruction locations of $\mathbf{SCM}_{\rm FSA}$.

(8) Let s_1, s_2 be states of $\mathbf{SCM}_{\rm FSA}$ and let I be a macro instruction. Suppose $s_1 \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = s_2 \restriction$

ISSN 1426–2630

(Int-Locations $\cup$ FinSeq-Locations). Suppose I is closed on s_1 and halting on s_1. Then I is closed on s_2 and halting on s_2.

(9) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ and for all macro instructions I, J holds I is closed on Initialize(s) iff I is closed on $s+\cdot$ Initialized(J).

(10) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I, J be macro instructions, and let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. Then I is closed on s if and only if I is closed on $s+\cdot(I+\cdot$ Start-At(l)).

(11) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$ and let I be a macro instruction. Suppose $I+\cdot$ Start-At(insloc(0)) $\subseteq s_1$ and I is closed on s_1. Let n be a natural number. Suppose ProgramPart(Relocated(I, n)) $\subseteq s_2$ and $\mathbf{IC}_{(s_2)} =$ insloc(n) and $s_1 \restriction$ (Int-Locations $\cup$ FinSeq-Locations) $=$ $s_2 \restriction$ (Int-Locations $\cup$ FinSeq-Locations). Let i be a natural number. Then $\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)} + n = \mathbf{IC}_{(\mathrm{Computation}(s_2))(i)}$ and IncAddr(CurInstr((Computation(s_1))(i)), n) $=$ CurInstr((Computation (s_2))(i)) and (Computation(s_1))(i) $\restriction$ (Int-Locations $\cup$ FinSeq-Locations) $=$ (Computation(s_2))(i) $\restriction$ (Int-Locations $\cup$ FinSeq-Locations).

(12) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let i be a keeping 0 parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, and let J be a parahalting macro instruction, and let a be an integer location. Then (IExec(i;J, s))(a) $=$ (IExec(J, Exec(i, Initialize(s))))(a).

(13) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let i be a keeping 0 parahalting instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, and let J be a parahalting macro instruction, and let f be a finite sequence location. Then (IExec(i;J, s))(f) $=$ (IExec(J, Exec(i, Initialize(s))))(f).

Let a be an integer location and let I, J be macro instructions. The functor $if = 0(a, I, J)$ yields a macro instruction and is defined by:

(Def. 1) $if = 0(a, I, J) =$ (**if** $a = 0$ **goto** insloc(card J+3));J; Goto(insloc(card $I + 1$));I;$\mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

The functor $if > 0(a, I, J)$ yields a macro instruction and is defined by:

(Def. 2) $if > 0(a, I, J) =$ (**if** $a > 0$ **goto** insloc(card J+3));J; Goto(insloc(card $I + 1$));I;$\mathrm{Stop}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

Let a be an integer location and let I, J be macro instructions. The functor $if < 0(a, I, J)$ yields a macro instruction and is defined as follows:

(Def. 3) $if < 0(a, I, J) = if = 0(a, J, if > 0(a, J, I))$.

The following propositions are true:

(14) For all macro instructions I, J and for every integer location a holds card $if = 0(a, I, J) =$ card $I +$ card $J + 4$.

(15) For all macro instructions I, J and for every integer location a holds card $if > 0(a, I, J) =$ card $I +$ card $J + 4$.

(16) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I, J be macro instructions, and let a be a read-write integer location. Suppose $s(a) = 0$ and I is closed on

s and halting on s. Then $if = 0(a, I, J)$ is closed on s and $if = 0(a, I, J)$ is halting on s.

(17) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I, J be macro instructions, and let a be a read-write integer location. Suppose $s(a) = 0$ and I is closed on Initialize(s) and halting on Initialize(s). Then $\mathrm{IExec}(if = 0(a, I, J), s) = \mathrm{IExec}(I, s) +\cdot \text{Start-At}(\mathrm{insloc}(\mathrm{card}\, I + \mathrm{card}\, J + 3))$.

(18) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I, J be macro instructions, and let a be a read-write integer location. Suppose $s(a) \neq 0$ and J is closed on s and halting on s. Then $if = 0(a, I, J)$ is closed on s and $if = 0(a, I, J)$ is halting on s.

(19) Let I, J be macro instructions, and let a be a read-write integer location, and let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $s(a) \neq 0$ and J is closed on Initialize(s) and halting on Initialize(s). Then $\mathrm{IExec}(if = 0(a, I, J), s) = \mathrm{IExec}(J, s) +\cdot \text{Start-At}(\mathrm{insloc}(\mathrm{card}\, I + \mathrm{card}\, J + 3))$.

(20) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I, J be parahalting macro instructions, and let a be a read-write integer location. Then $if = 0(a, I, J)$ is parahalting and if $s(a) = 0$, then $\mathrm{IExec}(if = 0(a, I, J), s) = \mathrm{IExec}(I, s) +\cdot \text{Start-At}(\mathrm{insloc}(\mathrm{card}\, I + \mathrm{card}\, J + 3))$ and if $s(a) \neq 0$, then $\mathrm{IExec}(if = 0(a, I, J), s) = \mathrm{IExec}(J, s) +\cdot \text{Start-At}(\mathrm{insloc}(\mathrm{card}\, I + \mathrm{card}\, J + 3))$.

(21) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I, J be parahalting macro instructions, and let a be a read-write integer location. Then

(i) $\mathbf{IC}_{\mathrm{IExec}(if = 0(a, I, J), s)} = \mathrm{insloc}(\mathrm{card}\, I + \mathrm{card}\, J + 3)$,

(ii) if $s(a) = 0$, then for every integer location d holds $(\mathrm{IExec}(if = 0(a, I, J), s))(d) = (\mathrm{IExec}(I, s))(d)$ and for every finite sequence location f holds $(\mathrm{IExec}(if = 0(a, I, J), s))(f) = (\mathrm{IExec}(I, s))(f)$, and

(iii) if $s(a) \neq 0$, then for every integer location d holds $(\mathrm{IExec}(if = 0(a, I, J), s))(d) = (\mathrm{IExec}(J, s))(d)$ and for every finite sequence location f holds $(\mathrm{IExec}(if = 0(a, I, J), s))(f) = (\mathrm{IExec}(J, s))(f)$.

(22) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I, J be macro instructions, and let a be a read-write integer location. Suppose $s(a) > 0$ and I is closed on s and halting on s. Then $if > 0(a, I, J)$ is closed on s and $if > 0(a, I, J)$ is halting on s.

(23) Let I, J be macro instructions, and let a be a read-write integer location, and let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $s(a) > 0$ and I is closed on Initialize(s) and halting on Initialize(s). Then $\mathrm{IExec}(if > 0(a, I, J), s) = \mathrm{IExec}(I, s) +\cdot \text{Start-At}(\mathrm{insloc}(\mathrm{card}\, I + \mathrm{card}\, J + 3))$.

(24) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I, J be macro instructions, and let a be a read-write integer location. Suppose $s(a) \leq 0$ and J is closed on s and halting on s. Then $if > 0(a, I, J)$ is closed on s and $if > 0(a, I, J)$ is halting on s.

(25) Let I, J be macro instructions, and let a be a read-write integer location, and let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $s(a) \leq 0$ and J is closed on

Initialize(s) and halting on Initialize(s). Then IExec($if > 0(a, I, J), s$) = IExec(J, s)+· Start-At(insloc(card I + card J + 3)).

(26) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let I, J be parahalting macro instructions, and let a be a read-write integer location. Then $if > 0(a, I, J)$ is parahalting and if $s(a) > 0$, then IExec($if > 0(a, I, J), s$) = IExec(I, s)+· Start-At(insloc(card I + card J + 3)) and if $s(a) \leq 0$, then IExec($if > 0(a, I, J), s$) = IExec(J, s)+· Start-At(insloc(card I + card J + 3)).

(27) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let I, J be parahalting macro instructions, and let a be a read-write integer location. Then

(i) $\mathbf{IC}_{{\rm IExec}(if>0(a,I,J),s)}$ = insloc(card I + card J + 3),

(ii) if $s(a) > 0$, then for every integer location d holds (IExec($if > 0(a, I, J), s$))(d) = (IExec(I, s))(d) and for every finite sequence location f holds (IExec($if > 0(a, I, J), s$))(f) = (IExec(I, s))(f), and

(iii) if $s(a) \leq 0$, then for every integer location d holds (IExec($if > 0(a, I, J), s$))(d) = (IExec(J, s))(d) and for every finite sequence location f holds (IExec($if > 0(a, I, J), s$))(f) = (IExec(J, s))(f).

(28) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let I, J be macro instructions, and let a be a read-write integer location. Suppose $s(a) < 0$ and I is closed on s and halting on s. Then $if < 0(a, I, J)$ is closed on s and $if < 0(a, I, J)$ is halting on s.

(29) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let I, J be macro instructions, and let a be a read-write integer location. Suppose $s(a) < 0$ and I is closed on Initialize(s) and halting on Initialize(s). Then IExec($if < 0(a, I, J), s$) = IExec(I, s)+· Start-At(insloc(card I + card J + card J + 7)).

(30) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let I, J be macro instructions, and let a be a read-write integer location. Suppose $s(a) = 0$ and J is closed on s and halting on s. Then $if < 0(a, I, J)$ is closed on s and $if < 0(a, I, J)$ is halting on s.

(31) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let I, J be macro instructions, and let a be a read-write integer location. Suppose $s(a) = 0$ and J is closed on Initialize(s) and halting on Initialize(s). Then IExec($if < 0(a, I, J), s$) = IExec(J, s)+· Start-At(insloc(card I + card J + card J + 7)).

(32) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let I, J be macro instructions, and let a be a read-write integer location. Suppose $s(a) > 0$ and J is closed on s and halting on s. Then $if < 0(a, I, J)$ is closed on s and $if < 0(a, I, J)$ is halting on s.

(33) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let I, J be macro instructions, and let a be a read-write integer location. Suppose $s(a) > 0$ and J is closed on Initialize(s) and halting on Initialize(s). Then IExec($if < 0(a, I, J), s$) = IExec(J, s)+· Start-At(insloc(card I + card J + card J + 7)).

(34) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let I, J be parahalting macro instructions, and let a be a read-write integer location. Then

(i) $if < 0(a, I, J)$ is parahalting,

(ii) if $s(a) < 0$, then $\mathrm{IExec}(if < 0(a, I, J), s) = \mathrm{IExec}(I, s) +\cdot$ Start-At(insloc (card I + card J + card $J + 7$)), and

(iii) if $s(a) \geq 0$, then $\mathrm{IExec}(if < 0(a, I, J), s) = \mathrm{IExec}(J, s) +\cdot$ Start-At(insloc (card I + card J + card $J + 7$)).

Let I, J be parahalting macro instructions and let a be a read-write integer location. Observe that $if = 0(a, I, J)$ is parahalting and $if > 0(a, I, J)$ is parahalting.

Let a, b be integer locations and let I, J be macro instructions. The functor $if = 0(a, b, I, J)$ yields a macro instruction and is defined as follows:

(Def. 4) $if = 0(a, b, I, J) = \mathrm{SubFrom}(a, b);if = 0(a, I, J)$.

The functor $if > 0(a, b, I, J)$ yields a macro instruction and is defined by:

(Def. 5) $if > 0(a, b, I, J) = \mathrm{SubFrom}(a, b);if > 0(a, I, J)$.

We introduce $if < 0(b, a, I, J)$ as a synonym of $if > 0(a, b, I, J)$.

Let I, J be parahalting macro instructions and let a, b be read-write integer locations. One can check that $if = 0(a, b, I, J)$ is parahalting and $if > 0(a, b, I, J)$ is parahalting.

Next we state several propositions:

(35) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every macro instruction I holds $\mathrm{Result}(s +\cdot \mathrm{Initialized}(I)) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations}) = \mathrm{IExec}(I, s) \restriction (\text{Int-Locations} \cup \text{FinSeq-Locations})$.

(36) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I be a macro instruction, and let a be an integer location. Then $\mathrm{Result}(s +\cdot \mathrm{Initialized}(I))$ and $\mathrm{IExec}(I, s)$ are equal outside the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(37) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$, and let i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, and let a be an integer location. Suppose that

(i) for every integer location b such that $a \neq b$ holds $s_1(b) = s_2(b)$,

(ii) for every finite sequence location f holds $s_1(f) = s_2(f)$,

(iii) i does not refer a, and

(iv) $\mathbf{IC}_{(s_1)} = \mathbf{IC}_{(s_2)}$.

Then

(v) for every integer location b such that $a \neq b$ holds $(\mathrm{Exec}(i, s_1))(b) = (\mathrm{Exec}(i, s_2))(b)$,

(vi) for every finite sequence location f holds $(\mathrm{Exec}(i, s_1))(f) = (\mathrm{Exec}(i, s_2))(f)$, and

(vii) $\mathbf{IC}_{\mathrm{Exec}(i, s_1)} = \mathbf{IC}_{\mathrm{Exec}(i, s_2)}$.

(38) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I be a macro instruction, and let a be an integer location. Suppose that

(i) I does not refer a,

(ii) for every integer location b such that $a \neq b$ holds $s_1(b) = s_2(b)$,

(iii) for every finite sequence location f holds $s_1(f) = s_2(f)$, and

(iv) I is closed on s_1 and halting on s_1.

Let k be a natural number. Then

(v) for every integer location b such that $a \neq b$ holds (Computation($s_1 +\cdot (I +\cdot$ Start-At(insloc(0)))))(k)(b) = (Computation($s_2 +\cdot (I +\cdot$ Start-At (insloc(0)))))(k)(b),

(vi) for every finite sequence location f holds (Computation($s_1 +\cdot (I +\cdot$ Start-At(insloc(0)))))(k)(f) = (Computation($s_2 +\cdot (I +\cdot$ Start-At (insloc(0)))))(k)(f),

(vii) $\mathbf{IC}_{(\mathrm{Computation}(s_1 +\cdot (I +\cdot \mathrm{Start\text{-}At(insloc(0)))))(k)}} = \mathbf{IC}_{(\mathrm{Computation}(s_2 +\cdot (I +\cdot \mathrm{Start\text{-}At(insloc(0)))))(k)}}$, and

(viii) CurInstr((Computation($s_1 +\cdot (I +\cdot$ Start-At(insloc(0)))))(k)) = CurInstr((Computation($s_2 +\cdot (I +\cdot$ Start-At(insloc(0)))))(k)).

(39) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I, J be macro instructions, and let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. Then I is closed on s and halting on s if and only if I is closed on $s +\cdot (I +\cdot$ Start-At(l)) and halting on $s +\cdot (I +\cdot$ Start-At(l)).

(40) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I be a macro instruction, and let a be an integer location. Suppose that

(i) I does not refer a,

(ii) for every integer location b such that $a \neq b$ holds $s_1(b) = s_2(b)$,

(iii) for every finite sequence location f holds $s_1(f) = s_2(f)$, and

(iv) I is closed on s_1 and halting on s_1.

Then I is closed on s_2 and halting on s_2.

(41) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I be a macro instruction, and let a be an integer location. Suppose that

(i) for every read-write integer location d such that $a \neq d$ holds $s_1(d) = s_2(d)$,

(ii) for every finite sequence location f holds $s_1(f) = s_2(f)$,

(iii) I does not refer a, and

(iv) I is closed on Initialize(s_1) and halting on Initialize(s_1).

Then

(v) for every integer location d such that $a \neq d$ holds (IExec(I, s_1))(d) = (IExec(I, s_2))(d),

(vi) for every finite sequence location f holds (IExec(I, s_1))(f) = (IExec(I, s_2))(f), and

(vii) $\mathbf{IC}_{\mathrm{IExec}(I,s_1)} = \mathbf{IC}_{\mathrm{IExec}(I,s_2)}$.

(42) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I, J be parahalting macro instructions, and let a, b be read-write integer locations. Suppose I does not refer a and J does not refer a. Then

(i) $\mathbf{IC}_{\mathrm{IExec}(if=0(a,b,I,J),s)} = \mathrm{insloc}(\mathrm{card}\, I + \mathrm{card}\, J + 5)$,

(ii) if $s(a) = s(b)$, then for every integer location d such that $a \neq d$ holds (IExec($if = 0(a, b, I, J), s$))(d) = (IExec(I, s))(d) and for every finite sequence location f holds (IExec($if = 0(a, b, I, J), s$))(f) = (IExec(I, s))(f), and

(iii) if $s(a) \neq s(b)$, then for every integer location d such that $a \neq d$ holds (IExec($if = 0(a, b, I, J), s$))(d) = (IExec(J, s))(d) and for every finite sequence location f holds $(\mathrm{IExec}(if = 0(a,b,I,J),s))(f) = (\mathrm{IExec}(J,s))(f)$.

(43) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I, J be parahalting macro instructions, and let a, b be read-write integer locations. Suppose I does not refer a and J does not refer a. Then

(i) $\mathbf{IC}_{\mathrm{IExec}(if>0(a,b,I,J),s)} = \mathrm{insloc}(\mathrm{card}\, I + \mathrm{card}\, J + 5)$,

(ii) if $s(a) > s(b)$, then for every integer location d such that $a \neq d$ holds $(\mathrm{IExec}(if > 0(a,b,I,J),s))(d) = (\mathrm{IExec}(I,s))(d)$ and for every finite sequence location f holds $(\mathrm{IExec}(if > 0(a,b,I,J),s))(f) = (\mathrm{IExec}(I,s))(f)$, and

(iii) if $s(a) \leq s(b)$, then for every integer location d such that $a \neq d$ holds $(\mathrm{IExec}(if > 0(a,b,I,J),s))(d) = (\mathrm{IExec}(J,s))(d)$ and for every finite sequence location f holds $(\mathrm{IExec}(if > 0(a,b,I,J),s))(f) = (\mathrm{IExec}(J,s))(f)$.

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

[3] Noriko Asamoto. Some multi instructions defined by sequence of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):615–619, 1996.

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

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

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

[9] Grzegorz Bancerek and Piotr Rudnicki. Development of terminology for **SCM**. *Formalized Mathematics*, 4(**1**):61–67, 1993.

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

[11] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.

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

[14] 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.

[15] Czesław Byliński. Products and coproducts in categories. *Formalized Mathematics*, 2(**5**):701–709, 1991.

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

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

[18] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.

[19] Piotr Rudnicki and Andrzej Trybulec. Memory handling for $\mathbf{SCM}_{\rm FSA}$. *Formalized Mathematics*, 6(**1**):29–36, 1997.
[20] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[21] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[22] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[23] Andrzej Trybulec and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\rm FSA}$. *Formalized Mathematics*, 5(**4**):571–576, 1996.
[24] Andrzej Trybulec and Yatsuka Nakamura. Relocability for $\mathbf{SCM}_{\rm FSA}$. *Formalized Mathematics*, 5(**4**):583–586, 1996.
[25] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[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. An extension of **SCM**. *Formalized Mathematics*, 5(**4**):507–512, 1996.
[28] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\rm FSA}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[29] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[30] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 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 August 27, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Bounds in Posets and Relational Substructures [1]

Grzegorz Bancerek
Institute of Mathematics
Polish Academy of Sciences

Summary. Notation and facts necessary to start with the formalization of continuous lattices according to [9] are introduced.

MML Identifier: `YELLOW_0`.

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

1. Reexamination of poset concepts

The scheme *RelStrEx* deals with a non empty set $\mathcal{A}$ and a binary predicate $\mathcal{P}$, and states that:

> There exists a non empty strict relational structure L such that the carrier of $L = \mathcal{A}$ and for all elements a, b of L holds $a \leq b$ iff $\mathcal{P}[a, b]$

for all values of the parameters.

Let A be a non empty relational structure. Let us observe that A is reflexive if and only if:

(Def. 1) For every element x of A holds $x \leq x$.

Let A be a relational structure. Let us observe that A is transitive if and only if:

(Def. 2) For all elements x, y, z of A such that $x \leq y$ and $y \leq z$ holds $x \leq z$.

Let us observe that A is antisymmetric if and only if:

(Def. 3) For all elements x, y of A such that $x \leq y$ and $y \leq x$ holds $x = y$.

[1] This work was partially supported by Office of Naval Research Grant N00014-95-1-1336.

ISSN 1426–2630

One can check that every non empty relational structure which is complete has l.u.b.'s and g.l.b.'s and every non empty reflexive relational structure which is trivial is also complete, transitive, and antisymmetric.

Let x be a set and let R be a binary relation on $\{x\}$. Observe that $\langle\{x\}, R\rangle$ is trivial.

Let us observe that there exists a relational structure which is strict, trivial, non empty, and reflexive.

Let L be a non empty 1-sorted structure. Observe that there exists a subset of L which is finite and non empty.

One can prove the following propositions:

(1) Let P_1, P_2 be relational structures. Suppose the relational structure of P_1 = the relational structure of P_2. Let a_1, b_1 be elements of P_1 and a_2, b_2 be elements of P_2 such that $a_1 = a_2$ and $b_1 = b_2$. Then

(i) if $a_1 \leq b_1$, then $a_2 \leq b_2$, and

(ii) if $a_1 < b_1$, then $a_2 < b_2$.

(2) Let P_1, P_2 be relational structures. Suppose the relational structure of P_1 = the relational structure of P_2. Let X be a set, a_1 be an element of P_1, and a_2 be an element of P_2 such that $a_1 = a_2$. Then

(i) if $X \leq a_1$, then $X \leq a_2$, and

(ii) if $X \geq a_1$, then $X \geq a_2$.

(3) Let P_1, P_2 be non empty relational structures. Suppose the relational structure of P_1 = the relational structure of P_2 and P_1 is complete. Then P_2 is complete.

(4) Let L be a transitive relational structure and x, y be elements of L. Suppose $x \leq y$. Let X be a set. Then

(i) if $y \leq X$, then $x \leq X$, and

(ii) if $x \geq X$, then $y \geq X$.

(5) Let L be a non empty relational structure, X be a set, and x be an element of L. Then

(i) $x \geq X$ iff $x \geq X \cap$ (the carrier of L), and

(ii) $x \leq X$ iff $x \leq X \cap$ (the carrier of L).

(6) For every relational structure L and for every element a of L holds $\emptyset \leq a$ and $\emptyset \geq a$.

(7) Let L be a relational structure and a, b be elements of L. Then

(i) $a \leq \{b\}$ iff $a \leq b$, and

(ii) $a \geq \{b\}$ iff $b \leq a$.

(8) Let L be a relational structure and a, b, c be elements of L. Then

(i) $a \leq \{b, c\}$ iff $a \leq b$ and $a \leq c$, and

(ii) $a \geq \{b, c\}$ iff $b \leq a$ and $c \leq a$.

(9) Let L be a relational structure and X, Y be sets. Suppose $X \subseteq Y$. Let x be an element of L. Then

(i) if $x \leq Y$, then $x \leq X$, and

(ii) if $x \geq Y$, then $x \geq X$.

(10) Let L be a relational structure, X, Y be sets, and x be an element of L. Then
(i) if $x \leq X$ and $x \leq Y$, then $x \leq X \cup Y$, and
(ii) if $x \geq X$ and $x \geq Y$, then $x \geq X \cup Y$.

(11) Let L be a non empty transitive relational structure, X be a set, and x, y be elements of L. If $X \leq x$ and $x \leq y$, then $X \leq y$.

(12) Let L be a non empty transitive relational structure, X be a set, and x, y be elements of L. If $X \geq x$ and $x \geq y$, then $X \geq y$.

Let L be a non empty relational structure. Note that Ω_L is non empty.

2. Least upper and greatest lower bounds

Let L be a relational structure. We say that L is lower-bounded if and only if:

(Def. 4) There exists an element x of L such that $x \leq$ the carrier of L

We say that L is upper-bounded if and only if:

(Def. 5) There exists an element x of L such that $x \geq$ the carrier of L

Let L be a relational structure. We say that L is bounded if and only if:

(Def. 6) L is lower-bounded upper-bounded.

The following proposition is true

(13) Let P_1, P_2 be relational structures such that the relational structure of $P_1 =$ the relational structure of P_2. Then
(i) if P_1 is lower-bounded, then P_2 is lower-bounded, and
(ii) if P_1 is upper-bounded, then P_2 is upper-bounded.

One can verify the following observations:

* every non empty relational structure which is complete is also bounded,
* every relational structure which is bounded is also lower-bounded and upper-bounded, and
* every relational structure which is lower-bounded and upper-bounded is also bounded.

One can verify that there exists a non empty poset which is complete.

Let L be a relational structure and let X be a set. We say that sup X exists in L if and only if the condition (Def. 7) is satisfied.

(Def. 7) There exists an element a of L such that
(i) $X \leq a$,
(ii) for every element b of L such that $X \leq b$ holds $b \geq a$, and
(iii) for every element c of L such that $X \leq c$ and for every element b of L such that $X \leq b$ holds $b \geq c$ holds $c = a$.

We say that inf X exists in L if and only if the condition (Def. 8) is satisfied.

(Def. 8) There exists an element a of L such that

(i) $X \geq a$,

(ii) for every element b of L such that $X \geq b$ holds $b \leq a$, and

(iii) for every element c of L such that $X \geq c$ and for every element b of L such that $X \geq b$ holds $b \leq c$ holds $c = a$.

One can prove the following propositions:

(14) Let L_1, L_2 be relational structures. Suppose the relational structure of $L_1 =$ the relational structure of L_2. Let X be a set. Then

(i) if sup X exists in L_1, then sup X exists in L_2, and

(ii) if inf X exists in L_1, then inf X exists in L_2.

(15) Let L be an antisymmetric relational structure and X be a set. Then sup X exists in L if and only if there exists an element a of L such that $X \leq a$ and for every element b of L such that $X \leq b$ holds $a \leq b$.

(16) Let L be an antisymmetric relational structure and X be a set. Then inf X exists in L if and only if there exists an element a of L such that $X \geq a$ and for every element b of L such that $X \geq b$ holds $a \geq b$.

(17) Let L be a complete antisymmetric non empty relational structure and X be a set. Then sup X exists in L and inf X exists in L.

(18) Let L be a non empty antisymmetric relational structure and a, b, c be elements of L. Then $c = a \sqcup b$ and sup $\{a, b\}$ exists in L if and only if $c \geq a$ and $c \geq b$ and for every element d of L such that $d \geq a$ and $d \geq b$ holds $c \leq d$.

(19) Let L be a non empty antisymmetric relational structure and a, b, c be elements of L. Then $c = a \sqcap b$ and inf $\{a, b\}$ exists in L if and only if $c \leq a$ and $c \leq b$ and for every element d of L such that $d \leq a$ and $d \leq b$ holds $c \geq d$.

(20) Let L be a non empty antisymmetric relational structure. Then L has l.u.b.'s if and only if for all elements a, b of L holds sup $\{a, b\}$ exists in L.

(21) Let L be a non empty antisymmetric relational structure. Then L has g.l.b.'s if and only if for all elements a, b of L holds inf $\{a, b\}$ exists in L.

(22) Let L be an antisymmetric relational structure with l.u.b.'s and a, b, c be elements of L. Then $c = a \sqcup b$ if and only if the following conditions are satisfied:

(i) $c \geq a$,

(ii) $c \geq b$, and

(iii) for every element d of L such that $d \geq a$ and $d \geq b$ holds $c \leq d$.

(23) Let L be an antisymmetric relational structure with g.l.b.'s and a, b, c be elements of L. Then $c = a \sqcap b$ if and only if the following conditions are satisfied:

(i) $c \leq a$,

(ii) $c \leq b$, and

(iii) for every element d of L such that $d \leq a$ and $d \leq b$ holds $c \geq d$.

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

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

Let L be a non empty relational structure and let X be a set. The functor $\bigsqcup_L X$ yielding an element of L is defined as follows:

(Def. 9) $X \leq \bigsqcup_L X$ and for every element a of L such that $X \leq a$ holds $\bigsqcup_L X \leq a$ if sup X exists in L.

The functor $\bigsqcap_L X$ yielding an element of L is defined as follows:

(Def. 10) $X \geq \bigsqcap_L X$ and for every element a of L such that $X \geq a$ holds $a \leq \bigsqcap_L X$ if inf X exists in L.

We now state a number of propositions:

(26) Let L_1, L_2 be non empty relational structures. Suppose the relational structure of $L_1 =$ the relational structure of L_2. Let X be a set. If sup X exists in L_1, then $\bigsqcup_{L_1} X = \bigsqcup_{L_2} X$.

(27) Let L_1, L_2 be non empty relational structures. Suppose the relational structure of $L_1 =$ the relational structure of L_2. Let X be a set. If inf X exists in L_1, then $\bigsqcap_{L_1} X = \bigsqcap_{L_2} X$.

(28) For every complete non empty poset L and for every set X holds $\bigsqcup_L X = \bigsqcup_{(\mathbb{L}_L)} X$ and $\bigsqcap_L X = \bigsqcap_{(\mathbb{L}_L)} X$.

(29) For every complete lattice L and for every set X holds $\bigsqcup_L X = \bigsqcup_{\mathrm{Poset}(L)} X$ and $\bigsqcap_L X = \bigsqcap_{\mathrm{Poset}(L)} X$.

(30) Let L be a non empty antisymmetric relational structure, a be an element of L, and X be a set. Then $a = \bigsqcup_L X$ and sup X exists in L if and only if $a \geq X$ and for every element b of L such that $b \geq X$ holds $a \leq b$.

