FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

Left and Right Component of the Complement of a Special Closed Curve

Andrzej Trybulec
Warsaw University
Białystok

Summary. In the article the concept of the left and right component are introduced. These are the auxiliary notions needed in the proof of Jordan Curve Theorem.

MML Identifier: GOBOARD9.

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

For simplicity we adopt the following rules: f will denote a non constant standard special circular sequence, i, j, k will denote natural numbers, p, q will denote points of $\mathcal{E}^2_{\mathrm{T}}$, and G will denote a Go-board.

The following propositions are true:

(1) $i -' i = 0$.

(2) $i -' j \leq i$.

(3) Let G_1 be a non empty topological space and let A_1, A_2, B be subsets of the carrier of G_1. Suppose A_1 is a component of B and A_2 is a component of B. Then $A_1 = A_2$ or A_1 misses A_2.

(4) Let G_1 be a non empty topological space, and let A, B be non empty subsets of the carrier of G_1, and let A_3 be a subset of the carrier of $G_1 \restriction B$.If $A = A_3$, then $G_1 \restriction A = G_1 \restriction B \restriction A_3$.

(5) Let G_1 be a non empty topological space and let A, B be non empty subsets of the carrier of G_1. Suppose $A \subseteq B$ and A is connected. Then there exists a subset C of the carrier of G_1 such that C is a component of B and $A \subseteq C$.

(6) Let G_1 be a non empty topological space and let A, B, C, D be subsets of the carrier of G_1. Suppose B is connected and C is a component of D and $A \subseteq C$ and A meets B and $B \subseteq D$. Then $B \subseteq C$.

ISSN 1426–2630

(7) $\mathcal{L}(p,q)$ is convex.

(8) $\mathcal{L}(p,q)$ is connected.

One can check that there exists a subset of the carrier of $\mathcal{E}^2_{\mathrm{T}}$ which is convex.

One can prove the following three propositions:

(9) For all convex subsets P, Q of the carrier of $\mathcal{E}^2_{\mathrm{T}}$ holds $P \cap Q$ is convex.

(10) For every finite sequence f of elements of $\mathcal{E}^2_{\mathrm{T}}$ holds $\mathrm{Rev}(\mathbf{X}\text{-coordinate}(f)) = \mathbf{X}\text{-coordinate}(\mathrm{Rev}(f))$.

(11) For every finite sequence f of elements of $\mathcal{E}^2_{\mathrm{T}}$ holds $\mathrm{Rev}(\mathbf{Y}\text{-coordinate}(f)) = \mathbf{Y}\text{-coordinate}(\mathrm{Rev}(f))$.

Let us mention that there exists a finite sequence which is non constant.

Let f be a non constant finite sequence. Note that $\mathrm{Rev}(f)$ is non constant.

Let f be a standard special circular sequence. Then $\mathrm{Rev}(f)$ is a standard special circular sequence.

We now state a number of propositions:

(12) If $i \geq 1$ and $j \geq 1$ and $i + j = \operatorname{len} f$, then $\mathrm{leftcell}(f, i) = \mathrm{rightcell}(\mathrm{Rev}(f), j)$.

(13) If $i \geq 1$ and $j \geq 1$ and $i + j = \operatorname{len} f$, then $\mathrm{leftcell}(\mathrm{Rev}(f), i) = \mathrm{rightcell}(f, j)$.

(14) Suppose $1 \leq k$ and $k + 1 \leq \operatorname{len} f$. Then there exist i, j such that $i \leq \operatorname{len}$ the Go-board of f and $j \leq \operatorname{width}$ the Go-board of f and cell(the Go-board of f, $i, j) = \mathrm{leftcell}(f, k)$.

(15) If $j \leq \operatorname{width} G$, then $\operatorname{Int} \mathrm{hstrip}(G, j)$ is convex.

(16) If $i \leq \operatorname{len} G$, then $\operatorname{Int} \mathrm{vstrip}(G, i)$ is convex.

(17) If $i \leq \operatorname{len} G$ and $j \leq \operatorname{width} G$, then $\operatorname{Int} \mathrm{cell}(G, i, j) \neq \emptyset$.

(18) If $1 \leq k$ and $k + 1 \leq \operatorname{len} f$, then $\operatorname{Int} \mathrm{leftcell}(f, k) \neq \emptyset$.

(19) If $1 \leq k$ and $k + 1 \leq \operatorname{len} f$, then $\operatorname{Int} \mathrm{rightcell}(f, k) \neq \emptyset$.

(20) If $i \leq \operatorname{len} G$ and $j \leq \operatorname{width} G$, then $\operatorname{Int} \mathrm{cell}(G, i, j)$ is convex.

(21) If $i \leq \operatorname{len} G$ and $j \leq \operatorname{width} G$, then $\operatorname{Int} \mathrm{cell}(G, i, j)$ is connected.

(22) If $1 \leq k$ and $k + 1 \leq \operatorname{len} f$, then $\operatorname{Int} \mathrm{leftcell}(f, k)$ is connected.

(23) If $1 \leq k$ and $k + 1 \leq \operatorname{len} f$, then $\operatorname{Int} \mathrm{rightcell}(f, k)$ is connected.

Let us consider f. The functor $\mathrm{LeftComp}(f)$ yields a subset of the carrier of $\mathcal{E}^2_{\mathrm{T}}$ and is defined as follows:

(Def. 1) $\mathrm{LeftComp}(f)$ is a component of $(\widetilde{\mathcal{L}}(f))^{\mathrm{c}}$ and $\operatorname{Int} \mathrm{leftcell}(f, 1) \subseteq \mathrm{LeftComp}(f)$.

The functor $\mathrm{RightComp}(f)$ yields a subset of the carrier of $\mathcal{E}^2_{\mathrm{T}}$ and is defined by:

(Def. 2) $\mathrm{RightComp}(f)$ is a component of $(\widetilde{\mathcal{L}}(f))^{\mathrm{c}}$ and $\operatorname{Int} \mathrm{rightcell}(f, 1) \subseteq \mathrm{RightComp}(f)$.

One can prove the following propositions:

(24) For every k such that $1 \leq k$ and $k + 1 \leq \operatorname{len} f$ holds $\operatorname{Int} \mathrm{leftcell}(f, k) \subseteq \mathrm{LeftComp}(f)$.

(25) The Go-board of $\mathrm{Rev}(f)$ = the Go-board of f.

(26) $\mathrm{RightComp}(f) = \mathrm{LeftComp}(\mathrm{Rev}(f))$.

(27) $\mathrm{RightComp}(\mathrm{Rev}(f)) = \mathrm{LeftComp}(f)$.

(28) For every k such that $1 \leq k$ and $k+1 \leq \mathrm{len}\, f$ holds $\mathrm{Int}\, \mathrm{rightcell}(f, k) \subseteq \mathrm{RightComp}(f)$.

References

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

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

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

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

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

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

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

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

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

[10] 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.

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

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

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

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

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

[16] Yatsuka Nakamura and Jarosław Kotowicz. The Jordan's property for certain subsets of the plane. *Formalized Mathematics*, 3(**2**):137–142, 1992.

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

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

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

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

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

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

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

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

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

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

[27] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[28] Mirosław Wysocki and Agata Darmochwał. Subsets of topological spaces. *Formalized Mathematics*, 1(**1**):231–237, 1990.

Received October 29, 1995

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

Reduction Relations

Grzegorz Bancerek
Institute of Mathematics
Polish Academy of Sciences

Summary. The goal of the article is to start the formalization of Knuth-Bendix completion method (see [2,11] or [1]; see also [12,10]), i.e. to formalize the concept of the completion of a reduction relation. The completion of a reduction relation R is a complete reduction relation equivalent to R such that convertible elements have the same normal forms. The theory formalized in the article includes concepts and facts concerning normal forms, terminating reductions, Church-Rosser property, and equivalence of reduction relations.

MML Identifier: `REWRITE1`.

The terminology and notation used here are introduced in the following articles: [16], [17], [9], [3], [6], [18], [19], [4], [13], [14], [5], [15], [7], and [8].

1. Forgetting concatenation and reduction sequence

Let p, q be finite sequences. The functor $p \mathbin{^{\$\frown}} q$ yielding a finite sequence is defined as follows:

(Def. 1) (i) $p \mathbin{^{\$\frown}} q = p \frown q$ if $p = \varepsilon$ or $q = \varepsilon$,

(ii) there exists a natural number i and there exists a finite sequence r such that $\mathrm{len}\, p = i + 1$ and $r = p \restriction \mathrm{Seg}\, i$ and $p \mathbin{^{\$\frown}} q = r \frown q$, otherwise.

In the sequel p, q are finite sequences and x, y are sets.

We now state several propositions:

(1) $\varepsilon \mathbin{^{\$\frown}} p = p$ and $p \mathbin{^{\$\frown}} \varepsilon = p$.

(2) If $q \neq \varepsilon$, then $(p \frown \langle x \rangle) \mathbin{^{\$\frown}} q = p \frown q$.

(3) $(p \frown \langle x \rangle) \mathbin{^{\$\frown}} (\langle y \rangle \frown q) = p \frown \langle y \rangle \frown q$.

(4) If $q \neq \varepsilon$, then $\langle x \rangle \mathbin{^{\$\frown}} q = q$.

(5) If $p \neq \varepsilon$, then there exist x, q such that $p = \langle x \rangle \frown q$ and $\mathrm{len}\, p = \mathrm{len}\, q + 1$.

ISSN 1426–2630

The scheme *PathCatenation* concerns finite sequences $\mathcal{A}$, $\mathcal{B}$ and a binary predicate $\mathcal{P}$, and states that:

Let i be a natural number. Suppose $i \in \mathrm{dom}(\mathcal{A} \,{}^{\$\frown}\, \mathcal{B})$ and $i+1 \in \mathrm{dom}(\mathcal{A} \,{}^{\$\frown}\, \mathcal{B})$. Let x, y be sets. If $x = (\mathcal{A} \,{}^{\$\frown}\, \mathcal{B})(i)$ and $y = (\mathcal{A} \,{}^{\$\frown}\, \mathcal{B})(i+1)$, then $\mathcal{P}[x, y]$

provided the parameters satisfy the following conditions:

- For every natural number i such that $i \in \mathrm{dom}\,\mathcal{A}$ and $i+1 \in \mathrm{dom}\,\mathcal{A}$ holds $\mathcal{P}[\mathcal{A}(i), \mathcal{A}(i+1)]$,
- For every natural number i such that $i \in \mathrm{dom}\,\mathcal{B}$ and $i+1 \in \mathrm{dom}\,\mathcal{B}$ holds $\mathcal{P}[\mathcal{B}(i), \mathcal{B}(i+1)]$,
- $\mathrm{len}\,\mathcal{A} > 0$ and $\mathrm{len}\,\mathcal{B} > 0$ and $\mathcal{A}(\mathrm{len}\,\mathcal{A}) = \mathcal{B}(1)$.

Let R be a binary relation. A finite sequence is said to be a reduction sequence w.r.t. R if:

(Def. 2) $\mathrm{len}\,\mathrm{it} > 0$ and for every natural number i such that $i \in \mathrm{dom}\,\mathrm{it}$ and $i+1 \in \mathrm{dom}\,\mathrm{it}$ holds $\langle \mathrm{it}(i), \mathrm{it}(i+1) \rangle \in R$.

Next we state the proposition

(6) For every binary relation R and for every reduction sequence p w.r.t. R holds $1 \in \mathrm{dom}\,p$ and $\mathrm{len}\,p \in \mathrm{dom}\,p$.

Let R be a binary relation. Note that every reduction sequence w.r.t. R is non empty.

One can prove the following propositions:

(7) For every binary relation R and for every set a holds $\langle a \rangle$ is a reduction sequence w.r.t. R.

(8) For every binary relation R and for all sets a, b such that $\langle a, b \rangle \in R$ holds $\langle a, b \rangle$ is a reduction sequence w.r.t. R.

(9) Let R be a binary relation and let p, q be reduction sequences w.r.t. R. If $p(\mathrm{len}\,p) = q(1)$, then $p \,{}^{\$\frown}\, q$ is a reduction sequence w.r.t. R.

(10) Let R be a binary relation and let p be a reduction sequence w.r.t. R. Then $\mathrm{Rev}(p)$ is a reduction sequence w.r.t. $R^{\smile}$.

(11) For all binary relations R, Q such that $R \subseteq Q$ holds every reduction sequence w.r.t. R is a reduction sequence w.r.t. Q.

2. Reducibility, convertibility and normal forms

Let R be a binary relation and let a, b be sets. We say that R reduces a to b if and only if:

(Def. 3) There exists a reduction sequence p w.r.t. R such that $p(1) = a$ and $p(\mathrm{len}\,p) = b$.

Let R be a binary relation and let a, b be sets. We say that a and b are convertible w.r.t. R if and only if:

(Def. 4) $R \cup R^{\smile}$ reduces a to b.

One can prove the following propositions:

(12) Let R be a binary relation and let a, b be sets. Then R reduces a to b if and only if there exists a finite sequence p such that $\operatorname{len} p > 0$ and $p(1) = a$ and $p(\operatorname{len} p) = b$ and for every natural number i such that $i \in \operatorname{dom} p$ and $i + 1 \in \operatorname{dom} p$ holds $\langle p(i), p(i+1)\rangle \in R$.

(13) For every binary relation R and for every set a holds R reduces a to a.

(14) For all sets a, b such that $\emptyset$ reduces a to b holds $a = b$.

(15) For every binary relation R and for all sets a, b such that R reduces a to b and $a \notin \operatorname{field} R$ holds $a = b$.

(16) For every binary relation R and for all sets a, b such that $\langle a, b\rangle \in R$ holds R reduces a to b.

(17) Let R be a binary relation and let a, b, c be sets. Suppose R reduces a to b and R reduces b to c. Then R reduces a to c.

(18) Let R be a binary relation, and let p be a reduction sequence w.r.t. R, and let i, j be natural numbers. If $i \in \operatorname{dom} p$ and $j \in \operatorname{dom} p$ and $i \leq j$, then R reduces $p(i)$ to $p(j)$.

(19) For every binary relation R and for all sets a, b such that R reduces a to b and $a \neq b$ holds $a \in \operatorname{field} R$ and $b \in \operatorname{field} R$.

(20) For every binary relation R and for all sets a, b such that R reduces a to b holds $a \in \operatorname{field} R$ iff $b \in \operatorname{field} R$.

(21) For every binary relation R and for all sets a, b holds R reduces a to b iff $a = b$ or $\langle a, b\rangle \in R^*$.

(22) For every binary relation R and for all sets a, b holds R reduces a to b iff R^* reduces a to b.

(23) Let R, Q be binary relations. Suppose $R \subseteq Q$. Let a, b be sets. If R reduces a to b, then Q reduces a to b.

(24) Let R be a binary relation, and let X be a set, and let a, b be sets. Then R reduces a to b if and only if $R \cup \triangle_X$ reduces a to b.

(25) For every binary relation R and for all sets a, b such that R reduces a to b holds $R^{\smile}$ reduces b to a.

(26) Let R be a binary relation and let a, b be sets. Suppose R reduces a to b. Then a and b are convertible w.r.t. R and b and a are convertible w.r.t. R.

(27) For every binary relation R and for every set a holds a and a are convertible w.r.t. R.

(28) For all sets a, b such that a and b are convertible w.r.t. $\emptyset$ holds $a = b$.

(29) Let R be a binary relation and let a, b be sets. If a and b are convertible w.r.t. R and $a \notin \operatorname{field} R$, then $a = b$.

(30) For every binary relation R and for all sets a, b such that $\langle a, b\rangle \in R$ holds a and b are convertible w.r.t. R.

(31) Let R be a binary relation and let a, b, c be sets. Suppose a and b are convertible w.r.t. R and b and c are convertible w.r.t. R. Then a and c

are convertible w.r.t. R.

(32) Let R be a binary relation and let a, b be sets. Suppose a and b are convertible w.r.t. R. Then b and a are convertible w.r.t. R.

(33) Let R be a binary relation and let a, b be sets. If a and b are convertible w.r.t. R and $a \neq b$, then $a \in \operatorname{field} R$ and $b \in \operatorname{field} R$.

Let R be a binary relation and let a be a set. We say that a is a normal form w.r.t. R if and only if:

(Def. 5) It is not true that there exists a set b such that $\langle a, b \rangle \in R$.

The following propositions are true:

(34) Let R be a binary relation and let a, b be sets. If a is a normal form w.r.t. R and R reduces a to b, then $a = b$.

(35) For every binary relation R and for every set a such that $a \notin \operatorname{field} R$ holds a is a normal form w.r.t. R.

Let R be a binary relation and let a, b be sets. We say that b is a normal form of a w.r.t. R if and only if:

(Def. 6) b is a normal form w.r.t. R and R reduces a to b.

We say that a and b are convergent w.r.t. R if and only if:

(Def. 7) There exists a set c such that R reduces a to c and R reduces b to c.

We say that a and b are divergent w.r.t. R if and only if:

(Def. 8) There exists a set c such that R reduces c to a and R reduces c to b.

We say that a and b are convergent at most in 1 step w.r.t. R if and only if:

(Def. 9) There exists a set c such that $\langle a, c \rangle \in R$ or $a = c$ but $\langle b, c \rangle \in R$ or $b = c$.

We say that a and b are divergent at most in 1 step w.r.t. R if and only if:

(Def. 10) There exists a set c such that $\langle c, a \rangle \in R$ or $a = c$ but $\langle c, b \rangle \in R$ or $b = c$.

Next we state a number of propositions:

(36) For every binary relation R and for every set a such that $a \notin \operatorname{field} R$ holds a is a normal form of a w.r.t. R.

(37) Let R be a binary relation and let a, b be sets. Suppose R reduces a to b. Then

(i) a and b are convergent w.r.t. R,

(ii) a and b are divergent w.r.t. R,

(iii) b and a are convergent w.r.t. R, and

(iv) b and a are divergent w.r.t. R.

(38) Let R be a binary relation and let a, b be sets. Suppose a and b are convergent w.r.t. R or a and b are divergent w.r.t. R. Then a and b are convertible w.r.t. R.

(39) Let R be a binary relation and let a be a set. Then a and a are convergent w.r.t. R and a and a are divergent w.r.t. R.

(40) For all sets a, b such that a and b are convergent w.r.t. $\emptyset$ or a and b are divergent w.r.t. $\emptyset$ holds $a = b$.

(41) Let R be a binary relation and let a, b be sets. Suppose a and b are convergent w.r.t. R. Then b and a are convergent w.r.t. R.

(42) Let R be a binary relation and let a, b be sets. Suppose a and b are divergent w.r.t. R. Then b and a are divergent w.r.t. R.

(43) Let R be a binary relation and let a, b, c be sets. Suppose that
(i) R reduces a to b and b and c are convergent w.r.t. R, or
(ii) a and b are convergent w.r.t. R and R reduces c to b.
Then a and c are convergent w.r.t. R.

(44) Let R be a binary relation and let a, b, c be sets. Suppose that
(i) R reduces b to a and b and c are divergent w.r.t. R, or
(ii) a and b are divergent w.r.t. R and R reduces b to c.
Then a and c are divergent w.r.t. R.

(45) Let R be a binary relation and let a, b be sets. Suppose a and b are convergent at most in 1 step w.r.t. R. Then a and b are convergent w.r.t. R.

(46) Let R be a binary relation and let a, b be sets. Suppose a and b are divergent at most in 1 step w.r.t. R. Then a and b are divergent w.r.t. R.

Let R be a binary relation and let a be a set. We say that a has a normal form w.r.t. R if and only if:

(Def. 11) There exists set which is a normal form of a w.r.t. R.

Next we state the proposition

(47) For every binary relation R and for every set a such that $a \notin \operatorname{field} R$ holds a has a normal form w.r.t. R.

Let R be a binary relation and let a be a set. Let us assume that a has a normal form w.r.t. R and for all sets b, c such that b is a normal form of a w.r.t. R and c is a normal form of a w.r.t. R holds $b = c$. The functor $\mathrm{nf}_R(a)$ is defined by:

(Def. 12) $\mathrm{nf}_R(a)$ is a normal form of a w.r.t. R.

3. Terminating reductions

Let R be a binary relation. We say that R is reversely well founded if and only if:

(Def. 13) $R^{\smile}$ is well founded.

We say that R is weakly-normalizing if and only if:

(Def. 14) For every set a such that $a \in \operatorname{field} R$ holds a has a normal form w.r.t. R.

We say that R is strongly-normalizing if and only if:

(Def. 15) For every many sorted set f indexed by $\mathbb{N}$ there exists a natural number i such that $\langle f(i), f(i+1)\rangle \notin R$.

Let R be a binary relation. Let us observe that R is reversely well founded if and only if the condition (Def. 16) is satisfied.

(Def. 16) Let Y be a set. Suppose $Y \subseteq \operatorname{field} R$ and $Y \neq \emptyset$. Then there exists a set a such that $a \in Y$ and for every set b such that $b \in Y$ and $a \neq b$ holds $\langle a, b\rangle \notin R$.

The scheme *coNoetherianInduction* deals with a binary relation $\mathcal{A}$ and a unary predicate $\mathcal{P}$, and states that:

For every set a such that $a \in \operatorname{field} \mathcal{A}$ holds $\mathcal{P}[a]$

provided the parameters meet the following conditions:

- $\mathcal{A}$ is reversely well founded,
- For every set a such that for every set b such that $\langle a, b\rangle \in \mathcal{A}$ and $a \neq b$ holds $\mathcal{P}[b]$ holds $\mathcal{P}[a]$.

One can check that every binary relation which is strongly-normalizing is also irreflexive and reversely well founded and every binary relation which is reversely well founded and irreflexive is also strongly-normalizing.

Let us note that every binary relation which is empty is also weakly-normalizing and strongly-normalizing.

Let us note that there exists a binary relation which is empty.

Next we state the proposition

(48) Let Q be a reversely well founded binary relation and let R be a binary relation. If $R \subseteq Q$, then R is reversely well founded.

Let us observe that every binary relation which is strongly-normalizing is also weakly-normalizing.

4. Church-Rosser property

Let R, Q be binary relations. We say that R commutes-weakly with Q if and only if the condition (Def. 17) is satisfied.

(Def. 17) Let a, b, c be sets. Suppose $\langle a, b\rangle \in R$ and $\langle a, c\rangle \in Q$. Then there exists a set d such that Q reduces b to d and R reduces c to d.

Let us notice that the predicate defined above is symmetric. We say that R commutes with Q if and only if the condition (Def. 18) is satisfied.

(Def. 18) Let a, b, c be sets. Suppose R reduces a to b and Q reduces a to c. Then there exists a set d such that Q reduces b to d and R reduces c to d.

Let us notice that the predicate introduced above is symmetric.

We now state the proposition

(49) For all binary relations R, Q such that R commutes with Q holds R commutes-weakly with Q.

Let R be a binary relation. We say that R has unique normal form property if and only if the condition (Def. 19) is satisfied.

(Def. 19) Let a, b be sets. Suppose a is a normal form w.r.t. R and b is a normal form w.r.t. R and a and b are convertible w.r.t. R. Then $a = b$.

We say that R has normal form property if and only if the condition (Def. 20) is satisfied.

(Def. 20) Let a, b be sets. Suppose a is a normal form w.r.t. R and a and b are convertible w.r.t. R. Then R reduces b to a.

We say that R is subcommutative if and only if:

(Def. 21) For all sets a, b, c such that $\langle a, b\rangle \in R$ and $\langle a, c\rangle \in R$ holds b and c are convergent at most in 1 step w.r.t. R.

We introduce R has diamond property as a synonym of R is subcommutative. We say that R is confluent if and only if:

(Def. 22) For all sets a, b such that a and b are divergent w.r.t. R holds a and b are convergent w.r.t. R.

We say that R has Church-Rosser property if and only if:

(Def. 23) For all sets a, b such that a and b are convertible w.r.t. R holds a and b are convergent w.r.t. R.

We say that R is locally-confluent if and only if:

(Def. 24) For all sets a, b, c such that $\langle a, b\rangle \in R$ and $\langle a, c\rangle \in R$ holds b and c are convergent w.r.t. R.

We introduce R has weak Church-Rosser property as a synonym of R is locally-confluent.

Next we state four propositions:

(50) Let R be a binary relation. Suppose R is subcommutative. Let a, b, c be sets. Suppose R reduces a to b and $\langle a, c\rangle \in R$. Then b and c are convergent w.r.t. R.

(51) For every binary relation R holds R is confluent iff R commutes with R.

(52) Let R be a binary relation. Then R is confluent if and only if for all sets a, b, c such that R reduces a to b and $\langle a, c\rangle \in R$ holds b and c are convergent w.r.t. R

(53) For every binary relation R holds R is locally-confluent iff R commutes-weakly with R.

One can verify the following observations:

* every binary relation which has Church-Rosser property is confluent,
* every binary relation which is confluent is also locally-confluent and has Church-Rosser property,
* every binary relation which is subcommutative is also confluent,
* every binary relation which has Church-Rosser property has also normal form property,
* every binary relation which has normal form property has also unique normal form property, and

* every binary relation which is weakly-normalizing and has unique normal form property has Church-Rosser property.

One can check that every binary relation which is empty is also subcommutative.

One can verify that there exists a binary relation which is empty.

The following three propositions are true:

(54) Let R be a binary relation with unique normal form property and let a, b, c be sets. Suppose b is a normal form of a w.r.t. R and c is a normal form of a w.r.t. R. Then $b = c$.

(55) Let R be a weakly-normalizing binary relation with unique normal form property and let a be a set. Then $\mathrm{nf}_R(a)$ is a normal form of a w.r.t. R.

(56) Let R be a weakly-normalizing binary relation with unique normal form property and let a, b be sets. If a and b are convertible w.r.t. R, then $\mathrm{nf}_R(a) = \mathrm{nf}_R(b)$.

Let us note that every binary relation which is strongly-normalizing and locally-confluent is also confluent.

Let R be a binary relation. We say that R is complete if and only if:

(Def. 25) R is confluent and strongly-normalizing.

Let us note that every binary relation which is complete is also confluent and strongly-normalizing and every binary relation which is confluent and strongly-normalizing is also complete.

Let us mention that there exists a binary relation which is empty.

Let us note that there exists a non empty binary relation which is complete.

We now state three propositions:

(57) Let R, Q be binary relations with Church-Rosser property. If R commutes with Q, then $R \cup Q$ has Church-Rosser property.

(58) For every binary relation R holds R is confluent iff R^* has weak Church-Rosser property.

(59) For every binary relation R holds R is confluent iff R^* is subcommutative.

5. Completion method

Let R, Q be binary relations. We say that R and Q are equivalent if and only if the condition (Def. 26) is satisfied.

(Def. 26) Let a, b be sets. Then a and b are convertible w.r.t. R if and only if a and b are convertible w.r.t. Q.

Let us observe that the predicate introduced above is symmetric.

Let R be a binary relation and let a, b be sets. We say that a and b are critical w.r.t. R if and only if:

(Def. 27) a and b are divergent at most in 1 step w.r.t. R and a and b are not convergent w.r.t. R.

We now state four propositions:

(60) Let R be a binary relation and let a, b be sets. Suppose a and b are critical w.r.t. R. Then a and b are convertible w.r.t. R.

(61) Let R be a binary relation. Suppose that it is not true that there exist sets a, b such that a and b are critical w.r.t. R Then R is locally-confluent.

(62) Let R, Q be binary relations. Suppose that for all sets a, b such that $\langle a, b\rangle \in Q$ holds a and b are critical w.r.t. R. Then R and $R \cup Q$ are equivalent.

(63) Let R be a binary relation. Then there exists a complete binary relation Q such that

(i) $\mathrm{field}\, Q \subseteq \mathrm{field}\, R$, and

(ii) for all sets a, b holds a and b are convertible w.r.t. R iff a and b are convergent w.r.t. Q.

Let R be a binary relation. A complete binary relation is said to be a completion of R if it satisfies the condition (Def. 28).

(Def. 28) Let a, b be sets. Then a and b are convertible w.r.t. R if and only if a and b are convergent w.r.t. it.

Next we state three propositions:

(64) For every binary relation R and for every completion C of R holds R and C are equivalent.

(65) Let R be a binary relation and let Q be a complete binary relation. If R and Q are equivalent, then Q is a completion of R.

(66) Let R be a binary relation, and let C be a completion of R, and let a, b be sets. Then a and b are convertible w.r.t. R if and only if $\mathrm{nf}_C(a) = \mathrm{nf}_C(b)$.

References

[1] S. Abramsky, D. M. Gabbay, and S. E. Maibaum, editors. *Handbook of Logic in Computer Science, vol. 2: Computational structures*. Clarendon Press, Oxford, 1992.

[2] Leo Bachmair and Nachum Dershowitz. Critical pair criteria for completion. *Journal of Symbolic Computation*, 6(1):1–18, 1988.

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

[4] Grzegorz Bancerek. The well ordering relations. *Formalized Mathematics*, 1(**1**):123–129, 1990.

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

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

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

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

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

[10] Gerard Huet. A complete proof of correctness of the knuth-bendix completion. *Journal of Computer and System Sciences*, 23(1):3–57, 1981.
[11] Jan Willem Klop and Aart Middeldrop. An introduction to knuth-bendix completion. *CWI Quarterly*, 1(3):31–52, 1988.
[12] Donald E. Knuth and Peter B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebras*, pages 263–297, Pergamon, Oxford, 1970.
[13] Konrad Raczkowski and Paweł Sadowski. Equivalence relations and classes of abstraction. *Formalized Mathematics*, 1(**3**):441–444, 1990.
[14] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[15] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[16] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[17] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[18] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[19] Edmund Woronowicz and Anna Zalewska. Properties of binary relations. *Formalized Mathematics*, 1(**1**):85–89, 1990.

Received November 14, 1995

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

Lattice of Congruences in Many Sorted Algebra

Robert Milewski
Warsaw University
Białystok

MML Identifier: MSUALG_5.

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

1. More on Equivalence Relations

For simplicity we adopt the following convention: I, X denote sets, M denotes a many sorted set indexed by I, R_1 denotes a binary relation on X, and E_1, E_2, E_3 denote equivalence relations of X.

We now state the proposition

(1) $(E_1 \sqcup E_2) \sqcup E_3 = E_1 \sqcup (E_2 \sqcup E_3)$.

Let X be a set and let R be a binary relation on X. The functor $\mathrm{EqCl}(R)$ yielding an equivalence relation of X is defined as follows:

(Def. 1) $R \subseteq \mathrm{EqCl}(R)$ and for every equivalence relation E_2 of X such that $R \subseteq E_2$ holds $\mathrm{EqCl}(R) \subseteq E_2$.

One can prove the following propositions:

(2) $E_1 \sqcup E_2 = \mathrm{EqCl}(E_1 \cup E_2)$.

(3) $\mathrm{EqCl}(E_1) = E_1$.

(4) $\nabla_X \cup R_1 = \nabla_X$.

ISSN 1426–2630

2. Lattice of Equivalence Relations

Let X be a set. The functor $\mathrm{EqRelLatt}(X)$ yields a strict lattice and is defined by the conditions (Def. 2).

(Def. 2) (i) The carrier of $\mathrm{EqRelLatt}(X) = \{x : x$ ranges over relations between X and X, x is an equivalence relation of $X\}$, and

(ii) for all equivalence relations x, y of X holds (the meet operation of $\mathrm{EqRelLatt}(X))(x, y) = x \cap y$ and (the join operation of $\mathrm{EqRelLatt}(X))(x, y) = x \sqcup y$.

3. Many Sorted Equivalence Relations

Let us consider I, M. Note that there exists a many sorted relation of M which is equivalence.

Let us consider I, M. An equivalence relation of M is an equivalence many sorted relation of M.

We adopt the following convention: I will denote a non empty set, M will denote a many sorted set indexed by I, and E_4, E_1, E_2, E_3 will denote equivalence relations of M.

Let I be a non empty set, let M be a many sorted set indexed by I, and let R be a many sorted relation of M. The functor $\mathrm{EqCl}(R)$ yields an equivalence relation of M and is defined as follows:

(Def. 3) For every element i of I holds $(\mathrm{EqCl}(R))(i) = \mathrm{EqCl}(R(i))$.

The following proposition is true

(5) $\mathrm{EqCl}(E_4) = E_4$.

4. Lattice of Many Sorted Equivalence Relations

Let I be a non empty set, let M be a many sorted set indexed by I, and let E_1, E_2 be equivalence relations of M. The functor $E_1 \sqcup E_2$ yielding an equivalence relation of M is defined as follows:

(Def. 4) There exists a many sorted relation E_3 of M such that $E_3 = E_1 \cup E_2$ and $E_1 \sqcup E_2 = \mathrm{EqCl}(E_3)$.

Let us observe that the functor introduced above is commutative.

Next we state several propositions:

(6) $E_1 \cup E_2 \subseteq E_1 \sqcup E_2$.

(7) For every equivalence relation E_4 of M such that $E_1 \cup E_2 \subseteq E_4$ holds $E_1 \sqcup E_2 \subseteq E_4$.

(8) If $E_1 \cup E_2 \subseteq E_3$ and for every equivalence relation E_4 of M such that $E_1 \cup E_2 \subseteq E_4$ holds $E_3 \subseteq E_4$, then $E_3 = E_1 \sqcup E_2$.

(9) $E_4 \sqcup E_4 = E_4$.

(10) $(E_1 \sqcup E_2) \sqcup E_3 = E_1 \sqcup (E_2 \sqcup E_3)$.

(11) $E_1 \cap (E_1 \sqcup E_2) = E_1$.

(12) For every equivalence relation E_4 of M such that $E_4 = E_1 \cap E_2$ holds $E_1 \sqcup E_4 = E_1$.

(13) For all equivalence relations E_1, E_2 of M holds $E_1 \cap E_2$ is an equivalence relation of M.

Let I be a non empty set and let M be a many sorted set indexed by I. The functor $\mathrm{EqRelLatt}(M)$ yielding a strict lattice is defined by the conditions (Def. 5).

(Def. 5) (i) For arbitrary x holds $x \in$ the carrier of $\mathrm{EqRelLatt}(M)$ iff x is an equivalence relation of M, and

(ii) for all equivalence relations x, y of M holds (the meet operation of $\mathrm{EqRelLatt}(M))(x, y) = x \cap y$ and (the join operation of $\mathrm{EqRelLatt}(M))(x, y) = x \sqcup y$.

5. Lattice of Congruences in Many Sorted Algebra

Let S be a non empty many sorted signature and let A be an algebra over S Note that every many sorted relation of A which is equivalence is also equivalence.

In the sequel S will denote a non void non empty many sorted signature and A will denote a non-empty algebra over S.

Next we state several propositions:

(14) Let o be an operation symbol of S, and let C_1, C_2 be congruences of A, and let x_1, y_1 be arbitrary, and let a_1, b_1 be finite sequences. Suppose $\langle x_1, y_1 \rangle \in C_1(\pi_{\mathrm{len}\, a_1+1} \mathrm{Arity}(o)) \cup C_2(\pi_{\mathrm{len}\, a_1+1} \mathrm{Arity}(o))$. Let x, y be elements of $\mathrm{Args}(o, A)$. Suppose $x = a_1 \frown \langle x_1 \rangle \frown b_1$ and $y = a_1 \frown \langle y_1 \rangle \frown b_1$. Then $\langle (\mathrm{Den}(o, A))(x), (\mathrm{Den}(o, A))(y) \rangle \in C_1(\text{the result sort of } o) \cup C_2(\text{the result sort of } o)$.

(15) Let o be an operation symbol of S, and let C_1, C_2 be congruences of A, and let C be an equivalence many sorted relation of A. Suppose $C = C_1 \sqcup C_2$. Let x_1, y_1 be arbitrary, and let n be a natural number, and let a_1, a_2, b_1 be finite sequences. Suppose $\mathrm{len}\, a_1 = n$ and $\mathrm{len}\, a_1 = \mathrm{len}\, a_2$ and for every natural number k such that $k \in \mathrm{dom}\, a_1$ holds $\langle a_1(k), a_2(k) \rangle \in C(\pi_k \mathrm{Arity}(o))$. Suppose $\langle (\mathrm{Den}(o, A))(a_1 \frown \langle x_1 \rangle \frown b_1), (\mathrm{Den}(o, A))(a_2 \frown \langle x_1 \rangle \frown b_1) \rangle \in C(\text{the result sort of } o)$ and $\langle x_1, y_1 \rangle \in C(\pi_{n+1} \mathrm{Arity}(o))$. Let x be an element of $\mathrm{Args}(o, A)$. If $x = a_1 \frown \langle x_1 \rangle \frown b_1$, then $\langle (\mathrm{Den}(o, A))(x), (\mathrm{Den}(o, A))(a_2 \frown \langle y_1 \rangle \frown b_1) \rangle \in C(\text{the result sort of } o)$.

(16) Let o be an operation symbol of S, and let C_1, C_2 be congruences of A, and let C be an equivalence many sorted relation of A. Suppose $C = C_1 \sqcup C_2$. Let x, y be elements of $\mathrm{Args}(o, A)$. Suppose that for every natural number n such that $n \in \mathrm{dom}\, x$ holds $\langle x(n), y(n)\rangle \in C(\pi_n \mathrm{Arity}(o))$. Then $\langle(\mathrm{Den}(o, A))(x), (\mathrm{Den}(o, A))(y)\rangle \in C(\text{the result sort of } o)$.

(17) For all congruences C_1, C_2 of A holds $C_1 \sqcup C_2$ is a congruence of A.

(18) For all congruences C_1, C_2 of A holds $C_1 \cap C_2$ is a congruence of A.

Let us consider S and let A be a non-empty algebra over S. The functor $\mathrm{CongrLatt}(A)$ yielding a strict sublattice of $\mathrm{EqRelLatt}(\text{the sorts of } A)$ is defined by:

(Def. 6) For arbitrary x holds $x \in$ the carrier of $\mathrm{CongrLatt}(A)$ iff x is a congruence of A.

We now state four propositions:

(19) $\mathrm{id}_{(\text{the sorts of } A)}$ is a congruence of A.

(20) $[\![\text{the sorts of } A, \text{the sorts of } A]\!]$ is a congruence of A.

(21) $\bot_{\mathrm{CongrLatt}(A)} = \mathrm{id}_{(\text{the sorts of } A)}$.

(22) $\top_{\mathrm{CongrLatt}(A)} = [\![\text{the sorts of } A, \text{the sorts of } A]\!]$.

Let us consider S and let us consider A. One can check that $\mathrm{CongrLatt}(A)$ is bounded.

References

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

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

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

[4] Ewa Burakowska. Subalgebras of many sorted algebra. Lattice of subalgebras. *Formalized Mathematics*, 5(**1**):47–54, 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] Marek Chmur. The lattice of natural numbers and the sublattice of it. The set of prime numbers. *Formalized Mathematics*, 2(**4**):453–459, 1991.

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

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

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

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

[16] Konrad Raczkowski and Paweł Sadowski. Equivalence relations and classes of abstraction. *Formalized Mathematics*, 1(**3**):441–444, 1990.
[17] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[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. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 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 January 11, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

Miscellaneous Facts about Functions

Grzegorz Bancerek
Institute of Mathematics
Polish Academy of Sciences

Andrzej Trybulec
Warsaw University
Białystok

MML Identifier: FUNCT_7.

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

1. Preliminaries

For simplicity we adopt the following rules: x is arbitrary, m, n are natural numbers, f, g are functions, and A, B are sets.

We now state several propositions:

(1) For every function f and for every set X such that $\operatorname{rng} f \subseteq X$ holds $\mathrm{id}_X \cdot f = f$.

(2) Let X be a set, and let Y be a non empty set, and let f be a function from X into Y. Suppose f is one-to-one. Let B be a subset of X and let C be a subset of Y. If $C \subseteq f^\circ B$, then $f^{-1} C \subseteq B$.

(3) Let X, Y be non empty sets and let f be a function from X into Y. Suppose f is one-to-one. Let x be an element of X and let A be a subset of X. If $f(x) \in f^\circ A$, then $x \in A$.

(4) Let X, Y be non empty sets and let f be a function from X into Y. Suppose f is one-to-one. Let x be an element of X, and let A be a subset of X, and let B be a subset of Y. If $f(x) \in f^\circ A \setminus B$, then $x \in A \setminus f^{-1} B$.

(5) Let X, Y be non empty sets and let f be a function from X into Y. Suppose f is one-to-one. Let y be an element of Y, and let A be a subset of X, and let B be a subset of Y. If $y \in f^\circ A \setminus B$, then $f^{-1}(y) \in A \setminus f^{-1} B$.

(6) For every function f and for arbitrary a such that $a \in \operatorname{dom} f$ holds $f \restriction \{a\} = a \dot{\longmapsto} f(a)$.

ISSN 1426–2630

Let x, y be arbitrary. Observe that $x \dot{\longmapsto} y$ is non empty.

Let x, y, a, b be arbitrary. One can check that $[x \longmapsto a, y \longmapsto b]$ is non empty.

One can prove the following propositions:

(7) For every set I and for every many sorted set M indexed by I and for arbitrary i such that $i \in I$ holds $i \dot{\longmapsto} M(i) = M \restriction \{i\}$.

(8) Let I, J be sets, and let M be a many sorted set indexed by $[\!: I, J :\!]$, and let i, j be arbitrary. If $i \in I$ and $j \in J$, then $[\langle i, j \rangle \mapsto M(i, j)] = M \restriction [\!: \{i\}, \{j\} :\!]$.

(9) If $x \in \operatorname{dom} f$ and $x \notin \operatorname{dom} g$, then $(f {+\cdot} g)(x) = f(x)$.

(10) For all functions f, g, h such that $\operatorname{rng} g \subseteq \operatorname{dom} f$ and $\operatorname{rng} h \subseteq \operatorname{dom} f$ holds $f \cdot (g {+\cdot} h) = f \cdot g {+\cdot} f \cdot h$.

(11) For all functions f, g, h holds $(g {+\cdot} h) \cdot f = g \cdot f {+\cdot} h \cdot f$.

(12) For all functions f, g, h such that $\operatorname{rng} f$ misses $\operatorname{dom} g$ holds $(h {+\cdot} g) \cdot f = h \cdot f$.

(13) For all sets A, B and for arbitrary y such that A meets $\operatorname{rng}(\mathrm{id}_B {+\cdot} (A \longmapsto y))$ holds $y \in A$.

(14) For arbitrary x, y and for every set A such that $x \neq y$ holds $x \notin \operatorname{rng}(\mathrm{id}_A {+\cdot} (x \dot{\longmapsto} y))$.

(15) For every set X and for arbitrary a and for every function f such that $\operatorname{dom} f = X \cup \{a\}$ holds $f = f \restriction X {+\cdot} (a \dot{\longmapsto} f(a))$.

(16) For every function f and for all sets X, y, z holds $f {+\cdot} (X \longmapsto y) {+\cdot} (X \longmapsto z) = f {+\cdot} (X \longmapsto z)$.

(17) If $0 < m$ and $m \leq n$, then $\mathbb{Z}_m \subseteq \mathbb{Z}_n$.

(18) $\mathbb{Z} \neq \mathbb{Z}^*$.

(19) $\emptyset^* = \{\emptyset\}$.

(20) $\langle x \rangle \in A^*$ iff $x \in A$.

(21) $A \subseteq B$ iff $A^* \subseteq B^*$.

(22) For every subset A of $\mathbb{N}$ such that for all n, m such that $n \in A$ and $m < n$ holds $m \in A$ holds A is a cardinal number.

(23) Let A be a finite set and let X be a non empty family of subsets of A. Then there exists an element C of X such that for every element B of X such that $B \subseteq C$ holds $B = C$.

(24) Let p, q be finite sequences. Suppose $\operatorname{len} p = \operatorname{len} q + 1$. Let i be a natural number. Then $i \in \operatorname{dom} q$ if and only if the following conditions are satisfied:

(i) $i \in \operatorname{dom} p$, and

(ii) $i + 1 \in \operatorname{dom} p$.

Let us note that there exists a finite sequence which is function yielding non empty and non-empty.

Note that ε is function yielding. Let f be a function. Observe that $\langle f \rangle$ is function yielding. Let g be a function. One can check that $\langle f, g \rangle$ is function yielding. Let h be a function. Observe that $\langle f, g, h \rangle$ is function yielding.

Let n be a natural number and let f be a function. One can verify that $n \mapsto f$ is function yielding.

Let p be a finite sequence and let q be a non empty finite sequence. One can verify that $p \frown q$ is non empty and $q \frown p$ is non empty.

Let p, q be function yielding finite sequences. Note that $p \frown q$ is function yielding.

Next we state the proposition

(25) Let p, q be finite sequences. Suppose $p \frown q$ is function yielding. Then p is function yielding and q is function yielding.

2. Some useful schemes

In this article we present several logical schemes. The scheme *KappaD* concerns non empty sets $\mathcal{A}$, $\mathcal{B}$ and a unary functor $\mathcal{F}$ yielding arbitrary, and states that:

There exists a function 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 meet the following condition:

- For every element x of $\mathcal{A}$ holds $\mathcal{F}(x) \in \mathcal{B}$.

The scheme *Kappa2D* deals with non empty sets $\mathcal{A}$, $\mathcal{B}$, $\mathcal{C}$ and a binary functor $\mathcal{F}$ yielding arbitrary, and states that:

There exists a function 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 parameters meet the following requirement:

- For every element x of $\mathcal{A}$ and for every element y of $\mathcal{B}$ holds $\mathcal{F}(x, y) \in \mathcal{C}$.

The scheme *FinMono* concerns a set $\mathcal{A}$, a non empty set $\mathcal{B}$, and two unary functors $\mathcal{F}$ and $\mathcal{G}$ yielding arbitrary, and states that:

$\{\mathcal{F}(d) : d$ ranges over elements of $\mathcal{B}$, $\mathcal{G}(d) \in \mathcal{A}\}$ is finite

provided the following conditions are satisfied:

- $\mathcal{A}$ is finite,
- For all elements d_1, d_2 of $\mathcal{B}$ such that $\mathcal{G}(d_1) = \mathcal{G}(d_2)$ holds $d_1 = d_2$.

The scheme *CardMono* concerns a set $\mathcal{A}$, a non empty set $\mathcal{B}$, and a unary functor $\mathcal{F}$ yielding arbitrary, and states that:

$\mathcal{A} \approx \{d : d$ ranges over elements of $\mathcal{B}$, $\mathcal{F}(d) \in \mathcal{A}\}$

provided the following requirements are met:

- For arbitrary x such that $x \in \mathcal{A}$ there exists an element d of $\mathcal{B}$ such that $x = \mathcal{F}(d)$,
- For all elements d_1, d_2 of $\mathcal{B}$ such that $\mathcal{F}(d_1) = \mathcal{F}(d_2)$ holds $d_1 = d_2$.

The scheme *CardMono'* concerns a set $\mathcal{A}$, a non empty set $\mathcal{B}$, and a unary functor $\mathcal{F}$ yielding arbitrary, and states that:

$\mathcal{A} \approx \{\mathcal{F}(d) : d$ ranges over elements of $\mathcal{B}$, $d \in \mathcal{A}\}$
provided the following conditions are satisfied:
- $\mathcal{A} \subseteq \mathcal{B}$,
- For all elements d_1, d_2 of $\mathcal{B}$ such that $\mathcal{F}(d_1) = \mathcal{F}(d_2)$ holds $d_1 = d_2$.

The scheme *FuncSeqInd* concerns a unary predicate $\mathcal{P}$, and states that:
For every function yielding finite sequence p holds $\mathcal{P}[p]$
provided the following conditions are satisfied:
- $\mathcal{P}[\varepsilon]$,
- For every function yielding finite sequence p such that $\mathcal{P}[p]$ and for every function f holds $\mathcal{P}[p \frown \langle f \rangle]$.

3. Some auxiliary concepts

Let x be arbitrary and let y be a set. Let us assume that $x \in y$. The functor $x(\in y)$ yielding an element of y is defined as follows:

(Def. 1) $x(\in y) = x$.

One can prove the following proposition

(26) If $x \in A \cap B$, then $x(\in A) = x(\in B)$.

Let f, g be functions and let A be a set. We say that f and g equal outside A if and only if:

(Def. 2) $f \restriction (\operatorname{dom} f \setminus A) = g \restriction (\operatorname{dom} g \setminus A)$.

Next we state several propositions:

(27) For every function f and for every set A holds f and f equal outside A.

(28) For all functions f, g and for every set A such that f and g equal outside A holds g and f equal outside A

(29) Let f, g, h be functions and let A be a set. Suppose f and g equal outside A and g and h equal outside A. Then f and h equal outside A.

(30) For all functions f, g and for every set A such that f and g equal outside A holds $\operatorname{dom} f \setminus A = \operatorname{dom} g \setminus A$.

(31) For all functions f, g and for every set A such that $\operatorname{dom} g \subseteq A$ holds f and $f {+\cdot} g$ equal outside A

Let f be a function and let i, x be arbitrary. The functor $f +\cdot (i, x)$ yields a function and is defined by:

(Def. 3) (i) $f +\cdot (i, x) = f {+\cdot} (i \longmapsto x)$ if $i \in \operatorname{dom} f$,
(ii) $f +\cdot (i, x) = f$, otherwise.

Next we state several propositions:

(32) For every function f and for arbitrary d, i holds $\operatorname{dom}(f +\cdot (i, d)) = \operatorname{dom} f$.

(33) For every function f and for arbitrary d, i such that $i \in \operatorname{dom} f$ holds $(f +\cdot (i, d))(i) = d$.

(34) For every function f and for arbitrary d, i, j such that $i \neq j$ and $j \in \operatorname{dom} f$ holds $(f +\cdot (i,d))(j) = f(j)$.