(31) Let L be a non empty antisymmetric relational structure, a be an element of L, and X be a set. Then $a = \bigsqcap_L X$ and inf X exists in L if and only if $a \leq X$ and for every element b of L such that $b \leq X$ holds $a \geq b$.

(32) Let L be a complete antisymmetric non empty relational structure, a be an element of L, and X be a set. Then $a = \bigsqcup_L X$ if and only if the following conditions are satisfied:

(i) $a \geq X$, and

(ii) for every element b of L such that $b \geq X$ holds $a \leq b$.

(33) Let L be a complete antisymmetric non empty relational structure, a be an element of L, and X be a set. Then $a = \bigsqcap_L X$ if and only if the following conditions are satisfied:

(i) $a \leq X$, and

(ii) for every element b of L such that $b \leq X$ holds $a \geq b$.

(34) Let L be a non empty relational structure and X, Y be sets. Suppose $X \subseteq Y$ and sup X exists in L and sup Y exists in L. Then $\bigsqcup_L X \leq \bigsqcup_L Y$.

(35) Let L be a non empty relational structure and X, Y be sets. Suppose $X \subseteq Y$ and inf X exists in L and inf Y exists in L. Then $\bigsqcap_L X \geq \bigsqcap_L Y$.

(36) Let L be a non empty antisymmetric transitive relational structure and X, Y be sets. Suppose sup X exists in L and sup Y exists in L and sup $X \cup Y$ exists in L. Then $\bigsqcup_L (X \cup Y) = \bigsqcup_L X \sqcup \bigsqcup_L Y$.

(37) Let L be a non empty antisymmetric transitive relational structure and X, Y be sets. Suppose inf X exists in L and inf Y exists in L and inf $X \cup Y$ exists in L. Then $\bigsqcap_L (X \cup Y) = \bigsqcap_L X \sqcap \bigsqcap_L Y$.

Let L be a non empty relational structure and let X be a subset of the carrier of L. We introduce $\sup X$ as a synonym of $\bigsqcup_L X$. We introduce $\inf X$ as a synonym of $\bigsqcap_L X$.

We now state several propositions:

(38) Let L be a non empty reflexive antisymmetric relational structure and a be an element of L. Then sup $\{a\}$ exists in L and inf $\{a\}$ exists in L.

(39) Let L be a non empty reflexive antisymmetric relational structure and a be an element of L. Then $\sup\{a\} = a$ and $\inf\{a\} = a$.

(40) For every poset L with g.l.b.'s and for all elements a, b of L holds $\inf\{a, b\} = a \sqcap b$.

(41) For every poset L with l.u.b.'s and for all elements a, b of L holds $\sup\{a, b\} = a \sqcup b$.

(42) Let L be a lower-bounded antisymmetric non empty relational structure. Then sup $\emptyset$ exists in L and inf the carrier of L exists in L.

(43) Let L be an upper-bounded antisymmetric non empty relational structure. Then inf $\emptyset$ exists in L and sup the carrier of L exists in L.

Let L be a non empty relational structure. The functor $\bot_L$ yielding an element of L is defined by:

(Def. 11) $\bot_L = \bigsqcup_L \emptyset$.

The functor $\top_L$ yields an element of L and is defined by:

(Def. 12) $\top_L = \bigsqcap_L \emptyset$.

The following propositions are true:

(44) For every lower-bounded antisymmetric non empty relational structure L and for every element x of L holds $\bot_L \leq x$.

(45) For every upper-bounded antisymmetric non empty relational structure L and for every element x of L holds $x \leq \top_L$.

(46) Let L be a non empty relational structure and X, Y be sets. Suppose that for every element x of L holds $x \geq X$ iff $x \geq Y$. If sup X exists in L, then sup Y exists in L.

(47) Let L be a non empty relational structure and X, Y be sets. Suppose sup X exists in L and for every element x of L holds $x \geq X$ iff $x \geq Y$. Then $\bigsqcup_L X = \bigsqcup_L Y$.

(48) Let L be a non empty relational structure and X, Y be sets. Suppose that for every element x of L holds $x \leq X$ iff $x \leq Y$. If inf X exists in L, then inf Y exists in L.

(49) Let L be a non empty relational structure and X, Y be sets. Suppose inf X exists in L and for every element x of L holds $x \leq X$ iff $x \leq Y$. Then $\bigsqcap_L X = \bigsqcap_L Y$.

(50) Let L be a non empty relational structure and X be a set. Then
 (i) sup X exists in L iff sup $X \cap$ (the carrier of L) exists in L, and
 (ii) inf X exists in L iff inf $X \cap$ (the carrier of L) exists in L.

(51) Let L be a non empty relational structure and X be a set. Suppose sup X exists in L or sup $X \cap$ (the carrier of L) exists in L. Then $\bigsqcup_L X = \bigsqcup_L (X \cap \text{(the carrier of } L))$.

(52) Let L be a non empty relational structure and X be a set. Suppose inf X exists in L or inf $X \cap$ (the carrier of L) exists in L. Then $\bigsqcap_L X = \bigsqcap_L (X \cap \text{(the carrier of } L))$.

(53) Let L be a non empty relational structure. If for every subset X of L holds sup X exists in L, then L is complete.

(54) Let L be a non empty poset. Then L has l.u.b.'s if and only if for every finite non empty subset X of L holds sup X exists in L.

(55) Let L be a non empty poset. Then L has g.l.b.'s if and only if for every finite non empty subset X of L holds inf X exists in L.

3. Relational substructures

We now state the proposition

(56) For every set X and for every binary relation R on X holds $R = R|^2 X$.

Let L be a relational structure. A relational structure is said to be a relational substructure of L if:

(Def. 13) The carrier of it $\subseteq$ the carrier of L and the internal relation of it $\subseteq$ the internal relation of L.

Let L be a relational structure and let S be a relational substructure of L. We say that S is full if and only if:

(Def. 14) The internal relation of S = (the internal relation of L) $|^2$ (the carrier of S).

Let L be a relational structure. Note that there exists a relational substructure of L which is strict and full.

Let L be a non empty relational structure. Observe that there exists a relational substructure of L which is non empty, full, and strict.

One can prove the following two propositions:

(57) Let L be a relational structure and X be a subset of the carrier of L. Then $\langle X, \text{(the internal relation of } L)|^2 (X)\rangle$ is a full relational substructure of L.

(58) Let L be a relational structure and S_1, S_2 be full relational substructures of L. Suppose the carrier of S_1 = the carrier of S_2. Then the relational structure of S_1 = the relational structure of S_2.

Let L be a relational structure and let X be a subset of the carrier of L. The functor $\mathrm{sub}(X)$ yields a full strict relational substructure of L and is defined by:

(Def. 15) The carrier of $\mathrm{sub}(X) = X$.

The following propositions are true:

(59) Let L be a non empty relational structure and S be a non empty relational substructure of L. Then every element of S is an element of L.

(60) Let L be a relational structure, S be a relational substructure of L, a, b be elements of L, and x, y be elements of S. If $x = a$ and $y = b$ and $x \leq y$, then $a \leq b$.

(61) Let L be a relational structure, S be a full relational substructure of L, a, b be elements of L, and x, y be elements of S. Suppose $x = a$ and $y = b$ and $a \leq b$ and $x \in$ the carrier of S and $y \in$ the carrier of S. Then $x \leq y$.

(62) Let L be a non empty relational structure, S be a non empty full relational substructure of L, X be a set, a be an element of L, and x be an element of S such that $x = a$. Then

(i) if $a \leq X$, then $x \leq X$, and

(ii) if $a \geq X$, then $x \geq X$.

(63) Let L be a non empty relational structure, S be a non empty relational substructure of L, X be a subset of S, a be an element of L, and x be an element of S such that $x = a$. Then

(i) if $x \leq X$, then $a \leq X$, and

(ii) if $x \geq X$, then $a \geq X$.

Let L be a reflexive relational structure. Note that every full relational substructure of L is reflexive.

Let L be a transitive relational structure. Note that every full relational substructure of L is transitive.

Let L be an antisymmetric relational structure. Note that every full relational substructure of L is antisymmetric.

Let L be a non empty relational structure and let S be a relational substructure of L. We say that S is meet-inheriting if and only if the condition (Def. 16) is satisfied.

(Def. 16) Let x, y be elements of L. Suppose $x \in$ the carrier of S and $y \in$ the carrier of S and $\inf \{x, y\}$ exists in L. Then $\inf\{x, y\} \in$ the carrier of S.

We say that S is join-inheriting if and only if the condition (Def. 17) is satisfied.

(Def. 17) Let x, y be elements of L. Suppose $x \in$ the carrier of S and $y \in$ the carrier of S and $\sup \{x, y\}$ exists in L. Then $\sup\{x, y\} \in$ the carrier of S.

Let L be a non empty relational structure and let S be a relational substructure of L. We say that S is infs-inheriting if and only if:

(Def. 18) For every subset X of S such that $\inf X$ exists in L holds $\bigsqcap_L X \in$ the carrier of S.

We say that S is sups-inheriting if and only if:

(Def. 19) For every subset X of S such that sup X exists in L holds $\bigsqcup_L X \in$ the carrier of S.

Let L be a non empty relational structure. One can check that every relational substructure of L which is infs-inheriting is also meet-inheriting and every relational substructure of L which is sups-inheriting is also join-inheriting.

Let L be a non empty relational structure. Note that there exists a relational substructure of L which is infs-inheriting, sups-inheriting, non empty, full, and strict.

Now we present two schemes. The scheme *InfsInheritingSch* concerns a non empty transitive relational structure $\mathcal{A}$, a non empty full relational substructure $\mathcal{B}$ of $\mathcal{A}$, a subset $\mathcal{C}$ of $\mathcal{B}$, and a unary predicate $\mathcal{P}$, and states that:

Inf $\mathcal{C}$ exists in $\mathcal{B}$ and $\bigsqcap_{\mathcal{B}}\mathcal{C} = \bigsqcap_{\mathcal{A}}\mathcal{C}$

provided the following conditions are met:

- For every subset Y of $\mathcal{B}$ such that $\mathcal{P}[Y]$ and inf Y exists in $\mathcal{A}$ holds $\bigsqcap_{\mathcal{A}}Y \in$ the carrier of $\mathcal{B}$,
- $\mathcal{P}[\mathcal{C}]$,
- Inf $\mathcal{C}$ exists in $\mathcal{A}$.

The scheme *SupsInheritingSch* deals with a non empty transitive relational structure $\mathcal{A}$, a non empty full relational substructure $\mathcal{B}$ of $\mathcal{A}$, a subset $\mathcal{C}$ of $\mathcal{B}$, and a unary predicate $\mathcal{P}$, and states that:

Sup $\mathcal{C}$ exists in $\mathcal{B}$ and $\bigsqcup_{\mathcal{B}}\mathcal{C} = \bigsqcup_{\mathcal{A}}\mathcal{C}$

provided the following conditions are satisfied:

- For every subset Y of $\mathcal{B}$ such that $\mathcal{P}[Y]$ and sup Y exists in $\mathcal{A}$ holds $\bigsqcup_{\mathcal{A}}Y \in$ the carrier of $\mathcal{B}$,
- $\mathcal{P}[\mathcal{C}]$,
- Sup $\mathcal{C}$ exists in $\mathcal{A}$.

One can prove the following propositions:

(64) Let L be a non empty transitive relational structure, S be an infs-inheriting non empty full relational substructure of L, and X be a subset of S. If inf X exists in L, then inf X exists in S and $\bigsqcap_S X = \bigsqcap_L X$.

(65) Let L be a non empty transitive relational structure, S be a sups-inheriting non empty full relational substructure of L, and X be a subset of S. If sup X exists in L, then sup X exists in S and $\bigsqcup_S X = \bigsqcup_L X$.

(66) Let L be a non empty transitive relational structure, S be a meet-inheriting non empty full relational substructure of L, and x, y be elements of S. Suppose inf $\{x, y\}$ exists in L. Then inf $\{x, y\}$ exists in S and $\bigsqcap_S\{x, y\} = \bigsqcap_L\{x, y\}$.

(67) Let L be a non empty transitive relational structure, S be a join-inheriting non empty full relational substructure of L, and x, y be elements of S. Suppose sup $\{x, y\}$ exists in L. Then sup $\{x, y\}$ exists in S and $\bigsqcup_S\{x, y\} = \bigsqcup_L\{x, y\}$.

Let L be an antisymmetric transitive relational structure with g.l.b.'s. Note that every non empty meet-inheriting full relational substructure of L has g.l.b.'s.

Let L be an antisymmetric transitive relational structure with l.u.b.'s. Observe that every non empty join-inheriting full relational substructure of L has l.u.b.'s.

The following four propositions are true:

(68) Let L be a complete non empty poset, S be an infs-inheriting non empty full relational substructure of L, and X be a subset of S. Then $\bigsqcap_S X = \bigsqcap_L X$.

(69) Let L be a complete non empty poset, S be a sups-inheriting non empty full relational substructure of L, and X be a subset of S. Then $\bigsqcup_S X = \bigsqcup_L X$.

(70) Let L be a poset with g.l.b.'s, S be a meet-inheriting non empty full relational substructure of L, x, y be elements of S, and a, b be elements of L. If $a = x$ and $b = y$, then $x \sqcap y = a \sqcap b$.

(71) Let L be a poset with l.u.b.'s, S be a join-inheriting non empty full relational substructure of L, x, y be elements of S, and a, b be elements of L. If $a = x$ and $b = y$, then $x \sqcup y = a \sqcup b$.

References

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

[2] Grzegorz Bancerek. Filters - Part II. Quotient lattices modulo filters and direct product of two lattices. *Formalized Mathematics*, 2(**3**):433–438, 1991.

[3] Grzegorz Bancerek. The well ordering relations. *Formalized Mathematics*, 1(**1**):123–129, 1990.

[4] Józef Białas. Group and field definitions. *Formalized Mathematics*, 1(**3**):433–439, 1990.

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

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

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

[8] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.

[9] 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.

[10] Krzysztof Hryniewiecki. Relations of tolerance. *Formalized Mathematics*, 2(**1**):105–109, 1991.

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

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

[13] 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.

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

[16] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

[17] Edmund Woronowicz and Anna Zalewska. Properties of binary relations. *Formalized Mathematics*, 1(**1**):85–89, 1990.

[18] Stanisław Żukowski. Introduction to lattice theory. *Formalized Mathematics*, 1(**1**):215–222, 1990.

Received September 10, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Directed Sets, Nets, Ideals, Filters, and Maps [1]

Grzegorz Bancerek
Institute of Mathematics
Polish Academy of Sciences

Summary. Notation and facts necessary to start with the formalization of continuous lattices according to [8] are introduced. The article contains among other things, the definition of directed and filtered subsets of a poset (see 1.1 in [8, p. 2]), the definition of nets on the poset (see 1.2 in [8, p. 2]), the definition of ideals and filters and the definition of maps preserving arbitrary and directed sups and arbitrary and filtered infs (1.9 also in [8, p. 4]). The concepts of semilattices, sup-semiletices and poset lattices (1.8 in [8, p. 4]) are also introduced. A number of facts concerning the above notion and including remarks 1.4, 1.5, and 1.10 from [8, pp. 3–5] is presented.

MML Identifier: WAYBEL_0.

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

1. Directed subsets

Let L be a relational structure and let X be a subset of L. We say that X is directed if and only if:

(Def. 1) For all elements x, y of L such that $x \in X$ and $y \in X$ there exists an element z of L such that $z \in X$ and $x \leq z$ and $y \leq z$.

We say that X is filtered if and only if:

(Def. 2) For all elements x, y of L such that $x \in X$ and $y \in X$ there exists an element z of L such that $z \in X$ and $z \leq x$ and $z \leq y$.

[1] This work has been partially supported by Office of Naval Research Grant N00014-95-1-1336.

ISSN 1426–2630

Next we state two propositions:

(1) Let L be a non empty transitive relational structure and X be a subset of L. Then X is non empty directed if and only if for every finite subset Y of X there exists an element x of L such that $x \in X$ and $x \geq Y$.

(2) Let L be a non empty transitive relational structure and X be a subset of L. Then X is non empty filtered if and only if for every finite subset Y of X there exists an element x of L such that $x \in X$ and $x \leq Y$.

Let L be a relational structure. One can verify that $\emptyset_L$ is directed and filtered.

Let L be a relational structure. Observe that there exists a subset of L which is directed and filtered.

One can prove the following three propositions:

(3) Let L_1, L_2 be relational structures. Suppose the relational structure of $L_1 =$ the relational structure of L_2. Let X_1 be a subset of L_1 and X_2 be a subset of L_2. If $X_1 = X_2$ and X_1 is directed, then X_2 is directed.

(4) Let L_1, L_2 be relational structures. Suppose the relational structure of $L_1 =$ the relational structure of L_2. Let X_1 be a subset of L_1 and X_2 be a subset of L_2. If $X_1 = X_2$ and X_1 is filtered, then X_2 is filtered.

(5) For every non empty reflexive relational structure L and for every element x of L holds $\{x\}$ is directed and filtered.

Let L be a non empty reflexive relational structure. Note that there exists a subset of L which is directed, filtered, non empty, and finite.

Let L be a relational structure with l.u.b.'s. Note that Ω_L is directed.

Let L be an upper-bounded non empty relational structure. Observe that Ω_L is directed.

Let L be a relational structure with g.l.b.'s. One can check that Ω_L is filtered.

Let L be a lower-bounded non empty relational structure. Note that Ω_L is filtered.

Let L be a non empty relational structure and let S be a relational substructure of L. We say that S is filtered-infs-inheriting if and only if:

(Def. 3) For every filtered subset X of S such that $X \neq \emptyset$ and inf X exists in L holds $\bigsqcap_L X \in$ the carrier of S.

We say that S is directed-sups-inheriting if and only if:

(Def. 4) For every directed subset X of S such that $X \neq \emptyset$ and sup X exists in L holds $\bigsqcup_L X \in$ the carrier of S.

Let L be a non empty relational structure. Observe that every relational substructure of L which is infs-inheriting is also filtered-infs-inheriting and every relational substructure of L which is sups-inheriting is also directed-sups-inheriting.

Let L be a non empty relational structure. Observe that there exists a relational substructure of L which is infs-inheriting, sups-inheriting, non empty, full, and strict.

We now state two propositions:

(6) Let L be a non empty transitive relational structure, S be a filtered-infs-inheriting non empty full relational substructure of L, and X be a filtered subset of S. Suppose $X \neq \emptyset$ and $\inf X$ exists in L. Then $\inf X$ exists in S and $\bigsqcap_S X = \bigsqcap_L X$.

(7) Let L be a non empty transitive relational structure, S be a directed-sups-inheriting non empty full relational substructure of L, and X be a directed subset of S. Suppose $X \neq \emptyset$ and $\sup X$ exists in L. Then $\sup X$ exists in S and $\bigsqcup_S X = \bigsqcup_L X$.

2. Nets

Let L_1, L_2 be non empty 1-sorted structures, let f be a map from L_1 into L_2, and let x be an element of L_1. Then $f(x)$ is an element of L_2.

Let L_1, L_2 be relational structures and let f be a map from L_1 into L_2. We say that f is antitone if and only if:

(Def. 5) For all elements x, y of L_1 such that $x \leq y$ and for all elements a, b of L_2 such that $a = f(x)$ and $b = f(y)$ holds $a \geq b$.

Let L be a 1-sorted structure. We consider net structures over L as extensions of relational structure as systems

$\langle$ a carrier, a internal relation, a mapping $\rangle$,

where the carrier is a set, the internal relation is a binary relation on the carrier, and the mapping is a function from the carrier into the carrier of L.

Let L be a 1-sorted structure, let X be a non empty set, let O be a binary relation on X, and let F be a function from X into the carrier of L. Note that $\langle X, O, F\rangle$ is non empty.

Let N be a relational structure. We say that N is directed if and only if:

(Def. 6) Ω_N is directed.

Let L be a 1-sorted structure. Note that there exists a strict net structure over L which is non empty, reflexive, transitive, antisymmetric, and directed.

Let L be a 1-sorted structure. A prenet over L is a directed non empty net structure over L.

Let L be a 1-sorted structure. A net in L is a transitive prenet over L.

Let L be a non empty 1-sorted structure and let N be a non empty net structure over L. The functor $\mathrm{netmap}(N, L)$ yields a map from N into L and is defined by:

(Def. 7) $\mathrm{netmap}(N, L) =$ the mapping of N.

Let i be an element of the carrier of N. The functor $N(i)$ yielding an element of L is defined by:

(Def. 8) $N(i) =$ (the mapping of $N)(i)$.

Let L be a non empty relational structure and let N be a non empty net structure over L. We say that N is monotone if and only if:

(Def. 9) $\mathrm{netmap}(N, L)$ is monotone.

We say that N is antitone if and only if:

(Def. 10) netmap(N, L) is antitone.

Let L be a non empty 1-sorted structure, let N be a non empty net structure over L, and let X be a set. We say that N is eventually in X if and only if:

(Def. 11) There exists an element i of N such that for every element j of N such that $i \leq j$ holds $N(j) \in X$.

We say that N is often in X if and only if:

(Def. 12) For every element i of N there exists an element j of N such that $i \leq j$ and $N(j) \in X$.

Next we state three propositions:

(8) Let L be a non empty 1-sorted structure, N be a non empty net structure over L, and X, Y be sets such that $X \subseteq Y$. Then

(i) if N is eventually in X, then N is eventually in Y, and

(ii) if N is often in X, then N is often in Y.

(9) Let L be a non empty 1-sorted structure, N be a non empty net structure over L, and X be a set. Then N is eventually in X if and only if N is not often in (the carrier of $L) \setminus (X)$.

(10) Let L be a non empty 1-sorted structure, N be a non empty net structure over L, and X be a set. Then N is often in X if and only if N is not eventually in (the carrier of $L) \setminus (X)$.

Let L be a non empty relational structure and let N be a non empty net structure over L. We say that N is eventually-directed if and only if:

(Def. 13) For every element i of N holds N is eventually in $\{N(j) : j$ ranges over elements of N, $N(i) \leq N(j)\}$.

We say that N is eventually-filtered if and only if:

(Def. 14) For every element i of N holds N is eventually in $\{N(j) : j$ ranges over elements of N, $N(i) \geq N(j)\}$.

One can prove the following propositions:

(11) Let L be a non empty relational structure and N be a non empty net structure over L. Then N is eventually-directed if and only if for every element i of N there exists an element j of N such that for every element k of N such that $j \leq k$ holds $N(i) \leq N(k)$.

(12) Let L be a non empty relational structure and N be a non empty net structure over L. Then N is eventually-filtered if and only if for every element i of N there exists an element j of N such that for every element k of N such that $j \leq k$ holds $N(i) \geq N(k)$.

Let L be a non empty relational structure. Observe that every prenet over L which is monotone is also eventually-directed and every prenet over L which is antitone is also eventually-filtered.

Let L be a non empty reflexive relational structure. Observe that there exists a prenet over L which is monotone, antitone, and strict.

3. Lower and upper subsets

Let L be a relational structure and let X be a subset of the carrier of L. The functor $\downarrow X$ yielding a subset of L is defined by:

(Def. 15) For every element x of L holds $x \in \downarrow X$ iff there exists an element y of L such that $y \geq x$ and $y \in X$.

The functor $\uparrow X$ yielding a subset of L is defined as follows:

(Def. 16) For every element x of L holds $x \in \uparrow X$ iff there exists an element y of L such that $y \leq x$ and $y \in X$.

One can prove the following three propositions:

(13) Let L_1, L_2 be relational structures. Suppose the relational structure of $L_1 =$ the relational structure of L_2. Let X be a subset of the carrier of L_1 and Y be a subset of the carrier of L_2. If $X = Y$, then $\downarrow X = \downarrow Y$ and $\uparrow X = \uparrow Y$.

(14) Let L be a non empty relational structure and X be a subset of L. Then $\downarrow X = \{x : x$ ranges over elements of L, $\bigvee_{y\,:\,\mathrm{element\ of}\ L}\ x \leq y\ \wedge\ y \in X\}$.

(15) Let L be a non empty relational structure and X be a subset of L. Then $\uparrow X = \{x : x$ ranges over elements of L, $\bigvee_{y\,:\,\mathrm{element\ of}\ L}\ x \geq y\ \wedge\ y \in X\}$.

Let L be a non empty reflexive relational structure and let X be a non empty subset of the carrier of L. Note that $\downarrow X$ is non empty and $\uparrow X$ is non empty.

We now state the proposition

(16) For every reflexive relational structure L and for every subset X of the carrier of L holds $X \subseteq \downarrow X$ and $X \subseteq \uparrow X$.

Let L be a non empty relational structure and let x be an element of the carrier of L. The functor $\downarrow x$ yields a subset of L and is defined by:

(Def. 17) $\downarrow x = \downarrow \{x\}$.

The functor $\uparrow x$ yields a subset of L and is defined by:

(Def. 18) $\uparrow x = \uparrow \{x\}$.

Next we state several propositions:

(17) For every non empty relational structure L and for all elements x, y of L holds $y \in \downarrow x$ iff $y \leq x$.

(18) For every non empty relational structure L and for all elements x, y of L holds $y \in \uparrow x$ iff $x \leq y$.

(19) Let L be a non empty reflexive antisymmetric relational structure and x, y be elements of the carrier of L. If $\downarrow x = \downarrow y$, then $x = y$.

(20) Let L be a non empty reflexive antisymmetric relational structure and x, y be elements of the carrier of L. If $\uparrow x = \uparrow y$, then $x = y$.

(21) For every non empty transitive relational structure L and for all elements x, y of L such that $x \leq y$ holds $\downarrow x \subseteq \downarrow y$.

(22) For every non empty transitive relational structure L and for all elements x, y of L such that $x \leq y$ holds $\uparrow y \subseteq \uparrow x$.

Let L be a non empty reflexive relational structure and let x be an element of the carrier of L. Note that $\downarrow x$ is non empty and directed and $\uparrow x$ is non empty and filtered.

Let L be a relational structure and let X be a subset of L. We say that X is lower if and only if:

(Def. 19) For all elements x, y of L such that $x \in X$ and $y \leq x$ holds $y \in X$.

We say that X is upper if and only if:

(Def. 20) For all elements x, y of L such that $x \in X$ and $x \leq y$ holds $y \in X$.

Let L be a relational structure. One can check that there exists a subset of L which is lower and upper.

Next we state several propositions:

(23) For every relational structure L and for every subset X of L holds X is lower iff $\downarrow X \subseteq X$.

(24) For every relational structure L and for every subset X of L holds X is upper iff $\uparrow X \subseteq X$.

(25) Let L_1, L_2 be relational structures. Suppose the relational structure of $L_1 =$ the relational structure of L_2. Let X_1 be a subset of L_1 and X_2 be a subset of L_2 such that $X_1 = X_2$. Then

(i) if X_1 is lower, then X_2 is lower, and

(ii) if X_1 is upper, then X_2 is upper.

(26) Let L be a relational structure and A be a subset of $2^{\text{the carrier of } L}$. Suppose that for every subset X of L such that $X \in A$ holds X is lower. Then $\bigcup A$ is a lower subset of L.

(27) Let L be a relational structure and X, Y be subsets of L. If X is lower and Y is lower, then $X \cap Y$ is lower and $X \cup Y$ is lower.

(28) Let L be a relational structure and A be a subset of $2^{\text{the carrier of } L}$. Suppose that for every subset X of L such that $X \in A$ holds X is upper. Then $\bigcup A$ is an upper subset of L.

(29) Let L be a relational structure and X, Y be subsets of L. If X is upper and Y is upper, then $X \cap Y$ is upper and $X \cup Y$ is upper.

Let L be a non empty transitive relational structure and let X be a subset of L. One can verify that $\downarrow X$ is lower and $\uparrow X$ is upper.

Let L be a non empty transitive relational structure and let x be an element of L. Observe that $\downarrow x$ is lower and $\uparrow x$ is upper.

Let L be a non empty relational structure. Observe that Ω_L is lower and upper.

Let L be a non empty relational structure. Note that there exists a subset of L which is non empty, lower, and upper.

Let L be a non empty reflexive transitive relational structure. Observe that there exists a subset of L which is non empty, lower, and directed and there exists a subset of L which is non empty, upper, and filtered.

Let L be a poset with g.l.b.'s and l.u.b.'s. One can verify that there exists a subset of L which is non empty, directed, filtered, lower, and upper.

Next we state the proposition

(30) Let L be a non empty transitive reflexive relational structure and X be a subset of L. Then X is directed if and only if $\downarrow X$ is directed.

Let L be a non empty transitive reflexive relational structure and let X be a directed subset of L. Note that $\downarrow X$ is directed.

We now state several propositions:

(31) Let L be a non empty transitive reflexive relational structure, X be a subset of L, and x be an element of L. Then $x \geq X$ if and only if $x \geq \downarrow X$.

(32) Let L be a non empty transitive reflexive relational structure and X be a subset of L. Then sup X exists in L if and only if sup $\downarrow X$ exists in L.

(33) Let L be a non empty transitive reflexive relational structure and X be a subset of L. If sup X exists in L, then $\sup X = \sup\downarrow X$.

(34) For every non empty poset L and for every element x of L holds sup $\downarrow x$ exists in L and $\sup\downarrow x = x$.

(35) Let L be a non empty transitive reflexive relational structure and X be a subset of L. Then X is filtered if and only if $\uparrow X$ is filtered.