(35) For every function f and for arbitrary d, e, i, j such that $i \neq j$ holds $f +\cdot (i,d) +\cdot (j,e) = f +\cdot (j,e) +\cdot (i,d)$.

(36) For every function f and for arbitrary d, e, i holds $f +\cdot (i,d) +\cdot (i,e) = f +\cdot (i,e)$.

(37) For every function f and for arbitrary i holds $f +\cdot (i, f(i)) = f$.

Let f be a finite sequence, let i be a natural number, and let x be arbitrary. One can check that $f +\cdot (i,x)$ is finite sequence-like.

Let D be a set, let f be a finite sequence of elements of D, let i be a natural number, and let d be an element of D. Then $f +\cdot (i,d)$ is a finite sequence of elements of D.

The following three propositions are true:

(38) Let D be a non empty set, and let f be a finite sequence of elements of D, and let d be an element of D, and let i be a natural number. If $i \in \operatorname{dom} f$, then $\pi_i(f +\cdot (i,d)) = d$.

(39) Let D be a non empty set, and let f be a finite sequence of elements of D, and let d be an element of D, and let i, j be natural numbers. If $i \neq j$ and $j \in \operatorname{dom} f$, then $\pi_j(f +\cdot (i,d)) = \pi_j f$.

(40) Let D be a non empty set, and let f be a finite sequence of elements of D, and let d, e be elements of D, and let i be a natural number. Then $f +\cdot (i, \pi_i f) = f$.

4. On the composition of a finite sequence of functions

Let X be a set and let p be a function yielding finite sequence. The functor $\operatorname{compose}_X p$ yielding a function is defined by the condition (Def. 4).

(Def. 4) There exists a many sorted function f of $\mathbb{N}$ such that

(i) $\operatorname{compose}_X p = f(\operatorname{len} p)$,

(ii) $f(0) = \operatorname{id}_X$, and

(iii) for every natural number i such that $i+1 \in \operatorname{dom} p$ and for all functions g, h such that $g = f(i)$ and $h = p(i+1)$ holds $f(i+1) = h \cdot g$.

Let p be a function yielding finite sequence and let x be a set. The functor $\operatorname{apply}(p,x)$ yields a finite sequence and is defined by the conditions (Def. 5).

(Def. 5) (i) $\operatorname{len} \operatorname{apply}(p,x) = \operatorname{len} p + 1$,

(ii) $(\operatorname{apply}(p,x))(1) = x$, and

(iii) for every natural number i and for every function f such that $i \in \operatorname{dom} p$ and $f = p(i)$ holds $(\operatorname{apply}(p,x))(i+1) = f((\operatorname{apply}(p,x))(i))$.

We adopt the following convention: X, Y, x denote sets, p, q denote function yielding finite sequences, and f, g, h denote functions.

The following propositions are true:

(41) $\mathrm{compose}_X\, \varepsilon = \mathrm{id}_X$.

(42) $\mathrm{apply}(\varepsilon, x) = \langle x \rangle$.

(43) $\mathrm{compose}_X (p \frown \langle f \rangle) = f \cdot \mathrm{compose}_X\, p$.

(44) $\mathrm{apply}(p \frown \langle f \rangle, x) = (\mathrm{apply}(p, x)) \frown \langle f((\mathrm{apply}(p, x))(\mathrm{len}\, p + 1)) \rangle$.

(45) $\mathrm{compose}_X (\langle f \rangle \frown p) = \mathrm{compose}_{f \circ X}\, p \cdot (f \restriction X)$.

(46) $\mathrm{apply}(\langle f \rangle \frown p, x) = \langle x \rangle \frown \mathrm{apply}(p, f(x))$.

(47) $\mathrm{compose}_X \langle f \rangle = f \cdot \mathrm{id}_X$.

(48) If $\mathrm{dom}\, f \subseteq X$, then $\mathrm{compose}_X \langle f \rangle = f$.

(49) $\mathrm{apply}(\langle f \rangle, x) = \langle x, f(x) \rangle$.

(50) If $\mathrm{rng}\, \mathrm{compose}_X\, p \subseteq Y$, then $\mathrm{compose}_X (p \frown q) = \mathrm{compose}_Y\, q \cdot \mathrm{compose}_X\, p$.

(51) $(\mathrm{apply}(p \frown q, x))(\mathrm{len}(p \frown q) + 1) = (\mathrm{apply}(q, (\mathrm{apply}(p, x))(\mathrm{len}\, p + 1)))(\mathrm{len}\, q + 1)$.

(52) $\mathrm{apply}(p \frown q, x) = (\mathrm{apply}(p, x))\ {}^{\$}\!\frown \mathrm{apply}(q, (\mathrm{apply}(p, x))(\mathrm{len}\, p + 1))$.

(53) $\mathrm{compose}_X \langle f, g \rangle = g \cdot f \cdot \mathrm{id}_X$.

(54) If $\mathrm{dom}\, f \subseteq X$ or $\mathrm{dom}(g \cdot f) \subseteq X$, then $\mathrm{compose}_X \langle f, g \rangle = g \cdot f$.

(55) $\mathrm{apply}(\langle f, g \rangle, x) = \langle x, f(x), g(f(x)) \rangle$.

(56) $\mathrm{compose}_X \langle f, g, h \rangle = h \cdot g \cdot f \cdot \mathrm{id}_X$.

(57) If $\mathrm{dom}\, f \subseteq X$ or $\mathrm{dom}(g \cdot f) \subseteq X$ or $\mathrm{dom}(h \cdot g \cdot f) \subseteq X$, then $\mathrm{compose}_X \langle f, g, h \rangle = h \cdot g \cdot f$.

(58) $\mathrm{apply}(\langle f, g, h \rangle, x) = \langle x \rangle \frown \langle f(x), g(f(x)), h(g(f(x))) \rangle$.

Let F be a finite sequence. The functor $\mathrm{firstdom}(F)$ is defined as follows:

(Def. 6) (i) $\mathrm{firstdom}(F)$ is empty if F is empty,

(ii) $\mathrm{firstdom}(F) = \pi_1(F(1))$, otherwise.

The functor $\mathrm{lastrng}(F)$ is defined by:

(Def. 7) (i) $\mathrm{lastrng}(F)$ is empty if F is empty,

(ii) $\mathrm{lastrng}(F) = \pi_2(F(\mathrm{len}\, F))$, otherwise.

Next we state three propositions:

(59) $\mathrm{firstdom}(\varepsilon) = \emptyset$ and $\mathrm{lastrng}(\varepsilon) = \emptyset$.

(60) For every finite sequence p holds $\mathrm{firstdom}(\langle f \rangle \frown p) = \mathrm{dom}\, f$ and $\mathrm{lastrng}(p \frown \langle f \rangle) = \mathrm{rng}\, f$.

(61) For every function yielding finite sequence p such that $p \neq \varepsilon$ holds $\mathrm{rng}\, \mathrm{compose}_X\, p \subseteq \mathrm{lastrng}(p)$.

Let I_1 be a finite sequence. We say that I_1 is composable if and only if:

(Def. 8) There exists a finite sequence p such that $\mathrm{len}\, p = \mathrm{len}\, I_1 + 1$ and for every natural number i such that $i \in \mathrm{dom}\, I_1$ holds $I_1(i) \in p(i+1)^{p(i)}$.

We now state the proposition

(62) For all finite sequences p, q such that $p \frown q$ is composable holds p is composable and q is composable.

One can verify that every finite sequence which is composable is also function yielding.

Let us observe that every finite sequence which is empty is also composable.

Let f be a function. One can check that $\langle f \rangle$ is composable.

Let us observe that there exists a finite sequence which is composable non empty and non-empty.

A composable sequence is a composable finite sequence.

Next we state several propositions:

(63) For every composable sequence p such that $p \neq \varepsilon$ holds $\mathrm{dom}\,\mathrm{compose}_X\, p = \mathrm{firstdom}(p) \cap X$.

(64) For every composable sequence p holds $\mathrm{dom}\,\mathrm{compose}_{\mathrm{firstdom}(p)}\, p = \mathrm{firstdom}(p)$.

(65) For every composable sequence p and for every function f such that $\mathrm{rng}\, f \subseteq \mathrm{firstdom}(p)$ holds $\langle f \rangle \frown p$ is a composable sequence.

(66) For every composable sequence p and for every function f such that $\mathrm{lastrng}(p) \subseteq \mathrm{dom}\, f$ holds $p \frown \langle f \rangle$ is a composable sequence.

(67) For every composable sequence p such that $x \in \mathrm{firstdom}(p)$ and $x \in X$ holds $(\mathrm{apply}(p, x))(\mathrm{len}\, p + 1) = (\mathrm{compose}_X\, p)(x)$.

Let X, Y be sets. Let us assume that if Y is empty, then X is empty. A composable sequence is called a composable sequence from X into Y if:

(Def. 9) $\mathrm{firstdom}(\mathrm{it}) = X$ and $\mathrm{lastrng}(\mathrm{it}) \subseteq Y$.

Let Y be a non empty set, let X be a set, and let F be a composable sequence from X into Y. Then $\mathrm{compose}_X\, F$ is a function from X into Y.

Let q be a non-empty non empty finite sequence. A finite sequence is said to be a composable sequence along q if:

(Def. 10) $\mathrm{len}\,\mathrm{it} + 1 = \mathrm{len}\, q$ and for every natural number i such that $i \in \mathrm{dom}\,\mathrm{it}$ holds $\mathrm{it}(i) \in q(i+1)^{q(i)}$.

Let q be a non-empty non empty finite sequence. Observe that every composable sequence along q is composable and non-empty.

One can prove the following three propositions:

(68) Let q be a non-empty non empty finite sequence and let p be a composable sequence along q. If $p \neq \varepsilon$, then $\mathrm{firstdom}(p) = q(1)$ and $\mathrm{lastrng}(p) \subseteq q(\mathrm{len}\, q)$.

(69) Let q be a non-empty non empty finite sequence and let p be a composable sequence along q. Then $\mathrm{dom}\,\mathrm{compose}_{q(1)}\, p = q(1)$ and $\mathrm{rng}\,\mathrm{compose}_{q(1)}\, p \subseteq q(\mathrm{len}\, q)$.

(70) For every function f and for every natural number n holds $f^n = \mathrm{compose}_{\mathrm{dom}\, f \cup \mathrm{rng}\, f}(n \mapsto f)$.

References

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

[2] Grzegorz Bancerek. Curried and uncurried functions. *Formalized Mathematics*, 1(**3**):537–541, 1990.
[3] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[4] Grzegorz Bancerek. Reduction relations. *Formalized Mathematics*, 5(**4**):469–478, 1996.
[5] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[6] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(**1**):175–180, 1990.
[7] Czesław Byliński. Cartesian categories. *Formalized Mathematics*, 3(**2**):161–169, 1992.
[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. Products and coproducts in categories. *Formalized Mathematics*, 2(**5**):701–709, 1991.
[14] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[15] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[16] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[17] Jarosław Kotowicz, Beata Madras, and Małgorzata Korolkiewicz. Basic notation of universal algebra. *Formalized Mathematics*, 3(**2**):251–253, 1992.
[18] Beata Madras. Product of family of universal algebras. *Formalized Mathematics*, 4(**1**):103–108, 1993.
[19] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[20] Dariusz Surowik. Cyclic groups and some of their properties - part I. *Formalized Mathematics*, 2(**5**):623–627, 1991.
[21] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[22] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.
[23] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[24] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[25] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[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. 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.

Received January 12, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

Examples of Category Structures

Andrzej Trybulec
Warsaw University
Białystok

Summary. We continue the formalization of the category theory.

MML Identifier: ALTCAT_2.

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

1. Preliminaries

One can prove the following proposition

(1) For all sets X_1, X_2 and for arbitrary a_1, a_2 holds $[\!: X_1 \longmapsto a_1, X_2 \longmapsto a_2 :\!] = [\!: X_1, X_2 :\!] \longmapsto \langle a_1, a_2 \rangle$.

Let I be a set. Observe that $\emptyset_I$ is function yielding.

The following two propositions are true:

(2) For all functions f, g holds $\curvearrowleft(g \cdot f) = g \cdot \curvearrowleft f$.

(3) For all functions f, g, h holds $\curvearrowleft(f \cdot [\!: g, h :\!]) = \curvearrowleft f \cdot [\!: h, g :\!]$.

Let f be a function yielding function. Observe that $\curvearrowleft f$ is function yielding.

One can prove the following proposition

(4) Let I be a set and let A, B, C be many sorted sets indexed by I. Suppose A is transformable to B. Let F be a many sorted function from A into B and let G be a many sorted function from B into C. Then $G \circ F$ is a many sorted function from A into C.

Let I be a set and let A be a many sorted set indexed by $[\!: I, I :\!]$. Then $\curvearrowleft A$ is a many sorted set indexed by $[\!: I, I :\!]$.

We now state the proposition

ISSN 1426–2630

(5) Let I_1 be a set, and let I_2 be a non empty set, and let f be a function from I_1 into I_2, and let B, C be many sorted sets indexed by I_2, and let G be a many sorted function from B into C. Then $G \cdot f$ is a many sorted function from $B \cdot f$ into $C \cdot f$.

Let I be a set, let A, B be many sorted sets indexed by $[\!: I, I :\!]$, and let F be a many sorted function from A into B. Then $\curvearrowleft F$ is a many sorted function from $\curvearrowleft A$ into $\curvearrowleft B$.

We now state the proposition

(6) Let I_1, I_2 be non empty sets, and let M be a many sorted set indexed by $[\!: I_1, I_2 :\!]$ and let o_1 be an element of I_1, and let o_2 be an element of I_2. Then $(\curvearrowleft M)(o_2, o_1) = M(o_1, o_2)$.

Let I_1 be a set and let f, g be many sorted functions of I_1 Then $g \circ f$ is a many sorted function of I_1.

2. An auxiliary notion

Let I, J be sets, let A be a many sorted set indexed by I, and let B be a many sorted set indexed by J. The predicate $A \mathrel{\dot{\subseteq}} B$ is defined as follows:

(Def. 1) $I \subseteq J$ and for arbitrary i such that $i \in I$ holds $A(i) \subseteq B(i)$.

One can prove the following four propositions:

(7) For every set I and for every many sorted set A indexed by I holds $A \mathrel{\dot{\subseteq}} A$.

(8) Let I, J be sets, and let A be a many sorted set indexed by I, and let B be a many sorted set indexed by J. If $A \mathrel{\dot{\subseteq}} B$ and $B \mathrel{\dot{\subseteq}} A$, then $A = B$.

(9) Let I, J, K be sets, and let A be a many sorted set indexed by I, and let B be a many sorted set indexed by J, and let C be a many sorted set indexed by K. If $A \mathrel{\dot{\subseteq}} B$ and $B \mathrel{\dot{\subseteq}} C$, then $A \mathrel{\dot{\subseteq}} C$.

(10) Let I be a set, and let A be a many sorted set indexed by I, and let B be a many sorted set indexed by I. Then $A \mathrel{\dot{\subseteq}} B$ if and only if $A \subseteq B$.

3. A bit of lambda calculus

In this article we present several logical schemes. The scheme *OnSingletons* deals with a non empty set $\mathcal{A}$, a unary functor $\mathcal{F}$ yielding arbitrary, and a unary predicate $\mathcal{P}$, and states that:

$\{\langle o, \mathcal{F}(o)\rangle : o$ ranges over elements of $\mathcal{A}$, $\mathcal{P}[o]\}$ is a function

for all values of the parameters.

The scheme *DomOnSingletons* deals with a non empty set $\mathcal{A}$, a function $\mathcal{B}$, a unary functor $\mathcal{F}$ yielding arbitrary, and a unary predicate $\mathcal{P}$, and states that:

$\operatorname{dom} \mathcal{B} = \{o : o$ ranges over elements of $\mathcal{A}$, $\mathcal{P}[o]\}$

provided the following condition is satisfied:

- $\mathcal{B} = \{\langle o, \mathcal{F}(o)\rangle : o$ ranges over elements of $\mathcal{A}$, $\mathcal{P}[o]\}$.

The scheme *ValOnSingletons* deals with a non empty set $\mathcal{A}$, a function $\mathcal{B}$, an element $\mathcal{C}$ of $\mathcal{A}$, a unary functor $\mathcal{F}$ yielding arbitrary, and a unary predicate $\mathcal{P}$, and states that:

$\mathcal{B}(\mathcal{C}) = \mathcal{F}(\mathcal{C})$

provided the following requirements are met:

- $\mathcal{B} = \{\langle o, \mathcal{F}(o)\rangle : o$ ranges over elements of $\mathcal{A}$, $\mathcal{P}[o]\}$,
- $\mathcal{P}[\mathcal{C}]$.

4. More on old categories

The following propositions are true:

(11) For every category C and for all objects i, j, k of C holds $[\!:\mathrm{hom}(j,k), \mathrm{hom}(i,j)\,:\!] \subseteq \mathrm{dom}\,(\text{the composition of } C)$.

(12) For every category C and for all objects i, j, k of C holds (the composition of $C)^\circ[\!:\mathrm{hom}(j,k), \mathrm{hom}(i,j)\,:\!] \subseteq \mathrm{hom}(i,k)$.

Let C be a category structure. The functor $\mathrm{HomSets}_C$ yields a many sorted set indexed by $[\!:$the objects of C, the objects of $C\,:\!]$ and is defined as follows:

(Def. 2) For all objects i, j of C holds $\mathrm{HomSets}_C(i, j) = \mathrm{hom}(i,j)$.

The following proposition is true

(13) For every category C and for every object i of C holds $\mathrm{id}_i \in \mathrm{HomSets}_C(i, i)$.

Let C be a category. The functor $\mathrm{Composition}_C$ yielding a binary composition of $\mathrm{HomSets}_C$ is defined by:

(Def. 3) For all objects i, j, k of C holds $\mathrm{Composition}_C(i, j, k) =$ (the composition of $C) \restriction [\!:\mathrm{HomSets}_C(j, k), \mathrm{HomSets}_C(i, j)\,:\!]$.

Next we state three propositions:

(14) Let C be a category and let i, j, k be objects of C Suppose $\mathrm{hom}(i,j) \neq \emptyset$ and $\mathrm{hom}(j,k) \neq \emptyset$. Let f be a morphism from i to j and let g be a morphism from j to k. Then $\mathrm{Composition}_C(i, j, k)(g, f) = g \cdot f$.

(15) For every category C holds $\mathrm{Composition}_C$ is associative.

(16) For every category C holds $\mathrm{Composition}_C$ has left units and right units.

5. Transforming an old category into a new one

Let C be a category. The functor $\mathrm{Alter}(C)$ yielding a strict non empty category structure is defined as follows:

(Def. 4) $\mathrm{Alter}(C) = \langle$the objects of C, $\mathrm{HomSets}_C, \mathrm{Composition}_C\rangle$.

We now state three propositions:

(17) For every category C holds $\mathrm{Alter}(C)$ is associative.

(18) For every category C holds $\mathrm{Alter}(C)$ has units.

(19) For every category C holds $\mathrm{Alter}(C)$ is transitive.

Let C be a category. Then $\mathrm{Alter}(C)$ is a strict category.

6. More on new categories

Let us note that there exists a graph which is non empty and strict.

Let C be a graph. We say that C is reflexive if and only if:

(Def. 5) For arbitrary x such that $x \in$ the carrier of C holds (the arrows of C)(x, x) $\neq \emptyset$.

Let C be a non empty graph. Let us observe that C is reflexive if and only if:

(Def. 6) For every object o of C holds $\langle o, o \rangle \neq \emptyset$.

Let C be a non empty category structure. Observe that the carrier of C is non empty.

Let C be a non empty transitive category structure. Let us observe that C is associative if and only if the condition (Def. 7) is satisfied.

(Def. 7) Let o_1, o_2, o_3, o_4 be objects of C and let f be a morphism from o_1 to o_2, and let g be a morphism from o_2 to o_3, and let h be a morphism from o_3 to o_4. If $\langle o_1, o_2 \rangle \neq \emptyset$ and $\langle o_2, o_3 \rangle \neq \emptyset$ and $\langle o_3, o_4 \rangle \neq \emptyset$, then $(h \cdot g) \cdot f = h \cdot (g \cdot f)$.

Let C be a non empty category structure. Let us observe that C has units if and only if the condition (Def. 8) is satisfied.

(Def. 8) Let o be an object of C. Then

(i) $\langle o, o \rangle \neq \emptyset$, and

(ii) there exists a morphism i from o to o such that for every object o' of C and for every morphism m' from o' to o and for every morphism m'' from o to o' holds if $\langle o', o \rangle \neq \emptyset$, then $i \cdot m' = m'$ and if $\langle o, o' \rangle \neq \emptyset$, then $m'' \cdot i = m''$.

Let us observe that every non empty category structure which has units is reflexive.

One can check that there exists a graph which is non empty and reflexive.

One can verify that there exists a category structure which is non empty and reflexive.

7. The empty category

The strict category structure $\emptyset_{CAT}$ is defined by:

(Def. 9) The carrier of $\emptyset_{CAT}$ is empty.

Let us note that $\emptyset_{CAT}$ is empty.

Let us mention that there exists a category structure which is empty and strict.

Next we state the proposition

(20) For every empty strict category structure E holds $E = \emptyset_{CAT}$.

8. Subcategories

Let C be a category structure. A category structure is said to be a substructure of C if it satisfies the conditions (Def. 10).

(Def. 10) (i) The carrier of it $\subseteq$ the carrier of C,

(ii) the arrows of it $\dot{\subseteq}$ the arrows of C, and

(iii) the composition of it $\dot{\subseteq}$ the composition of C.

In the sequel C, C_1, C_2, C_3 denote category structures.

The following propositions are true:

(21) C is a substructure of C.

(22) If C_1 is a substructure of C_2 and C_2 is a substructure of C_3, then C_1 is a substructure of C_3.

(23) Let C_1, C_2 be category structures. Suppose C_1 is a substructure of C_2 and C_2 is a substructure of C_1. Then the category structure of $C_1 =$ the category structure of C_2.

Let C be a category structure. One can check that there exists a substructure of C which is strict.

Let C be a non empty category structure and let o be an object of C. The functor $\Box\restriction o$ yielding a strict substructure of C is defined by the conditions (Def. 11).

(Def. 11) (i) The carrier of $\Box\restriction o = \{o\}$,

(ii) the arrows of $\Box\restriction o = [\langle o, o\rangle \mapsto \langle o, o\rangle]$, and

(iii) the composition of $\Box\restriction o = \langle o, o, o\rangle \dot{\longmapsto}(\text{the composition of } C)(o, o, o)$.

In the sequel C denotes a non empty category structure and o denotes an object of C.

One can prove the following proposition

(24) For every object o' of $\Box\restriction o$ holds $o' = o$.

Let C be a non empty category structure and let o be an object of C. Observe that $\Box\restriction o$ is transitive and non empty.

Let C be a non empty category structure. One can verify that there exists a substructure of C which is transitive non empty and strict.

We now state the proposition

(25) Let C be a transitive non empty category structure and let D_1, D_2 be transitive non empty substructures of C. Suppose the carrier of $D_1 \subseteq$ the carrier of D_2 and the arrows of $D_1 \dot{\subseteq}$ the arrows of D_2. Then D_1 is a substructure of D_2.

Let C be a category structure and let D be a substructure of C. We say that D is full if and only if:

(Def. 12) The arrows of $D =$ (the arrows of C) $\restriction$ [: the carrier of D, the carrier of D :].

Let C be a non empty category structure with units and let D be a substructure of C. We say that D is id-inheriting if and only if:

(Def. 13) For every object o of D and for every object o' of C such that $o = o'$ holds $\mathrm{id}_{o'} \in \langle o, o \rangle$.

Let C be a category structure. One can verify that there exists a substructure of C which is full and strict.

Let C be a non empty category structure. Observe that there exists a substructure of C which is full non empty and strict.

Let C be a category and let o be an object of C. Note that $\Box \restriction o$ is full and id-inheriting.

Let C be a category. One can verify that there exists a substructure of C which is full id-inheriting non empty and strict.

In the sequel C is a non empty transitive category structure.

The following propositions are true:

(26) Let D be a substructure of C. Suppose the carrier of $D =$ the carrier of C and the arrows of $D =$ the arrows of C. Then the category structure of $D =$ the category structure of C.

(27) Let D_1, D_2 be non empty transitive substructures of C. Suppose the carrier of $D_1 =$ the carrier of D_2 and the arrows of $D_1 =$ the arrows of D_2. Then the category structure of $D_1 =$ the category structure of D_2.

(28) Let D be a full substructure of C. Suppose the carrier of $D =$ the carrier of C. Then the category structure of $D =$ the category structure of C.

(29) Let C be a non empty category structure, and let D be a full non empty substructure of C, and let o_1, o_2 be objects of C and let p_1, p_2 be objects of D If $o_1 = p_1$ and $o_2 = p_2$, then $\langle o_1, o_2 \rangle = \langle p_1, p_2 \rangle$.