Let L be a non empty transitive reflexive relational structure and let X be a filtered subset of L. Note that $\uparrow X$ is filtered.

One can prove the following four propositions:

(36) Let L be a non empty transitive reflexive relational structure, X be a subset of L, and x be an element of L. Then $x \leq X$ if and only if $x \leq \uparrow X$.

(37) Let L be a non empty transitive reflexive relational structure and X be a subset of L. Then inf X exists in L if and only if inf $\uparrow X$ exists in L.

(38) Let L be a non empty transitive reflexive relational structure and X be a subset of L. If inf X exists in L, then $\inf X = \inf\uparrow X$.

(39) For every non empty poset L and for every element x of L holds inf $\uparrow x$ exists in L and $\inf\uparrow x = x$.

4. Ideals and filters

Let L be a non empty reflexive transitive relational structure. An ideal of L is a directed lower non empty subset of L. A filter of L is a filtered upper non empty subset of L.

Next we state several propositions:

(40) Let L be an antisymmetric relational structure with l.u.b.'s and X be a lower subset of L. Then X is directed if and only if for all elements x, y of L such that $x \in X$ and $y \in X$ holds $x \sqcup y \in X$.

(41) Let L be an antisymmetric relational structure with g.l.b.'s and X be an upper subset of L. Then X is filtered if and only if for all elements x, y of L such that $x \in X$ and $y \in X$ holds $x \sqcap y \in X$.

(42) Let L be a poset with l.u.b.'s and X be a non empty lower subset of L. Then X is directed if and only if for every finite subset Y of X such that $Y \neq \emptyset$ holds $\bigsqcup_L Y \in X$.

(43) Let L be a poset with g.l.b.'s and X be a non empty upper subset of L. Then X is filtered if and only if for every finite subset Y of X such that $Y \neq \emptyset$ holds $\bigsqcap_L Y \in X$.

(44) Let L be a non empty antisymmetric relational structure. Suppose L has l.u.b.'s or g.l.b.'s. Let X, Y be subsets of L. Suppose X is lower directed and Y is lower directed. Then $X \cap Y$ is directed.

(45) Let L be a non empty antisymmetric relational structure. Suppose L has l.u.b.'s or g.l.b.'s. Let X, Y be subsets of L. Suppose X is upper filtered and Y is upper filtered. Then $X \cap Y$ is filtered.

(46) Let L be a relational structure and A be a subset of $2^{\text{the carrier of } L}$. Suppose that

(i) for every subset X of L such that $X \in A$ holds X is directed, and

(ii) for all subsets X, Y of L such that $X \in A$ and $Y \in A$ there exists a subset Z of L such that $Z \in A$ and $X \cup Y \subseteq Z$.

Let X be a subset of L. If $X = \bigcup A$, then X is directed.

(47) Let L be a relational structure and A be a subset of $2^{\text{the carrier of } L}$. Suppose that

(i) for every subset X of L such that $X \in A$ holds X is filtered, and

(ii) for all subsets X, Y of L such that $X \in A$ and $Y \in A$ there exists a subset Z of L such that $Z \in A$ and $X \cup Y \subseteq Z$.

Let X be a subset of L. If $X = \bigcup A$, then X is filtered.

Let L be a non empty reflexive transitive relational structure and let I be an ideal of L. We say that I is principal if and only if:

(Def. 21) There exists an element x of L such that $x \in I$ and $x \geq I$.

Let L be a non empty reflexive transitive relational structure and let F be a filter of L. We say that F is principal if and only if:

(Def. 22) There exists an element x of L such that $x \in F$ and $x \leq F$.

Next we state two propositions:

(48) Let L be a non empty reflexive transitive relational structure and I be an ideal of L. Then I is principal if and only if there exists an element x of L such that $I = {\downarrow}x$.

(49) Let L be a non empty reflexive transitive relational structure and F be a filter of L. Then F is principal if and only if there exists an element x of L such that $F = {\uparrow}x$.

Let L be a non empty reflexive transitive relational structure. The functor $\mathrm{Ids}(L)$ yields a set and is defined by:

(Def. 23) $\mathrm{Ids}(L) = \{X : X$ ranges over ideals of $L\}$.

The functor $\mathrm{Filt}(L)$ yields a set and is defined as follows:

(Def. 24) $\mathrm{Filt}(L) = \{X : X$ ranges over filters of $L\}$.

Let L be a non empty reflexive transitive relational structure. The functor $\mathrm{Ids}_0(L)$ yielding a set is defined by:

(Def. 25) $\mathrm{Ids}_0(L) = \mathrm{Ids}(L) \cup \{\emptyset\}$.

The functor $\mathrm{Filt}_0(L)$ yielding a set is defined as follows:

(Def. 26) $\mathrm{Filt}_0(L) = \mathrm{Filt}(L) \cup \{\emptyset\}$.

Let L be a non empty relational structure and let X be a subset of the carrier of L. The functor finsups(X) yielding a subset of L is defined as follows:

(Def. 27) $\mathrm{finsups}(X) = \{\bigsqcup_L Y : Y$ ranges over finite subsets of X, sup Y exists in $L\}$.

The functor fininfs(X) yielding a subset of L is defined as follows:

(Def. 28) $\mathrm{fininfs}(X) = \{\bigsqcap_L Y : Y$ ranges over finite subsets of X, inf Y exists in $L\}$.

Let L be a non empty antisymmetric lower-bounded relational structure and let X be a subset of the carrier of L. Note that finsups(X) is non empty.

Let L be a non empty antisymmetric upper-bounded relational structure and let X be a subset of the carrier of L. Note that fininfs(X) is non empty.

Let L be a non empty reflexive antisymmetric relational structure and let X be a non empty subset of the carrier of L. Note that finsups(X) is non empty and fininfs(X) is non empty.

One can prove the following two propositions:

(50) Let L be a non empty reflexive antisymmetric relational structure and X be a subset of the carrier of L. Then $X \subseteq \mathrm{finsups}(X)$ and $X \subseteq \mathrm{fininfs}(X)$.

(51) Let L be a non empty transitive relational structure and X, F be subsets of L. Suppose that
 (i) for every finite subset Y of X such that $Y \neq \emptyset$ holds sup Y exists in L,
 (ii) for every element x of L such that $x \in F$ there exists a finite subset Y of X such that sup Y exists in L and $x = \bigsqcup_L Y$, and
 (iii) for every finite subset Y of X such that $Y \neq \emptyset$ holds $\bigsqcup_L Y \in F$.
 Then F is directed.

Let L be a poset with l.u.b.'s and let X be a subset of the carrier of L. Note that finsups(X) is directed.

The following propositions are true:

(52) Let L be a non empty transitive reflexive relational structure and X, F be subsets of L. Suppose that
 (i) for every finite subset Y of X such that $Y \neq \emptyset$ holds sup Y exists in L,
 (ii) for every element x of L such that $x \in F$ there exists a finite subset Y of X such that sup Y exists in L and $x = \bigsqcup_L Y$, and
 (iii) for every finite subset Y of X such that $Y \neq \emptyset$ holds $\bigsqcup_L Y \in F$.
 Let x be an element of L. Then $x \geq X$ if and only if $x \geq F$.

(53) Let L be a non empty transitive reflexive relational structure and X, F be subsets of L. Suppose that

(i) for every finite subset Y of X such that $Y \neq \emptyset$ holds sup Y exists in L,
(ii) for every element x of L such that $x \in F$ there exists a finite subset Y of X such that sup Y exists in L and $x = \bigsqcup_L Y$, and
(iii) for every finite subset Y of X such that $Y \neq \emptyset$ holds $\bigsqcup_L Y \in F$.
Then sup X exists in L if and only if sup F exists in L.

(54) Let L be a non empty transitive reflexive relational structure and X, F be subsets of L. Suppose that
(i) for every finite subset Y of X such that $Y \neq \emptyset$ holds sup Y exists in L,
(ii) for every element x of L such that $x \in F$ there exists a finite subset Y of X such that sup Y exists in L and $x = \bigsqcup_L Y$,
(iii) for every finite subset Y of X such that $Y \neq \emptyset$ holds $\bigsqcup_L Y \in F$, and
(iv) sup X exists in L.
Then $\sup F = \sup X$.

(55) Let L be a poset with l.u.b.'s and X be a subset of L. If sup X exists in L or L is complete, then $\sup X = \sup \mathrm{finsups}(X)$.

(56) Let L be a non empty transitive relational structure and X, F be subsets of L. Suppose that
(i) for every finite subset Y of X such that $Y \neq \emptyset$ holds inf Y exists in L,
(ii) for every element x of L such that $x \in F$ there exists a finite subset Y of X such that inf Y exists in L and $x = \bigsqcap_L Y$, and
(iii) for every finite subset Y of X such that $Y \neq \emptyset$ holds $\bigsqcap_L Y \in F$.
Then F is filtered.

Let L be a poset with g.l.b.'s and let X be a subset of the carrier of L. One can check that $\mathrm{fininfs}(X)$ is filtered.

The following propositions are true:

(57) Let L be a non empty transitive reflexive relational structure and X, F be subsets of L. Suppose that
(i) for every finite subset Y of X such that $Y \neq \emptyset$ holds inf Y exists in L,
(ii) for every element x of L such that $x \in F$ there exists a finite subset Y of X such that inf Y exists in L and $x = \bigsqcap_L Y$, and
(iii) for every finite subset Y of X such that $Y \neq \emptyset$ holds $\bigsqcap_L Y \in F$.
Let x be an element of L. Then $x \leq X$ if and only if $x \leq F$.

(58) Let L be a non empty transitive reflexive relational structure and X, F be subsets of L. Suppose that
(i) for every finite subset Y of X such that $Y \neq \emptyset$ holds inf Y exists in L,
(ii) for every element x of L such that $x \in F$ there exists a finite subset Y of X such that inf Y exists in L and $x = \bigsqcap_L Y$, and
(iii) for every finite subset Y of X such that $Y \neq \emptyset$ holds $\bigsqcap_L Y \in F$.
Then inf X exists in L if and only if inf F exists in L.

(59) Let L be a non empty transitive reflexive relational structure and X, F be subsets of L. Suppose that
(i) for every finite subset Y of X such that $Y \neq \emptyset$ holds inf Y exists in L,

(ii) for every element x of L such that $x \in F$ there exists a finite subset Y of X such that $\inf Y$ exists in L and $x = \bigsqcap_L Y$,

(iii) for every finite subset Y of X such that $Y \neq \emptyset$ holds $\bigsqcap_L Y \in F$, and

(iv) $\inf X$ exists in L.

Then $\inf F = \inf X$.

(60) Let L be a poset with g.l.b.'s and X be a subset of L. If $\inf X$ exists in L or L is complete, then $\inf X = \inf \mathrm{fininfs}(X)$.

(61) Let L be a poset with l.u.b.'s and X be a subset of the carrier of L. Then $X \subseteq {\downarrow}\mathrm{finsups}(X)$ and for every ideal I of L such that $X \subseteq I$ holds ${\downarrow}\mathrm{finsups}(X) \subseteq I$.

(62) Let L be a poset with g.l.b.'s and X be a subset of the carrier of L. Then $X \subseteq {\uparrow}\mathrm{fininfs}(X)$ and for every filter F of L such that $X \subseteq F$ holds ${\uparrow}\mathrm{fininfs}(X) \subseteq F$.

5. Chains

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

(Def. 29) For all elements x, y of L holds $x \leq y$ or $y \leq x$.

Let us observe that every non empty reflexive relational structure which is trivial is also connected.

Let us observe that there exists a non empty poset which is connected.

A chain is a connected non empty poset.

Let L be a chain. Observe that $L^{\smile}$ is connected.

6. Semilattices

A semilattice is a poset with g.l.b.'s. A sup-semilattice is a poset with l.u.b.'s. A lattice is a poset with g.l.b.'s and l.u.b.'s.

The following two propositions are true:

(63) Let L be a semilattice and X be an upper non empty subset of L. Then X is a filter of L if and only if $\mathrm{sub}(X)$ is meet-inheriting.

(64) Let L be a sup-semilattice and X be a lower non empty subset of L. Then X is an ideal of L if and only if $\mathrm{sub}(X)$ is join-inheriting.

7. Maps

Let S, T be non empty relational structures, let f be a map from S into T, and let X be a subset of S. We say that f preserves inf of X if and only if:

(Def. 30) If $\inf X$ exists in S, then $\inf f^\circ X$ exists in T and $\inf(f^\circ X) = f(\inf X)$.

We say that f preserves sup of X if and only if:

(Def. 31) If $\sup X$ exists in S, then $\sup f^\circ X$ exists in T and $\sup(f^\circ X) = f(\sup X)$.

We now state the proposition

(65) Let S_1, S_2, T_1, T_2 be non empty relational structures. Suppose that
 (i) the relational structure of S_1 = the relational structure of T_1, and
 (ii) the relational structure of S_2 = the relational structure of T_2.

 Let f be a map from S_1 into S_2 and g be a map from T_1 into T_2. Suppose $f = g$. Let X be a subset of S_1 and Y be a subset of T_1 such that $X = Y$. Then
 (iii) if f preserves sup of X, then g preserves sup of Y, and
 (iv) if f preserves inf of X, then g preserves inf of Y.

Let L_1, L_2 be non empty relational structures and let f be a map from L_1 into L_2. We say that f is infs-preserving if and only if:

(Def. 32) For every subset X of L_1 holds f preserves inf of X.

We say that f is sups-preserving if and only if:

(Def. 33) For every subset X of L_1 holds f preserves sup of X.

We say that f is meet-preserving if and only if:

(Def. 34) For all elements x, y of L_1 holds f preserves inf of $\{x, y\}$.

We say that f is join-preserving if and only if:

(Def. 35) For all elements x, y of L_1 holds f preserves sup of $\{x, y\}$.

We say that f is filtered-infs-preserving if and only if:

(Def. 36) For every subset X of L_1 such that X is non empty filtered holds f preserves inf of X.

We say that f is directed-sups-preserving if and only if:

(Def. 37) For every subset X of L_1 such that X is non empty directed holds f preserves sup of X.

Let L_1, L_2 be non empty relational structures. Note that every map from L_1 into L_2 which is infs-preserving is also filtered-infs-preserving and meet-preserving and every map from L_1 into L_2 which is sups-preserving is also directed-sups-preserving and join-preserving.

Let S, T be relational structures and let f be a map from S into T. We say that f is isomorphic if and only if:

(Def. 38) (i) f is one-to-one monotone and there exists a map g from T into S such that $g = f^{-1}$ and g is monotone if S is non empty and T is non empty,

(ii) S is empty and T is empty, otherwise.

The following proposition is true

(66) Let S, T be non empty relational structures and f be a map from S into T. Then f is isomorphic if and only if the following conditions are satisfied:

(i) f is one-to-one,

(ii) $\operatorname{rng} f =$ the carrier of T, and

(iii) for all elements x, y of S holds $x \leq y$ iff $f(x) \leq f(y)$.

Let S, T be non empty relational structures. Note that every map from S into T which is isomorphic is also one-to-one and monotone.

We now state several propositions:

(67) Let S, T be non empty relational structures and f be a map from S into T. Suppose f is isomorphic. Then f^{-1} is a map from T into S and $\operatorname{rng}(f^{-1}) =$ the carrier of S.

(68) Let S, T be non empty relational structures and f be a map from S into T. Suppose f is isomorphic. Let g be a map from T into S. If $g = f^{-1}$, then g is isomorphic.

(69) Let S, T be non empty posets and f be a map from S into T. Suppose that for every filter X of S holds f preserves inf of X. Then f is monotone.

(70) Let S, T be non empty posets and f be a map from S into T. Suppose that for every filter X of S holds f preserves inf of X. Then f is filtered-infs-preserving.

(71) Let S be a semilattice, T be a non empty poset, and f be a map from S into T. Suppose that

(i) for every finite subset X of S holds f preserves inf of X, and

(ii) for every non empty filtered subset X of S holds f preserves inf of X.

Then f is infs-preserving.

(72) Let S, T be non empty posets and f be a map from S into T. Suppose that for every ideal X of S holds f preserves sup of X. Then f is monotone.

(73) Let S, T be non empty posets and f be a map from S into T. Suppose that for every ideal X of S holds f preserves sup of X. Then f is directed-sups-preserving.

(74) Let S be a sup-semilattice, T be a non empty poset, and f be a map from S into T. Suppose that

(i) for every finite subset X of S holds f preserves sup of X, and

(ii) for every non empty directed subset X of S holds f preserves sup of X.

Then f is sups-preserving.

8. Completeness wrt directed sets

Let L be a non empty reflexive relational structure. We say that L is up-complete if and only if the condition (Def. 39) is satisfied.

(Def. 39) Let X be a non empty directed subset of L. Then there exists an element x of L such that $x \geq X$ and for every element y of L such that $y \geq X$ holds $x \leq y$.

One can verify that every reflexive relational structure with l.u.b.'s which is up-complete is also upper-bounded.

The following proposition is true

(75) Let L be a non empty reflexive antisymmetric relational structure. Then L is up-complete if and only if for every non empty directed subset X of L holds sup X exists in L.

Let L be a non empty reflexive relational structure. We say that L is inf-complete if and only if the condition (Def. 40) is satisfied.

(Def. 40) Let X be a non empty subset of L. Then there exists an element x of L such that $x \leq X$ and for every element y of L such that $y \leq X$ holds $x \geq y$.

Next we state the proposition

(76) Let L be a non empty reflexive antisymmetric relational structure. Then L is inf-complete if and only if for every non empty subset X of L holds inf X exists in L.

One can check the following observations:

* every non empty reflexive relational structure which is complete is also up-complete and inf-complete,
* every non empty reflexive relational structure which is inf-complete is also lower-bounded, and
* every non empty poset which is up-complete and lower-bounded and has l.u.b.'s is also complete.

Let us note that every non empty reflexive antisymmetric relational structure which is inf-complete has g.l.b.'s.

Let us note that every non empty reflexive antisymmetric upper-bounded relational structure which is inf-complete has l.u.b.'s.

One can check that there exists a lattice which is complete and strict.

References

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

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

[3] Grzegorz Bancerek. Filters - Part II. Quotient lattices modulo filters and direct product of two lattices. *Formalized Mathematics*, 2(**3**):433–438, 1991.

[4] Józef Białas. Group and field definitions. *Formalized Mathematics*, 1(**3**):433–439, 1990.

[5] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[6] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[7] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[8] 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.
[9] Adam Grabowski. On the category of posets. *Formalized Mathematics*, 5(**4**):501–505, 1996.
[10] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[11] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[12] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[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 and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[16] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[17] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.
[18] Edmund Woronowicz and Anna Zalewska. Properties of binary relations. *Formalized Mathematics*, 1(**1**):85–89, 1990.
[19] Stanisław Żukowski. Introduction to lattice theory. *Formalized Mathematics*, 1(**1**):215–222, 1990.

Received September 12, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Fixpoints in Complete Lattices [1]

Piotr Rudnicki
University of Alberta
Edmonton

Andrzej Trybulec
Warsaw University
Białystok

Summary. Theorem (5) states that if an iterate of a function has a unique fixpoint then it is also the fixpoint of the function. It has been included here in response to P. Andrews claim that such a proof in set theory takes thousands of lines when one starts with the axioms. While probably true, such a claim is misleading about the usefulness of proof-checking systems based on set theory.

Next, we prove the existence of the least and the greatest fixpoints for $\subseteq$-monotone functions from a powerset to a powerset of a set. Scheme *Knaster* is the Knaster theorem about the existence of fixpoints, cf. [14]. Theorem (11) is the Banach decomposition theorem which is then used to prove the Schröder-Bernstein theorem (12) (we followed Paulson's development of these theorems in Isabelle [16]). It is interesting to note that the last theorem when stated in Mizar in terms of cardinals becomes trivial to prove as in the Mizar development of cardinals the $\leq$ relation is synonymous with $\subseteq$.

Section 3 introduces the notion of the lattice of a lattice subset provided the subset has lubs and glbs.

The main theorem of Section 4 is the Tarski theorem (43) that every monotone function f over a complete lattice L has a complete lattice of fixpoints. As the consequence of this theorem we get the existence of the least fixpoint equal to $f^{\beta}(\bot_L)$ for some ordinal β with cardinality not bigger than the cardinality of the carrier of L, cf. [14], and analogously the existence of the greatest fixpoint equal to $f^{\beta}(\top_L)$.

Section 5 connects the fixpoint properties of monotone functions over complete lattices with the fixpoints of $\subseteq$-monotone functions over the lattice of subsets of a set (Boolean lattice).

MML Identifier: KNASTER.

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

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

ISSN 1426–2630

1. Preliminaries

In this paper f, g, h will be functions.

The following three propositions are true:

(1) If f is one-to-one and g is one-to-one and $\operatorname{rng} f$ misses $\operatorname{rng} g$, then $f {+\cdot} g$ is one-to-one.

(2) If $\operatorname{dom} f$ misses $\operatorname{dom} g$, then $f \cup g$ is a function.

(3) Suppose $h = f \cup g$ and $\operatorname{dom} f$ misses $\operatorname{dom} g$. Then h is one-to-one if and only if the following conditions are satisfied:

(i) f is one-to-one,

(ii) g is one-to-one, and

(iii) $\operatorname{rng} f$ misses $\operatorname{rng} g$.

2. Fixpoints in general

Let x be a set and let f be a function. We say that x is a fixpoint of f if and only if:

(Def. 1) $x \in \operatorname{dom} f$ and $x = f(x)$.

Let A be a non empty set, let a be an element of A, and let f be a function from A into A. Let us observe that a is a fixpoint of f if and only if:

(Def. 2) $a = f(a)$.

For simplicity we follow a convention: x, y, X will be sets, A will be a non empty set, n will be a natural number, and f will be a function from X into X.

Next we state two propositions:

(4) If x is a fixpoint of f^n, then $f(x)$ is a fixpoint of f^n.

(5) If there exists n such that x is a fixpoint of f^n and for every y such that y is a fixpoint of f^n holds $x = y$, then x is a fixpoint of f.

Let A, B be non empty sets and let f be a function from A into B. Let us observe that f is $\subseteq$-monotone if and only if:

(Def. 3) For all elements x, y of A such that $x \subseteq y$ holds $f(x) \subseteq f(y)$.

Let A be a set and let B be a non empty set. Observe that there exists a function from A into B which is $\subseteq$-monotone.

Let X be a set and let f be a $\subseteq$-monotone function from 2^X into 2^X. The functor $\operatorname{lfp}(X, f)$ yields a subset of X and is defined by:

(Def. 4) $\operatorname{lfp}(X, f) = \bigcap\{h : h \text{ ranges over subsets of } X,\ f(h) \subseteq h\}$.

The functor $\operatorname{gfp}(X, f)$ yielding a subset of X is defined by:

(Def. 5) $\operatorname{gfp}(X, f) = \bigcup\{h : h \text{ ranges over subsets of } X,\ h \subseteq f(h)\}$.

In the sequel f will be a $\subseteq$-monotone function from 2^X into 2^X and S will be a subset of X.

One can prove the following propositions:

(6) $\operatorname{lfp}(X, f)$ is a fixpoint of f.

(7) $\operatorname{gfp}(X, f)$ is a fixpoint of f.

(8) If $f(S) \subseteq S$, then $\operatorname{lfp}(X, f) \subseteq S$.

(9) If $S \subseteq f(S)$, then $S \subseteq \operatorname{gfp}(X, f)$.

(10) If S is a fixpoint of f, then $\operatorname{lfp}(X, f) \subseteq S$ and $S \subseteq \operatorname{gfp}(X, f)$.

The scheme *Knaster* deals with a set $\mathcal{A}$ and a unary functor $\mathcal{F}$ yielding a set, and states that:

There exists a set D such that $\mathcal{F}(D) = D$ and $D \subseteq \mathcal{A}$

provided the parameters meet the following requirements:

- For all sets X, Y such that $X \subseteq Y$ holds $\mathcal{F}(X) \subseteq \mathcal{F}(Y)$,
- $\mathcal{F}(\mathcal{A}) \subseteq \mathcal{A}$.

In the sequel X, Y are non empty sets, f is a function from X into Y, and g is a function from Y into X.

We now state several propositions:

(11) There exist sets X_1, X_2, Y_1, Y_2 such that X_1 misses X_2 and Y_1 misses Y_2 and $X_1 \cup X_2 = X$ and $Y_1 \cup Y_2 = Y$ and $f^{\circ}X_1 = Y_1$ and $g^{\circ}Y_2 = X_2$.

(12) If f is one-to-one and g is one-to-one, then there exists function from X into Y which is bijective.

(13) If there exists f which is bijective, then $X \approx Y$.

(14) If f is one-to-one and g is one-to-one, then $X \approx Y$.

(15) For all cardinal numbers N, M such that $N \leq M$ and $M \leq N$ holds $N = M$.

3. The lattice of lattice subset

Let L be a non empty lattice structure, let f be a unary operation on L, and let x be an element of L. Then $f(x)$ is an element of L.

Let L be a lattice, let f be a function from the carrier of L into the carrier of L, let x be an element of the carrier of L, and let O be an ordinal number. The functor $f_{\sqcup}^{O}(x)$ is defined by the condition (Def. 6).

(Def. 6) There exists a transfinite sequence L_0 such that

(i) $f_{\sqcup}^{O}(x) = \operatorname{last} L_0$,

(ii) $\operatorname{dom} L_0 = \operatorname{succ} O$,

(iii) $L_0(\emptyset) = x$,

(iv) for every ordinal number C and for arbitrary y such that $\operatorname{succ} C \in \operatorname{succ} O$ and $y = L_0(C)$ holds $L_0(\operatorname{succ} C) = f(y)$, and

(v) for every ordinal number C and for every transfinite sequence L_1 such that $C \in \operatorname{succ} O$ and $C \neq \emptyset$ and C is a limit ordinal number and $L_1 = L_0 \upharpoonright C$ holds $L_0(C) = \bigsqcup_L \operatorname{rng} L_1$.

The functor $f_{\sqcap}^{O}(x)$ is defined by the condition (Def. 7).

(Def. 7) There exists a transfinite sequence L_0 such that
(i) $f_{\sqcap}^{O}(x) = \mathrm{last}\,L_0$,
(ii) $\mathrm{dom}\,L_0 = \mathrm{succ}\,O$,
(iii) $L_0(\emptyset) = x$,
(iv) for every ordinal number C and for arbitrary y such that $\mathrm{succ}\,C \in \mathrm{succ}\,O$ and $y = L_0(C)$ holds $L_0(\mathrm{succ}\,C) = f(y)$, and
(v) for every ordinal number C and for every transfinite sequence L_1 such that $C \in \mathrm{succ}\,O$ and $C \neq \emptyset$ and C is a limit ordinal number and $L_1 = L_0 \restriction C$ holds $L_0(C) = \bigsqcap_L \mathrm{rng}\,L_1$.

For simplicity we adopt the following rules: L will denote a lattice, f will denote a function from the carrier of L into the carrier of L, x will denote an element of the carrier of L, O, O_1, O_2 will denote ordinal numbers, and T will denote a transfinite sequence.

One can prove the following propositions:

(16) $f_{\sqcup}^{\emptyset}(x) = x$.

(17) $f_{\sqcap}^{\emptyset}(x) = x$.

(18) $f_{\sqcup}^{\mathrm{succ}\,O}(x) = f(f_{\sqcup}^{O}(x))$.

(19) $f_{\sqcap}^{\mathrm{succ}\,O}(x) = f(f_{\sqcap}^{O}(x))$.

(20) Suppose $O \neq \emptyset$ and O is a limit ordinal number and $\mathrm{dom}\,T = O$ and for every ordinal number A such that $A \in O$ holds $T(A) = f_{\sqcup}^{A}(x)$. Then $f_{\sqcup}^{O}(x) = \bigsqcup_L \mathrm{rng}\,T$.

(21) Suppose $O \neq \emptyset$ and O is a limit ordinal number and $\mathrm{dom}\,T = O$ and for every ordinal number A such that $A \in O$ holds $T(A) = f_{\sqcap}^{A}(x)$. Then $f_{\sqcap}^{O}(x) = \bigsqcap_L \mathrm{rng}\,T$.

(22) $f^{n}(x) = f_{\sqcup}^{n}(x)$.

(23) $f^{n}(x) = f_{\sqcap}^{n}(x)$.

Let L be a lattice, let f be a unary operation on the carrier of L, let a be an element of the carrier of L, and let O be an ordinal number. Then $f_{\sqcup}^{O}(a)$ is an element of L.

Let L be a lattice, let f be a unary operation on the carrier of L, let a be an element of the carrier of L, and let O be an ordinal number. Then $f_{\sqcap}^{O}(a)$ is an element of L.

Let L be a non empty lattice structure and let P be a subset of L. We say that P has l.u.b.'s if and only if the condition (Def. 8) is satisfied.

(Def. 8) Let x, y be elements of L. Suppose $x \in P$ and $y \in P$. Then there exists an element z of L such that $z \in P$ and $x \sqsubseteq z$ and $y \sqsubseteq z$ and for every element z' of L such that $z' \in P$ and $x \sqsubseteq z'$ and $y \sqsubseteq z'$ holds $z \sqsubseteq z'$.

We say that P has g.l.b.'s if and only if the condition (Def. 9) is satisfied.