(30) For every non empty category structure C and for every non empty substructure D of C holds every object of D is an object of C.

Let C be a transitive non empty category structure. Note that every substructure of C which is full and non empty is also transitive.

The following propositions are true:

(31) Let D_1, D_2 be full non empty substructures of C. Suppose the carrier of D_1 = the carrier of D_2. Then the category structure of D_1 = the category structure of D_2.

(32) Let C be a non empty category structure, and let D be a non empty substructure of C, and let o_1, o_2 be objects of C and let p_1, p_2 be objects of D If $o_1 = p_1$ and $o_2 = p_2$, then $\langle p_1, p_2 \rangle \subseteq \langle o_1, o_2 \rangle$.

(33) Let C be a non empty transitive category structure, and let D be a non empty transitive substructure of C, and let p_1, p_2, p_3 be objects of D Suppose $\langle p_1, p_2 \rangle \neq \emptyset$ and $\langle p_2, p_3 \rangle \neq \emptyset$. Let o_1, o_2, o_3 be objects of C Suppose $o_1 = p_1$ and $o_2 = p_2$ and $o_3 = p_3$. Let f be a morphism from o_1 to o_2, and let g be a morphism from o_2 to o_3, and let f_1 be a morphism from p_1 to p_2, and let g_1 be a morphism from p_2 to p_3. If $f = f_1$ and $g = g_1$, then $g \cdot f = g_1 \cdot f_1$.

Let C be an associative transitive non empty category structure. Note that every non empty substructure of C which is transitive is also associative.

One can prove the following proposition

(34) Let C be a non empty category structure, and let D be a non empty substructure of C, and let o_1, o_2 be objects of C and let p_1, p_2 be objects of D If $o_1 = p_1$ and $o_2 = p_2$ and $\langle p_1, p_2 \rangle \neq \emptyset$, then every morphism from p_1 to p_2 is a morphism from o_1 to o_2.

Let C be a transitive non empty category structure with units. Note that every non empty substructure of C which is id-inheriting and transitive has units.

Let C be a category. Note that there exists a non empty substructure of C which is id-inheriting and transitive.

Let C be a category. A subcategory of C is an id-inheriting transitive substructure of C.

We now state the proposition

(35) Let C be a category, and let D be a non empty subcategory of C, and let o be an object of D, and let o' be an object of C. If $o = o'$, then $\mathrm{id}_o = \mathrm{id}_{o'}$.

References

[1] Czesław Byliński. Basic functions and operations on functions. *Formalized Mathematics*, 1(**1**):245–254, 1990.
[2] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(**1**):175–180, 1990.
[3] Czesław Byliński. Cartesian categories. *Formalized Mathematics*, 3(**2**):161–169, 1992.
[4] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 1990.
[5] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[6] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[7] Czesław Byliński. Introduction to categories and functors. *Formalized Mathematics*, 1(**2**):409–420, 1990.

[8] 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.
[9] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[10] Artur Korniłowicz. On the group of automorphisms of universal algebra & many sorted algebra. *Formalized Mathematics*, 5(**2**):221–226, 1996.
[11] Małgorzata Korolkiewicz. Homomorphisms of many sorted algebras. *Formalized Mathematics*, 5(**1**):61–65, 1996.
[12] Beata Madras. Product of family of universal algebras. *Formalized Mathematics*, 4(**1**):103–108, 1993.
[13] Michał Muzalewski and Wojciech Skaba. Three-argument operations and four-argument operations. *Formalized Mathematics*, 2(**2**):221–224, 1991.
[14] Andrzej Trybulec. Categories without uniqueness of **cod** and **dom**. *Formalized Mathematics*, 5(**2**):259–267, 1996.
[15] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[16] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[17] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[18] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[19] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[20] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received January 22, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

On the Category of Posets

Adam Grabowski
Warsaw University
Białystok

Summary. In the paper the construction of a category of partially ordered sets is shown: in the second section according to [8] and in the third section according to the definition given in [15]. Some of useful notions such as monotone map and the set of monotone maps between relational structures are given.

MML Identifier: `ORDERS_3`.

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

1. Preliminaries

Let I_1 be a relation structure. We say that I_1 is discrete if and only if:

(Def. 1) The internal relation of $I_1 = \triangle_{\text{the carrier of } I_1}$.

Let us mention that there exists a poset which is strict discrete and non empty and there exists a poset which is strict discrete and empty.

Let X be a set. Then $\triangle_X$ is an order in X.

Observe that $\langle \emptyset, \triangle_\emptyset \rangle$ is empty. Let P be an empty relation structure. One can check that the internal relation of P is empty.

Let us mention that every relation structure which is empty is also discrete.

Let P be a relation structure and let I_1 be a subset of P. We say that I_1 is disconnected if and only if the condition (Def. 2) is satisfied.

(Def. 2) There exist subsets A, B of P such that

(i) $A \neq \emptyset$,

(ii) $B \neq \emptyset$,

(iii) $I_1 = A \cup B$,

ISSN 1426–2630

(iv) A misses B, and

(v) the internal relation of P = (the internal relation of P) $|^2$ (A) $\cup$ (the internal relation of P) $|^2$ (B).

We introduce I_1 is connected as an antonym of I_1 is disconnected.

Let I_1 be a non empty relation structure. We say that I_1 is disconnected if and only if:

(Def. 3) $\Omega_{(I_1)}$ is disconnected.

We introduce I_1 is connected as an antonym of I_1 is disconnected.

In the sequel T will denote a non empty relation structure and a will denote an element of T.

One can prove the following propositions:

(1) For every discrete non empty relation structure D_1 and for all elements x, y of D_1 holds $x \leq y$ iff $x = y$.

(2) For every binary relation R and for arbitrary a such that R is an order in $\{a\}$ holds $R = \triangle_{\{a\}}$.

(3) If T is reflexive and $\Omega_T = \{a\}$, then T is discrete.

In the sequel a will be arbitrary.

One can prove the following two propositions:

(4) If $\Omega_T = \{a\}$, then T is connected.

(5) For every discrete non empty poset D_1 such that there exist elements a, b of D_1 such that $a \neq b$ holds D_1 is disconnected.

One can check that there exists a non empty poset which is strict and connected and there exists a non empty poset which is strict disconnected and discrete.

2. On the Category of Posets

Let I_1 be a set. We say that I_1 is poset-membered if and only if:

(Def. 4) For arbitrary a such that $a \in I_1$ holds a is a non empty poset.

One can check that there exists a set which is non empty and poset-membered.

A set of posets is a poset-membered set.

Let P be a non empty set of posets. We see that the element of P is a non empty poset.

Let L_1, L_2 be relation structures and let f be a map from L_1 into L_2. We say that f is monotone 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 $a \leq b$.

In the sequel P will denote a non empty set of posets and A, B will denote elements of P.

Let A, B be relation structures. The functor $B_{\leq}^{A}$ is defined by the condition (Def. 6).

(Def. 6) $a \in B^A_{\leq}$ if and only if there exists a map f from A into B such that $a = f$ and $f \in (\text{the carrier of } B)^{\text{the carrier of } A}$ and f is monotone.

The following propositions are true:

(6) For all non empty relation structures A, B, C and for all functions f, g such that $f \in B^A_{\leq}$ and $g \in C^B_{\leq}$ holds $g \cdot f \in C^A_{\leq}$.

(7) $\mathrm{id}_{(\text{the carrier of } T)} \in T^T_{\leq}$.

Let us consider T. Observe that $T^T_{\leq}$ is non empty.

Let X be a set. The functor $\mathrm{Carr}(X)$ yields a set and is defined by:

(Def. 7) $a \in \mathrm{Carr}(X)$ iff there exists a 1-sorted structure s such that $s \in X$ and $a =$ the carrier of s.

Let us consider P. Observe that $\mathrm{Carr}(P)$ is non empty.

The following propositions are true:

(8) For every 1-sorted structure f holds $\mathrm{Carr}(\{f\}) = \{\text{the carrier of } f\}$.

(9) For all 1-sorted structures f, g holds $\mathrm{Carr}(\{f, g\}) = \{\text{the carrier of } f, \text{the carrier of } g\}$.

(10) $B^A_{\leq} \subseteq \mathrm{Funcs}\,\mathrm{Carr}(P)$.

(11) For all relation structures A, B holds $B^A_{\leq} \subseteq (\text{the carrier of } B)^{\text{the carrier of } A}$.

Let A, B be non empty poset. Observe that $B^A_{\leq}$ is functional.

Let P be a non empty set of posets. The functor $\mathrm{POSCat}(P)$ yielding a strict category with triple-like morphisms is defined by the conditions (Def. 8).

(Def. 8) (i) The objects of $\mathrm{POSCat}(P) = P$,

(ii) for all elements a, b of P and for every element f of $\mathrm{Funcs}\,\mathrm{Carr}(P)$ such that $f \in b^a_{\leq}$ holds $\langle\langle a, b\rangle, f\rangle$ is a morphism of $\mathrm{POSCat}(P)$,

(iii) for every morphism m of $\mathrm{POSCat}(P)$ there exist elements a, b of P and there exists an element f of $\mathrm{Funcs}\,\mathrm{Carr}(P)$ such that $m = \langle\langle a, b\rangle, f\rangle$ and $f \in b^a_{\leq}$, and

(iv) for all morphisms m_1, m_2 of $\mathrm{POSCat}(P)$ and for all elements a_1, a_2, a_3 of P and for all elements f_1, f_2 of $\mathrm{Funcs}\,\mathrm{Carr}(P)$ such that $m_1 = \langle\langle a_1, a_2\rangle, f_1\rangle$ and $m_2 = \langle\langle a_2, a_3\rangle, f_2\rangle$ holds $m_2 \cdot m_1 = \langle\langle a_1, a_3\rangle, f_2 \cdot f_1\rangle$.

3. On the Alternative Category of Posets

In this article we present several logical schemes. The scheme *AltCatEx* concerns a non empty set $\mathcal{A}$ and a binary functor $\mathcal{F}$ yielding a functional set, and states that:

There exists a strict category structure C such that

(i) the carrier of $C = \mathcal{A}$, and

(ii) for all elements i, j of $\mathcal{A}$ holds (the arrows of C)$(i, j) = \mathcal{F}(i, j)$ and for all elements i, j, k of $\mathcal{A}$ holds (the composition of C)$(i, j, k) = \mathrm{FuncComp}(\mathcal{F}(i, j), \mathcal{F}(j, k))$

provided the following condition is met:

- For all elements i, j, k of $\mathcal{A}$ and for all functions f, g such that $f \in \mathcal{F}(i,j)$ and $g \in \mathcal{F}(j,k)$ holds $g \cdot f \in \mathcal{F}(i,k)$.

The scheme *AltCatUniq* deals with a non empty set $\mathcal{A}$ and a binary functor $\mathcal{F}$ yielding a functional set, and states that:

Let C_1, C_2 be strict category structures. Suppose that

(i) the carrier of $C_1 = \mathcal{A}$,

(ii) for all elements i, j of $\mathcal{A}$ holds (the arrows of C_1)$(i, j) = \mathcal{F}(i,j)$ and for all elements i, j, k of $\mathcal{A}$ holds (the composition of C_1)$(i, j, k) = \mathrm{FuncComp}(\mathcal{F}(i,j), \mathcal{F}(j,k))$,

(iii) the carrier of $C_2 = \mathcal{A}$, and

(iv) for all elements i, j of $\mathcal{A}$ holds (the arrows of C_2)$(i, j) = \mathcal{F}(i,j)$ and for all elements i, j, k of $\mathcal{A}$ holds (the composition of C_2)$(i, j, k) = \mathrm{FuncComp}(\mathcal{F}(i,j), \mathcal{F}(j,k))$.

Then $C_1 = C_2$

for all values of the parameters.

Let P be a non empty set of posets. The functor POSAltCat(P) yielding a strict category structure is defined by the conditions (Def. 9).

(Def. 9) (i) The carrier of POSAltCat$(P) = P$, and

(ii) for all elements i, j of P holds (the arrows of POSAltCat(P))$(i, j) = j^{i}_{\leq}$ and for all elements i, j, k of P holds (the composition of POSAltCat(P))$(i, j, k) = \mathrm{FuncComp}(j^{i}_{\leq}, k^{j}_{\leq})$.

Let P be a non empty set of posets. One can verify that POSAltCat(P) is transitive and non empty.

Let P be a non empty set of posets. Observe that POSAltCat(P) is associative and has units.

One can prove the following proposition

(12) Let o_1, o_2 be objects of POSAltCat(P) and let A, B be elements of P. If $o_1 = A$ and $o_2 = B$, then $\langle o_1, o_2 \rangle \subseteq$ (the carrier of B)$^{\text{the carrier of } A}$.

References

[1] Grzegorz Bancerek. Categorial categories and slice categories. *Formalized Mathematics*, 5(**2**):157–165, 1996.

[2] Grzegorz Bancerek. The well ordering relations. *Formalized Mathematics*, 1(**1**):123–129, 1990.

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

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

[5] Czesław Byliński. Category Ens. *Formalized Mathematics*, 2(**4**):527–533, 1991.

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

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

[8] Czesław Byliński. Introduction to categories and functors. *Formalized Mathematics*, 1(**2**):409–420, 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] Zbigniew Karno. Continuity of mappings over the union of subspaces. *Formalized Mathematics*, 3(**1**):1–16, 1992.
[12] Beata Madras. Product of family of universal algebras. *Formalized Mathematics*, 4(**1**):103–108, 1993.
[13] Michał Muzalewski and Wojciech Skaba. Three-argument operations and four-argument operations. *Formalized Mathematics*, 2(**2**):221–224, 1991.
[14] Beata Padlewska and Agata Darmochwał. Topological spaces and continuous functions. *Formalized Mathematics*, 1(**1**):223–230, 1990.
[15] Andrzej Trybulec. Categories without uniqueness of **cod** and **dom**. *Formalized Mathematics*, 5(**2**):259–267, 1996.
[16] Andrzej Trybulec. Function domains and Frænkel operator. *Formalized Mathematics*, 1(**3**):495–500, 1990.
[17] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[18] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[19] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 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.

Received January 22, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

An Extension of SCM

Andrzej Trybulec
Warsaw University
Białystok

Yatsuka Nakamura
Shinshu University
Nagano

Piotr Rudnicki
University of Alberta
Edmonton

MML Identifier: `SCMFSA_1`.

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

In this paper x will be arbitrary and k will denote a natural number.

The subset $\text{Data-Loc}_{\text{SCM}_{\text{FSA}}}$ of $\mathbb{Z}$ is defined as follows:

(Def. 1) $\text{Data-Loc}_{\text{SCM}_{\text{FSA}}} = \text{Data-Loc}_{\text{SCM}}$.

The subset $\text{Data}^*\text{-Loc}_{\text{SCM}_{\text{FSA}}}$ of $\mathbb{Z}$ is defined as follows:

(Def. 2) $\text{Data}^*\text{-Loc}_{\text{SCM}_{\text{FSA}}} = \mathbb{Z} \setminus \mathbb{N}$.

The subset $\text{Instr-Loc}_{\text{SCM}_{\text{FSA}}}$ of $\mathbb{Z}$ is defined as follows:

(Def. 3) $\text{Instr-Loc}_{\text{SCM}_{\text{FSA}}} = \text{Instr-Loc}_{\text{SCM}}$.

One can check the following observations:

- $\text{Data}^*\text{-Loc}_{\text{SCM}_{\text{FSA}}}$ is non empty,
- $\text{Data-Loc}_{\text{SCM}_{\text{FSA}}}$ is non empty, and
- $\text{Instr-Loc}_{\text{SCM}_{\text{FSA}}}$ is non empty.

For simplicity we adopt the following convention: J, K are elements of $\mathbb{Z}_{13}$, a is an element of $\text{Instr-Loc}_{\text{SCM}_{\text{FSA}}}$, b, c, c_1 are elements of $\text{Data-Loc}_{\text{SCM}_{\text{FSA}}}$, and f, f_1 are elements of $\text{Data}^*\text{-Loc}_{\text{SCM}_{\text{FSA}}}$.

The subset $\text{Instr}_{\text{SCM}_{\text{FSA}}}$ of $[\!: \mathbb{Z}_{13}, (\bigcup\{\mathbb{Z}, \mathbb{Z}^*\} \cup \mathbb{Z})^* :\!]$ is defined by:

(Def. 4) $\text{Instr}_{\text{SCM}_{\text{FSA}}} = \text{Instr}_{\text{SCM}} \cup \{\langle J, \langle c, f, b\rangle\rangle : J \in \{9, 10\}\} \cup \{\langle K, \langle c_1, f_1\rangle\rangle : K \in \{11, 12\}\}$.

The following two propositions are true:

(1) $\text{Instr}_{\text{SCM}_{\text{FSA}}} = \text{Instr}_{\text{SCM}} \cup \{\langle J, \langle c, f, b\rangle\rangle : J \in \{9, 10\}\} \cup \{\langle K, \langle c_1, f_1\rangle\rangle : K \in \{11, 12\}\}$.

(2) $\text{Instr}_{\text{SCM}} \subseteq \text{Instr}_{\text{SCM}_{\text{FSA}}}$.

ISSN 1426–2630

Let us observe that $\mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}$ is non empty.

Let I be an element of $\mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}$. The functor $\mathrm{InsCode}(I)$ yielding a natural number is defined by:

(Def. 5) $\mathrm{InsCode}(I) = I_{\mathbf{1}}$.

The following two propositions are true:

(3) For every element I of $\mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}$ such that $\mathrm{InsCode}(I) \leq 8$ holds $I \in \mathrm{Instr}_{\mathrm{SCM}}$.

(4) $\langle 0, \varepsilon \rangle \in \mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

The function $\mathrm{OK}_{\mathrm{SCM}_{\mathrm{FSA}}}$ from $\mathbb{Z}$ into $\{\mathbb{Z}, \mathbb{Z}^*\} \cup \{\mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}, \text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}\}$ is defined by:

(Def. 6) $\mathrm{OK}_{\mathrm{SCM}_{\mathrm{FSA}}} = (\mathbb{Z} \longmapsto \mathbb{Z}^*) + \cdot\, \mathrm{OK}_{\mathrm{SCM}} + \cdot\, (\mathrm{Instr}_{\mathrm{SCM}} \stackrel{\cdot}{\longmapsto} \mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}) \cdot (\mathrm{OK}_{\mathrm{SCM}} \restriction \text{Instr-Loc}_{\mathrm{SCM}})$.

One can prove the following propositions:

(5) $\mathrm{OK}_{\mathrm{SCM}_{\mathrm{FSA}}} = (\mathbb{Z} \longmapsto \mathbb{Z}^*) + \cdot\, \mathrm{OK}_{\mathrm{SCM}} + \cdot\, (\mathrm{Instr}_{\mathrm{SCM}} \stackrel{\cdot}{\longmapsto} \mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}) \cdot (\mathrm{OK}_{\mathrm{SCM}} \restriction \text{Instr-Loc}_{\mathrm{SCM}})$.

(6) If $x \in \{9, 10\}$, then $\langle x, \langle c, f, b \rangle \rangle \in \mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

(7) If $x \in \{11, 12\}$, then $\langle x, \langle c, f \rangle \rangle \in \mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

(8) $\mathbb{Z} = \{0\} \cup \text{Data-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}} \cup \text{Data}^*\text{-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}} \cup \text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

(9) $\mathrm{OK}_{\mathrm{SCM}_{\mathrm{FSA}}}(0) = \text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

(10) $\mathrm{OK}_{\mathrm{SCM}_{\mathrm{FSA}}}(b) = \mathbb{Z}$.

(11) $\mathrm{OK}_{\mathrm{SCM}_{\mathrm{FSA}}}(a) = \mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

(12) $\mathrm{OK}_{\mathrm{SCM}_{\mathrm{FSA}}}(f) = \mathbb{Z}^*$.

(13) $\text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}} \neq \mathbb{Z}$ and $\mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}} \neq \mathbb{Z}$ and $\text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}} \neq \mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}$ and $\text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}} \neq \mathbb{Z}^*$ and $\mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}} \neq \mathbb{Z}^*$.

(14) For every integer i such that $\mathrm{OK}_{\mathrm{SCM}_{\mathrm{FSA}}}(i) = \text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ holds $i = 0$.

(15) For every integer i such that $\mathrm{OK}_{\mathrm{SCM}_{\mathrm{FSA}}}(i) = \mathbb{Z}$ holds $i \in \text{Data-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

(16) For every integer i such that $\mathrm{OK}_{\mathrm{SCM}_{\mathrm{FSA}}}(i) = \mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}$ holds $i \in \text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

(17) For every integer i such that $\mathrm{OK}_{\mathrm{SCM}_{\mathrm{FSA}}}(i) = \mathbb{Z}^*$ holds $i \in \text{Data}^*\text{-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

An $\mathbf{SCM}_{\mathrm{FSA}}$-state is an element of $\prod(\mathrm{OK}_{\mathrm{SCM}_{\mathrm{FSA}}})$.

Next we state two propositions:

(18) For every $\mathbf{SCM}_{\mathrm{FSA}}$-state s and for every element I of $\mathrm{Instr}_{\mathrm{SCM}}$ holds $s \restriction \mathbb{N} + \cdot\, (\text{Instr-Loc}_{\mathrm{SCM}} \longmapsto I)$ is a state $_{\mathrm{SCM}}$.

(19) For every $\mathbf{SCM}_{\mathrm{FSA}}$-state s and for every state $_{\mathrm{SCM}}$ s' holds $s + \cdot\, s' + \cdot\, s \restriction \text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ is an $\mathbf{SCM}_{\mathrm{FSA}}$-state.

In the sequel s is an $\mathbf{SCM}_{\mathrm{FSA}}$-state.

Let s be an $\mathbf{SCM}_{\mathrm{FSA}}$-state and let u be an element of $\text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$. The functor $\mathrm{Chg}_{\mathrm{SCM}_{\mathrm{FSA}}}(s, u)$ yields an $\mathbf{SCM}_{\mathrm{FSA}}$-state and is defined as follows:

(Def. 7) $\mathrm{Chg}_{\mathrm{SCM}_{\mathrm{FSA}}}(s, u) = s{+}\cdot(0 \dot{\longmapsto} u)$.

Let s be an $\mathbf{SCM}_{\mathrm{FSA}}$-state, let t be an element of $\text{Data-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$, and let u be an integer. The functor $\mathrm{Chg}_{\mathrm{SCM}_{\mathrm{FSA}}}(s, t, u)$ yielding an $\mathbf{SCM}_{\mathrm{FSA}}$-state is defined as follows:

(Def. 8) $\mathrm{Chg}_{\mathrm{SCM}_{\mathrm{FSA}}}(s, t, u) = s{+}\cdot(t \dot{\longmapsto} u)$.

Let s be an $\mathbf{SCM}_{\mathrm{FSA}}$-state, let t be an element of $\text{Data}^*\text{-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$, and let u be a finite sequence of elements of $\mathbb{Z}$ The functor $\mathrm{Chg}_{\mathrm{SCM}_{\mathrm{FSA}}}(s, t, u)$ yielding an $\mathbf{SCM}_{\mathrm{FSA}}$-state is defined as follows:

(Def. 9) $\mathrm{Chg}_{\mathrm{SCM}_{\mathrm{FSA}}}(s, t, u) = s{+}\cdot(t \dot{\longmapsto} u)$.

Let s be an $\mathbf{SCM}_{\mathrm{FSA}}$-state and let a be an element of $\text{Data-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$. Then $s(a)$ is an integer.

Let s be an $\mathbf{SCM}_{\mathrm{FSA}}$-state and let a be an element of $\text{Data}^*\text{-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$. Then $s(a)$ is a finite sequence of elements of $\mathbb{Z}$.

Let x be an element of $\mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}$. Let us assume that there exist c, f, b, J such that $x = \langle\!\langle J, \langle c, f, b\rangle\rangle\!\rangle$. The functor x int-addr$_1$ yielding an element of $\text{Data-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ is defined by:

(Def. 10) There exist c, f, b such that $\langle c, f, b\rangle = x_{\mathbf{2}}$ and x int-addr$_1 = c$.

The functor x int-addr$_2$ yielding an element of $\text{Data-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ is defined as follows:

(Def. 11) There exist c, f, b such that $\langle c, f, b\rangle = x_{\mathbf{2}}$ and x int-addr$_2 = b$.

The functor x coll-addr$_1$ yields an element of $\text{Data}^*\text{-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ and is defined as follows:

(Def. 12) There exist c, f, b such that $\langle c, f, b\rangle = x_{\mathbf{2}}$ and x coll-addr$_1 = f$.

Let x be an element of $\mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}$. Let us assume that there exist c, f, J such that $x = \langle\!\langle J, \langle c, f\rangle\rangle\!\rangle$. The functor x int-addr$_3$ yielding an element of $\text{Data-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ is defined as follows:

(Def. 13) There exist c, f such that $\langle c, f\rangle = x_{\mathbf{2}}$ and x int-addr$_3 = c$.

The functor x coll-addr$_2$ yields an element of $\text{Data}^*\text{-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ and is defined as follows:

(Def. 14) There exist c, f such that $\langle c, f\rangle = x_{\mathbf{2}}$ and x coll-addr$_2 = f$.

Let l be an element of $\text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$. The functor $\mathrm{Next}(l)$ yielding an element of $\text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ is defined as follows:

(Def. 15) There exists an element L of $\text{Instr-Loc}_{\mathrm{SCM}}$ such that $L = l$ and $\mathrm{Next}(l) = \mathrm{Next}(L)$.

Let s be an $\mathbf{SCM}_{\mathrm{FSA}}$-state. The functor $\mathbf{IC}_s$ yielding an element of $\text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ is defined by:

(Def. 16) $\mathbf{IC}_s = s(0)$.

Let x be an element of $\mathrm{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}$ and let s be an $\mathbf{SCM}_{\mathrm{FSA}}$-state. The functor $\text{Exec-Res}_{\mathrm{SCM}_{\mathrm{FSA}}}(x, s)$ yielding an $\mathbf{SCM}_{\mathrm{FSA}}$-state is defined by:

(Def. 17) (i) There exists an element x' of $\mathrm{Instr}_{\mathrm{SCM}}$ and there exists a state $_{\mathrm{SCM}}$ s' such that $x = x'$ and $s' = s \restriction \mathbb{N}{+}\cdot(\text{Instr-Loc}_{\mathrm{SCM}} \longmapsto x')$ and

Exec-Res$_{\mathrm{SCM_{FSA}}}(x, s) = s{+}\!\cdot\,$Exec-Res$_{\mathrm{SCM}}(x', s'){+}\!\cdot s \restriction$ Instr-Loc$_{\mathrm{SCM_{FSA}}}$ if InsCode$(x) \leq 8$,

(ii) there exists an integer i and there exists k such that $k = |s(x \text{ int-addr}_2)|$ and $i = \pi_k s(x \text{ coll-addr}_1)$ and Exec-Res$_{\mathrm{SCM_{FSA}}}(x, s) =$ Chg$_{\mathrm{SCM_{FSA}}}$(Chg$_{\mathrm{SCM_{FSA}}}(s, x \text{ int-addr}_1, i)$, Next$(\mathbf{IC}_s)$) if InsCode$(x) = 9$,

(iii) there exists a finite sequence f of elements of $\mathbb{Z}$ and there exists k such that $k = |s(x \text{ int-addr}_2)|$ and $f = s(x \text{ coll-addr}_1) +\cdot (k, s(x \text{ int-addr}_1))$ and Exec-Res$_{\mathrm{SCM_{FSA}}}(x, s) =$ Chg$_{\mathrm{SCM_{FSA}}}$(Chg$_{\mathrm{SCM_{FSA}}}(s, x \text{ coll-addr}_1, f)$, Next$(\mathbf{IC}_s)$) if InsCode$(x) = 10$,

(iv) Exec-Res$_{\mathrm{SCM_{FSA}}}(x, s) =$ Chg$_{\mathrm{SCM_{FSA}}}$(Chg$_{\mathrm{SCM_{FSA}}}(s, x \text{ int-addr}_3, \operatorname{len} s(x \text{ coll-addr}_2))$, Next$(\mathbf{IC}_s)$) if InsCode$(x) = 11$,

(v) there exists a finite sequence f of elements of $\mathbb{Z}$ and there exists k such that $k = |s(x \text{ int-addr}_3)|$ and $f = k \mapsto 0$ and Exec-Res$_{\mathrm{SCM_{FSA}}}(x, s) =$ Chg$_{\mathrm{SCM_{FSA}}}$(Chg$_{\mathrm{SCM_{FSA}}}(s, x \text{ coll-addr}_2, f)$, Next$(\mathbf{IC}_s)$) if InsCode$(x) = 12$,

(vi) Exec-Res$_{\mathrm{SCM_{FSA}}}(x, s) = s$, otherwise.

The function Exec$_{\mathrm{SCM_{FSA}}}$ from Instr$_{\mathrm{SCM_{FSA}}}$ into $(\prod(\mathrm{OK}_{\mathrm{SCM_{FSA}}}))^{\prod(\mathrm{OK}_{\mathrm{SCM_{FSA}}})}$ is defined by:

(Def. 18) For every element x of Instr$_{\mathrm{SCM_{FSA}}}$ and for every $\mathbf{SCM}_{\mathrm{FSA}}$-state y holds (Exec$_{\mathrm{SCM_{FSA}}}(x)$ **qua** element of $(\prod(\mathrm{OK}_{\mathrm{SCM_{FSA}}}))^{\prod(\mathrm{OK}_{\mathrm{SCM_{FSA}}})})(y) =$ Exec-Res$_{\mathrm{SCM_{FSA}}}(x, y)$.

One can prove the following propositions:

(20) For every $\mathbf{SCM}_{\mathrm{FSA}}$-state s and for every element u of Instr-Loc$_{\mathrm{SCM_{FSA}}}$ holds $(\mathrm{Chg}_{\mathrm{SCM_{FSA}}}(s, u))(0) = u$.

(21) For every $\mathbf{SCM}_{\mathrm{FSA}}$-state s and for every element u of Instr-Loc$_{\mathrm{SCM_{FSA}}}$ and for every element m_1 of Data-Loc$_{\mathrm{SCM_{FSA}}}$ holds $(\mathrm{Chg}_{\mathrm{SCM_{FSA}}}(s, u))(m_1) = s(m_1)$.

(22) For every $\mathbf{SCM}_{\mathrm{FSA}}$-state s and for every element u of Instr-Loc$_{\mathrm{SCM_{FSA}}}$ and for every element p of Data*-Loc$_{\mathrm{SCM_{FSA}}}$ holds $(\mathrm{Chg}_{\mathrm{SCM_{FSA}}}(s, u))(p) = s(p)$.

(23) For every $\mathbf{SCM}_{\mathrm{FSA}}$-state s and for all elements u, v of Instr-Loc$_{\mathrm{SCM_{FSA}}}$ holds $(\mathrm{Chg}_{\mathrm{SCM_{FSA}}}(s, u))(v) = s(v)$.

(24) For every $\mathbf{SCM}_{\mathrm{FSA}}$-state s and for every element t of Data-Loc$_{\mathrm{SCM_{FSA}}}$ and for every integer u holds $(\mathrm{Chg}_{\mathrm{SCM_{FSA}}}(s, t, u))(0) = s(0)$.

(25) For every $\mathbf{SCM}_{\mathrm{FSA}}$-state s and for every element t of Data-Loc$_{\mathrm{SCM_{FSA}}}$ and for every integer u holds $(\mathrm{Chg}_{\mathrm{SCM_{FSA}}}(s, t, u))(t) = u$.

(26) Let s be an $\mathbf{SCM}_{\mathrm{FSA}}$-state, and let t be an element of Data-Loc$_{\mathrm{SCM_{FSA}}}$, and let u be an integer, and let m_1 be an element of Data-Loc$_{\mathrm{SCM_{FSA}}}$. If $m_1 \neq t$, then $(\mathrm{Chg}_{\mathrm{SCM_{FSA}}}(s, t, u))(m_1) = s(m_1)$.

(27) Let s be an $\mathbf{SCM}_{\mathrm{FSA}}$-state, and let t be an element of Data-Loc$_{\mathrm{SCM_{FSA}}}$, and let u be an integer, and let f be an element of Data*-Loc$_{\mathrm{SCM_{FSA}}}$. Then $(\mathrm{Chg}_{\mathrm{SCM_{FSA}}}(s, t, u))(f) = s(f)$.

(28) Let s be an $\mathbf{SCM}_{\rm FSA}$-state, and let t be an element of $\text{Data-Loc}_{\rm SCM_{FSA}}$, and let u be an integer, and let v be an element of $\text{Instr-Loc}_{\rm SCM_{FSA}}$. Then $(\text{Chg}_{\rm SCM_{FSA}}(s,t,u))(v) = s(v)$.

(29) Let s be an $\mathbf{SCM}_{\rm FSA}$-state, and let t be an element of $\text{Data}^*\text{-Loc}_{\rm SCM_{FSA}}$, and let u be a finite sequence of elements of $\mathbb{Z}$. Then $(\text{Chg}_{\rm SCM_{FSA}}(s,t,u))(t) = u$.

(30) Let s be an $\mathbf{SCM}_{\rm FSA}$-state, and let t be an element of $\text{Data}^*\text{-Loc}_{\rm SCM_{FSA}}$, and let u be a finite sequence of elements of $\mathbb{Z}$, and let m_1 be an element of $\text{Data}^*\text{-Loc}_{\rm SCM_{FSA}}$. If $m_1 \neq t$, then $(\text{Chg}_{\rm SCM_{FSA}}(s,t,u))(m_1) = s(m_1)$.

(31) Let s be an $\mathbf{SCM}_{\rm FSA}$-state, and let t be an element of $\text{Data}^*\text{-Loc}_{\rm SCM_{FSA}}$, and let u be a finite sequence of elements of $\mathbb{Z}$, and let a be an element of $\text{Data-Loc}_{\rm SCM_{FSA}}$. Then $(\text{Chg}_{\rm SCM_{FSA}}(s,t,u))(a) = s(a)$.

(32) Let s be an $\mathbf{SCM}_{\rm FSA}$-state, and let t be an element of $\text{Data}^*\text{-Loc}_{\rm SCM_{FSA}}$, and let u be a finite sequence of elements of $\mathbb{Z}$, and let v be an element of $\text{Instr-Loc}_{\rm SCM_{FSA}}$. Then $(\text{Chg}_{\rm SCM_{FSA}}(s,t,u))(v) = s(v)$.

References

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

[2] Grzegorz Bancerek. The reflection theorem. *Formalized Mathematics*, 1(**5**):973–977, 1990.

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

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

[5] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 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. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.

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

[10] Czesław Byliński. Subcategories and products of categories. *Formalized Mathematics*, 1(**4**):725–732, 1990.

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

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

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

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

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

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

[17] Andrzej Trybulec. Function domains and Frænkel operator. *Formalized Mathematics*, 1(**3**):495–500, 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] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[21] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[22] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[23] Wojciech A. Trybulec. Binary operations on finite sequences. *Formalized Mathematics*, 1(**5**):979–981, 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.

Received February 3, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

Components and Unions of Components

Yatsuka Nakamura
Shinshu University
Nagano

Andrzej Trybulec
Warsaw University
Białystok

Summary. First, we generalized **skl** function for a subset of topological spaces the value of which is the component including the set. Second, we introduced a concept of union of components a family of which has good algebraic properties. At the end, we discuss relationship between connectivity of a set as a subset in the whole space and as a subset of a subspace.

MML Identifier: `CONNSP_3`.

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

1. The Component of a Subset in a Topological Space

In this paper G_1 will denote a non empty topological space and V, A will denote subsets of the carrier of G_1.

Let G_1 be a non empty topological structure and let V be a subset of the carrier of G_1. The functor Component(V) yields a subset of the carrier of G_1 and is defined by the condition (Def. 1).

(Def. 1) There exists a family F of subsets of G_1 such that for every subset A of the carrier of G_1 holds $A \in F$ iff A is connected and $V \subseteq A$ and $\bigcup F = \mathrm{Component}(V)$.

One can prove the following propositions:

(1) If there exists A such that A is connected and $V \subseteq A$, then $V \subseteq \mathrm{Component}(V)$.

(2) If it is not true that there exists A such that A is connected and $V \subseteq A$, then $\mathrm{Component}(V) = \emptyset$.

(3) $\mathrm{Component}(\emptyset_{(G_1)})$ = the carrier of G_1.

ISSN 1426–2630

(4) For every subset V of the carrier of G_1 such that V is connected holds $\mathrm{Component}(V) \neq \emptyset$.

(5) For every subset V of the carrier of G_1 such that V is connected and $V \neq \emptyset$ holds $\mathrm{Component}(V)$ is connected.

(6) For all subsets V, C of the carrier of G_1 such that V is connected and C is connected holds if $\mathrm{Component}(V) \subseteq C$, then $C = \mathrm{Component}(V)$.

(7) For every subset A of the carrier of G_1 such that A is a component of G_1 holds $\mathrm{Component}(A) = A$.

(8) Let A be a subset of the carrier of G_1. Then A is a component of G_1 if and only if there exists a subset V of the carrier of G_1 such that V is connected and $V \neq \emptyset$ and $A = \mathrm{Component}(V)$.

(9) For every subset V of the carrier of G_1 such that V is connected and $V \neq \emptyset$ holds $\mathrm{Component}(V)$ is a component of G_1.

(10) If A is a component of G_1 and V is connected and $V \subseteq A$ and $V \neq \emptyset$, then $A = \mathrm{Component}(V)$.

(11) For every subset V of the carrier of G_1 such that V is connected and $V \neq \emptyset$ holds $\mathrm{Component}(\mathrm{Component}(V)) = \mathrm{Component}(V)$.

(12) Let A, B be subsets of the carrier of G_1. If A is connected and B is connected and $A \neq \emptyset$ and $A \subseteq B$, then $\mathrm{Component}(A) = \mathrm{Component}(B)$.

(13) For all subsets A, B of the carrier of G_1 such that A is connected and B is connected and $A \neq \emptyset$ and $A \subseteq B$ holds $B \subseteq \mathrm{Component}(A)$.

(14) For all subsets A, B of the carrier of G_1 such that A is connected and $A \cup B$ is connected and $A \neq \emptyset$ holds $A \cup B \subseteq \mathrm{Component}(A)$.

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

(16) Let A, B be subsets of the carrier of G_1. Suppose A is connected and B is connected and $A \cap B \neq \emptyset$. Then $A \cup B \subseteq \mathrm{Component}(A)$ and $A \cup B \subseteq \mathrm{Component}(B)$ and $A \subseteq \mathrm{Component}(B)$ and $B \subseteq \mathrm{Component}(A)$.

(17) For every subset A of the carrier of G_1 such that A is connected and $A \neq \emptyset$ holds $\overline{A} \subseteq \mathrm{Component}(A)$.

(18) Let A, B be subsets of the carrier of G_1. Suppose A is a component of G_1 and B is connected and $B \neq \emptyset$ and $A \cap B = \emptyset$. Then $A \cap \mathrm{Component}(B) = \emptyset$.

2. On Unions of Components

Let G_1 be a non empty topological structure. A subset of the carrier of G_1 is called a union of components of G_1 if it satisfies the condition (Def. 2).

(Def. 2) There exists a family F of subsets of G_1 such that for every subset B of the carrier of G_1 such that $B \in F$ holds B is a component of G_1 and it $= \bigcup F$.

The following propositions are true:

(19) $\emptyset_{(G_1)}$ is a union of components of G_1.

(20) Let A be a subset of the carrier of G_1. If $A =$ the carrier of G_1, then A is a union of components of G_1.

(21) Let A be a subset of the carrier of G_1 and let p be a point of G_1. If $p \in A$ and A is a union of components of G_1, then $\mathrm{Component}(p) \subseteq A$.

(22) Let A, B be subsets of the carrier of G_1. Suppose A is a union of components of G_1 and B is a union of components of G_1. Then $A \cup B$ is a union of components of G_1 and $A \cap B$ is a union of components of G_1

(23) Let F_1 be a family of subsets of G_1. Suppose that for every subset A of the carrier of G_1 such that $A \in F_1$ holds A is a union of components of G_1. Then $\bigcup F_1$ is a union of components of G_1.

(24) Let F_1 be a family of subsets of G_1. Suppose that for every subset A of the carrier of G_1 such that $A \in F_1$ holds A is a union of components of G_1. Then $\bigcap F_1$ is a union of components of G_1.

(25) Let A, B be subsets of the carrier of G_1. Suppose A is a union of components of G_1 and B is a union of components of G_1. Then $A \setminus B$ is a union of components of G_1.

3. Operations Down and Up

Let us consider G_1, let B be a subset of the carrier of G_1, and let p be a point of G_1. Let us assume that $p \in B$. The functor $\mathrm{Down}(p, B)$ yielding a point of $G_1 \restriction B$ is defined by:

(Def. 3) $\mathrm{Down}(p, B) = p$.

Let us consider G_1, let B be a subset of the carrier of G_1, and let p be a point of $G_1 \restriction B$. Let us assume that $B \neq \emptyset$. The functor $\mathrm{Up}(p)$ yielding a point of G_1 is defined as follows:

(Def. 4) $\mathrm{Up}(p) = p$.

Let us consider G_1 and let V, B be subsets of the carrier of G_1. Let us assume that $B \neq \emptyset$. The functor $\mathrm{Down}(V, B)$ yields a subset of the carrier of $G_1 \restriction B$ and is defined by:

(Def. 5) $\mathrm{Down}(V, B) = V \cap B$.

Let us consider G_1, let B be a subset of the carrier of G_1, and let V be a subset of the carrier of $G_1 \restriction B$. Let us assume that $B \neq \emptyset$. The functor $\mathrm{Up}(V)$ yielding a subset of the carrier of G_1 is defined as follows:

(Def. 6) $\mathrm{Up}(V) = V$.

Let us consider G_1, let B be a subset of the carrier of G_1, and let p be a point of G_1. Let us assume that $p \in B$. The functor $\mathrm{skl}(p, B)$ yields a subset of the carrier of G_1 and is defined as follows:

(Def. 7) For every point q of $G_1 \restriction B$ such that $q = p$ holds $\mathrm{skl}(p, B) = \mathrm{Component}(q)$.

The following propositions are true:

(26) For every subset B of the carrier of G_1 and for every point p of G_1 such that $p \in B$ holds $\mathrm{skl}(p, B) \neq \emptyset$.

(27) For every subset B of the carrier of G_1 and for every point p of G_1 such that $p \in B$ holds $\mathrm{skl}(p, B) = \mathrm{Component}(\mathrm{Down}(p, B))$.

(28) For all subsets V, B of the carrier of G_1 such that $B \neq \emptyset$ and $V \subseteq B$ holds $\mathrm{Down}(V, B) = V$.

(29) For all subsets V, B of the carrier of G_1 such that $B \neq \emptyset$ and V is open holds $\mathrm{Down}(V, B)$ is open.

(30) For all subsets V, B of the carrier of G_1 such that $B \neq \emptyset$ and $V \subseteq B$ holds $\overline{\mathrm{Down}(V, B)} = \overline{V} \cap B$.

(31) Let B be a subset of the carrier of G_1 and let V be a subset of the carrier of $G_1 \restriction B$.If $B \neq \emptyset$, then $\overline{V} = \overline{\mathrm{Up}(V)} \cap B$.

(32) For all subsets V, B of the carrier of G_1 such that $B \neq \emptyset$ and $V \subseteq B$ holds $\overline{\mathrm{Down}(V, B)} \subseteq \overline{V}$.

(33) Let B be a subset of the carrier of G_1 and let V be a subset of the carrier of $G_1 \restriction B$.If $B \neq \emptyset$ and $V \subseteq B$, then $\mathrm{Down}(\mathrm{Up}(V), B) = V$.

(34) Let V, B be subsets of the carrier of G_1 and let W be a subset of the carrier of $G_1 \restriction B$.If $V = W$ and $V \neq \emptyset$ and $B \neq \emptyset$ and W is connected, then V is connected.

(35) For every subset B of the carrier of G_1 and for every point p of G_1 such that $p \in B$ holds $\mathrm{skl}(p, B)$ is connected.

References

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

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

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

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

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

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

[7] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.

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

[9] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 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] Mirosław Wysocki and Agata Darmochwał. Subsets of topological spaces. *Formalized Mathematics*, 1(**1**):231–237, 1990.

Received February 5, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

The $\mathbf{SCM}_{\rm FSA}$ Computer

Andrzej Trybulec
Warsaw University
Białystok

Yatsuka Nakamura
Shinshu University
Nagano

Piotr Rudnicki
University of Alberta
Edmonton

MML Identifier: SCMFSA_2.

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

1. Preliminaries

One can prove the following propositions:

(1) Let N be a non empty set with non empty elements and let S be a von Neumann definite realistic AMI over N. Then $\mathbf{IC}_S \notin$ the instruction locations of S.

(2) Let N be a non empty set with non empty elements, and let S be a definite AMI over N, and let s be a state of S, and let i be an instruction-location of S. Then $s(i)$ is an instruction of S.

(3) Let N be a non empty set with non empty elements, and let S be an AMI over N, and let s be a state of S. Then the instruction locations of $S \subseteq \operatorname{dom} s$.

(4) Let N be a non empty set with non empty elements, and let S be a von Neumann AMI over N, and let s be a state of S. Then $\mathbf{IC}_s \in \operatorname{dom} s$.

(5) Let N be a non empty set with non empty elements, and let S be an AMI over N, and let s be a state of S, and let l be an instruction-location of S. Then $l \in \operatorname{dom} s$.

ISSN 1426–2630

2. The $\mathbf{SCM}_{\mathrm{FSA}}$ Computer

The strict AMI $\mathbf{SCM}_{\mathrm{FSA}}$ over $\{\mathbb{Z}, \mathbb{Z}^*\}$ is defined by:

(Def. 1) $\mathbf{SCM}_{\mathrm{FSA}} = \langle \mathbb{Z}, 0(\in \mathbb{Z}), \text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}, \mathbb{Z}_{13}, 0(\in \mathbb{Z}_{13}), \text{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}, \text{OK}_{\mathrm{SCM}_{\mathrm{FSA}}}, \text{Exec}_{\mathrm{SCM}_{\mathrm{FSA}}} \rangle$.

We now state two propositions:

(6) (i) The instruction locations of $\mathbf{SCM}_{\mathrm{FSA}} \neq \mathbb{Z}$,
(ii) the instructions of $\mathbf{SCM}_{\mathrm{FSA}} \neq \mathbb{Z}$,
(iii) the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}} \neq$ the instructions of $\mathbf{SCM}_{\mathrm{FSA}}$,
(iv) the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}} \neq \mathbb{Z}^*$, and
(v) the instructions of $\mathbf{SCM}_{\mathrm{FSA}} \neq \mathbb{Z}^*$.

(7) $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} = 0$.

3. The Memory Structure

In the sequel k, k_1, k_2 denote natural numbers.

The subset Int-Locations of the objects of $\mathbf{SCM}_{\mathrm{FSA}}$ is defined by:

(Def. 2) Int-Locations = $\text{Data-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

The subset FinSeq-Locations of the objects of $\mathbf{SCM}_{\mathrm{FSA}}$ is defined by:

(Def. 3) FinSeq-Locations = $\text{Data}^*\text{-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

The following proposition is true

(8) The objects of $\mathbf{SCM}_{\mathrm{FSA}}$ = Int-Locations $\cup$ FinSeq-Locations $\cup$ $\{\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\} \cup$ the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

An object of $\mathbf{SCM}_{\mathrm{FSA}}$ is called an integer location if:

(Def. 4) It $\in \text{Data-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

An object of $\mathbf{SCM}_{\mathrm{FSA}}$ is said to be a finite sequence location if:

(Def. 5) It $\in \text{Data}^*\text{-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

In the sequel d_1 denotes an integer location, f_1 denotes a finite sequence location, and x is arbitrary.

We now state several propositions:

(9) $d_1 \in$ Int-Locations.

(10) $f_1 \in$ FinSeq-Locations.

(11) If $x \in$ Int-Locations, then x is an integer location.

(12) If $x \in$ FinSeq-Locations, then x is a finite sequence location.

(13) Int-Locations misses the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(14) FinSeq-Locations misses the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

(15) Int-Locations misses FinSeq-Locations.

Let us consider k. The functor intloc(k) yields an integer location and is defined as follows:

(Def. 6) $\mathrm{intloc}(k) = \mathbf{d}_k$.

The functor insloc(k) yields an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$ and is defined by:

(Def. 7) $\mathrm{insloc}(k) = \mathbf{i}_k$.

The functor fsloc(k) yields a finite sequence location and is defined as follows:

(Def. 8) $\mathrm{fsloc}(k) = -(k + 1)$.

One can prove the following propositions:

(16) For all k_1, k_2 such that $k_1 \neq k_2$ holds $\mathrm{intloc}(k_1) \neq \mathrm{intloc}(k_2)$.

(17) For all k_1, k_2 such that $k_1 \neq k_2$ holds $\mathrm{fsloc}(k_1) \neq \mathrm{fsloc}(k_2)$.

(18) For all k_1, k_2 such that $k_1 \neq k_2$ holds $\mathrm{insloc}(k_1) \neq \mathrm{insloc}(k_2)$.

(19) For every integer location d_2 there exists a natural number i such that $d_2 = \mathrm{intloc}(i)$.

(20) For every finite sequence location f_2 there exists a natural number i such that $f_2 = \mathrm{fsloc}(i)$.

(21) For every instruction-location i_1 of $\mathbf{SCM}_{\mathrm{FSA}}$ there exists a natural number i such that $i_1 = \mathrm{insloc}(i)$.

(22) Int-Locations is infinite.

(23) FinSeq-Locations is infinite.