(Def. 9) Let x, y be elements of L. Suppose $x \in P$ and $y \in P$. Then there exists an element z of L such that $z \in P$ and $z \sqsubseteq x$ and $z \sqsubseteq y$ and for every element z' of L such that $z' \in P$ and $z' \sqsubseteq x$ and $z' \sqsubseteq y$ holds $z' \sqsubseteq z$.

Let L be a lattice. One can verify that there exists a subset of L which is non empty and has l.u.b.'s and g.l.b.'s.

Let L be a lattice and let P be a non empty subset of L with l.u.b.'s and g.l.b.'s. The functor $\mathbb{L}_P$ yields a strict lattice and is defined by the conditions (Def. 10).

(Def. 10) (i) The carrier of $\mathbb{L}_P = P$, and

(ii) for all elements x, y of $\mathbb{L}_P$ there exist elements x', y' of L such that $x = x'$ and $y = y'$ and $x \sqsubseteq y$ iff $x' \sqsubseteq y'$.

4. Complete lattices

Let us mention that every lattice which is complete is also bounded.

In the sequel L will be a complete lattice, f will be a monotone unary operation on L, and a, b will be elements of L.

The following propositions are true:

(24) There exists a which is a fixpoint of f.

(25) For every a such that $a \sqsubseteq f(a)$ and for every O holds $a \sqsubseteq f_{\sqcup}^{O}(a)$.

(26) For every a such that $f(a) \sqsubseteq a$ and for every O holds $f_{\sqcap}^{O}(a) \sqsubseteq a$.

(27) For every a such that $a \sqsubseteq f(a)$ and for all O_1, O_2 such that $O_1 \subseteq O_2$ holds $f_{\sqcup}^{O_1}(a) \sqsubseteq f_{\sqcup}^{O_2}(a)$.

(28) For every a such that $f(a) \sqsubseteq a$ and for all O_1, O_2 such that $O_1 \subseteq O_2$ holds $f_{\sqcap}^{O_2}(a) \sqsubseteq f_{\sqcap}^{O_1}(a)$.

(29) For every a such that $a \sqsubseteq f(a)$ and for all O_1, O_2 such that $O_1 \subseteq O_2$ and $O_1 \neq O_2$ and $f_{\sqcup}^{O_2}(a)$ is not a fixpoint of f holds $f_{\sqcup}^{O_1}(a) \neq f_{\sqcup}^{O_2}(a)$.

(30) For every a such that $f(a) \sqsubseteq a$ and for all O_1, O_2 such that $O_1 \subseteq O_2$ and $O_1 \neq O_2$ and $f_{\sqcap}^{O_2}(a)$ is not a fixpoint of f holds $f_{\sqcap}^{O_1}(a) \neq f_{\sqcap}^{O_2}(a)$.

(31) If $a \sqsubseteq f(a)$ and $f_{\sqcup}^{O_1}(a)$ is a fixpoint of f, then for every O_2 such that $O_1 \subseteq O_2$ holds $f_{\sqcup}^{O_1}(a) = f_{\sqcup}^{O_2}(a)$.

(32) If $f(a) \sqsubseteq a$ and $f_{\sqcap}^{O_1}(a)$ is a fixpoint of f, then for every O_2 such that $O_1 \subseteq O_2$ holds $f_{\sqcap}^{O_1}(a) = f_{\sqcap}^{O_2}(a)$.

(33) For every a such that $a \sqsubseteq f(a)$ there exists O such that $\overline{\overline{O}} \leq \overline{\overline{\text{the carrier of } L}}$ and $f_{\sqcup}^{O}(a)$ is a fixpoint of f.

(34) For every a such that $f(a) \sqsubseteq a$ there exists O such that $\overline{\overline{O}} \leq \overline{\overline{\text{the carrier of } L}}$ and $f_{\sqcap}^{O}(a)$ is a fixpoint of f.

(35) Given a, b. Suppose a is a fixpoint of f and b is a fixpoint of f. Then there exists O such that $\overline{\overline{O}} \leq \overline{\overline{\text{the carrier of } L}}$ and $f_{\sqcup}^{O}(a \sqcup b)$ is a fixpoint of f and $a \sqsubseteq f_{\sqcup}^{O}(a \sqcup b)$ and $b \sqsubseteq f_{\sqcup}^{O}(a \sqcup b)$.

(36) Given a, b. Suppose a is a fixpoint of f and b is a fixpoint of f. Then there exists O such that $\overline{\overline{O}} \leq \overline{\overline{\text{the carrier of } L}}$ and $f_{\sqcap}^{O}(a \sqcap b)$ is a fixpoint of f and $f_{\sqcap}^{O}(a \sqcap b) \sqsubseteq a$ and $f_{\sqcap}^{O}(a \sqcap b) \sqsubseteq b$.

(37) If $a \sqsubseteq f(a)$ and $a \sqsubseteq b$ and b is a fixpoint of f, then for every O_2 holds $f_{\sqcup}^{O_2}(a) \sqsubseteq b$.

(38) If $f(a) \sqsubseteq a$ and $b \sqsubseteq a$ and b is a fixpoint of f, then for every O_2 holds $b \sqsubseteq f_{\sqcap}^{O_2}(a)$.

Let L be a complete lattice and let f be a unary operation on L. Let us assume that f is monotone. The functor FixPoints(f) yielding a strict lattice is defined by:

(Def. 11) There exists a non empty subset P of L with l.u.b.'s and g.l.b.'s such that $P = \{x : x$ ranges over elements of L, x is a fixpoint of $f\}$ and FixPoints$(f) = \mathbb{L}_P$.

One can prove the following propositions:

(39) The carrier of FixPoints$(f) = \{x : x$ ranges over elements of L, x is a fixpoint of $f\}$.

(40) The carrier of FixPoints$(f) \subseteq$ the carrier of L.

(41) $a \in$ the carrier of FixPoints(f) iff a is a fixpoint of f.

(42) For all elements x, y of FixPoints(f) and for all a, b such that $x = a$ and $y = b$ holds $x \sqsubseteq y$ iff $a \sqsubseteq b$.

(43) FixPoints(f) is complete.

Let us consider L, f. The functor lfp(f) yields an element of L and is defined as follows:

(Def. 12) $\mathrm{lfp}(f) = f_{\sqcup}^{(\text{the carrier of } L)^+}(\perp_L)$.

The functor gfp(f) yielding an element of L is defined as follows:

(Def. 13) $\mathrm{gfp}(f) = f_{\sqcap}^{(\text{the carrier of } L)^+}(\top_L)$.

Next we state several propositions:

(44) lfp(f) is a fixpoint of f and there exists O such that $\overline{\overline{O}} \leq \overline{\overline{\text{the carrier of } L}}$ and $f_{\sqcup}^{O}(\perp_L) = \mathrm{lfp}(f)$.

(45) gfp(f) is a fixpoint of f and there exists O such that $\overline{\overline{O}} \leq \overline{\overline{\text{the carrier of } L}}$ and $f_{\sqcap}^{O}(\top_L) = \mathrm{gfp}(f)$.

(46) If a is a fixpoint of f, then $\mathrm{lfp}(f) \sqsubseteq a$ and $a \sqsubseteq \mathrm{gfp}(f)$.

(47) If $f(a) \sqsubseteq a$, then $\mathrm{lfp}(f) \sqsubseteq a$.

(48) If $a \sqsubseteq f(a)$, then $a \sqsubseteq \mathrm{gfp}(f)$.

5. Boolean lattices

In the sequel f is a monotone unary operation on the lattice of subsets of A.

Let A be a set. One can verify that the lattice of subsets of A is complete.

One can prove the following propositions:

(49) Let f be a unary operation on the lattice of subsets of A. Then f is monotone if and only if f is $\subseteq$-monotone.

(50) There exists a $\subseteq$-monotone function g from 2^A into 2^A such that $\mathrm{lfp}(A, g) = \mathrm{lfp}(f)$.

(51) There exists a $\subseteq$-monotone function g from 2^A into 2^A such that $\mathrm{gfp}(A, g) = \mathrm{gfp}(f)$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.
[2] Grzegorz Bancerek. Complete lattices. *Formalized Mathematics*, 2(**5**):719–725, 1991.
[3] Grzegorz Bancerek. Continuous, stable, and linear maps of coherence spaces. *Formalized Mathematics*, 5(**3**):381–393, 1996.
[4] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[5] Grzegorz Bancerek. The ordinal numbers. *Formalized Mathematics*, 1(**1**):91–96, 1990.
[6] Grzegorz Bancerek. Quantales. *Formalized Mathematics*, 5(**1**):85–91, 1996.
[7] Grzegorz Bancerek. Sequences of ordinal numbers. *Formalized Mathematics*, 1(**2**):281–290, 1990.
[8] Grzegorz Bancerek. The well ordering relations. *Formalized Mathematics*, 1(**1**):123–129, 1990.
[9] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(**1**):175–180, 1990.
[10] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[11] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[12] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[13] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[14] J.-L. Lassez, V. L. Nguyen, and E. A Sonenberg. Fixed point theorems and semantics: a folk tale. *Information Processing Letters*, 14(3):112–116, 1982.
[15] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[16] Lawrence C. Paulson. Set theory for verification: II, induction and recursion. *Journal of Automated Reasoning*, 15(2):167–215, 1995.
[17] Yozo Toda. The formalization of simple graphs. *Formalized Mathematics*, 5(**1**):137–144, 1996.
[18] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[19] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[20] Wojciech A. Trybulec. Partially ordered sets. *Formalized Mathematics*, 1(**2**):313–319, 1990.
[21] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[22] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[23] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.
[24] Edmund Woronowicz and Anna Zalewska. Properties of binary relations. *Formalized Mathematics*, 1(**1**):85–89, 1990.
[25] Stanisław Żukowski. Introduction to lattice theory. *Formalized Mathematics*, 1(**1**):215–222, 1990.

Received September 16, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Boolean Posets, Posets under Inclusion and Products of Relational Structures [1]

Adam Grabowski
Warsaw University
Białystok

Robert Milewski
Warsaw University
Białystok

Summary. In the paper some notions useful in formalization of [11] are introduced, e.g. the definition of the poset of subsets of a set with inclusion as an ordering relation. Using the theory of many sorted sets authors formulate the definition of product of relational structures.

MML Identifier: YELLOW_1.

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

1. Boolean Posets and Posets under Inclusion

In this paper X will be a set.

Let L be a lattice. Observe that Poset(L) has l.u.b.'s and g.l.b.'s.

Let L be an upper-bounded lattice. Note that Poset(L) is upper-bounded.

Let L be a lower-bounded lattice. One can check that Poset(L) is lower-bounded.

Let L be a complete lattice. One can verify that Poset(L) is complete.

Let X be a set. Then $\subseteq_X$ is an order in X.

Let X be a set. The functor $\langle X, \subseteq \rangle$ yielding a strict relational structure is defined as follows:

(Def. 1) $\langle X, \subseteq \rangle = \langle X, \subseteq_X \rangle$.

Let X be a set. Observe that $\langle X, \subseteq \rangle$ is reflexive antisymmetric and transitive.

Let X be a non empty set. Observe that $\langle X, \subseteq \rangle$ is non empty.

We now state the proposition

[1]This work was partially supported by Office of Naval Research Grant N00014-95-1-1336.

ISSN 1426–2630

(1) The carrier of $\langle X, \subseteq \rangle = X$ and the internal relation of $\langle X, \subseteq \rangle = {\subseteq}_X$.

Let X be a set. The functor $2^X_\subseteq$ yielding a strict relational structure is defined by:

(Def. 2) $2^X_\subseteq =$ Poset(the lattice of subsets of X).

Let X be a set. Note that $2^X_\subseteq$ is non empty reflexive antisymmetric and transitive.

Let X be a set. Note that $2^X_\subseteq$ is complete.

Next we state a number of propositions:

(2) For all elements x, y of $2^X_\subseteq$ holds $x \leq y$ iff $x \subseteq y$.

(3) For every non empty set X and for all elements x, y of $\langle X, \subseteq \rangle$ holds $x \leq y$ iff $x \subseteq y$.

(4) $2^X_\subseteq = \langle 2^X, \subseteq \rangle$.

(5) For every subset Y of 2^X holds $\langle Y, \subseteq \rangle$ is a full relational substructure of $2^X_\subseteq$.

(6) For every non empty set X such that $\langle X, \subseteq \rangle$ has l.u.b.'s and for all elements x, y of $\langle X, \subseteq \rangle$ holds $x \cup y \subseteq x \sqcup y$.

(7) For every non empty set X such that $\langle X, \subseteq \rangle$ has g.l.b.'s and for all elements x, y of $\langle X, \subseteq \rangle$ holds $x \sqcap y \subseteq x \cap y$.

(8) For every non empty set X and for all elements x, y of $\langle X, \subseteq \rangle$ such that $x \cup y \in X$ holds $x \sqcup y = x \cup y$.

(9) For every non empty set X and for all elements x, y of $\langle X, \subseteq \rangle$ such that $x \cap y \in X$ holds $x \sqcap y = x \cap y$.

(10) Let L be a relational structure. Suppose that for all elements x, y of L holds $x \leq y$ iff $x \subseteq y$. Then the internal relation of $L = {\subseteq}_{\text{the carrier of } L}$.

(11) For every non empty set X such that for all sets x, y such that $x \in X$ and $y \in X$ holds $x \cup y \in X$ holds $\langle X, \subseteq \rangle$ has l.u.b.'s.

(12) For every non empty set X such that for all sets x, y such that $x \in X$ and $y \in X$ holds $x \cap y \in X$ holds $\langle X, \subseteq \rangle$ has g.l.b.'s.

(13) For every non empty set X such that $\emptyset \in X$ holds $\perp_{\langle X, \subseteq \rangle} = \emptyset$.

(14) For every non empty set X such that $\bigcup X \in X$ holds $\top_{\langle X, \subseteq \rangle} = \bigcup X$.

(15) For every non empty set X such that $\langle X, \subseteq \rangle$ is upper-bounded holds $\bigcup X \in X$.

(16) For every non empty set X such that $\langle X, \subseteq \rangle$ is lower-bounded holds $\bigcap X \in X$.

(17) For all elements x, y of $2^X_\subseteq$ holds $x \sqcup y = x \cup y$ and $x \sqcap y = x \cap y$.

(18) $\perp_{2^X_\subseteq} = \emptyset$.

(19) $\top_{2^X_\subseteq} = X$.

(20) For every non empty subset Y of $2^X_\subseteq$ holds $\inf Y = \bigcap Y$.

(21) For every subset Y of $2^X_\subseteq$ holds $\sup Y = \bigcup Y$.

(22) For every non empty topological space T and for every subset X of $\langle$the topology of T, $\subseteq\rangle$ holds $\sup X = \bigcup X$.

(23) For every non empty topological space T holds $\perp_{\langle \text{the topology of } T,\, \subseteq\rangle} = \emptyset$.

(24) For every non empty topological space T holds $\top_{\langle \text{the topology of } T,\, \subseteq\rangle} =$ the carrier of T.

Let T be a non empty topological space. Observe that $\langle$the topology of T, $\subseteq\rangle$ is complete and non trivial.

We now state the proposition

(25) Let T be a topological space and let F be a family of subsets of T. Then F is open if and only if F is a subset of $\langle$the topology of T, $\subseteq\rangle$.

2. Products of Relational Structures

Let R be a binary relation. We say that R is relational structure yielding if and only if:

(Def. 3) For every set v such that $v \in \operatorname{rng} R$ holds v is a relational structure.

One can check that every function which is relational structure 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 relational structure yielding.

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

Let I be a set and let J be a relational structure yielding many sorted set indexed by I. The functor $\prod J$ yields a strict relational structure and is defined by the conditions (Def. 4).

(Def. 4) (i) The carrier of $\prod J = \prod \operatorname{support} J$, and

(ii) for all elements x, y of the carrier of $\prod J$ such that $x \in \prod \operatorname{support} J$ holds $x \leq y$ iff there exist functions f, g such that $f = x$ and $g = y$ and for every set i such that $i \in I$ there exists a relational structure R and there exist elements x_1, y_1 of R such that $R = J(i)$ and $x_1 = f(i)$ and $y_1 = g(i)$ and $x_1 \leq y_1$.

Let X be a set and let L be a relational structure. One can verify that $X \longmapsto L$ is relational structure yielding.

Let I be a set and let T be a relational structure. The functor T^I yielding a strict relational structure is defined by:

(Def. 5) $T^I = \prod (I \longmapsto T)$.

Next we state three propositions:

(26) For every relational structure yielding many sorted set J indexed by $\emptyset$ holds $\prod J = \langle \{\emptyset\}, \triangle_{\{\emptyset\}} \rangle$.

(27) For every relational structure Y holds $Y^{\emptyset} = \langle \{\emptyset\}, \triangle_{\{\emptyset\}} \rangle$.

(28) For every set X and for every relational structure Y holds (the carrier of $Y)^X =$ the carrier of Y^X.

Let X be a set and let Y be a non empty relational structure. Note that Y^X is non empty.

Let X be a set and let Y be a reflexive non empty relational structure. Observe that Y^X is reflexive.

Let Y be a non empty relational structure. Observe that $Y^{\emptyset}$ is trivial.

Let Y be a non empty reflexive relational structure. Note that $Y^{\emptyset}$ is antisymmetric and has g.l.b.'s and l.u.b.'s.

Let X be a set and let Y be a transitive non empty relational structure. Note that Y^X is transitive.

Let X be a set and let Y be an antisymmetric non empty relational structure. Note that Y^X is antisymmetric.

Let X be a non empty set and let Y be a non empty antisymmetric relational structure with g.l.b.'s. Observe that Y^X has g.l.b.'s.

Let X be a non empty set and let Y be a non empty antisymmetric relational structure with l.u.b.'s. Observe that Y^X has l.u.b.'s.

Let S, T be relational structures. The functor MonMaps(S,T) yielding a strict full relational substructure of $T^{\text{the carrier of } S}$ is defined by the condition (Def. 6).

(Def. 6) Let f be a map from S into T. Then $f \in$ the carrier of MonMaps(S,T) if and only if $f \in$ (the carrier of $T)^{\text{the carrier of } S}$ and f is monotone.

Acknowledgments

The authors would like to express their gratitude to Professor Andrzej Trybulec for his help in formulating mizared definition of the product.

References

[1] Grzegorz Bancerek. Bounds in posets and relational substructures. *Formalized Mathematics*, 6(**1**):81–91, 1997.
[2] Grzegorz Bancerek. Complete lattices. *Formalized Mathematics*, 2(**5**):719–725, 1991.
[3] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[4] Grzegorz Bancerek. Zermelo theorem and axiom of choice. *Formalized Mathematics*, 1(**2**):265–267, 1990.
[5] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[6] Józef Białas. Group and field definitions. *Formalized Mathematics*, 1(**3**):433–439, 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. Families of subsets, subspaces and mappings in topological spaces. *Formalized Mathematics*, 1(**2**):257–261, 1990.
[11] 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.

[12] Adam Grabowski. On the category of posets. *Formalized Mathematics*, 5(**4**):501–505, 1996.
[13] Krzysztof Hryniewiecki. Relations of tolerance. *Formalized Mathematics*, 2(**1**):105–109, 1991.
[14] Beata Madras. Product of family of universal algebras. *Formalized Mathematics*, 4(**1**):103–108, 1993.
[15] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, I. *Formalized Mathematics*, 5(**2**):167–172, 1996.
[16] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[17] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[18] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[19] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[20] Wojciech A. Trybulec. Partially ordered sets. *Formalized Mathematics*, 1(**2**):313–319, 1990.
[21] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[22] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[23] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.
[24] Edmund Woronowicz and Anna Zalewska. Properties of binary relations. *Formalized Mathematics*, 1(**1**):85–89, 1990.
[25] Stanisław Żukowski. Introduction to lattice theory. *Formalized Mathematics*, 1(**1**):215–222, 1990.

Received September 20, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Properties of Relational Structures, Posets, Lattices and Maps [1]

Mariusz Żynel
Warsaw University
Białystok

Czesław Byliński
Warsaw University
Białystok

Summary. In the paper we present some auxiliary facts concerning posets and maps between them. Our main purpose, however is to give an account on complete lattices and lattices of ideals. A sufficient condition that a lattice might be complete, the fixed-point theorem and two remarks upon images of complete lattices in monotone maps, introduced in [10, pp. 8–9], can be found in Section 7. Section 8 deals with lattices of ideals. We examine the meet and join of two ideals. In order to show that the lattice of ideals is complete, the infinite intersection of ideals is investigated.

MML Identifier: YELLOW_2.

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

1. Basic Facts

In this paper x will be arbitrary and X, Y will denote sets.

The scheme *RelStrSubset* deals with a non empty relational structure $\mathcal{A}$ and a unary predicate $\mathcal{P}$, and states that:

$\{x : x$ ranges over elements of $\mathcal{A}$, $\mathcal{P}[x]\}$ is a subset of $\mathcal{A}$

for all values of the parameters.

Let S be a non empty 1-sorted structure and let X be a non empty subset of the carrier of S. We see that the element of X is an element of S.

One can prove the following four propositions:

[1]This work was partially supported by Office of Naval Research Grant N00014-95-1-1336.

ISSN 1426–2630

(1) Let L be a non empty relational structure, and let x be an element of L, and let X be a subset of L. Then $X \subseteq {\downarrow}x$ if and only if $X \leq x$.

(2) Let L be a non empty relational structure, and let x be an element of L, and let X be a subset of L. Then $X \subseteq {\uparrow}x$ if and only if $x \leq X$.

(3) Let L be an antisymmetric transitive relational structure with l.u.b.'s and let X, Y be sets. Suppose sup X exists in L and sup Y exists in L. Then sup $X \cup Y$ exists in L and $\bigsqcup_L(X \cup Y) = \bigsqcup_L X \sqcup \bigsqcup_L Y$.

(4) Let L be an antisymmetric transitive relational structure with g.l.b.'s and let X, Y be sets. Suppose inf X exists in L and inf Y exists in L. Then inf $X \cup Y$ exists in L and $\bigsqcap_L(X \cup Y) = \bigsqcap_L X \sqcap \bigsqcap_L Y$.

2. Relational Substructures

The following propositions are true:

(5) For every binary relation R and for all sets X, Y such that $X \subseteq Y$ holds $R\,|^2\, X \subseteq R\,|^2\, Y$.

(6) Let L be a relational structure and let S, T be full relational substructures of L. Suppose the carrier of $S \subseteq$ the carrier of T. Then the internal relation of $S \subseteq$ the internal relation of T.

(7) Let L be a non empty relational structure and let S be a non empty relational substructure of L. Then

(i) if X is a directed subset of S, then X is a directed subset of L, and

(ii) if X is a filtered subset of S, then X is a filtered subset of L.

(8) Let L be a non empty relational structure and let S, T be non empty full relational substructures of L. Suppose the carrier of $S \subseteq$ the carrier of T. Let X be a subset of S. Then

(i) X is a subset of T, and

(ii) for every subset Y of T such that $X = Y$ holds if X is filtered, then Y is filtered and if X is directed, then Y is directed.

3. Maps

Now we present three schemes. The scheme *LambdaMD* deals with non empty relational structures $\mathcal{A}$, $\mathcal{B}$ and a unary functor $\mathcal{F}$ yielding an element of $\mathcal{B}$, and states that:

> There exists a map f from $\mathcal{A}$ into $\mathcal{B}$ such that for every element x of $\mathcal{A}$ holds $f(x) = \mathcal{F}(x)$

for all values of the parameters.

The scheme *KappaMD* concerns non empty relational structures $\mathcal{A}$, $\mathcal{B}$ and a unary functor $\mathcal{F}$ yielding arbitrary, and states that:

There exists a map f from $\mathcal{A}$ into $\mathcal{B}$ such that for every element x of $\mathcal{A}$ holds $f(x) = \mathcal{F}(x)$

provided the parameters satisfy the following condition:

- For every element x of $\mathcal{A}$ holds $\mathcal{F}(x)$ is an element of $\mathcal{B}$.

The scheme *NonUniqExMD* deals with non empty relational structures $\mathcal{A}$, $\mathcal{B}$ and a binary predicate $\mathcal{P}$, and states that:

There exists a map f from $\mathcal{A}$ into $\mathcal{B}$ such that for every element x of $\mathcal{A}$ holds $\mathcal{P}[x, f(x)]$

provided the following requirement is met:

- For every element x of $\mathcal{A}$ there exists an element y of $\mathcal{B}$ such that $\mathcal{P}[x, y]$.

Let S, T be 1-sorted structures and let f be a map from S into T. Then $\operatorname{rng} f$ is a subset of T.

One can prove the following proposition

(9) Let S, T be non empty 1-sorted structures and let f, g be maps from S into T. If for every element s of S holds $f(s) = g(s)$, then $f = g$.

Let J be a set, let L be a relational structure, and let f, g be functions from J into the carrier of L. The predicate $f \leq g$ is defined by:

(Def. 1) For arbitrary j such that $j \in J$ there exist elements a, b of L such that $a = f(j)$ and $b = g(j)$ and $a \leq b$.

We introduce $g \geq f$ as a synonym of $f \leq g$.

Next we state the proposition

(10) Let L, M be non empty relational structures and let f, g be maps from L into M. Then $f \leq g$ if and only if for every element x of L holds $f(x) \leq g(x)$.

4. The Image of a Map

Let L, M be non empty relational structures and let f be a map from L into M. The functor $\operatorname{Im} f$ yields a strict full relational substructure of M and is defined as follows:

(Def. 2) $\operatorname{Im} f = \operatorname{sub}(\operatorname{rng} f)$.

The following two propositions are true:

(11) For all non empty relational structures L, M and for every map f from L into M holds $\operatorname{rng} f =$ the carrier of $\operatorname{Im} f$.

(12) Let L, M be non empty relational structures, and let f be a map from L into M, and let y be an element of $\operatorname{Im} f$. Then there exists an element x of L such that $f(x) = y$.

Let L be a non empty relational structure and let X be a non empty subset of L. One can verify that $\operatorname{sub}(X)$ is non empty.

Let L, M be non empty relational structures and let f be a map from L into M. Observe that $\operatorname{Im} f$ is non empty.

5. Monotone Maps

One can prove the following propositions:

(13) For every non empty relational structure L holds id_L is monotone.

(14) Let L, M, N be non empty relational structures, and let f be a map from L into M, and let g be a map from M into N. If f is monotone and g is monotone, then $g \cdot f$ is monotone.

(15) Let L, M be non empty relational structures, and let f be a map from L into M, and let X be a subset of L, and let x be an element of L. If f is monotone and $x \leq X$, then $f(x) \leq f^{\circ}X$.

(16) Let L, M be non empty relational structures, and let f be a map from L into M, and let X be a subset of L, and let x be an element of L. If f is monotone and $X \leq x$, then $f^{\circ}X \leq f(x)$.

(17) Let S, T be non empty relational structures, and let f be a map from S into T, and let X be a directed subset of S. If f is monotone, then $f^{\circ}X$ is directed.

(18) Let L be a poset with l.u.b.'s and let f be a map from L into L. If f is directed-sups-preserving, then f is monotone.

(19) Let L be a poset with g.l.b.'s and let f be a map from L into L. If f is filtered-infs-preserving, then f is monotone.

6. Idempotent Maps

One can prove the following propositions:

(20) Let S be a non empty 1-sorted structure and let f be a map from S into S. If f is idempotent, then for every element x of S holds $f(f(x)) = f(x)$.

(21) Let S be a non empty 1-sorted structure and let f be a map from S into S. If f is idempotent, then $\mathrm{rng}\, f = \{x : x$ ranges over elements of S, $x = f(x)\}$.

(22) Let S be a non empty 1-sorted structure and let f be a map from S into S. If f is idempotent, then if $X \subseteq \mathrm{rng}\, f$, then $f^{\circ}X = X$.

(23) For every non empty relational structure L holds id_L is idempotent.

7. Complete Lattices

In the sequel L denotes a complete lattice and a denotes an element of L.

The following propositions are true:

(24) If $a \in X$, then $a \leq \bigsqcup_L X$ and $\bigsqcap_L X \leq a$.

(25) Let L be a non empty relational structure. Then for every X holds sup X exists in L if and only if for every Y holds inf Y exists in L.

(26) For every non empty relational structure L such that for every X holds sup X exists in L holds L is complete.

(27) For every non empty relational structure L such that for every X holds inf X exists in L holds L is complete.

(28) Let L be a non empty relational structure. Suppose that for every subset A of L holds inf A exists in L. Given X. Then inf X exists in L and $\bigsqcap_L X = \bigsqcap_L (X \cap (\text{the carrier of } L))$.

(29) Let L be a non empty relational structure. Suppose that for every subset A of L holds sup A exists in L. Given X. Then sup X exists in L and $\bigsqcup_L X = \bigsqcup_L (X \cap (\text{the carrier of } L))$.

(30) Let L be a non empty relational structure. If for every subset A of L holds inf A exists in L, then L is complete.

One can check that every non empty poset which is up-complete, inf-complete, and upper-bounded is also complete.

Next we state several propositions:

(31) Let f be a map from L into L. Suppose f is monotone. Let M be a subset of L. If $M = \{x : x \text{ ranges over elements of } L,\ x = f(x)\}$, then $\mathrm{sub}(M)$ is a complete lattice.

(32) Every infs-inheriting non empty full relational substructure of L is a complete lattice.

(33) Every sups-inheriting non empty full relational substructure of L is a complete lattice.