(24) The instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$ is infinite.

(25) Every integer location is a data-location.

(26) For every integer location l holds $\mathrm{ObjectKind}(l) = \mathbb{Z}$.

(27) For every finite sequence location l holds $\mathrm{ObjectKind}(l) = \mathbb{Z}^*$.

(28) For arbitrary x such that $x \in \mathrm{Data\text{-}Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ holds x is an integer location.

(29) For arbitrary x such that $x \in \mathrm{Data^*\text{-}Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ holds x is a finite sequence location.

(30) For arbitrary x such that $x \in \mathrm{Instr\text{-}Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ holds x is an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$.

Let l_1 be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. The functor $\mathrm{Next}(l_1)$ yields an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$ and is defined by:

(Def. 9) There exists an element m_1 of $\mathrm{Instr\text{-}Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ such that $m_1 = l_1$ and $\mathrm{Next}(l_1) = \mathrm{Next}(m_1)$.

Next we state two propositions:

(31) For every instruction-location l_1 of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every element m_1 of $\mathrm{Instr\text{-}Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ such that $m_1 = l_1$ holds $\mathrm{Next}(m_1) = \mathrm{Next}(l_1)$.

(32) $\mathrm{Next}(\mathrm{insloc}(k)) = \mathrm{insloc}(k + 1)$.

For simplicity we adopt the following convention: l_2, l_3 are instructions-locations of $\mathbf{SCM}_{\mathrm{FSA}}$, L_1 is an instruction-location of $\mathbf{SCM}$, i is an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, I is an instruction of $\mathbf{SCM}$, l is an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$, f, f_1, g are finite sequence locations, A, B are data-locations, and a, b, c, d_1, d_3 are integer locations.

We now state the proposition

(33) If $l_2 = L_1$, then $\mathrm{Next}(l_2) = \mathrm{Next}(L_1)$.

4. The Instruction Structure

Let I be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. The functor $\mathrm{InsCode}(I)$ yielding a natural number is defined as follows:

(Def. 10) $\mathrm{InsCode}(I) = I_{\mathbf{1}}$.

The following propositions are true:

(34) For every instruction I of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $\mathrm{InsCode}(I) \leq 8$ holds I is an instruction of $\mathbf{SCM}$.

(35) For every instruction I of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{InsCode}(I) \leq 12$.

(36) For every instruction i of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $\mathrm{InsCode}(i) = 0$ holds $i = \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(37) For every instruction i of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every instruction I of $\mathbf{SCM}$ such that $i = I$ holds $\mathrm{InsCode}(i) = \mathrm{InsCode}(I)$.

(38) Every instruction of $\mathbf{SCM}$ is an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$.

Let us consider a, b. The functor $a{:=}b$ yields an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and is defined as follows:

(Def. 11) There exist A, B such that $a = A$ and $b = B$ and $a{:=}b = A{:=}B$.

The functor $\mathrm{AddTo}(a, b)$ yields an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and is defined by:

(Def. 12) There exist A, B such that $a = A$ and $b = B$ and $\mathrm{AddTo}(a, b) = \mathrm{AddTo}(A, B)$.

The functor $\mathrm{SubFrom}(a, b)$ yields an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and is defined as follows:

(Def. 13) There exist A, B such that $a = A$ and $b = B$ and $\mathrm{SubFrom}(a, b) = \mathrm{SubFrom}(A, B)$.

The functor $\mathrm{MultBy}(a, b)$ yields an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and is defined as follows:

(Def. 14) There exist A, B such that $a = A$ and $b = B$ and $\mathrm{MultBy}(a, b) = \mathrm{MultBy}(A, B)$.

The functor $\mathrm{Divide}(a, b)$ yielding an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ is defined as follows:

(Def. 15) There exist A, B such that $a = A$ and $b = B$ and $\mathrm{Divide}(a, b) = \mathrm{Divide}(A, B)$.

We now state the proposition

(39) The instruction locations of $\mathbf{SCM}$ = the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$.

Let us consider l_2. The functor goto l_2 yields an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and is defined as follows:

(Def. 16) There exists L_1 such that $l_2 = L_1$ and goto l_2 = goto L_1.

Let us consider a. The functor **if** $a = 0$ **goto** l_2 yields an instruction of $\mathbf{SCM}_{\rm FSA}$ and is defined by:

(Def. 17) There exist A, L_1 such that $a = A$ and $l_2 = L_1$ and **if** $a = 0$ **goto** $l_2 =$ **if** $A = 0$ **goto** L_1.

The functor **if** $a > 0$ **goto** l_2 yields an instruction of $\mathbf{SCM}_{\rm FSA}$ and is defined as follows:

(Def. 18) There exist A, L_1 such that $a = A$ and $l_2 = L_1$ and **if** $a > 0$ **goto** $l_2 =$ **if** $A > 0$ **goto** L_1.

Let c, i be integer locations and let a be a finite sequence location. The functor $c{:=}a_i$ yielding an instruction of $\mathbf{SCM}_{\rm FSA}$ is defined by:

(Def. 19) $c{:=}a_i = \langle 9, \langle c, a, i\rangle\rangle$.

The functor $a_i{:=}c$ yielding an instruction of $\mathbf{SCM}_{\rm FSA}$ is defined by:

(Def. 20) $a_i{:=}c = \langle 10, \langle c, a, i\rangle\rangle$.

Let i be an integer location and let a be a finite sequence location. The functor $i{:=}\mathrm{len}a$ yielding an instruction of $\mathbf{SCM}_{\rm FSA}$ is defined as follows:

(Def. 21) $i{:=}\mathrm{len}a = \langle 11, \langle i, a\rangle\rangle$.

The functor $a{:=}\langle\underbrace{0, \ldots, 0}_{i}\rangle$ yields an instruction of $\mathbf{SCM}_{\rm FSA}$ and is defined as follows:

(Def. 22) $a{:=}\langle\underbrace{0, \ldots, 0}_{i}\rangle = \langle 12, \langle i, a\rangle\rangle$.

We now state a number of propositions:

(40) $\mathbf{halt}_{\mathbf{SCM}} = \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}$.

(41) $\mathrm{InsCode}(\mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}) = 0$.

(42) $\mathrm{InsCode}(a{:=}b) = 1$.

(43) $\mathrm{InsCode}(\mathrm{AddTo}(a, b)) = 2$.

(44) $\mathrm{InsCode}(\mathrm{SubFrom}(a, b)) = 3$.

(45) $\mathrm{InsCode}(\mathrm{MultBy}(a, b)) = 4$.

(46) $\mathrm{InsCode}(\mathrm{Divide}(a, b)) = 5$.

(47) $\mathrm{InsCode}(\mathrm{goto}\ l_3) = 6$.

(48) InsCode(**if** $a = 0$ **goto** l_3) $= 7$.

(49) InsCode(**if** $a > 0$ **goto** l_3) $= 8$.

(50) $\mathrm{InsCode}(c{:=}f_a) = 9$.

(51) $\mathrm{InsCode}(f_a{:=}c) = 10$.

(52) $\mathrm{InsCode}(a{:=}\mathrm{len}f_1) = 11$.

(53) $\mathrm{InsCode}(f_1{:=}\langle\underbrace{0, \ldots, 0}_{a}\rangle) = 12$.

(54) For every instruction i_2 of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{InsCode}(i_2) = 1$ there exist d_1, d_3 such that $i_2 = d_1{:=}d_3$.

(55) For every instruction i_2 of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{InsCode}(i_2) = 2$ there exist d_1, d_3 such that $i_2 = \mathrm{AddTo}(d_1, d_3)$.

(56) For every instruction i_2 of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{InsCode}(i_2) = 3$ there exist d_1, d_3 such that $i_2 = \mathrm{SubFrom}(d_1, d_3)$.

(57) For every instruction i_2 of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{InsCode}(i_2) = 4$ there exist d_1, d_3 such that $i_2 = \mathrm{MultBy}(d_1, d_3)$.

(58) For every instruction i_2 of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{InsCode}(i_2) = 5$ there exist d_1, d_3 such that $i_2 = \mathrm{Divide}(d_1, d_3)$.

(59) For every instruction i_2 of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{InsCode}(i_2) = 6$ there exists l_3 such that $i_2 = \mathrm{goto}\ l_3$.

(60) For every instruction i_2 of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{InsCode}(i_2) = 7$ there exist l_3, d_1 such that $i_2 = \mathbf{if}\ d_1 = 0\ \mathbf{goto}\ l_3$.

(61) For every instruction i_2 of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{InsCode}(i_2) = 8$ there exist l_3, d_1 such that $i_2 = \mathbf{if}\ d_1 > 0\ \mathbf{goto}\ l_3$.

(62) For every instruction i_2 of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{InsCode}(i_2) = 9$ there exist a, b, f_1 such that $i_2 = b{:=}{f_1}_a$.

(63) For every instruction i_2 of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{InsCode}(i_2) = 10$ there exist a, b, f_1 such that $i_2 = {f_1}_a{:=}b$.

(64) For every instruction i_2 of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{InsCode}(i_2) = 11$ there exist a, f_1 such that $i_2 = a{:=}\mathrm{len} f_1$.

(65) For every instruction i_2 of $\mathbf{SCM}_{\rm FSA}$ such that $\mathrm{InsCode}(i_2) = 12$ there exist a, f_1 such that $i_2 = f_1{:=}\langle\underbrace{0, \ldots, 0}_{a}\rangle$.

5. Relationship to SCM

In the sequel S denotes a state of $\mathbf{SCM}$ and s, s_1 denote states of $\mathbf{SCM}_{\rm FSA}$.

We now state a number of propositions:

(66) For every state s of $\mathbf{SCM}_{\rm FSA}$ and for every integer location d holds $d \in \mathrm{dom}\, s$.

(67) $f \in \mathrm{dom}\, s$.

(68) $f \notin \mathrm{dom}\, S$.

(69) For every state s of $\mathbf{SCM}_{\rm FSA}$ holds $\text{Int-Locations} \subseteq \mathrm{dom}\, s$.

(70) For every state s of $\mathbf{SCM}_{\rm FSA}$ holds $\text{FinSeq-Locations} \subseteq \mathrm{dom}\, s$.

(71) For every state s of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{dom}(s \restriction \text{Int-Locations}) = \text{Int-Locations}$.

(72) For every state s of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{dom}(s \restriction \text{FinSeq-Locations}) = \text{FinSeq-Locations}$.

(73) For every state s of $\mathbf{SCM}_{\rm FSA}$ and for every instruction i of $\mathbf{SCM}$ holds $s \restriction \mathbb{N} {+\cdot} (\text{Instr-Loc}_{\mathbf{SCM}} \longmapsto i)$ is a state of $\mathbf{SCM}$.

(74) For every state s of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every state s' of $\mathbf{SCM}$ holds $s{+\cdot}s'{+\cdot}s \restriction \text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ is a state of $\mathbf{SCM}_{\mathrm{FSA}}$.

(75) Let i be an instruction of $\mathbf{SCM}$, and let i_3 be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$, and let s be a state of $\mathbf{SCM}$, and let s_2 be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. If $i = i_3$ and $s = s_2 \restriction \mathbb{N}{+\cdot}(\text{Instr-Loc}_{\mathrm{SCM}} \longmapsto i)$, then $\text{Exec}(i_3, s_2) = s_2{+\cdot}\text{Exec}(i, s){+\cdot}s_2 \restriction \text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let d be an integer location. Then $s(d)$ is an integer.

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

Next we state several propositions:

(76) If $S = s \restriction \mathbb{N}{+\cdot}(\text{Instr-Loc}_{\mathrm{SCM}} \longmapsto I)$, then $s = s{+\cdot}S{+\cdot}s \restriction \text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$.

(77) For every element I of $\text{Instr}_{\mathrm{SCM}_{\mathrm{FSA}}}$ such that $I = i$ and for every $\mathbf{SCM}_{\mathrm{FSA}}$-state S such that $S = s$ holds $\text{Exec}(i, s) = \text{Exec-Res}_{\mathrm{SCM}_{\mathrm{FSA}}}(I, S)$.

(78) If $s_1 = s{+\cdot}S{+\cdot}s \restriction \text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$, then $s_1(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = S(\mathbf{IC}_{\mathbf{SCM}})$.

(79) If $s_1 = s{+\cdot}S{+\cdot}s \restriction \text{Instr-Loc}_{\mathrm{SCM}_{\mathrm{FSA}}}$ and $A = a$, then $S(A) = s_1(a)$.

(80) If $S = s \restriction \mathbb{N}{+\cdot}(\text{Instr-Loc}_{\mathrm{SCM}} \longmapsto I)$ and $A = a$, then $S(A) = s(a)$.

Let us note that $\mathbf{SCM}_{\mathrm{FSA}}$ is halting realistic von Neumann data-oriented definite and steady-programmed.

The following propositions are true:

(81) For every integer location d_2 holds $d_2 \neq \mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(82) For every finite sequence location d_2 holds $d_2 \neq \mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(83) For every integer location i_1 and for every finite sequence location d_2 holds $i_1 \neq d_2$.

(84) For every instruction-location i_1 of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every integer location d_2 holds $i_1 \neq d_2$.

(85) For every instruction-location i_1 of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every finite sequence location d_2 holds $i_1 \neq d_2$.

(86) Let s_1, s_3 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose that
  (i) $\mathbf{IC}_{(s_1)} = \mathbf{IC}_{(s_3)}$,
  (ii) for every integer location a holds $s_1(a) = s_3(a)$,
  (iii) for every finite sequence location f holds $s_1(f) = s_3(f)$, and
  (iv) for every instruction-location i of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $s_1(i) = s_3(i)$.

  Then $s_1 = s_3$.

(87) If $S = s$, then $\mathbf{IC}_s = \mathbf{IC}_S$.

(88) If $S = s \restriction \mathbb{N}{+\cdot}(\text{Instr-Loc}_{\mathrm{SCM}} \longmapsto I)$, then $\mathbf{IC}_s = \mathbf{IC}_S$.

6. Users Guide

One can prove the following propositions:

(89) $(\mathrm{Exec}(a{:=}b, s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(a{:=}b, s))(a) = s(b)$ and for every c such that $c \neq a$ holds $(\mathrm{Exec}(a{:=}b, s))(c) = s(c)$ and for every f holds $(\mathrm{Exec}(a{:=}b, s))(f) = s(f)$.

(90) $(\mathrm{Exec}(\mathrm{AddTo}(a, b), s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(\mathrm{AddTo}(a, b), s))(a) = s(a) + s(b)$ and for every c such that $c \neq a$ holds $(\mathrm{Exec}(\mathrm{AddTo}(a, b), s))(c) = s(c)$ and for every f holds $(\mathrm{Exec}(\mathrm{AddTo}(a, b), s))(f) = s(f)$.

(91) $(\mathrm{Exec}(\mathrm{SubFrom}(a, b), s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(\mathrm{SubFrom}(a, b), s))(a) = s(a) - s(b)$ and for every c such that $c \neq a$ holds $(\mathrm{Exec}(\mathrm{SubFrom}(a, b), s))(c) = s(c)$ and for every f holds $(\mathrm{Exec}(\mathrm{SubFrom}(a, b), s))(f) = s(f)$.

(92) $(\mathrm{Exec}(\mathrm{MultBy}(a, b), s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(\mathrm{MultBy}(a, b), s))(a) = s(a) \cdot s(b)$ and for every c such that $c \neq a$ holds $(\mathrm{Exec}(\mathrm{MultBy}(a, b), s))(c) = s(c)$ and for every f holds $(\mathrm{Exec}(\mathrm{MultBy}(a, b), s))(f) = s(f)$.

(93) Suppose $a \neq b$. Then
- (i) $(\mathrm{Exec}(\mathrm{Divide}(a, b), s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{Next}(\mathbf{IC}_s)$,
- (ii) $(\mathrm{Exec}(\mathrm{Divide}(a, b), s))(a) = s(a) \div s(b)$,
- (iii) $(\mathrm{Exec}(\mathrm{Divide}(a, b), s))(b) = s(a) \bmod s(b)$,
- (iv) for every c such that $c \neq a$ and $c \neq b$ holds $(\mathrm{Exec}(\mathrm{Divide}(a, b), s))(c) = s(c)$, and
- (v) for every f holds $(\mathrm{Exec}(\mathrm{Divide}(a, b), s))(f) = s(f)$.

(94) $(\mathrm{Exec}(\mathrm{Divide}(a, a), s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(\mathrm{Divide}(a, a), s))(a) = s(a) \bmod s(a)$ and for every c such that $c \neq a$ holds $(\mathrm{Exec}(\mathrm{Divide}(a, a), s))(c) = s(c)$ and for every f holds $(\mathrm{Exec}(\mathrm{Divide}(a, a), s))(f) = s(f)$.

(95) $(\mathrm{Exec}(\mathrm{goto}\ l, s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = l$ and for every c holds $(\mathrm{Exec}(\mathrm{goto}\ l, s))(c) = s(c)$ and for every f holds $(\mathrm{Exec}(\mathrm{goto}\ l, s))(f) = s(f)$.

(96)
- (i) If $s(a) = 0$, then $(\mathrm{Exec}(\mathbf{if}\ a = 0\ \mathbf{goto}\ l, s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = l$,
- (ii) if $s(a) \neq 0$, then $(\mathrm{Exec}(\mathbf{if}\ a = 0\ \mathbf{goto}\ l, s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{Next}(\mathbf{IC}_s)$,
- (iii) for every c holds $(\mathrm{Exec}(\mathbf{if}\ a = 0\ \mathbf{goto}\ l, s))(c) = s(c)$, and
- (iv) for every f holds $(\mathrm{Exec}(\mathbf{if}\ a = 0\ \mathbf{goto}\ l, s))(f) = s(f)$.

(97)
- (i) If $s(a) > 0$, then $(\mathrm{Exec}(\mathbf{if}\ a > 0\ \mathbf{goto}\ l, s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = l$,
- (ii) if $s(a) \leq 0$, then $(\mathrm{Exec}(\mathbf{if}\ a > 0\ \mathbf{goto}\ l, s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{Next}(\mathbf{IC}_s)$,
- (iii) for every c holds $(\mathrm{Exec}(\mathbf{if}\ a > 0\ \mathbf{goto}\ l, s))(c) = s(c)$, and
- (iv) for every f holds $(\mathrm{Exec}(\mathbf{if}\ a > 0\ \mathbf{goto}\ l, s))(f) = s(f)$.

(98)
- (i) $(\mathrm{Exec}(c{:=}g_a, s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{Next}(\mathbf{IC}_s)$,
- (ii) there exists k such that $k = |s(a)|$ and $(\mathrm{Exec}(c{:=}g_a, s))(c) = \pi_k s(g)$,
- (iii) for every b such that $b \neq c$ holds $(\mathrm{Exec}(c{:=}g_a, s))(b) = s(b)$, and
- (iv) for every f holds $(\mathrm{Exec}(c{:=}g_a, s))(f) = s(f)$.

(99)
- (i) $(\mathrm{Exec}(g_a{:=}c, s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{Next}(\mathbf{IC}_s)$,
- (ii) there exists k such that $k = |s(a)|$ and $(\mathrm{Exec}(g_a{:=}c, s))(g) = s(g) +\cdot (k, s(c))$,

(iii) for every b holds $(\mathrm{Exec}(g_a{:=}c, s))(b) = s(b)$, and

(iv) for every f such that $f \neq g$ holds $(\mathrm{Exec}(g_a{:=}c, s))(f) = s(f)$.

(100) $(\mathrm{Exec}(c{:=}\mathrm{len}g, s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{Next}(\mathbf{IC}_s)$ and $(\mathrm{Exec}(c{:=}\mathrm{len}g, s))(c) = \mathrm{len}\, s(g)$ and for every b such that $b \neq c$ holds $(\mathrm{Exec}(c{:=}\mathrm{len}g, s))(b) = s(b)$ and for every f holds $(\mathrm{Exec}(c{:=}\mathrm{len}g, s))(f) = s(f)$.

(101) (i) $(\mathrm{Exec}(g{:=}\langle\underbrace{0, \ldots, 0}_{c}\rangle, s))(\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}) = \mathrm{Next}(\mathbf{IC}_s)$,

(ii) there exists k such that $k = |s(c)|$ and $(\mathrm{Exec}(g{:=}\langle\underbrace{0, \ldots, 0}_{c}\rangle, s))(g) = k \mapsto 0$,

(iii) for every b holds $(\mathrm{Exec}(g{:=}\langle\underbrace{0, \ldots, 0}_{c}\rangle, s))(b) = s(b)$, and

(iv) for every f such that $f \neq g$ holds $(\mathrm{Exec}(g{:=}\langle\underbrace{0, \ldots, 0}_{c}\rangle, s))(f) = s(f)$.

References

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

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

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

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

[5] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 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. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.

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

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

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

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

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

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

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

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

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

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

[20] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[21] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[22] Andrzej Trybulec 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 Piotr Rudnicki. An extension of **SCM**. *Formalized Mathematics*, 5(**4**):507–512, 1996.
[24] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[25] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[26] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[27] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received February 7, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

On the Many Sorted Closure Operator and the Many Sorted Closure System

Artur Korniłowicz
Warsaw University
Białystok

MML Identifier: `CLOSURE1`.

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

1. Preliminaries

For simplicity we follow a convention: I is a set, i, x are arbitrary, A, M are many sorted sets indexed by I, f is a function, and F is a many sorted function of I.

The scheme *MSSUBSET* concerns a set $\mathcal{A}$, a non-empty many sorted set $\mathcal{B}$ indexed by $\mathcal{A}$, a many sorted set $\mathcal{C}$ indexed by $\mathcal{A}$, and a unary predicate $\mathcal{P}$, and states that:

If for every many sorted set X indexed by $\mathcal{A}$ holds $X \in \mathcal{B}$ iff $X \in \mathcal{C}$ and $\mathcal{P}[X]$, then $\mathcal{B} \subseteq \mathcal{C}$

for all values of the parameters.

The following two propositions are true:

(1) Let X be a non empty set and let x, y be arbitrary. If $x \subseteq y$, then $\{t : t$ ranges over elements of X, $y \subseteq t\} \subseteq \{z : z$ ranges over elements of X, $x \subseteq z\}$.

(2) If there exists A such that $A \in M$, then M is non-empty.

Let us consider I, F, A. Then $F \looparrowleft A$ is a many sorted set indexed by I.

Let us consider I, let A, B be non-empty many sorted sets indexed by I, let F be a many sorted function from A into B, and let X be an element of A. Then $F \looparrowleft X$ is an element of B.

One can prove the following propositions:

ISSN 1426–2630

(3) Let A, X be many sorted sets indexed by I, and let B be a non-empty many sorted set indexed by I and let F be a many sorted function from A into B. If $X \in A$, then $F \leftarrow\!\!\!\!\!\!\; \mathrel{\raisebox{0pt}{$\circ$}} X \in B$.

(4) Let F, G be many sorted functions of I and let A be a many sorted set indexed by I. If $A \in \mathrm{dom}_\kappa\, G(\kappa)$, then $F \looparrowleft (G \looparrowleft A) = (F \circ G) \looparrowleft A$.

(5) If F is "1-1", then for all many sorted sets A, B indexed by I such that $A \in \mathrm{dom}_\kappa\, F(\kappa)$ and $B \in \mathrm{dom}_\kappa\, F(\kappa)$ and $F \looparrowleft A = F \looparrowleft B$ holds $A = B$.

(6) Suppose $\mathrm{dom}_\kappa\, F(\kappa)$ is non-empty and for all many sorted sets A, B indexed by I such that $A \in \mathrm{dom}_\kappa\, F(\kappa)$ and $B \in \mathrm{dom}_\kappa\, F(\kappa)$ and $F \looparrowleft A = F \looparrowleft B$ holds $A = B$. Then F is "1-1".

(7) Let A, B be non-empty many sorted sets indexed by I and let F, G be many sorted functions from A into B. If for every M such that $M \in A$ holds $F \looparrowleft M = G \looparrowleft M$, then $F = G$.

Let us consider I, M. One can verify that there exists an element of 2^M which is empty yielding and locally-finite.

2. Properties of Many Sorted Closure Operators

Let us consider I, M.

(Def. 1) A many sorted function from 2^M into 2^M is called a set many sorted operation in M.

Let us consider I, M, let O be a set many sorted operation in M, and let X be an element of 2^M. Then $O \looparrowleft X$ is an element of 2^M.

Let us consider I, M and let I_1 be a set many sorted operation in M. We say that I_1 is reflexive if and only if:

(Def. 2) For every element X of 2^M holds $X \subseteq I_1 \looparrowleft X$.

We say that I_1 is monotonic if and only if:

(Def. 3) For all elements X, Y of 2^M such that $X \subseteq Y$ holds $I_1 \looparrowleft X \subseteq I_1 \looparrowleft Y$.

We say that I_1 is idempotent if and only if:

(Def. 4) For every element X of 2^M holds $I_1 \looparrowleft X = I_1 \looparrowleft (I_1 \looparrowleft X)$.