(34) Let M be a non empty relational structure and let f be a map from L into M. If f is sups-preserving, then $\mathrm{Im}\, f$ is sups-inheriting.

(35) Let M be a non empty relational structure and let f be a map from L into M. If f is infs-preserving, then $\mathrm{Im}\, f$ is infs-inheriting.

(36) Let L, M be complete lattices and let f be a map from L into M. Suppose f is sups-preserving or infs-preserving. Then $\mathrm{Im}\, f$ is a complete lattice.

(37) Let f be a map from L into L. Suppose f is idempotent and directed-sups-preserving. Then $\mathrm{Im}\, f$ is directed-sups-inheriting and $\mathrm{Im}\, f$ is a complete lattice.

8. Lattices of Ideals

Next we state several propositions:

(38) Let L be a relational structure and let F be a subset of $2^{\text{the carrier of } L}$. Suppose that for every subset X of L such that $X \in F$ holds X is lower. Then $\bigcap F$ is a lower subset of L.

(39) Let L be a relational structure and let F be a subset of $2^{\text{the carrier of } L}$. Suppose that for every subset X of L such that $X \in F$ holds X is upper. Then $\bigcap F$ is an upper subset of L.

(40) Let L be an antisymmetric relational structure with l.u.b.'s and let F be a subset of $2^{\text{the carrier of } L}$. Suppose that for every subset X of L such that $X \in F$ holds X is lower and directed. Then $\bigcap F$ is a directed subset of L.

(41) Let L be an antisymmetric relational structure with g.l.b.'s and let F be a subset of $2^{\text{the carrier of } L}$. Suppose that for every subset X of L such that $X \in F$ holds X is upper and filtered. Then $\bigcap F$ is a filtered subset of L.

(42) For every poset L with g.l.b.'s and for all ideals I, J of L holds $I \cap J$ is an ideal of L.

Let L be a non empty reflexive transitive relational structure. One can check that $\mathrm{Ids}(L)$ is non empty.

We now state three propositions:

(43) Let L be a non empty reflexive transitive relational structure. Then x is an element of $\langle \mathrm{Ids}(L), \subseteq \rangle$ if and only if x is an ideal of L.

(44) Let L be a non empty reflexive transitive relational structure and let I be an element of $\langle \mathrm{Ids}(L), \subseteq \rangle$. If $x \in I$, then x is an element of L.

(45) For every poset L with g.l.b.'s and for all elements x, y of $\langle \mathrm{Ids}(L), \subseteq \rangle$ holds $x \sqcap y = x \cap y$.

Let L be a poset with g.l.b.'s. One can verify that $\langle \mathrm{Ids}(L), \subseteq \rangle$ has g.l.b.'s.

We now state the proposition

(46) Let L be a poset with l.u.b.'s and let x, y be elements of $\langle \mathrm{Ids}(L), \subseteq \rangle$. Then there exists a subset Z of L such that

(i) $Z = \{z : z$ ranges over elements of L, $z \in x \ \vee \ z \in y \ \vee \ \bigvee_{a,b\,:\,\text{element of } L} \ a \in x \ \wedge \ b \in y \ \wedge \ z = a \sqcup b\}$,

(ii) $\sup \{x, y\}$ exists in $\langle \mathrm{Ids}(L), \subseteq \rangle$, and

(iii) $x \sqcup y = {\downarrow} Z$.

Let L be a poset with l.u.b.'s. One can check that $\langle \mathrm{Ids}(L), \subseteq \rangle$ has l.u.b.'s.

One can prove the following four propositions:

(47) For every lower-bounded sup-semilattice L and for every non empty subset X of $\mathrm{Ids}(L)$ holds $\bigcap X$ is an ideal of L.

(48) Let L be a lower-bounded sup-semilattice and let A be a non empty subset of $\langle \mathrm{Ids}(L), \subseteq \rangle$. Then $\inf A$ exists in $\langle \mathrm{Ids}(L), \subseteq \rangle$ and $\inf A = \bigcap A$.

(49) For every poset L with l.u.b.'s holds $\inf \emptyset$ exists in $\langle \mathrm{Ids}(L), \subseteq \rangle$ and $\bigsqcap_{(\langle \mathrm{Ids}(L), \subseteq \rangle)} \emptyset = \Omega_L$.

(50) For every lower-bounded sup-semilattice L holds $\langle \mathrm{Ids}(L), \subseteq \rangle$ is complete.

Let L be a lower-bounded sup-semilattice. Note that $\langle \mathrm{Ids}(L), \subseteq \rangle$ is complete.

9. Special Maps

Let L be a non empty poset. The functor SupMap(L) yielding a map from $\langle \mathrm{Ids}(L), \subseteq \rangle$ into L is defined as follows:

(Def. 3) For every ideal I of L holds $(\mathrm{SupMap}(L))(I) = \sup I$.

We now state three propositions:

(51) For every non empty poset L holds $\mathrm{dom}\,\mathrm{SupMap}(L) = \mathrm{Ids}(L)$ and $\mathrm{rng}\,\mathrm{SupMap}(L)$ is a subset of L.

(52) For every non empty poset L holds $x \in \mathrm{dom}\,\mathrm{SupMap}(L)$ iff x is an ideal of L.

(53) For every up-complete non empty poset L holds SupMap(L) is monotone.

Let L be an up-complete non empty poset. Observe that SupMap(L) is monotone.

Let L be a non empty poset. The functor IdsMap(L) yielding a map from L into $\langle \mathrm{Ids}(L), \subseteq \rangle$ is defined by:

(Def. 4) For every element x of L holds $(\mathrm{IdsMap}(L))(x) = {\downarrow} x$.

The following proposition is true

(54) For every non empty poset L holds IdsMap(L) is monotone.

Let L be a non empty poset. Observe that IdsMap(L) is monotone.

10. The Family of Elements in a Lattice

Let L be a non empty relational structure and let F be a binary relation. The functor $\bigsqcup_L F$ yielding an element of L is defined as follows:

(Def. 5) $\bigsqcup_L F = \bigsqcup_L \mathrm{rng}\, F$.

The functor $\bigsqcap_L F$ yields an element of L and is defined by:

(Def. 6) $\bigsqcap_L F = \bigsqcap_L \mathrm{rng}\, F$.

Let J be a set, let L be a non empty relational structure, and let F be a function from J into the carrier of L. We introduce Sup(F) as a synonym of $\bigsqcup_L F$. We introduce Inf(F) as a synonym of $\bigsqcap_L F$.

Let J be a non empty set, let S be a non empty 1-sorted structure, let F be a function from J into the carrier of S, and let j be an element of J. Then $F(j)$ is an element of S.

Let J be a set, let S be a non empty 1-sorted structure, and let F be a function from J into the carrier of S. Then $\mathrm{rng}\, F$ is a subset of S.

In the sequel J is a non empty set and j is an element of J.

We now state three propositions:

(55) For every function F from J into the carrier of L holds $F(j) \leq \mathrm{Sup}(F)$ and $\mathrm{Inf}(F) \leq F(j)$.

(56) For every function F from J into the carrier of L such that for every j holds $F(j) \leq a$ holds $\mathrm{Sup}(F) \leq a$.

(57) For every function F from J into the carrier of L such that for every j holds $a \leq F(j)$ holds $a \leq \mathrm{Inf}(F)$.

References

[1] Grzegorz Bancerek. Bounds in posets and relational substructures. *Formalized Mathematics*, 6(**1**):81–91, 1997.
[2] Grzegorz Bancerek. Cartesian product of functions. *Formalized Mathematics*, 2(**4**):547–552, 1991.
[3] Grzegorz Bancerek. Complete lattices. *Formalized Mathematics*, 2(**5**):719–725, 1991.
[4] Grzegorz Bancerek. Directed sets, nets, ideals, filters, and maps. *Formalized Mathematics*, 6(**1**):93–107, 1997.
[5] Grzegorz Bancerek. Quantales. *Formalized Mathematics*, 5(**1**):85–91, 1996.
[6] Grzegorz Bancerek. The well ordering relations. *Formalized Mathematics*, 1(**1**):123–129, 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] Agata Darmochwał. Families of subsets, subspaces and mappings in topological spaces. *Formalized Mathematics*, 1(**2**):257–261, 1990.
[10] 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.
[11] Adam Grabowski. On the category of posets. *Formalized Mathematics*, 5(**4**):501–505, 1996.
[12] Adam Grabowski and Robert Milewski. Boolean posets, posets under inclusion and products of relational structures. *Formalized Mathematics*, 6(**1**):117–121, 1997.
[13] Krzysztof Hryniewiecki. Relations of tolerance. *Formalized Mathematics*, 2(**1**):105–109, 1991.
[14] Zbigniew Karno. Continuity of mappings over the union of subspaces. *Formalized Mathematics*, 3(**1**):1–16, 1992.
[15] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[16] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[17] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[18] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[19] Wojciech A. Trybulec. Partially ordered sets. *Formalized Mathematics*, 1(**2**):313–319, 1990.
[20] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[21] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received September 20, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Galois Connections [1]

Czesław Byliński
Warsaw University
Białystok

Summary. The paper is the Mizar encoding of the chapter 0 section 3 of [12] In the paper the following concept are defined: Galois connections, Heyting algebras, and Boolean algebras.

MML Identifier: `WAYBEL_1`.

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

1. Preliminaries

Let A, B be non empty sets. One can check that every function from A into B is non empty.

Let L_1, L_2 be non empty 1-sorted structures and let f be a map from L_1 into L_2. Let us observe that f is one-to-one if and only if:

(Def. 1) For all elements x, y of L_1 such that $f(x) = f(y)$ holds $x = y$.

One can prove the following proposition

(1) Let L be a non empty 1-sorted structure and let f be a map from L into L. If for every element x of L holds $f(x) = x$, then $f = \mathrm{id}_L$.

Let L_1, L_2 be non empty relational structures and let f be a map from L_1 into L_2. Let us observe that f is monotone if and only if:

(Def. 2) For all elements x, y of L_1 such that $x \leq y$ holds $f(x) \leq f(y)$.

We now state four propositions:

(2) Let L be a non empty antisymmetric transitive relational structure with g.l.b.'s and let x, y, z be elements of L. If $x \leq y$, then $x \sqcap z \leq y \sqcap z$.

[1] This work was partially supported by Office of Naval Research Grant N00014-95-1-1336.

ISSN 1426–2630

(3) Let L be a non empty antisymmetric transitive relational structure with l.u.b.'s and let x, y, z be elements of L. If $x \leq y$, then $x \sqcup z \leq y \sqcup z$.

(4) Let L be a non empty lower-bounded antisymmetric relational structure and let x be an element of L. Then if L has g.l.b.'s, then $\bot_L \sqcap x = \bot_L$ and if L is reflexive and transitive and has l.u.b.'s, then $\bot_L \sqcup x = x$.

(5) Let L be a non empty upper-bounded antisymmetric relational structure and let x be an element of L. Then if L is transitive and reflexive and has g.l.b.'s, then $\top_L \sqcap x = x$ and if L has l.u.b.'s, then $\top_L \sqcup x = \top_L$.

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

(Def. 3) For all elements x, y, z of L holds $x \sqcap (y \sqcup z) = x \sqcap y \sqcup x \sqcap z$.

We now state the proposition

(6) For every lattice L holds L is distributive iff for all elements x, y, z of L holds $x \sqcup y \sqcap z = (x \sqcup y) \sqcap (x \sqcup z)$.

Let X be a set. One can verify that $2^X_\subseteq$ is distributive.

Let S be a non empty relational structure and let X be a set. We say that min X exists in S if and only if:

(Def. 4) Inf X exists in S and $\bigsqcap_S X \in X$.

We introduce X has the minimum in S as a synonym of min X exists in S. We say that max X exists in S if and only if:

(Def. 5) Sup X exists in S and $\bigsqcup_S X \in X$.

We introduce X has the maximum in S as a synonym of max X exists in S.

Let S be a non empty relational structure, let s be an element of S, and let X be a set. We say that s is a minimum of X if and only if:

(Def. 6) Inf X exists in S and $s = \bigsqcap_S X$ and $\bigsqcap_S X \in X$.

We say that s is a maximum of X if and only if:

(Def. 7) Sup X exists in S and $s = \bigsqcup_S X$ and $\bigsqcup_S X \in X$.

Let L be a relational structure. Note that id_L is isomorphic.

Let L_1, L_2 be relational structures. We say that L_1 and L_2 are isomorphic if and only if:

(Def. 8) There exists map from L_1 into L_2 which is isomorphic.

Let us notice that the predicate defined above is reflexive.

We now state two propositions:

(7) For all non empty relational structures L_1, L_2 such that L_1 and L_2 are isomorphic holds L_2 and L_1 are isomorphic.

(8) Let L_1, L_2, L_3 be relational structures. Suppose L_1 and L_2 are isomorphic and L_2 and L_3 are isomorphic. Then L_1 and L_3 are isomorphic.

2. Galois Connections

Let S, T be relational structures. A set is said to be a connection between S and T if:

(Def. 9) There exists a map g from S into T and there exists a map d from T into S such that it $= \langle g, d \rangle$.

Let S, T be relational structures, let g be a map from S into T, and let d be a map from T into S. Then $\langle g, d \rangle$ is a connection between S and T.

Let S, T be non empty relational structures and let g_1 be a connection between S and T. We say that g_1 is Galois if and only if the condition (Def. 10) is satisfied.

(Def. 10) There exists a map g from S into T and there exists a map d from T into S such that
 (i) $g_1 = \langle g, d \rangle$,
 (ii) g is monotone,
 (iii) d is monotone, and
 (iv) for every element t of T and for every element s of S holds $t \leq g(s)$ iff $d(t) \leq s$.

Next we state the proposition

(9) Let S, T be non empty posets, and let g be a map from S into T, and let d be a map from T into S. Then $\langle g, d \rangle$ is Galois if and only if the following conditions are satisfied:
 (i) g is monotone,
 (ii) d is monotone, and
 (iii) for every element t of T and for every element s of S holds $t \leq g(s)$ iff $d(t) \leq s$.

Let S, T be non empty relational structures and let g be a map from S into T. We say that g is upper adjoint if and only if:

(Def. 11) There exists a map d from T into S such that $\langle g, d \rangle$ is Galois.

We introduce g has a lower adjoint as a synonym of g is upper adjoint.

Let S, T be non empty relational structures and let d be a map from T into S. We say that d is lower adjoint if and only if:

(Def. 12) There exists a map g from S into T such that $\langle g, d \rangle$ is Galois.

We introduce d has an upper adjoint as a synonym of d is lower adjoint.

One can prove the following four propositions:

(10) Let S, T be non empty posets, and let g be a map from S into T, and let d be a map from T into S. If $\langle g, d \rangle$ is Galois, then g is upper adjoint and d is lower adjoint.

(11) Let S, T be non empty posets, and let g be a map from S into T, and let d be a map from T into S. Then $\langle g, d \rangle$ is Galois if and only if the following conditions are satisfied:
 (i) g is monotone, and

(ii) for every element t of T holds $d(t)$ is a minimum of $g^{-1}\uparrow t$.

(12) Let S, T be non empty posets, and let g be a map from S into T, and let d be a map from T into S. Then $\langle g, d\rangle$ is Galois if and only if the following conditions are satisfied:

(i) d is monotone, and

(ii) for every element s of S holds $g(s)$ is a maximum of $d^{-1}\downarrow s$.

(13) Let S, T be non empty posets and let g be a map from S into T. If g is upper adjoint, then g is infs-preserving.

Let S, T be non empty posets. Observe that every map from S into T which is upper adjoint is also infs-preserving.

We now state the proposition

(14) Let S, T be non empty posets and let d be a map from T into S. If d is lower adjoint, then d is sups-preserving.

Let S, T be non empty posets. Note that every map from S into T which is lower adjoint is also sups-preserving.

Next we state a number of propositions:

(15) Let S, T be non empty posets and let g be a map from S into T. Suppose S is complete and g is infs-preserving. Then there exists a map d from T into S such that $\langle g, d\rangle$ is Galois and for every element t of T holds $d(t)$ is a minimum of $g^{-1}\uparrow t$.

(16) Let S, T be non empty posets and let d be a map from T into S. Suppose T is complete and d is sups-preserving. Then there exists a map g from S into T such that $\langle g, d\rangle$ is Galois and for every element s of S holds $g(s)$ is a maximum of $d^{-1}\downarrow s$.

(17) Let S, T be non empty posets and let g be a map from S into T. Suppose S is complete. Then g is infs-preserving if and only if g is monotone and g has a lower adjoint.

(18) Let S, T be non empty posets and let d be a map from T into S. Suppose T is complete. Then d is sups-preserving if and only if d is monotone and d has an upper adjoint.

(19) Let S, T be non empty posets, and let g be a map from S into T, and let d be a map from T into S. If $\langle g, d\rangle$ is Galois, then $d \cdot g \leq \mathrm{id}_S$ and $\mathrm{id}_T \leq g \cdot d$.

(20) Let S, T be non empty posets, and let g be a map from S into T, and let d be a map from T into S. Suppose g is monotone and d is monotone and $d \cdot g \leq \mathrm{id}_S$ and $\mathrm{id}_T \leq g \cdot d$. Then $\langle g, d\rangle$ is Galois.

(21) Let S, T be non empty posets, and let g be a map from S into T, and let d be a map from T into S. Suppose g is monotone and d is monotone and $d \cdot g \leq \mathrm{id}_S$ and $\mathrm{id}_T \leq g \cdot d$. Then $d = d \cdot g \cdot d$ and $g = g \cdot d \cdot g$.

(22) Let S, T be non empty relational structures, and let g be a map from S into T, and let d be a map from T into S. If $d = d \cdot g \cdot d$ and $g = g \cdot d \cdot g$, then $g \cdot d$ is idempotent and $d \cdot g$ is idempotent.

(23) Let S, T be non empty posets, and let g be a map from S into T, and let d be a map from T into S. Suppose $\langle g, d\rangle$ is Galois and g is onto. Let t be an element of T. Then $d(t)$ is a minimum of $g^{-1}\{t\}$.

(24) Let S, T be non empty posets, and let g be a map from S into T, and let d be a map from T into S. If for every element t of T holds $d(t)$ is a minimum of $g^{-1}\{t\}$, then $g \cdot d = \mathrm{id}_T$.

(25) Let L_1, L_2 be non empty 1-sorted structures, and let g_3 be a map from L_1 into L_2, and let g_2 be a map from L_2 into L_1. If $g_2 \cdot g_3 = \mathrm{id}_{(L_1)}$, then g_3 is one-to-one and g_2 is onto.

(26) Let S, T be non empty posets, and let g be a map from S into T, and let d be a map from T into S. If $\langle g, d\rangle$ is Galois, then g is onto iff d is one-to-one.

(27) Let S, T be non empty posets, and let g be a map from S into T, and let d be a map from T into S. Suppose $\langle g, d\rangle$ is Galois and d is onto. Let s be an element of S. Then $g(s)$ is a maximum of $d^{-1}\{s\}$.

(28) Let S, T be non empty posets, and let g be a map from S into T, and let d be a map from T into S. If for every element s of S holds $g(s)$ is a maximum of $d^{-1}\{s\}$, then $d \cdot g = \mathrm{id}_S$.

(29) Let S, T be non empty posets, and let g be a map from S into T, and let d be a map from T into S. If $\langle g, d\rangle$ is Galois, then g is one-to-one iff d is onto.

Let L be a non empty relational structure and let p be a map from L into L. We say that p is projection if and only if:

(Def. 13) p is idempotent and monotone.

We introduce p is a projection operator as a synonym of p is projection.

Let L be a non empty relational structure. Note that id_L is projection.

Let L be a non empty relational structure. Observe that there exists a map from L into L which is projection.

Let L be a non empty relational structure and let c be a map from L into L. We say that c is closure if and only if:

(Def. 14) c is projection and $\mathrm{id}_L \leq c$.

We introduce c is a closure operator as a synonym of c is closure.

Let L be a non empty relational structure. Note that every map from L into L which is closure is also projection.

Let L be a non empty reflexive relational structure. Note that there exists a map from L into L which is closure.

Let L be a non empty reflexive relational structure. Note that id_L is closure.

Let L be a non empty relational structure and let k be a map from L into L. We say that k is kernel if and only if:

(Def. 15) k is projection and $k \leq \mathrm{id}_L$.

We introduce k is a kernel operator as a synonym of k is kernel.

Let L be a non empty relational structure. One can check that every map from L into L which is kernel is also projection.

Let L be a non empty reflexive relational structure. Note that there exists a map from L into L which is kernel.

Let L be a non empty reflexive relational structure. One can check that id_L is kernel.

One can prove the following two propositions:

(30) Let L be a non empty poset, and let c be a map from L into L, and let X be a subset of L. Suppose c is a closure operator and $\inf X$ exists in L and $X \subseteq \operatorname{rng} c$. Then $\inf X = c(\inf X)$.

(31) Let L be a non empty poset, and let k be a map from L into L, and let X be a subset of L. Suppose k is a kernel operator and $\sup X$ exists in L and $X \subseteq \operatorname{rng} k$. Then $\sup X = k(\sup X)$.

Let L_1, L_2 be non empty relational structures and let g be a map from L_1 into L_2. The functor g° yields a map from L_1 into $\operatorname{Im} g$ and is defined as follows:

(Def. 16) $g^\circ = (\text{the carrier of } \operatorname{Im} g) \upharpoonright (g)$.

One can prove the following proposition

(32) For all non empty relational structures L_1, L_2 and for every map g from L_1 into L_2 holds $g^\circ = g$.

Let L_1, L_2 be non empty relational structures and let g be a map from L_1 into L_2. Observe that g° is onto.

The following proposition is true

(33) Let L_1, L_2 be non empty relational structures and let g be a map from L_1 into L_2. If g is monotone, then g° is monotone.

Let L_1, L_2 be non empty relational structures and let g be a map from L_1 into L_2. The functor $g_\circ$ yields a map from $\operatorname{Im} g$ into L_2 and is defined by:

(Def. 17) $g_\circ = \mathrm{id}_{\operatorname{Im} g}$.

Next we state the proposition

(34) Let L_1, L_2 be non empty relational structures, and let g be a map from L_1 into L_2, and let s be an element of $\operatorname{Im} g$. Then $g_\circ(s) = s$.

Let L_1, L_2 be non empty relational structures and let g be a map from L_1 into L_2. One can check that $g_\circ$ is one-to-one and monotone.

We now state a number of propositions:

(35) For every non empty relational structure L and for every map f from L into L holds $f_\circ \cdot f^\circ = f$.

(36) For every non empty poset L and for every map f from L into L such that f is idempotent holds $f^\circ \cdot f_\circ = \mathrm{id}_{\operatorname{Im} f}$.

(37) Let L be a non empty poset and let f be a map from L into L. Suppose f is a projection operator. Then there exists a non empty poset T and there exists a map q from L into T and there exists a map i from T into L such that q is monotone and onto and i is monotone and one-to-one and $f = i \cdot q$ and $\mathrm{id}_T = q \cdot i$.

(38) Let L be a non empty poset and let f be a map from L into L. Given a non empty poset T and a map q from L into T and a map i from T into

L such that q is monotone and i is monotone and $f = i \cdot q$ and $\mathrm{id}_T = q \cdot i$. Then f is a projection operator.

(39) For every non empty poset L and for every map f from L into L such that f is a closure operator holds $\langle f_\circ, f^\circ \rangle$ is Galois.

(40) Let L be a non empty poset and let f be a map from L into L. Suppose f is a closure operator. Then there exists a non empty poset S and there exists a map g from S into L and there exists a map d from L into S such that $\langle g, d \rangle$ is Galois and $f = g \cdot d$.

(41) Let L be a non empty poset and let f be a map from L into L. Suppose that

(i) f is monotone, and

(ii) there exists a non empty poset S and there exists a map g from S into L and there exists a map d from L into S such that $\langle g, d \rangle$ is Galois and $f = g \cdot d$.

Then f is a closure operator.

(42) For every non empty poset L and for every map f from L into L such that f is a kernel operator holds $\langle f^\circ, f_\circ \rangle$ is Galois.

(43) Let L be a non empty poset and let f be a map from L into L. Suppose f is a kernel operator. Then there exists a non empty poset T and there exists a map g from L into T and there exists a map d from T into L such that $\langle g, d \rangle$ is Galois and $f = d \cdot g$.

(44) Let L be a non empty poset and let f be a map from L into L. Suppose that

(i) f is monotone, and

(ii) there exists a non empty poset T and there exists a map g from L into T and there exists a map d from T into L such that $\langle g, d \rangle$ is Galois and $f = d \cdot g$.

Then f is a kernel operator.

(45) Let L be a non empty poset and let p be a map from L into L. Suppose p is a projection operator. Then $\mathrm{rng}\, p = \{c : c$ ranges over elements of L, $c \leq p(c)\} \cap \{k : k$ ranges over elements of L, $p(k) \leq k\}$.

(46) Let L be a non empty poset and let p be a map from L into L. Suppose p is a projection operator. Then

(i) $\{c : c$ ranges over elements of L, $c \leq p(c)\}$ is a non empty subset of L, and

(ii) $\{k : k$ ranges over elements of L, $p(k) \leq k\}$ is a non empty subset of L.

(47) Let L be a non empty poset and let p be a map from L into L. Suppose p is a projection operator. Then $\mathrm{rng}(p \restriction \{c : c$ ranges over elements of L, $c \leq p(c)\}) = \mathrm{rng}\, p$ and $\mathrm{rng}(p \restriction \{k : k$ ranges over elements of L, $p(k) \leq k\}) = \mathrm{rng}\, p$.

(48) Let L be a non empty poset and let p be a map from L into L. Suppose p is a projection operator. Let L_4 be a non empty subset of L and let L_5

be a non empty subset of L. Suppose $L_4 = \{c : c$ ranges over elements of L, $c \leq p(c)\}$. Then $p \restriction L_4$ is a map from $\mathrm{sub}(L_4)$ into $\mathrm{sub}(L_4)$.

(49) Let L be a non empty poset and let p be a map from L into L. Suppose p is a projection operator. Let L_5 be a non empty subset of L. Suppose $L_5 = \{k : k$ ranges over elements of L, $p(k) \leq k\}$. Then $p \restriction L_5$ is a map from $\mathrm{sub}(L_5)$ into $\mathrm{sub}(L_5)$.

(50) Let L be a non empty poset and let p be a map from L into L. Suppose p is a projection operator. Let L_4 be a non empty subset of L. Suppose $L_4 = \{c : c$ ranges over elements of L, $c \leq p(c)\}$. Let p_1 be a map from $\mathrm{sub}(L_4)$ into $\mathrm{sub}(L_4)$. If $p_1 = p \restriction L_4$, then p_1 is a closure operator.

(51) Let L be a non empty poset and let p be a map from L into L. Suppose p is a projection operator. Let L_5 be a non empty subset of L. Suppose $L_5 = \{k : k$ ranges over elements of L, $p(k) \leq k\}$. Let p_2 be a map from $\mathrm{sub}(L_5)$ into $\mathrm{sub}(L_5)$. If $p_2 = p \restriction L_5$, then p_2 is a kernel operator.

(52) Let L be a non empty poset and let p be a map from L into L. Suppose p is monotone. Let L_4 be a subset of L. If $L_4 = \{c : c$ ranges over elements of L, $c \leq p(c)\}$, then $\mathrm{sub}(L_4)$ is sups-inheriting.

(53) Let L be a non empty poset and let p be a map from L into L. Suppose p is monotone. Let L_5 be a subset of L. If $L_5 = \{k : k$ ranges over elements of L, $p(k) \leq k\}$, then $\mathrm{sub}(L_5)$ is infs-inheriting.

(54) Let L be a non empty poset and let p be a map from L into L. Suppose p is a projection operator. Let L_4 be a non empty subset of L. Suppose $L_4 = \{c : c$ ranges over elements of L, $c \leq p(c)\}$. Then

(i) if p is infs-preserving, then $\mathrm{sub}(L_4)$ is infs-inheriting and $\mathrm{Im}\, p$ is infs-inheriting, and

(ii) if p is filtered-infs-preserving, then $\mathrm{sub}(L_4)$ is filtered-infs-inheriting and $\mathrm{Im}\, p$ is filtered-infs-inheriting.

(55) Let L be a non empty poset and let p be a map from L into L. Suppose p is a projection operator. Let L_5 be a non empty subset of L. Suppose $L_5 = \{k : k$ ranges over elements of L, $p(k) \leq k\}$. Then

(i) if p is sups-preserving, then $\mathrm{sub}(L_5)$ is sups-inheriting and $\mathrm{Im}\, p$ is sups-inheriting, and

(ii) if p is directed-sups-preserving, then $\mathrm{sub}(L_5)$ is directed-sups-inheriting and $\mathrm{Im}\, p$ is directed-sups-inheriting.

(56) Let L be a non empty poset and let p be a map from L into L. Then if p is a closure operator, then $\mathrm{Im}\, p$ is infs-inheriting and if p is a kernel operator, then $\mathrm{Im}\, p$ is sups-inheriting.

(57) Let L be a complete non empty poset and let p be a map from L into L. If p is a projection operator, then $\mathrm{Im}\, p$ is complete.

(58) Let L be a non empty poset and let c be a map from L into L. Suppose c is a closure operator. Then

(i) c° is sups-preserving, and

(ii) for every subset X of L such that $X \subseteq$ the carrier of $\operatorname{Im} c$ and sup X exists in L holds sup X exists in $\operatorname{Im} c$ and $\bigsqcup_{\operatorname{Im} c} X = c(\bigsqcup_L X)$.

(59) Let L be a non empty poset and let k be a map from L into L. Suppose k is a kernel operator. Then

(i) k° is infs-preserving, and

(ii) for every subset X of L such that $X \subseteq$ the carrier of $\operatorname{Im} k$ and inf X exists in L holds inf X exists in $\operatorname{Im} k$ and $\bigsqcap_{\operatorname{Im} k} X = k(\bigsqcap_L X)$.

3. Heyting Algebra

Next we state two propositions:

(60) For every complete non empty poset L holds $\langle \operatorname{IdsMap}(L), \operatorname{SupMap}(L) \rangle$ is Galois and $\operatorname{SupMap}(L)$ is sups-preserving.

(61) For every complete non empty poset L holds $\operatorname{IdsMap}(L) \cdot \operatorname{SupMap}(L)$ is a closure operator and $\operatorname{Im}(\operatorname{IdsMap}(L) \cdot \operatorname{SupMap}(L))$ and L are isomorphic.

Let S be a non empty relational structure and let x be an element of S. The functor $x \sqcap \square$ yields a map from S into S and is defined as follows:

(Def. 18) For every element s of S holds $(x \sqcap \square)(s) = x \sqcap s$.

Next we state two propositions:

(62) For every non empty relational structure S and for all elements x, t of S holds $\{s : s$ ranges over elements of S, $x \sqcap s \leq t\} = (x \sqcap \square)^{-1} {\downarrow} t$.

(63) For every non empty semilattice S and for every element x of S holds $x \sqcap \square$ is monotone.

Let S be a non empty semilattice and let x be an element of S. Note that $x \sqcap \square$ is monotone.

The following propositions are true:

(64) Let S be a non empty relational structure, and let x be an element of S, and let X be a subset of S. Then $(x \sqcap \square)^{\circ} X = \{x \sqcap y : y$ ranges over elements of S, $y \in X\}$.

(65) Let S be a non empty semilattice. Then for every element x of S holds $x \sqcap \square$ has an upper adjoint if and only if for all elements x, t of S holds max $\{s : s$ ranges over elements of S, $x \sqcap s \leq t\}$ exists in S.

(66) Let S be a non empty semilattice. Suppose that for every element x of S holds $x \sqcap \square$ has an upper adjoint. Let X be a subset of S. Suppose sup X exists in S. Let x be an element of S. Then $x \sqcap \bigsqcup_S X = \bigsqcup_S \{x \sqcap y : y$ ranges over elements of S, $y \in X\}$.

(67) Let S be a complete non empty poset. Then for every element x of S holds $x \sqcap \square$ has an upper adjoint if and only if for every subset X of S and for every element x of S holds $x \sqcap \bigsqcup_S X = \bigsqcup_S \{x \sqcap y : y$ ranges over elements of S, $y \in X\}$.

(68) Let S be a non empty lattice. Suppose that for every subset X of S such that sup X exists in S and for every element x of S holds $x \sqcap \bigsqcup_S X = \bigsqcup_S \{x \sqcap y : y$ ranges over elements of S, $y \in X\}$. Then S is distributive.

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

(Def. 19) H is a lattice and for every element x of H holds $x \sqcap \Box$ has an upper adjoint.

We introduce H is a Heyting algebra as a synonym of H is Heyting.

Let us observe that every non empty relational structure which is Heyting is also reflexive, transitive, and antisymmetric and has g.l.b.'s and l.u.b.'s.

Let H be a non empty relational structure and let a be an element of H. Let us assume that H is Heyting. The functor $a \Rightarrow \Box$ yielding a map from H into H is defined as follows:

(Def. 20) $\langle a \Rightarrow \Box, a \sqcap \Box \rangle$ is Galois.

We now state the proposition

(69) For every non empty relational structure H such that H is a Heyting algebra holds H is distributive.

Let us observe that every non empty relational structure which is Heyting is also distributive.

Let H be a non empty relational structure and let a, y be elements of H. The functor $a \Rightarrow y$ yields an element of H and is defined by:

(Def. 21) $a \Rightarrow y = (a \Rightarrow \Box)(y)$.

One can prove the following two propositions:

(70) Let H be a non empty relational structure. Suppose H is a Heyting algebra. Let x, a, y be elements of H. Then $x \geq a \sqcap y$ if and only if $a \Rightarrow x \geq y$.

(71) For every non empty relational structure H such that H is a Heyting algebra holds H is upper-bounded.

Let us mention that every non empty relational structure which is Heyting is also upper-bounded.

Next we state a number of propositions:

(72) Let H be a non empty relational structure. Suppose H is a Heyting algebra. Let a, b be elements of H. Then $\top_H = a \Rightarrow b$ if and only if $a \leq b$.

(73) For every non empty relational structure H such that H is a Heyting algebra and for every element a of H holds $\top_H = a \Rightarrow a$.

(74) Let H be a non empty relational structure. Suppose H is a Heyting algebra. Let a, b be elements of H. If $\top_H = a \Rightarrow b$ and $\top_H = b \Rightarrow a$, then $a = b$.

(75) Let H be a non empty relational structure. If H is a Heyting algebra, then for all elements a, b of H holds $b \leq a \Rightarrow b$.

(76) Let H be a non empty relational structure. If H is a Heyting algebra, then for every element a of H holds $\top_H = a \Rightarrow \top_H$.

(77) For every non empty relational structure H such that H is a Heyting algebra and for every element b of H holds $b = \top_H \Rightarrow b$.

(78) Let H be a non empty relational structure. Suppose H is a Heyting algebra. Let a, b, c be elements of H. If $a \leq b$, then $b \Rightarrow c \leq a \Rightarrow c$.

(79) Let H be a non empty relational structure. Suppose H is a Heyting algebra. Let a, b, c be elements of H. If $b \leq c$, then $a \Rightarrow b \leq a \Rightarrow c$.

(80) Let H be a non empty relational structure. Suppose H is a Heyting algebra. Let a, b be elements of H. Then $a \sqcap (a \Rightarrow b) = a \sqcap b$.

(81) Let H be a non empty relational structure. Suppose H is a Heyting algebra. Let a, b, c be elements of H. Then $a \sqcup b \Rightarrow c = (a \Rightarrow c) \sqcap (b \Rightarrow c)$.

Let H be a non empty relational structure and let a be an element of H. The functor $\neg a$ yields an element of H and is defined as follows:

(Def. 22) $\neg a = a \Rightarrow \bot_H$.

The following propositions are true:

(82) Let H be a non empty relational structure. Suppose H is a Heyting algebra and lower-bounded. Let a be an element of H. Then $\neg a$ is a maximum of $\{x : x$ ranges over elements of H, $a \sqcap x = \bot_H\}$.

(83) Let H be a non empty relational structure. If H is a Heyting algebra and lower-bounded, then $\neg(\bot_H) = \top_H$ and $\neg(\top_H) = \bot_H$.

(84) Let H be a non empty lower-bounded relational structure. Suppose H is a Heyting algebra. Let a, b be elements of H. Then $\neg a \geq b$ if and only if $\neg b \geq a$.

(85) Let H be a non empty lower-bounded relational structure. Suppose H is a Heyting algebra. Let a, b be elements of H. Then $\neg a \geq b$ if and only if $a \sqcap b = \bot_H$.

(86) Let H be a non empty lower-bounded relational structure. Suppose H is a Heyting algebra. Let a, b be elements of H. If $a \leq b$, then $\neg b \leq \neg a$.

(87) Let H be a non empty lower-bounded relational structure. Suppose H is a Heyting algebra. Let a, b be elements of H. Then $\neg(a \sqcup b) = \neg a \sqcap \neg b$.

(88) Let H be a non empty lower-bounded relational structure. Suppose H is a Heyting algebra. Let a, b be elements of H. Then $\neg(a \sqcap b) \geq \neg a \sqcup \neg b$.

Let L be a non empty relational structure and let x, y be elements of L. We say that y is a complement of x if and only if:

(Def. 23) $x \sqcup y = \top_L$ and $x \sqcap y = \bot_L$.

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

(Def. 24) For every element x of L holds there exists element of L which is a complement of x.

Let X be a set. Observe that $2^X_\subseteq$ is complemented.

Next we state two propositions:

(89) Let L be a non empty bounded lattice. Suppose L is a Heyting algebra and for every element x of L holds $\neg\neg x = x$. Let x be an element of L. Then $\neg x$ is a complement of x.

(90) Let L be a non empty bounded lattice. Then L is distributive and complemented if and only if L is a Heyting algebra and for every element x of L holds $\neg\neg x = x$.

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

(Def. 25) B is a lattice bounded distributive and complemented.

We introduce B is a Boolean algebra and B is a Boolean lattice as synonyms of B is Boolean.

Let us note that every non empty relational structure which is Boolean is also reflexive, transitive, antisymmetric, bounded, distributive, and complemented and has g.l.b.'s and l.u.b.'s.

Let us observe that every non empty relational structure which is reflexive, transitive, antisymmetric, bounded, distributive, and complemented and has g.l.b.'s and l.u.b.'s is also Boolean.

Let us note that every non empty relational structure which is Boolean is also Heyting.

One can verify that there exists a lattice which is strict, Boolean, and non empty.

Let us observe that there exists a lattice which is strict, Heyting, and non empty.

References

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

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

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

[4] Grzegorz Bancerek. Filters - Part II. Quotient lattices modulo filters and direct product of two lattices. *Formalized Mathematics*, 2(**3**):433–438, 1991.

[5] Grzegorz Bancerek. Quantales. *Formalized Mathematics*, 5(**1**):85–91, 1996.

[6] Grzegorz Bancerek. The well ordering relations. *Formalized Mathematics*, 1(**1**):123–129, 1990.

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

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

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

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

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

[12] 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.

[13] Adam Grabowski. On the category of posets. *Formalized Mathematics*, 5(**4**):501–505, 1996.
[14] Adam Grabowski and Robert Milewski. Boolean posets, posets under inclusion and products of relational structures. *Formalized Mathematics*, 6(**1**):117–121, 1997.
[15] Krzysztof Hryniewiecki. Relations of tolerance. *Formalized Mathematics*, 2(**1**):105–109, 1991.
[16] Zbigniew Karno. Continuity of mappings over the union of subspaces. *Formalized Mathematics*, 3(**1**):1–16, 1992.
[17] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[18] Yozo Toda. The formalization of simple graphs. *Formalized Mathematics*, 5(**1**):137–144, 1996.
[19] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[20] Wojciech A. Trybulec. Partially ordered sets. *Formalized Mathematics*, 1(**2**):313–319, 1990.
[21] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[22] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[23] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.
[24] Mariusz Żynel and Czesław Byliński. Properties of relational structures, posets, lattices and maps. *Formalized Mathematics*, 6(**1**):123–130, 1997.

Received September 25, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Cartesian Products of Relations and Relational Structures [1]

Artur Korniłowicz
Warsaw University
Białystok

Summary. In this paper the definitions of cartesian products of relations and relational structures are introduced. Facts about these notions are proved. This work is the continuation of formalization of [8].

MML Identifier: `YELLOW_3`.

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

1. Preliminaries

In this article we present several logical schemes. The scheme *FraenkelA2* concerns a non empty set $\mathcal{A}$, a binary functor $\mathcal{F}$ yielding a set, and two binary predicates $\mathcal{P}$, $\mathcal{Q}$, and states that:

$\{\mathcal{F}(s,t) : s$ ranges over elements of $\mathcal{A}$, t ranges over elements of $\mathcal{A}$, $\mathcal{P}[s,t]\}$ is a subset of $\mathcal{A}$

provided the following condition is met:

- For every element s of $\mathcal{A}$ and for every element t of $\mathcal{A}$ holds $\mathcal{F}(s,t) \in \mathcal{A}$.

The scheme *ExtensionalityR* deals with binary relations $\mathcal{A}$, $\mathcal{B}$ and a binary predicate $\mathcal{P}$, and states that:

$\mathcal{A} = \mathcal{B}$

provided the following requirements are met:

- For all sets a, b holds $\langle a, b\rangle \in \mathcal{A}$ iff $\mathcal{P}[a,b]$,
- For all sets a, b holds $\langle a, b\rangle \in \mathcal{B}$ iff $\mathcal{P}[a,b]$.

[1]This work was partially supported by Office of Naval Research Grant N00014-95-1-1336.

ISSN 1426–2630

Let X be an empty set. Observe that $\pi_1(X)$ is empty and $\pi_2(X)$ is empty.

Let X, Y be non empty sets and let D be a non empty subset of $[\!: X, Y :\!]$. Observe that $\pi_1(D)$ is non empty and $\pi_2(D)$ is non empty.

Let L be a non empty relational structure and let X be an empty subset of L. Observe that $\downarrow X$ is empty.

Let L be a non empty relational structure and let X be an empty subset of L. Observe that $\uparrow X$ is empty.

The following propositions are true:

(1) For all sets X, Y and for every subset D of $[\!: X, Y :\!]$ holds $D \subseteq [\!: \pi_1(D), \pi_2(D) :\!]$.

(2) Let L be a transitive antisymmetric relational structure with g.l.b.'s and let a, b, c, d be elements of L. If $a \leq c$ and $b \leq d$, then $a \sqcap b \leq c \sqcap d$.

(3) Let L be a transitive antisymmetric relational structure with l.u.b.'s and let a, b, c, d be elements of L. If $a \leq c$ and $b \leq d$, then $a \sqcup b \leq c \sqcup d$.

(4) Let L be a complete reflexive antisymmetric non empty relational structure, and let D be a subset of L, and let x be an element of L. If $x \in D$, then $\sup D \sqcap x = x$.

(5) Let L be a complete reflexive antisymmetric non empty relational structure, and let D be a subset of L, and let x be an element of L. If $x \in D$, then $\inf D \sqcup x = x$.

(6) For every non empty relational structure L and for all subsets X, Y of L such that $X \subseteq Y$ holds $\downarrow X \subseteq \downarrow Y$.

(7) For every non empty relational structure L and for all subsets X, Y of L such that $X \subseteq Y$ holds $\uparrow X \subseteq \uparrow Y$.

(8) Let S, T be posets with g.l.b.'s, and let f be a map from S into T, and let x, y be elements of S. Then f preserves inf of $\{x, y\}$ if and only if $f(x \sqcap y) = f(x) \sqcap f(y)$.

(9) Let S, T be posets with l.u.b.'s, and let f be a map from S into T, and let x, y be elements of S. Then f preserves sup of $\{x, y\}$ if and only if $f(x \sqcup y) = f(x) \sqcup f(y)$.

Now we present four schemes. The scheme *Inf Union* concerns a complete antisymmetric non empty relational structure $\mathcal{A}$ and a unary predicate $\mathcal{P}$, and states that:

$\bigsqcap_{\mathcal{A}}\{\bigsqcap_{\mathcal{A}} X : X$ ranges over subsets of $\mathcal{A}$, $\mathcal{P}[X]\} \geq \bigsqcap_{\mathcal{A}} \bigcup\{X : X$ ranges over subsets of $\mathcal{A}$, $\mathcal{P}[X]\}$

for all values of the parameters.

The scheme *Inf of Infs* deals with a complete lattice $\mathcal{A}$ and a unary predicate $\mathcal{P}$, and states that:

$\bigsqcap_{\mathcal{A}}\{\bigsqcap_{\mathcal{A}} X : X$ ranges over subsets of $\mathcal{A}$, $\mathcal{P}[X]\} = \bigsqcap_{\mathcal{A}} \bigcup\{X : X$ ranges over subsets of $\mathcal{A}$, $\mathcal{P}[X]\}$

for all values of the parameters.

The scheme *Sup Union* concerns a complete antisymmetric non empty relational structure $\mathcal{A}$ and a unary predicate $\mathcal{P}$, and states that:

$\bigsqcup_{\mathcal{A}}\{\bigsqcup_{\mathcal{A}} X : X$ ranges over subsets of $\mathcal{A}$, $\mathcal{P}[X]\} \leq \bigsqcup_{\mathcal{A}} \bigcup\{X : X$ ranges over subsets of $\mathcal{A}$, $\mathcal{P}[X]\}$

for all values of the parameters.

The scheme *Sup of Sups* concerns a complete lattice $\mathcal{A}$ and a unary predicate $\mathcal{P}$, and states that:

$\bigsqcup_{\mathcal{A}}\{\bigsqcup_{\mathcal{A}} X : X$ ranges over subsets of $\mathcal{A}$, $\mathcal{P}[X]\} = \bigsqcup_{\mathcal{A}} \bigcup\{X : X$ ranges over subsets of $\mathcal{A}$, $\mathcal{P}[X]\}$

for all values of the parameters.

2. Properties of Cartesian Products of Relational Structures

Let P, R be binary relations. The functor $P \times R$ yielding a binary relation is defined by:

(Def. 1) For all sets x, y holds $\langle x, y\rangle \in P \times R$ iff there exist sets p, q, s, t such that $x = \langle p, q\rangle$ and $y = \langle s, t\rangle$ and $\langle p, s\rangle \in P$ and $\langle q, t\rangle \in R$.

One can prove the following proposition

(10) Let P, R be binary relations and let x be a set. Then $x \in P \times R$ if and only if the following conditions are satisfied:

- (i) $\langle (x_{\mathbf{1}})_{\mathbf{1}}, (x_{\mathbf{2}})_{\mathbf{1}}\rangle \in P$,
- (ii) $\langle (x_{\mathbf{1}})_{\mathbf{2}}, (x_{\mathbf{2}})_{\mathbf{2}}\rangle \in R$,
- (iii) there exist sets a, b such that $x = \langle a, b\rangle$,
- (iv) there exist sets c, d such that $x_{\mathbf{1}} = \langle c, d\rangle$, and
- (v) there exist sets e, f such that $x_{\mathbf{2}} = \langle e, f\rangle$.

Let A, B, X, Y be sets, let P be a relation between A and B, and let R be a relation between X and Y. Then $P \times R$ is a relation between $[\!: A, X :\!]$ and $[\!: B, Y :\!]$.

Let X, Y be relational structures. The functor $[\!: X, Y :\!]$ yielding a strict relational structure is defined by the conditions (Def. 2).

(Def. 2) (i) The carrier of $[\!: X, Y :\!] = [\!:$ the carrier of X, the carrier of $Y :\!]$, and

(ii) the internal relation of $[\!: X, Y :\!] =$ (the internal relation of X)$\times$(the internal relation of Y).

Let L_1, L_2 be relational structures and let D be a subset of the carrier of $[\!: L_1, L_2 :\!]$. Then $\pi_1(D)$ is a subset of L_1. Then $\pi_2(D)$ is a subset of L_2.

Let S_1, S_2 be relational structures, let D_1 be a subset of the carrier of S_1, and let D_2 be a subset of the carrier of S_2. Then $[\!: D_1, D_2 :\!]$ is a subset of $[\!: S_1, S_2 :\!]$.

Let S_1, S_2 be non empty relational structures, let x be an element of the carrier of S_1, and let y be an element of the carrier of S_2. Then $\langle x, y\rangle$ is an element of $[\!: S_1, S_2 :\!]$.

Let L_1, L_2 be non empty relational structures and let x be an element of the carrier of $[\!: L_1, L_2 :\!]$. Then $x_{\mathbf{1}}$ is an element of L_1. Then $x_{\mathbf{2}}$ is an element of L_2.

The following three propositions are true:

(11) Let S_1, S_2 be non empty relational structures, and let a, c be elements of S_1, and let b, d be elements of S_2. Then $a \leq c$ and $b \leq d$ if and only if $\langle a, b\rangle \leq \langle c, d\rangle$.

(12) Let S_1, S_2 be non empty relational structures and let x, y be elements of $[: S_1, S_2 :]$. Then $x \leq y$ if and only if the following conditions are satisfied:

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

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

(13) Let A, B be relational structures, and let C be a non empty relational structure, and let f, g be maps from $[: A, B :]$ into C. Suppose that for every element x of A and for every element y of B holds $f(\langle x, y\rangle) = g(\langle x, y\rangle)$. Then $f = g$.

Let X, Y be non empty relational structures. Note that $[: X, Y :]$ is non empty.

Let X, Y be reflexive relational structures. Note that $[: X, Y :]$ is reflexive.

Let X, Y be antisymmetric relational structures. Note that $[: X, Y :]$ is antisymmetric.

Let X, Y be transitive relational structures. One can verify that $[: X, Y :]$ is transitive.

Let X, Y be relational structures with l.u.b.'s. One can verify that $[: X, Y :]$ has l.u.b.'s.

Let X, Y be relational structures with g.l.b.'s. One can verify that $[: X, Y :]$ has g.l.b.'s.

The following propositions are true:

(14) For all relational structures X, Y such that $[: X, Y :]$ is non empty holds X is non empty and Y is non empty.

(15) For all non empty relational structures X, Y such that $[: X, Y :]$ is reflexive holds X is reflexive and Y is reflexive.

(16) Let X, Y be non empty reflexive relational structures. If $[: X, Y :]$ is antisymmetric, then X is antisymmetric and Y is antisymmetric.

(17) Let X, Y be non empty reflexive relational structures. If $[: X, Y :]$ is transitive, then X is transitive and Y is transitive.

(18) For all non empty reflexive relational structures X, Y such that $[: X, Y :]$ has l.u.b.'s holds X has l.u.b.'s and Y has l.u.b.'s.

(19) For all non empty reflexive relational structures X, Y such that $[: X, Y :]$ has g.l.b.'s holds X has g.l.b.'s and Y has g.l.b.'s.

Let S_1, S_2 be relational structures, let D_1 be a directed subset of S_1, and let D_2 be a directed subset of S_2. Then $[: D_1, D_2 :]$ is a directed subset of $[: S_1, S_2 :]$.

We now state three propositions:

(20) Let S_1, S_2 be non empty relational structures, and let D_1 be a non empty subset of S_1, and let D_2 be a non empty subset of S_2. If $[: D_1, D_2 :]$ is directed, then D_1 is directed and D_2 is directed.

(21) For all non empty relational structures S_1, S_2 and for every non empty subset D of $[: S_1, S_2 :]$ holds $\pi_1(D)$ is non empty and $\pi_2(D)$ is non empty.

(22) Let S_1, S_2 be non empty reflexive relational structures and let D be a non empty directed subset of $[\!: S_1, S_2 :\!]$. Then $\pi_1(D)$ is directed and $\pi_2(D)$ is directed.

Let S_1, S_2 be relational structures, let D_1 be a filtered subset of S_1, and let D_2 be a filtered subset of S_2. Then $[\!: D_1, D_2 :\!]$ is a filtered subset of $[\!: S_1, S_2 :\!]$.

Next we state two propositions:

(23) Let S_1, S_2 be non empty relational structures, and let D_1 be a non empty subset of S_1, and let D_2 be a non empty subset of S_2. If $[\!: D_1, D_2 :\!]$ is filtered, then D_1 is filtered and D_2 is filtered.

(24) Let S_1, S_2 be non empty reflexive relational structures and let D be a non empty filtered subset of $[\!: S_1, S_2 :\!]$. Then $\pi_1(D)$ is filtered and $\pi_2(D)$ is filtered.

Let S_1, S_2 be relational structures, let D_1 be an upper subset of S_1, and let D_2 be an upper subset of S_2. Then $[\!: D_1, D_2 :\!]$ is an upper subset of $[\!: S_1, S_2 :\!]$.

We now state two propositions:

(25) Let S_1, S_2 be non empty reflexive relational structures, and let D_1 be a non empty subset of S_1, and let D_2 be a non empty subset of S_2. If $[\!: D_1, D_2 :\!]$ is upper, then D_1 is upper and D_2 is upper.

(26) Let S_1, S_2 be non empty reflexive relational structures and let D be a non empty upper subset of $[\!: S_1, S_2 :\!]$. Then $\pi_1(D)$ is upper and $\pi_2(D)$ is upper.

Let S_1, S_2 be relational structures, let D_1 be a lower subset of S_1, and let D_2 be a lower subset of S_2. Then $[\!: D_1, D_2 :\!]$ is a lower subset of $[\!: S_1, S_2 :\!]$.

Next we state two propositions:

(27) Let S_1, S_2 be non empty reflexive relational structures, and let D_1 be a non empty subset of S_1, and let D_2 be a non empty subset of S_2. If $[\!: D_1, D_2 :\!]$ is lower, then D_1 is lower and D_2 is lower.

(28) Let S_1, S_2 be non empty reflexive relational structures and let D be a non empty lower subset of $[\!: S_1, S_2 :\!]$. Then $\pi_1(D)$ is lower and $\pi_2(D)$ is lower.

Let R be a relational structure. We say that R is void if and only if:

(Def. 3) The internal relation of R is empty.

Let us observe that every relational structure which is empty is also void.

Let us note that there exists a poset which is non void, non empty, and strict.

One can check that every relational structure which is non void is also non empty.

Let us observe that every relational structure which is non empty and reflexive is also non void.

Let R be a non void relational structure. One can check that the internal relation of R is non empty.

Next we state a number of propositions:

(29) Let S_1, S_2 be non empty relational structures, and let D_1 be a non empty subset of S_1, and let D_2 be a non empty subset of S_2, and let x be

an element of S_1, and let y be an element of S_2. If $\langle x, y\rangle \geq [: D_1, D_2 :]$, then $x \geq D_1$ and $y \geq D_2$.

(30) Let S_1, S_2 be non empty relational structures, and let D_1 be a subset of S_1, and let D_2 be a subset of S_2, and let x be an element of S_1, and let y be an element of S_2. If $x \geq D_1$ and $y \geq D_2$, then $\langle x, y\rangle \geq [: D_1, D_2 :]$.

(31) Let S_1, S_2 be non empty relational structures, and let D be a subset of $[: S_1, S_2 :]$, and let x be an element of S_1, and let y be an element of S_2. Then $\langle x, y\rangle \geq D$ if and only if $x \geq \pi_1(D)$ and $y \geq \pi_2(D)$.

(32) Let S_1, S_2 be non empty relational structures, and let D_1 be a non empty subset of S_1, and let D_2 be a non empty subset of S_2, and let x be an element of S_1, and let y be an element of S_2. If $\langle x, y\rangle \leq [: D_1, D_2 :]$, then $x \leq D_1$ and $y \leq D_2$.

(33) Let S_1, S_2 be non empty relational structures, and let D_1 be a subset of S_1, and let D_2 be a subset of S_2, and let x be an element of S_1, and let y be an element of S_2. If $x \leq D_1$ and $y \leq D_2$, then $\langle x, y\rangle \leq [: D_1, D_2 :]$.

(34) Let S_1, S_2 be non empty relational structures, and let D be a subset of $[: S_1, S_2 :]$, and let x be an element of S_1, and let y be an element of S_2. Then $\langle x, y\rangle \leq D$ if and only if $x \leq \pi_1(D)$ and $y \leq \pi_2(D)$.

(35) Let S_1, S_2 be antisymmetric non empty relational structures, and let D_1 be a subset of S_1, and let D_2 be a subset of S_2, and let x be an element of S_1, and let y be an element of S_2. Suppose sup D_1 exists in S_1 and sup D_2 exists in S_2 and for every element b of $[: S_1, S_2 :]$ such that $b \geq [: D_1, D_2 :]$ holds $\langle x, y\rangle \leq b$. Then for every element c of S_1 such that $c \geq D_1$ holds $x \leq c$ and for every element d of S_2 such that $d \geq D_2$ holds $y \leq d$.

(36) Let S_1, S_2 be antisymmetric non empty relational structures, and let D_1 be a subset of S_1, and let D_2 be a subset of S_2, and let x be an element of S_1, and let y be an element of S_2. Suppose inf D_1 exists in S_1 and inf D_2 exists in S_2 and for every element b of $[: S_1, S_2 :]$ such that $b \leq [: D_1, D_2 :]$ holds $\langle x, y\rangle \geq b$. Then for every element c of S_1 such that $c \leq D_1$ holds $x \geq c$ and for every element d of S_2 such that $d \leq D_2$ holds $y \geq d$.

(37) Let S_1, S_2 be antisymmetric non empty relational structures, and let D_1 be a non empty subset of S_1, and let D_2 be a non empty subset of S_2, and let x be an element of S_1, and let y be an element of S_2. Suppose for every element c of S_1 such that $c \geq D_1$ holds $x \leq c$ and for every element d of S_2 such that $d \geq D_2$ holds $y \leq d$. Let b be an element of $[: S_1, S_2 :]$. If $b \geq [: D_1, D_2 :]$, then $\langle x, y\rangle \leq b$.

(38) Let S_1, S_2 be antisymmetric non empty relational structures, and let D_1 be a non empty subset of S_1, and let D_2 be a non empty subset of S_2, and let x be an element of S_1, and let y be an element of S_2. Suppose for every element c of S_1 such that $c \geq D_1$ holds $x \geq c$ and for every element d of S_2 such that $d \geq D_2$ holds $y \geq d$. Let b be an element of $[: S_1, S_2 :]$. If $b \geq [: D_1, D_2 :]$, then $\langle x, y\rangle \geq b$.

(39) Let S_1, S_2 be antisymmetric non empty relational structures, and let

D_1 be a non empty subset of S_1, and let D_2 be a non empty subset of S_2. Then $\sup D_1$ exists in S_1 and $\sup D_2$ exists in S_2 if and only if $\sup [\!:D_1, D_2:\!]$ exists in $[\!:S_1, S_2:\!]$.