We say that I_1 is topological if and only if:

(Def. 5) For all elements X, Y of 2^M holds $I_1 \looparrowleft (X \cup Y) = I_1 \looparrowleft X \cup I_1 \looparrowleft Y$.

One can prove the following propositions:

(8) For every non-empty many sorted set M indexed by I and for every element X of M holds $X = \mathrm{id}_M \looparrowleft X$.

(9) Let M be a non-empty many sorted set indexed by I and let X, Y be elements of M. If $X \subseteq Y$, then $\mathrm{id}_M \looparrowleft X \subseteq \mathrm{id}_M \looparrowleft Y$.

(10) Let M be a non-empty many sorted set indexed by I and let X, Y be elements of M. If $X \cup Y$ is an element of M, then $\mathrm{id}_M \looparrowleft (X \cup Y) = \mathrm{id}_M \looparrowleft X \cup \mathrm{id}_M \looparrowleft Y$.

(11) Let X be an element of 2^M and let i, x be arbitrary. Suppose $i \in I$ and $x \in (\mathrm{id}_{2^M} \looparrowleft X)(i)$. Then there exists a locally-finite element Y of 2^M such that $Y \subseteq X$ and $x \in (\mathrm{id}_{2^M} \looparrowleft Y)(i)$.

Let us consider I, M. Note that there exists a set many sorted operation in M which is reflexive monotonic idempotent and topological.

Next we state four propositions:

(12) id_{2^A} is a reflexive set many sorted operation in A.

(13) id_{2^A} is a monotonic set many sorted operation in A.

(14) id_{2^A} is an idempotent set many sorted operation in A.

(15) id_{2^A} is a topological set many sorted operation in A.

In the sequel P, R will denote set many sorted operations in M and E, T will denote elements of 2^M.

One can prove the following three propositions:

(16) If $E = M$ and P is reflexive, then $E = P \looparrowleft E$.

(17) If P is reflexive and for every element X of 2^M holds $P \looparrowleft X \subseteq X$, then P is idempotent.

(18) If P is monotonic, then $P \looparrowleft (E \cap T) \subseteq P \looparrowleft E \cap P \looparrowleft T$.

Let us consider I, M. Observe that every set many sorted operation in M which is topological is also monotonic.

One can prove the following proposition

(19) If P is topological, then $P \looparrowleft E \setminus P \looparrowleft T \subseteq P \looparrowleft (E \setminus T)$.

Let us consider I, M, R, P. Then $P \circ R$ is a set many sorted operation in M.

One can prove the following propositions:

(20) If P is reflexive and R is reflexive, then $P \circ R$ is reflexive.

(21) If P is monotonic and R is monotonic, then $P \circ R$ is monotonic.

(22) If P is idempotent and R is idempotent and $P \circ R = R \circ P$, then $P \circ R$ is idempotent.

(23) If P is topological and R is topological, then $P \circ R$ is topological.

(24) If P is reflexive and $i \in I$ and $f = P(i)$, then for every element x of $2^{M(i)}$ holds $x \subseteq f(x)$.

(25) If P is monotonic and $i \in I$ and $f = P(i)$, then for all elements x, y of $2^{M(i)}$ such that $x \subseteq y$ holds $f(x) \subseteq f(y)$.

(26) If P is idempotent and $i \in I$ and $f = P(i)$, then for every element x of $2^{M(i)}$ holds $f(x) = f(f(x))$.

(27) If P is topological and $i \in I$ and $f = P(i)$, then for all elements x, y of $2^{M(i)}$ holds $f(x \cup y) = f(x) \cup f(y)$.

3. On the Many Sorted Closure Operator and the Many Sorted Closure System

In the sequel S will be a 1-sorted structure.

Let us consider S. We consider many sorted closure system structures over S as extensions of many-sorted structure over S as systems

⟨ sorts, a family ⟩,

where the sorts constitute a many sorted set indexed by the carrier of S and the family is a subset family of the sorts.

In the sequel M_1 will be a many-sorted structure over S.

Let us consider S and let I_1 be a many sorted closure system structure over S. We say that I_1 is additive if and only if:

(Def. 6) The family of I_1 is additive.

We say that I_1 is absolutely-additive if and only if:

(Def. 7) The family of I_1 is absolutely-additive.

We say that I_1 is multiplicative if and only if:

(Def. 8) The family of I_1 is multiplicative.

We say that I_1 is absolutely-multiplicative if and only if:

(Def. 9) The family of I_1 is absolutely-multiplicative.

We say that I_1 is properly upper bound if and only if:

(Def. 10) The family of I_1 is properly upper bound.

We say that I_1 is properly lower bound if and only if:

(Def. 11) The family of I_1 is properly lower bound.

Let us consider S, M_1. The functor $\mathrm{MSFull}(M_1)$ yields a many sorted closure system structure over S and is defined as follows:

(Def. 12) $\mathrm{MSFull}(M_1) = \langle \text{the sorts of } M_1, 2^{\text{the sorts of } M_1} \rangle$.

Let us consider S, M_1. One can check that $\mathrm{MSFull}(M_1)$ is strict additive absolutely-additive multiplicative absolutely-multiplicative properly upper bound and properly lower bound.

Let us consider S and let M_1 be a non-empty many-sorted structure over S. One can check that $\mathrm{MSFull}(M_1)$ is non-empty.

Let us consider S. Observe that there exists a many sorted closure system structure over S which is strict non-empty additive absolutely-additive multiplicative absolutely-multiplicative properly upper bound and properly lower bound.

Let us consider S and let C_1 be an additive many sorted closure system structure over S. Note that the family of C_1 is additive.

Let us consider S and let C_1 be an absolutely-additive many sorted closure system structure over S. Observe that the family of C_1 is absolutely-additive.

Let us consider S and let C_1 be a multiplicative many sorted closure system structure over S. One can verify that the family of C_1 is multiplicative.

Let us consider S and let C_1 be an absolutely-multiplicative many sorted closure system structure over S. One can check that the family of C_1 is absolutely-multiplicative.

Let us consider S and let C_1 be a properly upper bound many sorted closure system structure over S. One can check that the family of C_1 is properly upper bound.

Let us consider S and let C_1 be a properly lower bound many sorted closure system structure over S. Note that the family of C_1 is properly lower bound.

Let us consider S, let M be a non-empty many sorted set indexed by the carrier of S, and let F be a subset family of M. Observe that $\langle M, F\rangle$ is non-empty.

Let us consider S, M_1 and let F be an additive subset family of the sorts of M_1. Observe that $\langle$the sorts of M_1, $F\rangle$ is additive.

Let us consider S, M_1 and let F be an absolutely-additive subset family of the sorts of M_1. One can check that $\langle$the sorts of M_1, $F\rangle$ is absolutely-additive.

Let us consider S, M_1 and let F be a multiplicative subset family of the sorts of M_1. Note that $\langle$the sorts of M_1, $F\rangle$ is multiplicative.

Let us consider S, M_1 and let F be an absolutely-multiplicative subset family of the sorts of M_1. Observe that $\langle$the sorts of M_1, $F\rangle$ is absolutely-multiplicative.

Let us consider S, M_1 and let F be a properly upper bound subset family of the sorts of M_1. One can verify that $\langle$the sorts of M_1, $F\rangle$ is properly upper bound.

Let us consider S, M_1 and let F be a properly lower bound subset family of the sorts of M_1. Observe that $\langle$the sorts of M_1, $F\rangle$ is properly lower bound.

Let us consider S. Observe that every many sorted closure system structure over S which is absolutely-additive is also additive.

Let us consider S. One can check that every many sorted closure system structure over S which is absolutely-multiplicative is also multiplicative.

Let us consider S. Observe that every many sorted closure system structure over S which is absolutely-multiplicative is also properly upper bound.

Let us consider S. One can verify that every many sorted closure system structure over S which is absolutely-additive is also properly lower bound.

Let us consider S. A many sorted closure system of S is an absolutely-multiplicative many sorted closure system structure over S.

Let us consider I, M. A many sorted closure operator of M is a reflexive monotonic idempotent set many sorted operation in M.

Let us consider I, M and let F be a many sorted function from M into M. The functor FixPoints(F) yielding a many sorted subset of M is defined by:

(Def. 13) For every i such that $i \in I$ holds $x \in (\mathrm{FixPoints}(F))(i)$ iff there exists a function f such that $f = F(i)$ and $x \in \mathrm{dom}\, f$ and $f(x) = x$.

Let us consider I, let M be an empty yielding many sorted set indexed by I, and let F be a many sorted function from M into M. One can verify that FixPoints(F) is empty yielding.

Next we state a number of propositions:

(28) For every many sorted function F from M into M holds $A \in M$ and $F \looparrowleft A = A$ iff $A \in \mathrm{FixPoints}(F)$.

(29) $\mathrm{FixPoints}(\mathrm{id}_A) = A$.

(30) Let A be a many sorted set indexed by the carrier of S, and let J be a reflexive monotonic set many sorted operation in A, and let D be a subset family of A. If $D = \mathrm{FixPoints}(J)$, then $\langle A, D \rangle$ is a many sorted closure system of S.

(31) Let D be a properly upper bound subset family of M and let X be an element of 2^M. Then there exists a non-empty subset family S_1 of M such that for every many sorted set Y indexed by I holds $Y \in S_1$ if and only if the following conditions are satisfied:

(i) $Y \in D$, and

(ii) $X \subseteq Y$.

(32) Let D be a properly upper bound subset family of M, and let X be an element of 2^M, and let S_1 be a non-empty subset family of M. Suppose that for every many sorted set Y indexed by I holds $Y \in S_1$ iff $Y \in D$ and $X \subseteq Y$. Let i be arbitrary and let D_1 be a non empty set. If $i \in I$ and $D_1 = D(i)$, then $S_1(i) = \{z : z$ ranges over elements of D_1, $X(i) \subseteq z\}$.

(33) Let D be a properly upper bound subset family of M. Then there exists a set many sorted operation J in M such that for every element X of 2^M and for every non-empty subset family S_1 of M if for every many sorted set Y indexed by I holds $Y \in S_1$ iff $Y \in D$ and $X \subseteq Y$, then $J \looparrowleft X = \bigcap S_1$.

(34) Let D be a properly upper bound subset family of M, and let A be an element of 2^M, and let J be a set many sorted operation in M. Suppose that

(i) $A \in D$, and

(ii) for every element X of 2^M and for every non-empty subset family S_1 of M such that for every many sorted set Y indexed by I holds $Y \in S_1$ iff $Y \in D$ and $X \subseteq Y$ holds $J \looparrowleft X = \bigcap S_1$.

Then $J \looparrowleft A = A$.

(35) Let D be an absolutely-multiplicative subset family of M, and let A be an element of 2^M, and let J be a set many sorted operation in M. Suppose that

(i) $J \looparrowleft A = A$, and

(ii) for every element X of 2^M and for every non-empty subset family S_1 of M such that for every many sorted set Y indexed by I holds $Y \in S_1$ iff $Y \in D$ and $X \subseteq Y$ holds $J \looparrowleft X = \bigcap S_1$.

Then $A \in D$.

(36) Let D be a properly upper bound subset family of M and let J be a set many sorted operation in M. Suppose that for every element X of 2^M and for every non-empty subset family S_1 of M such that for every many sorted set Y indexed by I holds $Y \in S_1$ iff $Y \in D$ and $X \subseteq Y$ holds

$J \leftrightarrow X = \bigcap S_1$. Then J is reflexive and monotonic.

(37) Let D be an absolutely-multiplicative subset family of M and let J be a set many sorted operation in M. Suppose that for every element X of 2^M and for every non-empty subset family S_1 of M such that for every many sorted set Y indexed by I holds $Y \in S_1$ iff $Y \in D$ and $X \subseteq Y$ holds $J \leftrightarrow X = \bigcap S_1$. Then J is idempotent.

(38) Let D be a many sorted closure system of S and let J be a set many sorted operation in the sorts of D. Suppose that for every element X of $2^{\text{the sorts of } D}$ and for every non-empty subset family S_1 of the sorts of D such that for every many sorted set Y indexed by the carrier of S holds $Y \in S_1$ iff $Y \in$ the family of D and $X \subseteq Y$ holds $J \leftrightarrow X = \bigcap S_1$. Then J is a many sorted closure operator of the sorts of D.

Let us consider S, let A be a many sorted set indexed by the carrier of S, and let C be a many sorted closure operator of A. The functor $\mathrm{ClSys}(C)$ yielding a many sorted closure system of S is defined as follows:

(Def. 14) There exists a subset family D of A such that $D = \mathrm{FixPoints}(C)$ and $\mathrm{ClSys}(C) = \langle A, D \rangle$.

Let us consider S, let A be a many sorted set indexed by the carrier of S, and let C be a many sorted closure operator of A. One can verify that $\mathrm{ClSys}(C)$ is strict.

Let us consider S, let A be a non-empty many sorted set indexed by the carrier of S, and let C be a many sorted closure operator of A. Note that $\mathrm{ClSys}(C)$ is non-empty.

Let us consider S and let D be a many sorted closure system of S. The functor $\mathrm{ClOp}(D)$ yielding a many sorted closure operator of the sorts of D is defined by the condition (Def. 15).

(Def. 15) Let X be an element of $2^{\text{the sorts of } D}$ and let S_1 be a non-empty subset family of the sorts of D. Suppose that for every many sorted set Y indexed by the carrier of S holds $Y \in S_1$ iff $Y \in$ the family of D and $X \subseteq Y$. Then $(\mathrm{ClOp}(D)) \leftrightarrow X = \bigcap S_1$.

The following two propositions are true:

(39) Let A be a many sorted set indexed by the carrier of S and let J be a many sorted closure operator of A. Then $\mathrm{ClOp}(\mathrm{ClSys}(J)) = J$.

(40) For every many sorted closure system D of S holds $\mathrm{ClSys}(\mathrm{ClOp}(D)) =$ the many sorted closure system structure of D.

References

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

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

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

[4] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[5] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[6] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.
[7] Czesław Byliński. 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] Artur Korniłowicz. Certain facts about families of subsets of many sorted sets. *Formalized Mathematics*, 5(**3**):451–456, 1996.
[10] Artur Korniłowicz. Definitions and basic properties of boolean & union of many sorted sets. *Formalized Mathematics*, 5(**2**):279–281, 1996.
[11] Artur Korniłowicz. Extensions of mappings on generator set. *Formalized Mathematics*, 5(**2**):269–272, 1996.
[12] Małgorzata Korolkiewicz. Homomorphisms of many sorted algebras. *Formalized Mathematics*, 5(**1**):61–65, 1996.
[13] Beata Madras. Product of family of universal algebras. *Formalized Mathematics*, 4(**1**):103–108, 1993.
[14] Beata Madras. Products of many sorted algebras. *Formalized Mathematics*, 5(**1**):55–60, 1996.
[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] Alexander Yu. Shibakov and Andrzej Trybulec. The Cantor set. *Formalized Mathematics*, 5(**2**):233–236, 1996.
[18] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[19] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[20] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[21] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[22] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received February 7, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

Computation in $\mathbf{SCM}_{\mathrm{FSA}}$

Andrzej Trybulec
Warsaw University
Białystok

Yatsuka Nakamura
Shinshu University
Nagano

Summary. The properties of computations in $\mathbf{SCM}_{\mathrm{FSA}}$ are investigated.

MML Identifier: SCMFSA_3.

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

1. Preliminaries

One can prove the following propositions:

(1) $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \notin$ Int-Locations.

(2) $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \notin$ FinSeq-Locations.

(3) Let i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let I be an instruction of $\mathbf{SCM}$. Suppose $i = I$. Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let S be a state of $\mathbf{SCM}$. Suppose $S = s \restriction$ (the objects of $\mathbf{SCM}$)$+\cdot$((the instruction locations of $\mathbf{SCM}$) $\longmapsto (I)$). Then $\mathrm{Exec}(i, s) = s{+\cdot}\mathrm{Exec}(I, S){+\cdot}s \restriction$ (the instruction locations of $\mathbf{SCM}_{\mathrm{FSA}}$).

(4) Let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $s_1 \restriction$ (Int-Locations $\cup$ FinSeq-Locations $\cup \{\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\}) = s_2 \restriction$(Int-Locations $\cup$ FinSeq-Locations $\cup \{\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\}$). Let l be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $\mathrm{Exec}(l, s_1) \restriction$ (Int-Locations $\cup$ FinSeq-Locations $\cup \{\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\}) =$ $\mathrm{Exec}(l, s_2) \restriction$ (Int-Locations $\cup$ FinSeq-Locations $\cup \{\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}}\}$).

(5) Let N be a non empty set with non empty elements, and let S be a steady-programmed AMI over N, and let i be an instruction of S, and let s

ISSN 1426–2630

be a state of S. Then $\mathrm{Exec}(i,s)\restriction$(the instruction locations of S) $= s\restriction$(the instruction locations of S).

2. Finite partial states of $\mathbf{SCM}_{\rm FSA}$

One can prove the following two propositions:

(6) For every finite partial state p of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{DataPart}(p) = p\restriction(\text{Int-Locations}\cup\text{FinSeq-Locations})$.

(7) For every finite partial state p of $\mathbf{SCM}_{\rm FSA}$ holds p is data-only iff $\mathrm{dom}\,p\subseteq\text{Int-Locations}\cup\text{FinSeq-Locations}$.

Let us observe that there exists a finite partial state of $\mathbf{SCM}_{\rm FSA}$ which is data-only.

We now state two propositions:

(8) For every finite partial state p of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{dom}\,\mathrm{DataPart}(p)\subseteq\text{Int-Locations}\cup\text{FinSeq-Locations}$.

(9) For every finite partial state p of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{dom}\,\mathrm{ProgramPart}(p)\subseteq$ the instruction locations of $\mathbf{SCM}_{\rm FSA}$.

Let I_1 be a partial function from $\mathrm{FinPartSt}(\mathbf{SCM}_{\rm FSA})$ to $\mathrm{FinPartSt}(\mathbf{SCM}_{\rm FSA})$. We say that I_1 is data-only if and only if the condition (Def. 1) is satisfied.

(Def. 1) Let p be a finite partial state of $\mathbf{SCM}_{\rm FSA}$. Suppose $p\in\mathrm{dom}\,I_1$. Then p is data-only and for every finite partial state q of $\mathbf{SCM}_{\rm FSA}$ such that $q = I_1(p)$ holds q is data-only.

One can verify that there exists a partial function from $\mathrm{FinPartSt}(\mathbf{SCM}_{\rm FSA})$ to $\mathrm{FinPartSt}(\mathbf{SCM}_{\rm FSA})$ which is data-only.

One can prove the following four propositions:

(10) Let i be an instruction of $\mathbf{SCM}_{\rm FSA}$, and let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let p be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$. Then $\mathrm{Exec}(i, s{+\cdot}p) = \mathrm{Exec}(i,s){+\cdot}p$.

(11) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let i_1 be an instruction-location of $\mathbf{SCM}_{\rm FSA}$, and let a be an integer location. Then $s(a) = (s{+\cdot}\,\text{Start-At}(i_1))(a)$.

(12) Let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let i_1 be an instruction-location of $\mathbf{SCM}_{\rm FSA}$, and let a be a finite sequence location. Then $s(a) = (s{+\cdot}\,\text{Start-At}(i_1))(a)$.

(13) For all states s, t of $\mathbf{SCM}_{\rm FSA}$ holds $s{+\cdot}t\restriction(\text{Int-Locations}\cup\text{FinSeq-Locations})$ is a state of $\mathbf{SCM}_{\rm FSA}$.

3. Autonomic finite partial states of $\mathbf{SCM}_{\mathrm{FSA}}$

Let l_1 be an integer location and let a be an integer. Then $l_1 \dot{\longmapsto} a$ is a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$.

The following proposition is true

(14) For every autonomic finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $\mathrm{DataPart}(p) \neq \emptyset$ holds $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\, p$.

Let us observe that there exists a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ which is autonomic and non programmed.

We now state a number of propositions:

(15) For every autonomic non programmed finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\, p$.

(16) For every autonomic finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ such that $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\, p$ holds $\mathbf{IC}_p \in \mathrm{dom}\, p$.

(17) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. If $p \subseteq s$, then for every natural number i holds $\mathbf{IC}_{(\mathrm{Computation}(s))(i)} \in \mathrm{dom}\, \mathrm{ProgramPart}(p)$.

(18) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number. Then $\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)} = \mathbf{IC}_{(\mathrm{Computation}(s_2))(i)}$ and $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{CurInstr}((\mathrm{Computation}(s_2))(i))$.

(19) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number and let d_1, d_2 be integer locations. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = d_1{:=}d_2$ and $d_1 \in \mathrm{dom}\, p$, then $(\mathrm{Computation}(s_1))(i)(d_2) = (\mathrm{Computation}(s_2))(i)(d_2)$.

(20) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number and let d_1, d_2 be integer locations. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{AddTo}(d_1, d_2)$ and $d_1 \in \mathrm{dom}\, p$, then $(\mathrm{Computation}(s_1))(i)(d_1) + (\mathrm{Computation}(s_1))(i)(d_2) = (\mathrm{Computation}(s_2))(i)(d_1) + (\mathrm{Computation}(s_2))(i)(d_2)$.

(21) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number and let d_1, d_2 be integer locations. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{SubFrom}(d_1, d_2)$ and $d_1 \in \mathrm{dom}\, p$, then $(\mathrm{Computation}(s_1))(i)(d_1) - (\mathrm{Computation}(s_1))(i)(d_2) = (\mathrm{Computation}(s_2))(i)(d_1) - (\mathrm{Computation}(s_2))(i)(d_2)$.

(22) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number and let d_1, d_2 be integer locations. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{MultBy}(d_1, d_2)$ and

$d_1 \in \operatorname{dom} p$, then $(\mathrm{Computation}(s_1))(i)(d_1) \cdot (\mathrm{Computation}(s_1))(i)(d_2) = (\mathrm{Computation}(s_2))(i)(d_1) \cdot (\mathrm{Computation}(s_2))(i)(d_2)$.

(23) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number and let d_1, d_2 be integer locations. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{Divide}(d_1, d_2)$ and $d_1 \in \operatorname{dom} p$ and $d_1 \neq d_2$, then $(\mathrm{Computation}(s_1))(i)(d_1) \div (\mathrm{Computation}(s_1))(i)(d_2) = (\mathrm{Computation}(s_2))(i)(d_1) \div (\mathrm{Computation}(s_2))(i)(d_2)$.

(24) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number and let d_1, d_2 be integer locations. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathrm{Divide}(d_1, d_2)$ and $d_2 \in \operatorname{dom} p$ and $d_1 \neq d_2$, then $(\mathrm{Computation}(s_1))(i)(d_1) \bmod (\mathrm{Computation}(s_1))(i)(d_2) = (\mathrm{Computation}(s_2))(i)(d_1) \bmod (\mathrm{Computation}(s_2))(i)(d_2)$.

(25) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, and let d_1 be an integer location, and let l_2 be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathbf{if}\ d_1 = 0\ \mathbf{goto}\ l_2$ and $l_2 \neq \mathrm{Next}(\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)})$, then $(\mathrm{Computation}(s_1))(i)(d_1) = 0$ iff $(\mathrm{Computation}(s_2))(i)(d_1) = 0$.

(26) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, and let d_1 be an integer location, and let l_2 be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = \mathbf{if}\ d_1 > 0\ \mathbf{goto}\ l_2$ and $l_2 \neq \mathrm{Next}(\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)})$, then $(\mathrm{Computation}(s_1))(i)(d_1) > 0$ iff $(\mathrm{Computation}(s_2))(i)(d_1) > 0$.

(27) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, and let d_1, d_2 be integer locations, and let f be a finite sequence location. Suppose $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = d_1{:=}f_{d_2}$ and $d_1 \in \operatorname{dom} p$. Let k_1, k_2 be natural numbers. If $k_1 = |(\mathrm{Computation}(s_1))(i)(d_2)|$ and $k_2 = |(\mathrm{Computation}(s_2))(i)(d_2)|$, then $\pi_{k_1}(\mathrm{Computation}(s_1))(i)(f) = \pi_{k_2}(\mathrm{Computation}(s_2))(i)(f)$.

(28) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, and let d_1, d_2 be integer locations, and let f be a finite sequence location. Suppose $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = f_{d_2}{:=}d_1$ and $f \in \operatorname{dom} p$. Let k_1, k_2 be natural numbers. If $k_1 = |(\mathrm{Computation}(s_1))(i)(d_2)|$ and $k_2 = |(\mathrm{Computation}(s_2))(i)(d_2)|$, then $(\mathrm{Computation}(s_1))(i)(f) +\cdot (k_1, (\mathrm{Computation}(s_1))(i)(d_1)) = (\mathrm{Computation}(s_2))(i)(f) +\cdot (k_2, (\mathrm{Computation}(s_2))(i)(d_1))$.