(40) Let S_1, S_2 be antisymmetric non empty relational structures, and let D_1 be a non empty subset of S_1, and let D_2 be a non empty subset of S_2. Then $\inf D_1$ exists in S_1 and $\inf D_2$ exists in S_2 if and only if $\inf [\!:D_1, D_2:\!]$ exists in $[\!:S_1, S_2:\!]$.

(41) Let S_1, S_2 be antisymmetric non empty relational structures and let D be a subset of $[\!:S_1, S_2:\!]$. Then $\sup \pi_1(D)$ exists in S_1 and $\sup \pi_2(D)$ exists in S_2 if and only if $\sup D$ exists in $[\!:S_1, S_2:\!]$.

(42) Let S_1, S_2 be antisymmetric non empty relational structures and let D be a subset of $[\!:S_1, S_2:\!]$. Then $\inf \pi_1(D)$ exists in S_1 and $\inf \pi_2(D)$ exists in S_2 if and only if $\inf D$ exists in $[\!:S_1, S_2:\!]$.

(43) Let S_1, S_2 be antisymmetric non empty relational structures, and let D_1 be a non empty subset of S_1, and let D_2 be a non empty subset of S_2. If $\sup D_1$ exists in S_1 and $\sup D_2$ exists in S_2, then $\sup[\!:D_1, D_2:\!] = \langle \sup D_1, \sup D_2 \rangle$.

(44) Let S_1, S_2 be antisymmetric non empty relational structures, and let D_1 be a non empty subset of S_1, and let D_2 be a non empty subset of S_2. If $\inf D_1$ exists in S_1 and $\inf D_2$ exists in S_2, then $\inf[\!:D_1, D_2:\!] = \langle \inf D_1, \inf D_2 \rangle$.

Let X, Y be complete antisymmetric non empty relational structures. Observe that $[\!:X, Y:\!]$ is complete.

We now state several propositions:

(45) Let X, Y be non empty lower-bounded antisymmetric relational structures. If $[\!:X, Y:\!]$ is complete, then X is complete and Y is complete.

(46) Let L_1, L_2 be antisymmetric non empty relational structures and let D be a non empty subset of $[\!:L_1, L_2:\!]$. If $[\!:L_1, L_2:\!]$ is complete or $\sup D$ exists in $[\!:L_1, L_2:\!]$, then $\sup D = \langle \sup \pi_1(D), \sup \pi_2(D) \rangle$.

(47) Let L_1, L_2 be antisymmetric non empty relational structures and let D be a non empty subset of $[\!:L_1, L_2:\!]$. If $[\!:L_1, L_2:\!]$ is complete or $\inf D$ exists in $[\!:L_1, L_2:\!]$, then $\inf D = \langle \inf \pi_1(D), \inf \pi_2(D) \rangle$.

(48) For all non empty reflexive relational structures S_1, S_2 and for every non empty directed subset D of $[\!:S_1, S_2:\!]$ holds $[\!:\pi_1(D), \pi_2(D):\!] \subseteq {\downarrow}D$.

(49) For all non empty reflexive relational structures S_1, S_2 and for every non empty filtered subset D of $[\!:S_1, S_2:\!]$ holds $[\!:\pi_1(D), \pi_2(D):\!] \subseteq {\uparrow}D$.

The scheme *Kappa2DS* concerns non empty relational structures $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and a binary functor $\mathcal{F}$ yielding a set, and states that:

There exists a map f from $[\!:\mathcal{A}, \mathcal{B}:\!]$ into $\mathcal{C}$ such that for every element x of $\mathcal{A}$ and for every element y of $\mathcal{B}$ holds $f(\langle x, y \rangle) = \mathcal{F}(x, y)$

provided the following requirement is met:

- For every element x of $\mathcal{A}$ and for every element y of $\mathcal{B}$ holds $\mathcal{F}(x, y)$ is an element of $\mathcal{C}$.

References

[1] Grzegorz Bancerek. Bounds in posets and relational substructures. *Formalized Mathematics*, 6(**1**):81–91, 1997.
[2] Grzegorz Bancerek. Complete lattices. *Formalized Mathematics*, 2(**5**):719–725, 1991.
[3] Grzegorz Bancerek. Curried and uncurried functions. *Formalized Mathematics*, 1(**3**):537–541, 1990.
[4] Grzegorz Bancerek. Directed sets, nets, ideals, filters, and maps. *Formalized Mathematics*, 6(**1**):93–107, 1997.
[5] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[6] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[7] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[8] 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.
[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] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[12] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[13] Wojciech A. Trybulec. Partially ordered sets. *Formalized Mathematics*, 1(**2**):313–319, 1990.
[14] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[15] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.
[16] Edmund Woronowicz and Anna Zalewska. Properties of binary relations. *Formalized Mathematics*, 1(**1**):85–89, 1990.
[17] Mariusz Żynel and Czesław Byliński. Properties of relational structures, posets, lattices and maps. *Formalized Mathematics*, 6(**1**):123–130, 1997.

Received September 25, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Definitions and Properties of the Join and Meet of Subsets [1]

Artur Korniłowicz
Warsaw University
Białystok

Summary. This paper is the continuation of formalization of [6]. The definitions of meet and join of subsets of relational structures are introduced. The properties of these notions are proved.

MML Identifier: `YELLOW_4`.

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

1. Preliminaries

The following propositions are true:

(1) Let L be a non empty relational structure, X be a set, and a be an element of L. If $a \in X$ and sup X exists in L, then $a \leq \bigsqcup_L X$.

(2) Let L be a non empty relational structure, X be a set, and a be an element of L. If $a \in X$ and inf X exists in L, then $\bigsqcap_L X \leq a$.

Let L be a relational structure and let A, B be subsets of the carrier of L. We say that A is finer than B if and only if:

(Def. 1) For every element a of L such that $a \in A$ there exists an element b of L such that $b \in B$ and $a \leq b$.

We say that B is coarser than A if and only if:

(Def. 2) For every element b of L such that $b \in B$ there exists an element a of L such that $a \in A$ and $a \leq b$.

[1]This work was partially supported by Office of Naval Research Grant N00014-95-1-1336.

ISSN 1426–2630

Let us note that in the case when the relational structure L is reflexive and non empty, both predicates defined above are reflexive.

Next we state several propositions:

(3) For every relational structure L and for every subset B of L holds $\emptyset_L$ is finer than B.

(4) Let L be a transitive relational structure and A, B, C be subsets of L. If A is finer than B and B is finer than C, then A is finer than C.

(5) For every relational structure L and for all subsets A, B of L such that B is finer than A and A is lower holds $B \subseteq A$.

(6) For every relational structure L and for every subset A of L holds $\emptyset_L$ is coarser than A.

(7) Let L be a transitive relational structure and A, B, C be subsets of L. If C is coarser than B and B is coarser than A, then C is coarser than A.

(8) Let L be a relational structure and A, B be subsets of L. If A is coarser than B and B is upper, then $A \subseteq B$.

2. The Join of Subsets

Let L be a non empty relational structure and let D_1, D_2 be subsets of the carrier of L. The functor $D_1 \sqcup D_2$ yielding a subset of L is defined by:

(Def. 3) $D_1 \sqcup D_2 = \{x \sqcup y : x$ ranges over elements of L, y ranges over elements of L, $x \in D_1 \ \wedge \ y \in D_2\}$.

Let L be an antisymmetric relational structure with l.u.b.'s and let D_1, D_2 be subsets of the carrier of L. Let us note that the functor $D_1 \sqcup D_2$ is commutative.

One can prove the following propositions:

(9) For every non empty relational structure L and for every subset X of L holds $X \sqcup \emptyset_L = \emptyset$.

(10) Let L be a non empty relational structure, X, Y be subsets of L, and x, y be elements of L. If $x \in X$ and $y \in Y$, then $x \sqcup y \in X \sqcup Y$.

(11) Let L be an antisymmetric relational structure with l.u.b.'s, A be a subset of L, and B be a non empty subset of L. Then A is finer than $A \sqcup B$.

(12) For every antisymmetric relational structure L with l.u.b.'s and for all subsets A, B of L holds $A \sqcup B$ is coarser than A.

(13) For every antisymmetric reflexive relational structure L with l.u.b.'s and for every subset A of L holds $A \subseteq A \sqcup A$.

(14) Let L be an antisymmetric transitive relational structure with l.u.b.'s and D_1, D_2, D_3 be subsets of L. Then $(D_1 \sqcup D_2) \sqcup D_3 = D_1 \sqcup (D_2 \sqcup D_3)$.

Let L be a non empty relational structure and let D_1, D_2 be non empty subsets of the carrier of L. Note that $D_1 \sqcup D_2$ is non empty.

Let L be a transitive antisymmetric relational structure with l.u.b.'s and let D_1, D_2 be directed subsets of L. Note that $D_1 \sqcup D_2$ is directed.

Let L be a transitive antisymmetric relational structure with l.u.b.'s and let D_1, D_2 be filtered subsets of L. Note that $D_1 \sqcup D_2$ is filtered.

Let L be a poset with l.u.b.'s and let D_1, D_2 be upper subsets of L. Observe that $D_1 \sqcup D_2$ is upper.

We now state a number of propositions:

(15) Let L be a non empty relational structure, Y be a subset of L, and x be an element of L. Then $\{x\} \sqcup Y = \{x \sqcup y : y$ ranges over elements of L, $y \in Y\}$.

(16) For every non empty relational structure L and for all subsets A, B, C of L holds $A \sqcup (B \cup C) = (A \sqcup B) \cup (A \sqcup C)$.

(17) Let L be an antisymmetric reflexive relational structure with l.u.b.'s and A, B, C be subsets of L. Then $A \cup (B \sqcup C) \subseteq (A \cup B) \sqcup (A \cup C)$.

(18) Let L be an antisymmetric relational structure with l.u.b.'s, A be an upper subset of L, and B, C be subsets of L. Then $(A \cup B) \sqcup (A \cup C) \subseteq A \cup (B \sqcup C)$.

(19) For every non empty relational structure L and for all elements x, y of L holds $\{x\} \sqcup \{y\} = \{x \sqcup y\}$.

(20) For every non empty relational structure L and for all elements x, y, z of L holds $\{x\} \sqcup \{y, z\} = \{x \sqcup y, x \sqcup z\}$.

(21) For every non empty relational structure L and for all subsets X_1, X_2, Y_1, Y_2 of L such that $X_1 \subseteq Y_1$ and $X_2 \subseteq Y_2$ holds $X_1 \sqcup X_2 \subseteq Y_1 \sqcup Y_2$.

(22) Let L be a reflexive antisymmetric relational structure with l.u.b.'s, D be a subset of L, and x be an element of L. If $x \leq D$, then $\{x\} \sqcup D = D$.

(23) Let L be an antisymmetric relational structure with l.u.b.'s, D be a subset of L, and x be an element of L. Then $x \leq \{x\} \sqcup D$.

(24) Let L be a poset with l.u.b.'s, X be a subset of L, and x be an element of L. If $\inf \{x\} \sqcup X$ exists in L and $\inf X$ exists in L, then $x \sqcup \inf X \leq \inf(\{x\} \sqcup X)$.

(25) Let L be a complete transitive antisymmetric non empty relational structure, A be a subset of L, and B be a non empty subset of L. Then $A \leq \sup(A \sqcup B)$.

(26) Let L be a transitive antisymmetric relational structure with l.u.b.'s, a, b be elements of L, and A, B be subsets of L. If $a \leq A$ and $b \leq B$, then $a \sqcup b \leq A \sqcup B$.

(27) Let L be a transitive antisymmetric relational structure with l.u.b.'s, a, b be elements of L, and A, B be subsets of L. If $a \geq A$ and $b \geq B$, then $a \sqcup b \geq A \sqcup B$.

(28) For every complete non empty poset L and for all non empty subsets A, B of L holds $\sup(A \sqcup B) = \sup A \sqcup \sup B$.

(29) Let L be an antisymmetric relational structure with l.u.b.'s, X be a

subset of L, and Y be a non empty subset of L. Then $X \subseteq \downarrow(X \sqcup Y)$.

(30) Let L be a poset with l.u.b.'s, x, y be elements of $\langle \mathrm{Ids}(L), \subseteq \rangle$, and X, Y be subsets of L. If $x = X$ and $y = Y$, then $x \sqcup y = \downarrow(X \sqcup Y)$.

(31) Let L be a non empty relational structure and D be a subset of $[: L, L :]$. Then $\bigcup\{X : X$ ranges over subsets of L, $\bigvee_{x\,:\,\mathrm{element\ of}\ L}\ X = \{x\} \sqcup \pi_2(D)\ \wedge\ x \in \pi_1(D)\} = \pi_1(D) \sqcup \pi_2(D)$.

(32) Let L be a transitive antisymmetric relational structure with l.u.b.'s and D_1, D_2 be subsets of L. Then $\downarrow(\downarrow D_1 \sqcup \downarrow D_2) \subseteq \downarrow(D_1 \sqcup D_2)$.

(33) For every poset L with l.u.b.'s and for all subsets D_1, D_2 of L holds $\downarrow(\downarrow D_1 \sqcup \downarrow D_2) = \downarrow(D_1 \sqcup D_2)$.

(34) Let L be a transitive antisymmetric relational structure with l.u.b.'s and D_1, D_2 be subsets of L. Then $\uparrow(\uparrow D_1 \sqcup \uparrow D_2) \subseteq \uparrow(D_1 \sqcup D_2)$.

(35) For every poset L with l.u.b.'s and for all subsets D_1, D_2 of L holds $\uparrow(\uparrow D_1 \sqcup \uparrow D_2) = \uparrow(D_1 \sqcup D_2)$.

3. The Meet of Subsets

Let L be a non empty relational structure and let D_1, D_2 be subsets of the carrier of L. The functor $D_1 \sqcap D_2$ yields a subset of L and is defined by:

(Def. 4) $D_1 \sqcap D_2 = \{x \sqcap y : x$ ranges over elements of L, y ranges over elements of L, $x \in D_1\ \wedge\ y \in D_2\}$.

Let L be an antisymmetric relational structure with g.l.b.'s and let D_1, D_2 be subsets of the carrier of L. Let us notice that the functor $D_1 \sqcap D_2$ is commutative.

Next we state several propositions:

(36) For every non empty relational structure L and for every subset X of L holds $X \sqcap \emptyset_L = \emptyset$.

(37) Let L be a non empty relational structure, X, Y be subsets of L, and x, y be elements of L. If $x \in X$ and $y \in Y$, then $x \sqcap y \in X \sqcap Y$.

(38) Let L be an antisymmetric relational structure with g.l.b.'s, A be a subset of L, and B be a non empty subset of L. Then A is coarser than $A \sqcap B$.

(39) For every antisymmetric relational structure L with g.l.b.'s and for all subsets A, B of L holds $A \sqcap B$ is finer than A.

(40) For every antisymmetric reflexive relational structure L with g.l.b.'s and for every subset A of L holds $A \subseteq A \sqcap A$.

(41) Let L be an antisymmetric transitive relational structure with g.l.b.'s and D_1, D_2, D_3 be subsets of L. Then $(D_1 \sqcap D_2) \sqcap D_3 = D_1 \sqcap (D_2 \sqcap D_3)$.

Let L be a non empty relational structure and let D_1, D_2 be non empty subsets of the carrier of L. Observe that $D_1 \sqcap D_2$ is non empty.

Let L be a transitive antisymmetric relational structure with g.l.b.'s and let D_1, D_2 be directed subsets of L. One can check that $D_1 \sqcap D_2$ is directed.

Let L be a transitive antisymmetric relational structure with g.l.b.'s and let D_1, D_2 be filtered subsets of L. One can check that $D_1 \sqcap D_2$ is filtered.

Let L be a semilattice and let D_1, D_2 be lower subsets of L. One can verify that $D_1 \sqcap D_2$ is lower.

One can prove the following propositions:

(42) Let L be a non empty relational structure, Y be a subset of L, and x be an element of L. Then $\{x\} \sqcap Y = \{x \sqcap y : y$ ranges over elements of L, $y \in Y\}$.

(43) For every non empty relational structure L and for all subsets A, B, C of L holds $A \sqcap (B \cup C) = A \sqcap B \cup A \sqcap C$.

(44) Let L be an antisymmetric reflexive relational structure with g.l.b.'s and A, B, C be subsets of L. Then $A \cup B \sqcap C \subseteq (A \cup B) \sqcap (A \cup C)$.

(45) Let L be an antisymmetric relational structure with g.l.b.'s, A be a lower subset of L, and B, C be subsets of L. Then $(A \cup B) \sqcap (A \cup C) \subseteq A \cup B \sqcap C$.

(46) For every non empty relational structure L and for all elements x, y of L holds $\{x\} \sqcap \{y\} = \{x \sqcap y\}$.

(47) For every non empty relational structure L and for all elements x, y, z of L holds $\{x\} \sqcap \{y, z\} = \{x \sqcap y, x \sqcap z\}$.

(48) For every non empty relational structure L and for all subsets X_1, X_2, Y_1, Y_2 of L such that $X_1 \subseteq Y_1$ and $X_2 \subseteq Y_2$ holds $X_1 \sqcap X_2 \subseteq Y_1 \sqcap Y_2$.

(49) For every antisymmetric reflexive relational structure L with g.l.b.'s and for all subsets A, B of L holds $A \cap B \subseteq A \sqcap B$.

(50) Let L be an antisymmetric reflexive relational structure with g.l.b.'s and A, B be lower subsets of L. Then $A \sqcap B = A \cap B$.

(51) Let L be a reflexive antisymmetric relational structure with g.l.b.'s, D be a subset of L, and x be an element of L. If $x \geq D$, then $\{x\} \sqcap D = D$.

(52) Let L be an antisymmetric relational structure with g.l.b.'s, D be a subset of L, and x be an element of L. Then $\{x\} \sqcap D \leq x$.

(53) Let L be a semilattice, X be a subset of L, and x be an element of L. If $\sup \{x\} \sqcap X$ exists in L and $\sup X$ exists in L, then $\sup(\{x\} \sqcap X) \leq x \sqcap \sup X$.

(54) Let L be a complete transitive antisymmetric non empty relational structure, A be a subset of L, and B be a non empty subset of L. Then $A \geq \inf(A \sqcap B)$.

(55) Let L be a transitive antisymmetric relational structure with g.l.b.'s, a, b be elements of L, and A, B be subsets of L. If $a \leq A$ and $b \leq B$, then $a \sqcap b \leq A \sqcap B$.

(56) Let L be a transitive antisymmetric relational structure with g.l.b.'s, a, b be elements of L, and A, B be subsets of L. If $a \geq A$ and $b \geq B$, then $a \sqcap b \geq A \sqcap B$.

(57) For every complete non empty poset L and for all non empty subsets A, B of L holds $\inf(A \sqcap B) = \inf A \sqcap \inf B$.

(58) Let L be a semilattice, x, y be elements of $\langle \mathrm{Ids}(L), \subseteq \rangle$, and x_1, y_1 be subsets of L. If $x = x_1$ and $y = y_1$, then $x \sqcap y = x_1 \sqcap y_1$.

(59) Let L be an antisymmetric relational structure with g.l.b.'s, X be a subset of L, and Y be a non empty subset of L. Then $X \subseteq \uparrow(X \sqcap Y)$.

(60) Let L be a non empty relational structure and D be a subset of $[\!: L, L :\!]$. Then $\bigcup\{X : X$ ranges over subsets of L, $\bigvee_{x\,:\,\text{element of } L}\; X = \{x\} \sqcap \pi_2(D) \;\wedge\; x \in \pi_1(D)\} = \pi_1(D) \sqcap \pi_2(D)$.

(61) Let L be a transitive antisymmetric relational structure with g.l.b.'s and D_1, D_2 be subsets of L. Then $\downarrow(\downarrow D_1 \sqcap \downarrow D_2) \subseteq \downarrow(D_1 \sqcap D_2)$.

(62) For every semilattice L and for all subsets D_1, D_2 of L holds $\downarrow(\downarrow D_1 \sqcap \downarrow D_2) = \downarrow(D_1 \sqcap D_2)$.

(63) Let L be a transitive antisymmetric relational structure with g.l.b.'s and D_1, D_2 be subsets of L. Then $\uparrow(\uparrow D_1 \sqcap \uparrow D_2) \subseteq \uparrow(D_1 \sqcap D_2)$.

(64) For every semilattice L and for all subsets D_1, D_2 of L holds $\uparrow(\uparrow D_1 \sqcap \uparrow D_2) = \uparrow(D_1 \sqcap D_2)$.

References

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

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

[3] Grzegorz Bancerek. Curried and uncurried functions. *Formalized Mathematics*, 1(**3**):537–541, 1990.

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

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

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

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

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

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

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

[13] Mariusz Żynel and Czesław Byliński. Properties of relational structures, posets, lattices and maps. *Formalized Mathematics*, 6(**1**):123–130, 1997.

Received September 25, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

Meet – Continuous Lattices [1]

Artur Korniłowicz
Warsaw University
Białystok

Summary. The aim of this work is the formalization of Chapter 0 Section 4 of [11]. In this paper the definition of meet-continuous lattices is introduced. Theorem 4.2 and Remark 4.3 are proved.

MML Identifier: `WAYBEL_2`.

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

1. Preliminaries

Let X, Y be non empty sets, let f be a function from X into Y, and let Z be a non empty subset of X. One can verify that $f^\circ Z$ is non empty.

One can check that every non empty relational structure which is reflexive and connected has g.l.b.'s and l.u.b.'s.

Let C be a chain. One can verify that Ω_C is directed.

Let X be a set. Note that every binary relation on X which is ordering is also reflexive, antisymmetric, and transitive.

Let X be a non empty set. One can verify that there exists a binary relation on X which is ordering.

The following propositions are true:

(1) Let L be an up-complete semilattice, and let D be a non empty directed subset of L, and let x be an element of L. Then $\sup \{x\} \sqcap D$ exists in L.

(2) Let L be an up-complete sup-semilattice, and let D be a non empty directed subset of L, and let x be an element of L. Then $\sup \{x\} \sqcup D$ exists in L.

[1]This work was partially supported by Office of Naval Research Grant N00014-95-1-1336.

ISSN 1426–2630

(3) For every up-complete sup-semilattice L and for all non empty directed subsets A, B of L holds $A \leq \sup(A \sqcup B)$.

(4) For every up-complete sup-semilattice L and for all non empty directed subsets A, B of L holds $\sup(A \sqcup B) = \sup A \sqcup \sup B$.

(5) Let L be an up-complete semilattice and let D be a non empty directed subset of $[\!: L, L :\!]$. Then $\{\sup(\{x\} \sqcap \pi_2(D)) : x$ ranges over elements of L, $x \in \pi_1(D)\} = \{\sup X : X$ ranges over non empty directed subsets of L, $\bigvee_{x\,:\,\text{element of } L}\ X = \{x\} \sqcap \pi_2(D)\ \wedge\ x \in \pi_1(D)\}$.

(6) Let L be a semilattice and let D be a non empty directed subset of $[\!: L, L :\!]$. Then $\bigcup\{X : X$ ranges over non empty directed subsets of L, $\bigvee_{x\,:\,\text{element of } L}\ X = \{x\} \sqcap \pi_2(D)\ \wedge\ x \in \pi_1(D)\} = \pi_1(D) \sqcap \pi_2(D)$.

(7) Let L be an up-complete semilattice and let D be a non empty directed subset of $[\!: L, L :\!]$. Then $\sup \bigcup\{X : X$ ranges over non empty directed subsets of L, $\bigvee_{x\,:\,\text{element of } L}\ X = \{x\} \sqcap \pi_2(D)\ \wedge\ x \in \pi_1(D)\}$ exists in L.

(8) Let L be an up-complete semilattice and let D be a non empty directed subset of $[\!: L, L :\!]$. Then $\sup \{\sup X : X$ ranges over non empty directed subsets of L, $\bigvee_{x\,:\,\text{element of } L}\ X = \{x\} \sqcap \pi_2(D)\ \wedge\ x \in \pi_1(D)\}$ exists in L.

(9) Let L be an up-complete semilattice and let D be a non empty directed subset of $[\!: L, L :\!]$. Then $\bigsqcup_L\{\sup X : X$ ranges over non empty directed subsets of L, $\bigvee_{x\,:\,\text{element of } L}\ X = \{x\} \sqcap \pi_2(D)\ \wedge\ x \in \pi_1(D)\} \leq \bigsqcup_L \bigcup\{X :$ X ranges over non empty directed subsets of L, $\bigvee_{x\,:\,\text{element of } L}\ X = \{x\} \sqcap \pi_2(D)\ \wedge\ x \in \pi_1(D)\}$.

(10) Let L be an up-complete semilattice and let D be a non empty directed subset of $[\!: L, L :\!]$. Then $\bigsqcup_L\{\sup X : X$ ranges over non empty directed subsets of L, $\bigvee_{x\,:\,\text{element of } L}\ X = \{x\} \sqcap \pi_2(D)\ \wedge\ x \in \pi_1(D)\} = \bigsqcup_L \bigcup\{X :$ X ranges over non empty directed subsets of L, $\bigvee_{x\,:\,\text{element of } L}\ X = \{x\} \sqcap \pi_2(D)\ \wedge\ x \in \pi_1(D)\}$.

Let S, T be up-complete non empty reflexive relational structures. One can verify that $[\!: S, T :\!]$ is up-complete.

The following four propositions are true:

(11) Let S, T be non empty reflexive antisymmetric relational structures. If $[\!: S, T :\!]$ is up-complete, then S is up-complete and T is up-complete.

(12) Let L be an up-complete antisymmetric non empty reflexive relational structure and let D be a non empty directed subset of $[\!: L, L :\!]$. Then $\sup D = \langle \sup \pi_1(D), \sup \pi_2(D) \rangle$.

(13) Let S_1, S_2 be non empty relational structures, and let D be a subset of S_1, and let f be a map from S_1 into S_2. If f is monotone, then $f^\circ{\downarrow}D \subseteq {\downarrow}(f^\circ D)$.

(14) Let S_1, S_2 be non empty relational structures, and let D be a subset of S_1, and let f be a map from S_1 into S_2. If f is monotone, then $f^\circ{\uparrow}D \subseteq {\uparrow}(f^\circ D)$.

Let us observe that every non empty reflexive relational structure which is trivial is also distributive and complemented.

Let us note that there exists a lattice which is strict, non empty, and trivial.

One can prove the following three propositions:

(15) Let H be a distributive complete lattice, and let a be an element of H, and let X be a finite subset of H. Then $\sup(\{a\} \sqcap X) = a \sqcap \sup X$.

(16) Let H be a distributive complete lattice, and let a be an element of H, and let X be a finite subset of H. Then $\inf(\{a\} \sqcup X) = a \sqcup \inf X$.

(17) Let H be a complete distributive lattice, and let a be an element of H, and let X be a finite subset of H. Then $a \sqcap \square$ preserves sup of X.

2. The properties of nets

The scheme *ExNet* concerns a non empty relational structure $\mathcal{A}$, a prenet $\mathcal{B}$ over $\mathcal{A}$, and a unary functor $\mathcal{F}$ yielding an element of the carrier of $\mathcal{A}$, and states that:

There exists a prenet M over $\mathcal{A}$ such that

(i) the relational structure of M = the relational structure of $\mathcal{B}$,

and

(ii) for every element i of the carrier of M holds (the mapping of $M)(i) = \mathcal{F}(($the mapping of $\mathcal{B})(i))$

for all values of the parameters.

The following three propositions are true:

(18) Let L be a non empty relational structure and let N be a prenet over L. If N is eventually-directed, then $\operatorname{rng} \operatorname{netmap}(N, L)$ is directed.

(19) Let L be a non empty reflexive relational structure, and let D be a non empty directed subset of L, and let n be a function from D into the carrier of L. Then $\langle D, (\text{the internal relation of } L) \mid^2(D), n\rangle$ is a prenet over L.

(20) Let L be a non empty reflexive relational structure, and let D be a non empty directed subset of L, and let n be a function from D into the carrier of L, and let N be a prenet over L. Suppose $n = \mathrm{id}_D$ and $N = \langle D, (\text{the internal relation of } L) \mid^2(D), n\rangle$. Then N is eventually-directed.

Let L be a non empty relational structure and let N be a net structure over L. The functor $\sup N$ yielding an element of L is defined by:

(Def. 1) $\sup N = \operatorname{Sup}(\text{the mapping of } N)$.

Let L be a non empty relational structure, let J be a set, and let f be a function from J into the carrier of L. The functor $\operatorname{FinSups}(f)$ yields a prenet over L and is defined by the condition (Def. 2).

(Def. 2) There exists a function g from $\operatorname{Fin} J$ into the carrier of L such that for every element x of $\operatorname{Fin} J$ holds $g(x) = \sup(f^\circ x)$ and $\operatorname{FinSups}(f) = \langle \operatorname{Fin} J, \subseteq_{\operatorname{Fin} J}, g\rangle$.

The following proposition is true

(21) Let L be a non empty relational structure, and let J, x be sets, and let f be a function from J into the carrier of L. Then x is an element of $\mathrm{FinSups}(f)$ if and only if x is an element of $\mathrm{Fin}\, J$.

Let L be a complete antisymmetric non empty reflexive relational structure, let J be a set, and let f be a function from J into the carrier of L. Note that $\mathrm{FinSups}(f)$ is monotone.

Let L be a non empty relational structure, let x be an element of L, and let N be a non empty net structure over L. The functor $x \sqcap N$ yielding a strict net structure over L is defined by the conditions (Def. 3).

(Def. 3) (i) The relational structure of $x \sqcap N$ = the relational structure of N, and

(ii) for every element i of the carrier of $x \sqcap N$ there exists an element y of L such that $y =$ (the mapping of N)(i) and (the mapping of $x \sqcap N$)$(i) = x \sqcap y$.

We now state the proposition

(22) Let L be a non empty relational structure, and let N be a non empty net structure over L, and let x be an element of L, and let y be a set. Then y is an element of N if and only if y is an element of $x \sqcap N$.

Let L be a non empty relational structure, let x be an element of L, and let N be a non empty net structure over L. Observe that $x \sqcap N$ is non empty.

Let L be a non empty relational structure, let x be an element of L, and let N be a prenet over L. Note that $x \sqcap N$ is directed.

Next we state several propositions:

(23) Let L be a non empty relational structure, and let x be an element of L, and let F be a non empty net structure over L. Then $\mathrm{rng}\,$(the mapping of $x \sqcap F$) $= \{x\} \sqcap \mathrm{rng}\,$(the mapping of F).

(24) Let L be a non empty relational structure, and let J be a set, and let f be a function from J into the carrier of L. If for every set x holds sup $f^\circ x$ exists in L, then $\mathrm{rng}\,\mathrm{netmap}(\mathrm{FinSups}(f), L) \subseteq \mathrm{finsups}(\mathrm{rng}\, f)$.

(25) Let L be a non empty reflexive antisymmetric relational structure, and let J be a set, and let f be a function from J into the carrier of L. Then $\mathrm{rng}\, f \subseteq \mathrm{rng}\,\mathrm{netmap}(\mathrm{FinSups}(f), L)$.

(26) Let L be a non empty reflexive antisymmetric relational structure, and let J be a set, and let f be a function from J into the carrier of L. Suppose sup $\mathrm{rng}\, f$ exists in L and sup $\mathrm{rng}\,\mathrm{netmap}(\mathrm{FinSups}(f), L)$ exists in L and for every element x of $\mathrm{Fin}\, J$ holds sup $f^\circ x$ exists in L. Then $\mathrm{Sup}(f) = \sup \mathrm{FinSups}(f)$.

(27) Let L be an antisymmetric transitive relational structure with g.l.b.'s, and let N be a prenet over L, and let x be an element of L. If N is eventually-directed, then $x \sqcap N$ is eventually-directed.

(28) Let L be an up-complete semilattice. Suppose that for every element x of L and for every non empty directed subset E of L such that $x \leq \sup E$ holds $x \leq \sup(\{x\} \sqcap E)$. Let D be a non empty directed subset of L and let x be an element of L. Then $x \sqcap \sup D = \sup(\{x\} \sqcap D)$.

(29) Let L be a poset with l.u.b.'s. Suppose that for every directed subset X of L and for every element x of L holds $x \sqcap \sup X = \sup(\{x\} \sqcap X)$. Let X be a subset of L and let x be an element of L. If sup X exists in L, then $x \sqcap \sup X = \sup(\{x\} \sqcap \mathrm{finsups}(X))$.

(30) Let L be an up-complete lattice. Suppose that for every subset X of L and for every element x of L holds $x \sqcap \sup X = \sup(\{x\} \sqcap \mathrm{finsups}(X))$. Let X be a non empty directed subset of L and let x be an element of L. Then $x \sqcap \sup X = \sup(\{x\} \sqcap X)$.

3. On the inf and sup operation

Let L be a non empty relational structure. The functor inf_op(L) yields a map from $[: L, L :]$ into L and is defined as follows:

(Def. 4) For all elements x, y of L holds $(\mathrm{inf\_op}(L))(\langle x, y \rangle) = x \sqcap y$.

One can prove the following proposition

(31) For every non empty relational structure L and for every element x of $[: L, L :]$ holds $(\mathrm{inf\_op}(L))(x) = x_{\mathbf{1}} \sqcap x_{\mathbf{2}}$.

Let L be a transitive antisymmetric relational structure with g.l.b.'s. Note that inf_op(L) is monotone.

The following two propositions are true:

(32) For every non empty relational structure S and for all subsets D_1, D_2 of S holds $(\mathrm{inf\_op}(S))^\circ[: D_1, D_2 :] = D_1 \sqcap D_2$.

(33) For every up-complete semilattice L and for every non empty directed subset D of $[: L, L :]$ holds $\sup((\mathrm{inf\_op}(L))^\circ D) = \sup(\pi_1(D) \sqcap \pi_2(D))$.

Let L be a non empty relational structure. The functor sup_op(L) yielding a map from $[: L, L :]$ into L is defined by:

(Def. 5) For all elements x, y of L holds $(\mathrm{sup\_op}(L))(\langle x, y \rangle) = x \sqcup y$.

We now state the proposition

(34) For every non empty relational structure L and for every element x of $[: L, L :]$ holds $(\mathrm{sup\_op}(L))(x) = x_{\mathbf{1}} \sqcup x_{\mathbf{2}}$.

Let L be a transitive antisymmetric relational structure with l.u.b.'s. Observe that sup_op(L) is monotone.

The following two propositions are true:

(35) For every non empty relational structure S and for all subsets D_1, D_2 of S holds $(\mathrm{sup\_op}(S))^\circ[: D_1, D_2 :] = D_1 \sqcup D_2$.

(36) For every complete non empty poset L and for every non empty filtered subset D of $[: L, L :]$ holds $\inf((\mathrm{sup\_op}(L))^\circ D) = \inf(\pi_1(D) \sqcup \pi_2(D))$.

4. Meet-continuous lattices

Let R be a non empty reflexive relational structure. We say that R satisfies MC if and only if:

(Def. 6) For every element x of R and for every non empty directed subset D of R holds $x \sqcap \sup D = \sup(\{x\} \sqcap D)$.

Let R be a non empty reflexive relational structure. We say that R is meet-continuous if and only if:

(Def. 7) R is up-complete and satisfies MC.

One can check that every non empty reflexive relational structure which is trivial satisfies MC.

Let us observe that every non empty reflexive relational structure which is meet-continuous is also up-complete and satisfies MC and every non empty reflexive relational structure which is up-complete and satisfies MC is also meet-continuous.

Let us observe that there exists a lattice which is strict, non empty, and trivial.

Next we state two propositions:

(37) Let S be a non empty reflexive relational structure. Suppose that for every subset X of S and for every element x of S holds $x \sqcap \sup X = \bigsqcup_S \{x \sqcap y : y$ ranges over elements of S, $y \in X\}$. Then S satisfies MC.

(38) Let L be an up-complete semilattice. If $\mathrm{SupMap}(L)$ is meet-preserving, then for all ideals I_1, I_2 of L holds $\sup I_1 \sqcap \sup I_2 = \sup(I_1 \sqcap I_2)$.

Let L be an up-complete sup-semilattice. Note that $\mathrm{SupMap}(L)$ is join-preserving.

One can prove the following propositions:

(39) Let L be an up-complete semilattice. If for all ideals I_1, I_2 of L holds $\sup I_1 \sqcap \sup I_2 = \sup(I_1 \sqcap I_2)$, then $\mathrm{SupMap}(L)$ is meet-preserving.

(40) Let L be an up-complete semilattice. Suppose that for all ideals I_1, I_2 of L holds $\sup I_1 \sqcap \sup I_2 = \sup(I_1 \sqcap I_2)$. Let D_1, D_2 be directed non empty subsets of L. Then $\sup D_1 \sqcap \sup D_2 = \sup(D_1 \sqcap D_2)$.

(41) Let L be a non empty reflexive relational structure. Suppose L satisfies MC. Let x be an element of L and let N be a non empty prenet over L. If N is eventually-directed, then $x \sqcap \sup N = \sup(\{x\} \sqcap \mathrm{rng}\,\mathrm{netmap}(N, L))$.

(42) Let L be a non empty reflexive relational structure. Suppose that for every element x of L and for every prenet N over L such that N is eventually-directed holds $x \sqcap \sup N = \sup(\{x\} \sqcap \mathrm{rng}\,\mathrm{netmap}(N, L))$. Then L satisfies MC.

(43) Let L be an up-complete antisymmetric non empty reflexive relational structure. Suppose $\mathrm{inf\_op}(L)$ is directed-sups-preserving. Let D_1, D_2 be non empty directed subsets of L. Then $\sup D_1 \sqcap \sup D_2 = \sup(D_1 \sqcap D_2)$.

(44) Let L be a non empty reflexive antisymmetric relational structure. If for all non empty directed subsets D_1, D_2 of L holds $\sup D_1 \sqcap \sup D_2 = \sup(D_1 \sqcap D_2)$, then L satisfies MC.

(45) Let L be an antisymmetric non empty reflexive relational structure with g.l.b.'s, satisfying MC, and let x be an element of L, and let D be a non empty directed subset of L. If $x \leq \sup D$, then $x = \sup(\{x\} \sqcap D)$.

(46) Let L be an up-complete semilattice. Suppose that for every element x of L and for every non empty directed subset E of L such that $x \leq \sup E$ holds $x \leq \sup(\{x\} \sqcap E)$. Then inf_op($L$) is directed-sups-preserving.

(47) Let L be a complete antisymmetric non empty reflexive relational structure. Suppose that for every element x of L and for every prenet N over L such that N is eventually-directed holds $x \sqcap \sup N = \sup(\{x\} \sqcap \operatorname{rng} \operatorname{netmap}(N, L))$. Let x be an element of L, and let J be a set, and let f be a function from J into the carrier of L. Then $x \sqcap \operatorname{Sup}(f) = \sup(x \sqcap \operatorname{FinSups}(f))$.

(48) Let L be a complete semilattice. Suppose that for every element x of L and for every set J and for every function f from J into the carrier of L holds $x \sqcap \operatorname{Sup}(f) = \sup(x \sqcap \operatorname{FinSups}(f))$. Let x be an element of L and let N be a prenet over L. If N is eventually-directed, then $x \sqcap \sup N = \sup(\{x\} \sqcap \operatorname{rng} \operatorname{netmap}(N, L))$.

(49) For every up-complete lattice L holds L is meet-continuous iff SupMap(L) is meet-preserving and join-preserving.

Let L be a meet-continuous lattice. One can verify that SupMap(L) is meet-preserving and join-preserving.

We now state four propositions:

(50) Let L be an up-complete lattice. Then L is meet-continuous if and only if for all ideals I_1, I_2 of L holds $\sup I_1 \sqcap \sup I_2 = \sup(I_1 \sqcap I_2)$.

(51) Let L be an up-complete lattice. Then L is meet-continuous if and only if for all non empty directed subsets D_1, D_2 of L holds $\sup D_1 \sqcap \sup D_2 = \sup(D_1 \sqcap D_2)$.

(52) Let L be an up-complete lattice. Then L is meet-continuous if and only if for every element x of L and for every non empty directed subset D of L such that $x \leq \sup D$ holds $x = \sup(\{x\} \sqcap D)$.

(53) For every up-complete semilattice L holds L is meet-continuous iff inf_op(L) is directed-sups-preserving.

Let L be a meet-continuous semilattice. Observe that inf_op(L) is directed-sups-preserving.

The following two propositions are true:

(54) Let L be an up-complete semilattice. Then L is meet-continuous if and only if for every element x of L and for every non empty prenet N over L such that N is eventually-directed holds $x \sqcap \sup N = \sup(\{x\} \sqcap \operatorname{rng} \operatorname{netmap}(N, L))$.

(55) Let L be a complete semilattice. Then L is meet-continuous if and only if for every element x of L and for every set J and for every function f from J into the carrier of L holds $x \sqcap \mathrm{Sup}(f) = \sup(x \sqcap \mathrm{FinSups}(f))$.

Let L be a meet-continuous semilattice and let x be an element of L. One can verify that $x \sqcap \square$ is directed-sups-preserving.

The following proposition is true

(56) For every complete non empty poset H holds H is Heyting iff H is meet-continuous and distributive.

Let us mention that every non empty poset which is complete and Heyting is also meet-continuous and distributive and every non empty poset which is complete, meet-continuous, and distributive is also Heyting.

References

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

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

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

[4] Grzegorz Bancerek. The well ordering relations. *Formalized Mathematics*, 1(**1**):123–129, 1990.

[5] Józef Białas. Group and field definitions. *Formalized Mathematics*, 1(**3**):433–439, 1990.

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

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

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

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

[10] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.

[11] 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.

[12] Adam Grabowski. On the category of posets. *Formalized Mathematics*, 5(**4**):501–505, 1996.

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

[14] Krzysztof Hryniewiecki. Relations of tolerance. *Formalized Mathematics*, 2(**1**):105–109, 1991.

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

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

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

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

[19] Andrzej Trybulec and Agata Darmochwał. Boolean domains. *Formalized Mathematics*, 1(**1**):187–190, 1990.

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

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

[22] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[23] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.
[24] Edmund Woronowicz and Anna Zalewska. Properties of binary relations. *Formalized Mathematics*, 1(**1**):85–89, 1990.
[25] Mariusz Żynel and Czesław Byliński. Properties of relational structures, posets, lattices and maps. *Formalized Mathematics*, 6(**1**):123–130, 1997.

Received October 10, 1996

FORMALIZED MATHEMATICS
Volume 6, Number 1, 1997
Warsaw University - Białystok

The "Way-Below" Relation [1]

Grzegorz Bancerek
Warsaw University
Białystok

Summary. In the paper the "way-below" relation, in symbols $x \ll y$, is introduced. Some authors prefer the term "relatively compact" or "way inside", since in the poset of open sets of a topology it is natural to read $U \ll V$ as "U is relatively compact in V". A compact element of a poset (or an element isolated from below) is defined to be way below itself. So, the compactness in the poset of open sets of a topology is equivalent to the compactness in that topology.

The article includes definitions, facts and examples 1.1–1.8 presented in [15, pp. 38–42].

MML Identifier: `WAYBEL_3`.

The terminology and notation used in this paper have been introduced in the following articles: [5], [25], [29], [30], [31], [20], [14], [23], [8], [28], [10], [11], [22], [24], [6], [19], [7], [26], [33], [27], [21], [32], [13], [12], [9], [4], [2], [1], [16], [3], [17], and [18].

1. The "Way-Below" Relation

Let L be a non empty reflexive relational structure and let x, y be elements of L. We say that x is way below y if and only if:

(Def. 1) For every non empty directed subset D of L such that $y \leq \sup D$ there exists an element d of L such that $d \in D$ and $x \leq d$.

We introduce $x \ll y$ and $y \gg x$ as synonyms of x is way below y.

Let L be a non empty reflexive relational structure and let x be an element of L. We say that x is compact if and only if:

(Def. 2) x is way below x.

[1]This work has been partially supported by Office of Naval Research Grant N00014-95-1-1336.

ISSN 1426–2630

We introduce x is isolated from below as a synonym of x is compact.

Next we state several propositions:

(1) Let L be a non empty reflexive antisymmetric relational structure and let x, y be elements of L. If $x \ll y$, then $x \leq y$.

(2) Let L be a non empty reflexive transitive relational structure and let u, x, y, z be elements of L. If $u \leq x$ and $x \ll y$ and $y \leq z$, then $u \ll z$.

(3) Let L be a non empty poset. Suppose L is inf-complete or has l.u.b.'s. Let x, y, z be elements of L. If $x \ll z$ and $y \ll z$, then sup $\{x, y\}$ exists in L and $x \sqcup y \ll z$.

(4) Let L be a lower-bounded antisymmetric reflexive non empty relational structure and let x be an element of L. Then $\perp_L \ll x$.

(5) For every non empty poset L and for all elements x, y, z of L such that $x \ll y$ and $y \ll z$ holds $x \ll z$.

(6) Let L be a non empty reflexive antisymmetric relational structure and let x, y be elements of L. If $x \ll y$ and $x \gg y$, then $x = y$.

Let L be a non empty reflexive relational structure and let x be an element of L. The functor $\downdownarrows x$ yields a subset of L and is defined as follows:

(Def. 3) $\downdownarrows x = \{y : y$ ranges over elements of L, $y \ll x\}$.

The functor $\upuparrows x$ yielding a subset of L is defined by:

(Def. 4) $\upuparrows x = \{y : y$ ranges over elements of L, $y \gg x\}$.

We now state several propositions:

(7) For every non empty reflexive relational structure L and for all elements x, y of L holds $x \in \downdownarrows y$ iff $x \ll y$.

(8) For every non empty reflexive relational structure L and for all elements x, y of L holds $x \in \upuparrows y$ iff $x \gg y$.

(9) For every non empty reflexive antisymmetric relational structure L and for every element x of L holds $x \geq \downdownarrows x$.

(10) For every non empty reflexive antisymmetric relational structure L and for every element x of L holds $x \leq \upuparrows x$.

(11) Let L be a non empty reflexive antisymmetric relational structure and let x be an element of L. Then $\downdownarrows x \subseteq \downarrow x$ and $\upuparrows x \subseteq \uparrow x$.

(12) Let L be a non empty reflexive transitive relational structure and let x, y be elements of L. If $x \leq y$, then $\downdownarrows x \subseteq \downdownarrows y$ and $\upuparrows y \subseteq \upuparrows x$.

Let L be a lower-bounded non empty reflexive antisymmetric relational structure and let x be an element of L. Note that $\downdownarrows x$ is non empty.

Let L be a non empty reflexive transitive relational structure and let x be an element of L. Note that $\downdownarrows x$ is lower and $\upuparrows x$ is upper.

Let L be a sup-semilattice and let x be an element of L. One can verify that $\downdownarrows x$ is directed.

Let L be an inf-complete non empty poset and let x be an element of L. Note that $\downdownarrows x$ is directed.

Let L be a connected non empty relational structure. One can check that every subset of L is directed and filtered.

Let us note that every non empty chain which is up-complete and lower-bounded is also complete.

One can verify that there exists a non empty chain which is complete.

We now state several propositions:

(13) For every up-complete non empty chain L and for all elements x, y of L such that $x < y$ holds $x \ll y$.

(14) Let L be a non empty reflexive antisymmetric relational structure and let x, y be elements of L. If x is not compact and $x \ll y$, then $x < y$.

(15) For every non empty lower-bounded reflexive antisymmetric relational structure L holds $\bot_L$ is compact.

(16) For every up-complete non empty poset L and for every non empty finite directed subset D of L holds $\sup D \in D$.

(17) For every up-complete non empty poset L such that L is finite holds every element of L is isolated from below.

2. The Way-Below Relation in Other Terms

The scheme *SSubsetEx* deals with a non empty relational structure $\mathcal{A}$ and a unary predicate $\mathcal{P}$, and states that:

There exists a subset X of $\mathcal{A}$ such that for every element x of $\mathcal{A}$ holds $x \in X$ iff $\mathcal{P}[x]$

for all values of the parameters.

We now state several propositions:

(18) Let L be a complete lattice and let x, y be elements of L. Suppose $x \ll y$. Let X be a subset of L. If $y \leq \sup X$, then there exists a finite subset A of L such that $A \subseteq X$ and $x \leq \sup A$.

(19) Let L be a complete lattice and let x, y be elements of L. Suppose that for every subset X of L such that $y \leq \sup X$ there exists a finite subset A of L such that $A \subseteq X$ and $x \leq \sup A$. Then $x \ll y$.

(20) Let L be a non empty reflexive transitive relational structure and let x, y be elements of L. If $x \ll y$, then for every ideal I of L such that $y \leq \sup I$ holds $x \in I$.

(21) Let L be an up-complete non empty poset and let x, y be elements of L. If for every ideal I of L such that $y \leq \sup I$ holds $x \in I$, then $x \ll y$.

(22) Let L be a lower-bounded lattice. Suppose L is meet-continuous. Let x, y be elements of L. Then $x \ll y$ if and only if for every ideal I of L such that $y = \sup I$ holds $x \in I$.

(23) Let L be a complete lattice. Then every element of L is compact if and only if for every non empty subset X of L there exists an element x of

L such that $x \in X$ and for every element y of L such that $y \in X$ holds $x \not< y$.

3. Continuous Lattices

Let L be a non empty reflexive relational structure. We say that L satisfies axiom of approximation if and only if:

(Def. 5) For every element x of L holds $x = \sup \twoheaddownarrow x$.

Let us note that every non empty reflexive relational structure which is trivial satisfies axiom of approximation.

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

(Def. 6) For every element x of L holds $\twoheaddownarrow x$ is non empty and directed and L is up-complete and satisfies axiom of approximation.

One can check that every non empty reflexive relational structure which is continuous is also up-complete and satisfies axiom of approximation and every lower-bounded sup-semilattice which is up-complete and satisfies axiom of approximation is also continuous.

Let us note that there exists a lattice which is continuous, complete, and strict.

Let L be a continuous non empty reflexive relational structure and let x be an element of L. One can verify that $\twoheaddownarrow x$ is non empty and directed.

Next we state two propositions:

(24) Let L be an up-complete semilattice. Suppose that for every element x of L holds $\twoheaddownarrow x$ is non empty and directed. Then L satisfies axiom of approximation if and only if for all elements x, y of L such that $x \not\leq y$ there exists an element u of L such that $u \ll x$ and $u \not\leq y$.

(25) For every continuous lattice L and for all elements x, y of L holds $x \leq y$ iff $\twoheaddownarrow x \subseteq \twoheaddownarrow y$.

One can verify that every non empty chain which is complete satisfies axiom of approximation.

The following proposition is true

(26) For every complete lattice L such that every element of L is compact holds L satisfies axiom of approximation.

4. The Way-Below Relation in Direct Powers

Let f be a binary relation. We say that f is nonempty if and only if:

(Def. 7) For every 1-sorted structure S such that $S \in \operatorname{rng} f$ holds S is non empty.

We say that f is reflexive-yielding if and only if:

(Def. 8) For every relational structure S such that $S \in \operatorname{rng} f$ holds S is reflexive.

Let I be a set. Observe that there exists a many sorted set indexed by I which is relational structure yielding, nonempty, and reflexive-yielding.

Let I be a set and let J be a relational structure yielding nonempty many sorted set indexed by I. Observe that $\prod J$ is non empty.

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

Let I be a set and let J be a relational structure yielding nonempty many sorted set indexed by I. Note that every element of $\prod J$ is function-like and relation-like.

Let I be a non empty set, let J be a relational structure yielding nonempty many sorted set indexed by I, let x be an element 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 relational structure yielding nonempty many sorted set indexed by I, let i be an element of I, and let X be a subset of $\prod J$. Then $\pi_i X$ is a subset of $J(i)$.

Next we state two propositions:

(27) Let I be a non empty set, and let J be a relational structure yielding nonempty many sorted set indexed by I, and let x be a function. Then x is an element of $\prod J$ if and only if $\operatorname{dom} x = I$ and for every element i of I holds $x(i)$ is an element of $J(i)$.

(28) Let I be a non empty set, and let J be a relational structure yielding nonempty many sorted set indexed by I, and let x, y be elements of $\prod J$. Then $x \leq y$ if and only if for every element i of I holds $x(i) \leq y(i)$.

Let I be a non empty set and let J be a relational structure yielding nonempty reflexive-yielding many sorted set indexed by I. Note that $\prod J$ is reflexive. Let i be an element of I. Then $J(i)$ is a non empty reflexive relational structure.

Let I be a non empty set, let J be a relational structure yielding nonempty reflexive-yielding many sorted set indexed by I, let x be an element of $\prod J$, and let i be an element of I. Then $x(i)$ is an element of $J(i)$.

One can prove the following propositions:

(29) Let I be a non empty set and let J be a relational structure yielding nonempty many sorted set indexed by I. If for every element i of I holds $J(i)$ is transitive, then $\prod J$ is transitive.

(30) Let I be a non empty set and let J be a relational structure yielding nonempty many sorted set indexed by I. Suppose that for every element i of I holds $J(i)$ is antisymmetric. Then $\prod J$ is antisymmetric.

(31) Let I be a non empty set and let J be a relational structure yielding nonempty reflexive-yielding many sorted set indexed by I. Suppose that for every element i of I holds $J(i)$ is a complete lattice. Then $\prod J$ is a complete lattice.

(32) Let I be a non empty set and let J be a relational structure yielding

nonempty reflexive-yielding many sorted set indexed by I. Suppose that for every element i of I holds $J(i)$ is a complete lattice. Let X be a subset of $\prod J$ and let i be an element of I. Then $(\sup X)(i) = \sup \pi_i X$.

(33) Let I be a non empty set and let J be a relational structure yielding nonempty reflexive-yielding many sorted set indexed by I. Suppose that for every element i of I holds $J(i)$ is a complete lattice. Let x, y be elements of $\prod J$. Then $x \ll y$ if and only if the following conditions are satisfied:

(i) for every element i of I holds $x(i) \ll y(i)$, and

(ii) there exists a finite subset K of I such that for every element i of I such that $i \notin K$ holds $x(i) = \perp_{J(i)}$.

5. The Way-Below Relation in Topological Spaces

One can prove the following four propositions:

(34) Let T be a non empty topological space and let x, y be elements of $\langle$the topology of T, $\subseteq\rangle$. Suppose x is way below y. Let F be a family of subsets of T. If F is open and $y \subseteq \bigcup F$, then there exists a finite subset G of F such that $x \subseteq \bigcup G$.

(35) Let T be a non empty topological space and let x, y be elements of $\langle$the topology of T, $\subseteq\rangle$. Suppose that for every family F of subsets of T such that F is open and $y \subseteq \bigcup F$ there exists a finite subset G of F such that $x \subseteq \bigcup G$. Then x is way below y.

(36) Let T be a non empty topological space, and let x be an element of $\langle$the topology of T, $\subseteq\rangle$, and let X be a subset of T. If $x = X$, then x is compact iff X is compact.

(37) Let T be a non empty topological space and let x be an element of $\langle$the topology of T, $\subseteq\rangle$. Suppose $x =$ the carrier of T. Then x is compact if and only if T is compact.

Let T be a non empty topological space. We say that T is locally-compact if and only if the condition (Def. 9) is satisfied.

(Def. 9) Let x be a point of T and let X be a subset of T. Suppose $x \in X$ and X is open. Then there exists a subset Y of T such that $x \in \operatorname{Int} Y$ and $Y \subseteq X$ and Y is compact.

Let us observe that every non empty topological space which is compact and T_2 is also T_3 , T_4 , and locally-compact.

We now state the proposition

(38) For every set x holds $\{x\}_{\rm top}$ is T_2.

One can verify that there exists a non empty topological space which is compact and T_2 .

One can prove the following two propositions:

(39) Let T be a non empty topological space and let x, y be elements of $\langle$the topology of T, $\subseteq\rangle$. If there exists a subset Z of T such that $x \subseteq Z$ and $Z \subseteq y$ and Z is compact, then $x \ll y$.

(40) Let T be a non empty topological space. Suppose T is locally-compact. Let x, y be elements of $\langle$the topology of T, $\subseteq\rangle$. If $x \ll y$, then there exists a subset Z of T such that $x \subseteq Z$ and $Z \subseteq y$ and Z is compact.

Let T be a topological structure and let X be a subset of the carrier of T. Then $\overline{X}$ is a subset of T.

The following three propositions are true:

(41) Let T be a non empty topological space. Suppose T is locally-compact and a T_2 space. Let x, y be elements of $\langle$the topology of T, $\subseteq\rangle$. If $x \ll y$, then there exists a subset Z of T such that $Z = x$ and $\overline{Z} \subseteq y$ and $\overline{Z}$ is compact.

(42) Let X be a non empty topological space. Suppose X is a T_3 space and $\langle$the topology of X, $\subseteq\rangle$ is continuous. Then X is locally-compact.

(43) For every non empty topological space T such that T is locally-compact holds $\langle$the topology of T, $\subseteq\rangle$ is continuous.

References

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

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

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

[4] Grzegorz Bancerek. Filters - Part II. Quotient lattices modulo filters and direct product of two lattices. *Formalized Mathematics*, 2(**3**):433–438, 1991.

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

[8] Józef Białas. Group and field definitions. *Formalized Mathematics*, 1(**3**):433–439, 1990.

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

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

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

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

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

[14] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.

[15] 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.

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

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

[18] Artur Korniłowicz. Meet – continuous lattices. *Formalized Mathematics*, 6(**1**):159–167, 1997.
[19] Beata Madras. Product of family of universal algebras. *Formalized Mathematics*, 4(**1**):103–108, 1993.
[20] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[21] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[22] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[23] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.
[24] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[25] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[26] Wojciech A. Trybulec. Groups. *Formalized Mathematics*, 1(**5**):821–827, 1990.
[27] Wojciech A. Trybulec. Partially ordered sets. *Formalized Mathematics*, 1(**2**):313–319, 1990.
[28] Zinaida Trybulec. Properties of subsets. *Formalized Mathematics*, 1(**1**):67–71, 1990.
[29] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[30] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[31] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.
[32] Mirosław Wysocki and Agata Darmochwał. Subsets of topological spaces. *Formalized Mathematics*, 1(**1**):231–237, 1990.
[33] Stanisław Żukowski. Introduction to lattice theory. *Formalized Mathematics*, 1(**1**):215–222, 1990.

Received October 11, 1996

alalalalalal alalalalalalal

Index of MML Identifiers

178

Contents

Formaliz. Math. 6 (**1**)

Continued on inside back cover