(29) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, and let d_1 be an integer location, and let f be a finite sequence location. If $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = d_1{:=}\mathrm{len}f$ and $d_1 \in \mathrm{dom}\, p$, then $\mathrm{len}(\mathrm{Computation}(s_1))(i)(f) = \mathrm{len}(\mathrm{Computation}(s_2))(i)(f)$.

(30) Let p be an autonomic non programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s_1$ and $p \subseteq s_2$. Let i be a natural number, and let d_1 be an integer location, and let f be a finite sequence location. Suppose $\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)) = f{:=}\langle\underbrace{0,\ldots,0}_{d_1}\rangle$ and $f \in \mathrm{dom}\, p$. Let k_1, k_2 be natural numbers. If $k_1 = |(\mathrm{Computation}(s_1))(i)(d_1)|$ and $k_2 = |(\mathrm{Computation}(s_2))(i)(d_1)|$, then $k_1 \mapsto 0 = k_2 \mapsto 0$.

References

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

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

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

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

[5] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 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. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.

[10] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 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] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.

[14] Rafał Kwiatek and Grzegorz Zwara. The divisibility of integers and integer relative primes. *Formalized Mathematics*, 1(**5**):829–832, 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] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.

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

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

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

[21] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 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 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. An extension of **SCM**. *Formalized Mathematics*, 5(**4**):507–512, 1996.
[26] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\rm FSA}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[27] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[28] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 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.

Received February 7, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

On the Closure Operator and the Closure System of Many Sorted Sets

Artur Korniłowicz
Warsaw University
Białystok

Summary. In this paper definitions of many sorted closure system and many sorted closure operator are introduced. These notations are also introduced in [11], but in another meaning. In this article closure system is absolutely multiplicative subset family of many sorted sets and in [11] is many sorted absolutely multiplicative subset family of many sorted sets. Analogously, closure operator is function between many sorted sets and in [11] is many sorted function from a many sorted set into a many sorted set.

MML Identifier: CLOSURE2.

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

1. Preliminaries

For simplicity we follow a convention: I will denote a set, i, x will be arbitrary, A, B, M will denote many sorted sets indexed by I, and f, f_1 will denote functions.

One can prove the following three propositions:

(1) For every non empty set M and for all elements X, Y of M such that $X \subseteq Y$ holds $\mathrm{id}_M(X) \subseteq \mathrm{id}_M(Y)$.

(2) If $A \subseteq B$, then $A \setminus M \subseteq B$.

(3) Let I be a non empty set, and let A be a many sorted set indexed by I, and let B be a many sorted subset of A. Then $\mathrm{rng}\, B \subseteq \bigcup \mathrm{rng}(2^A)$.

ISSN 1426–2630

One can check that every set which is empty is also functional.

One can verify that there exists a set which is empty and functional.

Let f, g be functions. Note that $\{f, g\}$ is functional.

2. Set of Many Sorted Subsets of a Many Sorted Set

Let us consider I, M. The functor $\mathrm{Bool}(M)$ yields a set and is defined by:

(Def. 1) $x \in \mathrm{Bool}(M)$ iff x is a many sorted subset of M.

Let us consider I, M. One can verify that $\mathrm{Bool}(M)$ is non empty and functional and has common domain.

Let us consider I, M.

(Def. 2) A subset of $\mathrm{Bool}(M)$ is called a family of many sorted subsets of M.

Let us consider I, M. Then $\mathrm{Bool}(M)$ is a family of many sorted subsets of M.

Let us consider I, M. One can check that there exists a family of many sorted subsets of M which is non empty and functional and has common domain.

Let us consider I, M. One can check that there exists a family of many sorted subsets of M which is empty and finite.

In the sequel S_1, S_2 will denote families of many sorted subsets of M.

Let us consider I, M and let S be a non empty family of many sorted subsets of M. We see that the element of S is a many sorted subset of M.

We now state several propositions:

(4) $S_1 \cup S_2$ is a family of many sorted subsets of M.

(5) $S_1 \cap S_2$ is a family of many sorted subsets of M.

(6) $S_1 \setminus x$ is a family of many sorted subsets of M.

(7) $S_1 \dot{-} S_2$ is a family of many sorted subsets of M.

(8) If $A \subseteq M$, then $\{A\}$ is a family of many sorted subsets of M.

(9) If $A \subseteq M$ and $B \subseteq M$, then $\{A, B\}$ is a family of many sorted subsets of M.

In the sequel E, T are elements of $\mathrm{Bool}(M)$.

One can prove the following four propositions:

(10) $E \cap T \in \mathrm{Bool}(M)$.

(11) $E \cup T \in \mathrm{Bool}(M)$.

(12) $E \setminus A \in \mathrm{Bool}(M)$.

(13) $E \dot{-} T \in \mathrm{Bool}(M)$.

3. Many Sorted Operator corresponding to the Operator on Many Sorted Subsets

Let S be a functional set. The functor $|S|$ yielding a function is defined as follows:

(Def. 3) (i) There exists a non empty functional set A such that $A = S$ and $\mathrm{dom}\,|S| = \bigcap\{\mathrm{dom}\,x : x$ ranges over elements of $A\}$ and for every i such that $i \in \mathrm{dom}\,|S|$ holds $|S|(i) = \{x(i) : x$ ranges over elements of $A\}$ if $S \neq \emptyset$,

(ii) $|S| = \emptyset$, otherwise.

Next we state the proposition

(14) For every non empty family S_1 of many sorted subsets of M holds $\mathrm{dom}\,|S_1| = I$.

Let S be an empty functional set. Observe that $|S|$ is empty.

Let us consider I, M and let S be a family of many sorted subsets of M. The functor $|{:}S{:}|$ yielding a many sorted set indexed by I is defined as follows:

(Def. 4) (i) $|{:}S{:}| = |S|$ if $S \neq \emptyset$,

(ii) $|{:}S{:}| = \emptyset_I$, otherwise.

Let us consider I, M and let S be an empty family of many sorted subsets of M. Note that $|{:}S{:}|$ is empty yielding.

The following proposition is true

(15) If S_1 is non empty, then for every i such that $i \in I$ holds $|{:}S_1{:}|(i) = \{x(i) : x$ ranges over elements of $\mathrm{Bool}(M)$, $x \in S_1\}$.

Let us consider I, M and let S_1 be a non empty family of many sorted subsets of M. Note that $|{:}S_1{:}|$ is non-empty.

One can prove the following propositions:

(16) $\mathrm{dom}\,|\{f\}| = \mathrm{dom}\,f$.

(17) $\mathrm{dom}\,|\{f, f_1\}| = \mathrm{dom}\,f \cap \mathrm{dom}\,f_1$.

(18) If $i \in \mathrm{dom}\,f$, then $|\{f\}|(i) = \{f(i)\}$.

(19) If $i \in I$ and $S_1 = \{f\}$, then $|{:}S_1{:}|(i) = \{f(i)\}$.

(20) If $i \in \mathrm{dom}\,|\{f, f_1\}|$, then $|\{f, f_1\}|(i) = \{f(i), f_1(i)\}$.

(21) If $i \in I$ and $S_1 = \{f, f_1\}$, then $|{:}S_1{:}|(i) = \{f(i), f_1(i)\}$.

Let us consider I, M, S_1. Then $|{:}S_1{:}|$ is a subset family of M.

We now state several propositions:

(22) If $A \in S_1$, then $A \in |{:}S_1{:}|$.

(23) If $S_1 = \{A, B\}$, then $\bigcup |{:}S_1{:}| = A \cup B$.

(24) If $S_1 = \{E, T\}$, then $\bigcap |{:}S_1{:}| = E \cap T$.

(25) Let Z be a many sorted subset of M. Suppose that for every many sorted set Z_1 indexed by I such that $Z_1 \in S_1$ holds $Z \subseteq Z_1$. Then $Z \subseteq \bigcap |{:}S_1{:}|$.

(26) $|{:}\,\mathrm{Bool}(M){:}| = 2^M$.

Let us consider I, M and let I_1 be a family of many sorted subsets of M. We say that I_1 is additive if and only if:

(Def. 5) For all A, B such that $A \in I_1$ and $B \in I_1$ holds $A \cup B \in I_1$.

We say that I_1 is absolutely-additive if and only if:

(Def. 6) For every family F of many sorted subsets of M such that $F \subseteq I_1$ holds $\bigcup|{:}F{:}| \in I_1$.

We say that I_1 is multiplicative if and only if:

(Def. 7) For all A, B such that $A \in I_1$ and $B \in I_1$ holds $A \cap B \in I_1$.

We say that I_1 is absolutely-multiplicative if and only if:

(Def. 8) For every family F of many sorted subsets of M such that $F \subseteq I_1$ holds $\bigcap|{:}F{:}| \in I_1$.

We say that I_1 is properly upper bound if and only if:

(Def. 9) $M \in I_1$.

We say that I_1 is properly lower bound if and only if:

(Def. 10) $\emptyset_I \in I_1$.

Let us consider I, M. Observe that there exists a family of many sorted subsets of M which is non empty functional additive absolutely-additive multiplicative absolutely-multiplicative properly upper bound and properly lower bound and has common domain.

Let us consider I, M. Then Bool(M) is an additive absolutely-additive multiplicative absolutely-multiplicative properly upper bound properly lower bound family of many sorted subsets of M.

Let us consider I, M. Observe that every family of many sorted subsets of M which is absolutely-additive is also additive.

Let us consider I, M. One can verify that every family of many sorted subsets of M which is absolutely-multiplicative is also multiplicative.

Let us consider I, M. One can check that every family of many sorted subsets of M which is absolutely-multiplicative is also properly upper bound.

Let us consider I, M. One can check that every family of many sorted subsets of M which is properly upper bound is also non empty.

Let us consider I, M. One can check that every family of many sorted subsets of M which is absolutely-additive is also properly lower bound.

Let us consider I, M. Note that every family of many sorted subsets of M which is properly lower bound is also non empty.

4. Properties of Closure Operators

Let us consider I, M.

(Def. 11) A function from Bool(M) into Bool(M) is called a set operation in M.

Let us consider I, M, let f be a set operation in M, and let x be an element of Bool(M). Then $f(x)$ is an element of Bool(M).

Let us consider I, M and let I_1 be a set operation in M. We say that I_1 is reflexive if and only if:

(Def. 12) For every element x of $\mathrm{Bool}(M)$ holds $x \subseteq I_1(x)$.

We say that I_1 is monotonic if and only if:

(Def. 13) For all elements x, y of $\mathrm{Bool}(M)$ such that $x \subseteq y$ holds $I_1(x) \subseteq I_1(y)$.

We say that I_1 is idempotent if and only if:

(Def. 14) For every element x of $\mathrm{Bool}(M)$ holds $I_1(x) = I_1(I_1(x))$.

We say that I_1 is topological if and only if:

(Def. 15) For all elements x, y of $\mathrm{Bool}(M)$ holds $I_1(x \cup y) = I_1(x) \cup I_1(y)$.

Let us consider I, M. Observe that there exists a set operation in M which is reflexive monotonic idempotent and topological.

Next we state four propositions:

(27) $\mathrm{id}_{\mathrm{Bool}(A)}$ is a reflexive set operation in A.

(28) $\mathrm{id}_{\mathrm{Bool}(A)}$ is a monotonic set operation in A.

(29) $\mathrm{id}_{\mathrm{Bool}(A)}$ is an idempotent set operation in A.

(30) $\mathrm{id}_{\mathrm{Bool}(A)}$ is a topological set operation in A.

In the sequel g, h are set operations in M.

One can prove the following three propositions:

(31) If $E = M$ and g is reflexive, then $E = g(E)$.

(32) If g is reflexive and for every element X of $\mathrm{Bool}(M)$ holds $g(X) \subseteq X$, then g is idempotent.

(33) For every element A of $\mathrm{Bool}(M)$ such that $A = E \cap T$ holds if g is monotonic, then $g(A) \subseteq g(E) \cap g(T)$.

Let us consider I, M. One can check that every set operation in M which is topological is also monotonic.

Next we state the proposition

(34) For every element A of $\mathrm{Bool}(M)$ such that $A = E \setminus T$ holds if g is topological, then $g(E) \setminus g(T) \subseteq g(A)$.

Let us consider I, M, h, g. Then $g \cdot h$ is a set operation in M.

The following four propositions are true:

(35) If g is reflexive and h is reflexive, then $g \cdot h$ is reflexive.

(36) If g is monotonic and h is monotonic, then $g \cdot h$ is monotonic.

(37) If g is idempotent and h is idempotent and $g \cdot h = h \cdot g$, then $g \cdot h$ is idempotent.

(38) If g is topological and h is topological, then $g \cdot h$ is topological.

5. On the Closure Operator and the Closure System

In the sequel S will be a 1-sorted structure.

Let us consider S. We consider closure system structures over S as extensions of many-sorted structure over S as systems

$\langle$ sorts, a family $\rangle$,

where the sorts constitute a many sorted set indexed by the carrier of S and the family is a family of many sorted subsets of the sorts.

In the sequel M_1 is a many-sorted structure over S.

Let us consider S and let I_1 be a closure system structure over S. We say that I_1 is additive if and only if:

(Def. 16) The family of I_1 is additive.

We say that I_1 is absolutely-additive if and only if:

(Def. 17) The family of I_1 is absolutely-additive.

We say that I_1 is multiplicative if and only if:

(Def. 18) The family of I_1 is multiplicative.

We say that I_1 is absolutely-multiplicative if and only if:

(Def. 19) The family of I_1 is absolutely-multiplicative.

We say that I_1 is properly upper bound if and only if:

(Def. 20) The family of I_1 is properly upper bound.

We say that I_1 is properly lower bound if and only if:

(Def. 21) The family of I_1 is properly lower bound.

Let us consider S, M_1. The functor $\mathrm{Full}(M_1)$ yielding a closure system structure over S is defined as follows:

(Def. 22) $\mathrm{Full}(M_1) = \langle$the sorts of M_1, Bool(the sorts of M_1)$\rangle$.

Let us consider S, M_1. Note that $\mathrm{Full}(M_1)$ is strict additive absolutely-additive multiplicative absolutely-multiplicative properly upper bound and properly lower bound.

Let us consider S and let M_1 be a non-empty many-sorted structure over S. Observe that $\mathrm{Full}(M_1)$ is non-empty.

Let us consider S. Note that there exists a closure system structure over S which is strict non-empty additive absolutely-additive multiplicative absolutely-multiplicative properly upper bound and properly lower bound.

Let us consider S and let C_1 be an additive closure system structure over S. Note that the family of C_1 is additive.

Let us consider S and let C_1 be an absolutely-additive closure system structure over S. Note that the family of C_1 is absolutely-additive.

Let us consider S and let C_1 be a multiplicative closure system structure over S. Note that the family of C_1 is multiplicative.

Let us consider S and let C_1 be an absolutely-multiplicative closure system structure over S. Note that the family of C_1 is absolutely-multiplicative.

Let us consider S and let C_1 be a properly upper bound closure system structure over S. One can verify that the family of C_1 is properly upper bound.

Let us consider S and let C_1 be a properly lower bound closure system structure over S. Observe that the family of C_1 is properly lower bound.

Let us consider S, let M be a non-empty many sorted set indexed by the carrier of S, and let F be a family of many sorted subsets of M. Note that $\langle M, F\rangle$ is non-empty.

Let us consider S, M_1 and let F be an additive family of many sorted subsets of the sorts of M_1. Note that $\langle$the sorts of M_1, $F\rangle$ is additive.

Let us consider S, M_1 and let F be an absolutely-additive family of many sorted subsets of the sorts of M_1. Note that $\langle$the sorts of M_1, $F\rangle$ is absolutely-additive.

Let us consider S, M_1 and let F be a multiplicative family of many sorted subsets of the sorts of M_1. Observe that $\langle$the sorts of M_1, $F\rangle$ is multiplicative.

Let us consider S, M_1 and let F be an absolutely-multiplicative family of many sorted subsets of the sorts of M_1. One can check that $\langle$the sorts of M_1, $F\rangle$ is absolutely-multiplicative.

Let us consider S, M_1 and let F be a properly upper bound family of many sorted subsets of the sorts of M_1. Note that $\langle$the sorts of M_1, $F\rangle$ is properly upper bound.

Let us consider S, M_1 and let F be a properly lower bound family of many sorted subsets of the sorts of M_1. Note that $\langle$the sorts of M_1, $F\rangle$ is properly lower bound.

Let us consider S. Observe that every closure system structure over S which is absolutely-additive is also additive.

Let us consider S. Note that every closure system structure over S which is absolutely-multiplicative is also multiplicative.

Let us consider S. Observe that every closure system structure over S which is absolutely-multiplicative is also properly upper bound.

Let us consider S. One can check that every closure system structure over S which is absolutely-additive is also properly lower bound.

Let us consider S. A closure system of S is an absolutely-multiplicative closure system structure over S.

Let us consider I, M. A closure operator of M is a reflexive monotonic idempotent set operation in M.

Next we state the proposition

(39) Let A be a many sorted set indexed by the carrier of S, and let f be a reflexive monotonic set operation in A, and let D be a family of many sorted subsets of A. Suppose $D = \{x : x$ ranges over elements of $\mathrm{Bool}(A)$, $f(x) = x\}$. Then $\langle A, D\rangle$ is a closure system of S.

Let us consider S, let A be a many sorted set indexed by the carrier of S, and let g be a closure operator of A. The functor $\mathrm{ClSys}(g)$ yielding a strict closure system of S is defined by:

(Def. 23) The sorts of $\mathrm{ClSys}(g) = A$ and the family of $\mathrm{ClSys}(g) = \{x : x$ ranges over elements of $\mathrm{Bool}(A)$, $g(x) = x\}$.

Let us consider S, let A be a closure system of S, and let C be a many sorted subset of the sorts of A. The functor $\overline{C}$ yielding an element of Bool(the sorts of A) is defined by the condition (Def. 24).

(Def. 24) There exists a family F of many sorted subsets of the sorts of A such that $\overline{C} = \bigcap|{:}F{:}|$ and $F = \{X : X$ ranges over elements of Bool(the sorts of A), $C \subseteq X \ \wedge \ X \in$ the family of $A\}$.

One can prove the following propositions:

(40) Let D be a closure system of S, and let a be an element of Bool(the sorts of D), and let f be a set operation in the sorts of D. Suppose $a \in$ the family of D and for every element x of Bool(the sorts of D) holds $f(x) = \overline{x}$. Then $f(a) = a$.

(41) Let D be a closure system of S, and let a be an element of Bool(the sorts of D), and let f be a set operation in the sorts of D. Suppose $f(a) = a$ and for every element x of Bool(the sorts of D) holds $f(x) = \overline{x}$. Then $a \in$ the family of D.

(42) Let D be a closure system of S and let f be a set operation in the sorts of D. Suppose that for every element x of Bool(the sorts of D) holds $f(x) = \overline{x}$. Then f is reflexive monotonic and idempotent.

Let us consider S and let D be a closure system of S. The functor $\mathrm{ClOp}(D)$ yields a closure operator of the sorts of D and is defined by:

(Def. 25) For every element x of Bool(the sorts of D) holds $(\mathrm{ClOp}(D))(x) = \overline{x}$.

Next we state two propositions:

(43) For every many sorted set A indexed by the carrier of S and for every closure operator f of A holds $\mathrm{ClOp}(\mathrm{ClSys}(f)) = f$.

(44) For every closure system D of S holds $\mathrm{ClSys}(\mathrm{ClOp}(D)) =$ the closure system structure of D.

References

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

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

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

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

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

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

[7] Czesław Byliński. 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] Artur Korniłowicz. Certain facts about families of subsets of many sorted sets. *Formalized Mathematics*, 5(**3**):451–456, 1996.
[10] Artur Korniłowicz. Definitions and basic properties of boolean & union of many sorted sets. *Formalized Mathematics*, 5(**2**):279–281, 1996.
[11] Artur Korniłowicz. On the many sorted closure operator and the many sorted closure system. *Formalized Mathematics*, 5(**4**):529–536, 1996.
[12] Małgorzata Korolkiewicz. Homomorphisms of many sorted algebras. *Formalized Mathematics*, 5(**1**):61–65, 1996.
[13] Beata Madras. Product of family of universal algebras. *Formalized Mathematics*, 4(**1**):103–108, 1993.
[14] Beata Madras. Products of many sorted algebras. *Formalized Mathematics*, 5(**1**):55–60, 1996.
[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] Alexander Yu. Shibakov and Andrzej Trybulec. The Cantor set. *Formalized Mathematics*, 5(**2**):233–236, 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] 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 February 7, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

Translations, Endomorphisms, and Stable Equational Theories

Grzegorz Bancerek
Institute of Mathematics
Polish Academy of Sciences

Summary. Equational theories of an algebra, i.e. the equivalence relation closed under translations and endomorphisms, are formalized. The correspondence between equational theories and term rewriting systems is discussed in the paper. We get as the main result that any pair of elements of an algebra belongs to the equational theory generated by a set A of axioms iff the elements are convertible w.r.t. term rewriting reduction determined by A.

The theory is developed according to [24].

MML Identifier: MSUALG_6.

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

1. Endomorphisms and translations

Let S be a non empty many sorted signature, let A be an algebra over S, and let s be a sort symbol of S. An element of A, s is an element of (the sorts of A)(s).

Let I be a set, let A be a many sorted set indexed by I, and let h_1, h_2 be many sorted functions from A into A. Then $h_2 \circ h_1$ is a many sorted function from A into A.

The following two propositions are true:

(1) Let S be a non empty non void many sorted signature, and let A be an algebra over S, and let o be an operation symbol of S, and let a be a set. If $a \in \mathrm{Args}(o, A)$, then a is a function.

ISSN 1426–2630

(2) Let S be a non empty non void many sorted signature, and let A be an algebra over S, and let o be an operation symbol of S, and let a be a function. Suppose $a \in \mathrm{Args}(o, A)$. Then $\mathrm{dom}\, a = \mathrm{dom}\, \mathrm{Arity}(o)$ and for every natural number i such that $i \in \mathrm{dom}\, \mathrm{Arity}(o)$ holds $a(i) \in$ (the sorts of $A)(\pi_i\, \mathrm{Arity}(o))$.

Let S be a non empty non void many sorted signature and let A be an algebra over S. We say that A is feasible if and only if:

(Def. 1) For every operation symbol o of S such that $\mathrm{Args}(o, A) \neq \emptyset$ holds $\mathrm{Result}(o, A) \neq \emptyset$.

Next we state the proposition

(3) Let S be a non empty non void many sorted signature, and let o be an operation symbol of S, and let A be an algebra over S. Then $\mathrm{Args}(o, A) \neq \emptyset$ if and only if for every natural number i such that $i \in \mathrm{dom}\, \mathrm{Arity}(o)$ holds (the sorts of $A)(\pi_i\, \mathrm{Arity}(o)) \neq \emptyset$.

Let S be a non empty non void many sorted signature. One can check that every algebra over S which is non-empty is also feasible.

Let S be a non empty non void many sorted signature. One can check that there exists an algebra over S which is non-empty.

Let S be a non empty non void many sorted signature and let A be an algebra over S. A many sorted function from A into A is called an endomorphism of A if:

(Def. 2) It is a homomorphism of A into A.

In the sequel S is a non empty non void many sorted signature and A is an algebra over S.

Next we state three propositions:

(4) $\mathrm{id}_{(\text{the sorts of } A)}$ is an endomorphism of A.

(5) Let h_1, h_2 be many sorted functions from A into A, and let o be an operation symbol of S, and let a be an element of $\mathrm{Args}(o, A)$. If $a \in \mathrm{Args}(o, A)$, then $h_2\#(h_1\#a) = (h_2 \circ h_1)\#a$.

(6) For all endomorphisms h_1, h_2 of A holds $h_2 \circ h_1$ is an endomorphism of A.

Let S be a non empty non void many sorted signature, let A be an algebra over S, and let h_1, h_2 be endomorphisms of A. Then $h_2 \circ h_1$ is an endomorphism of A.

Let S be a non empty non void many sorted signature. The functor $\mathrm{TranslRel}(S)$ is a binary relation on the carrier of S and is defined by the condition (Def. 3).

(Def. 3) Let s_1, s_2 be sort symbols of S. Then $\langle s_1, s_2\rangle \in \mathrm{TranslRel}(S)$ if and only if there exists an operation symbol o of S such that the result sort of $o = s_2$ and there exists a natural number i such that $i \in \mathrm{dom}\, \mathrm{Arity}(o)$ and $\pi_i\, \mathrm{Arity}(o) = s_1$.

We now state three propositions:

(7) Let S be a non empty non void many sorted signature, and let o be an operation symbol of S, and let A be an algebra over S, and let a be a function. Suppose $a \in \mathrm{Args}(o, A)$. Let i be a natural number and let x be an element of A, $\pi_i \mathrm{Arity}(o)$. Then $a +\cdot (i, x) \in \mathrm{Args}(o, A)$.

(8) Let A_1, A_2 be algebras over S, and let h be a many sorted function from A_1 into A_2, and let o be an operation symbol of S. Suppose $\mathrm{Args}(o, A_1) \neq \emptyset$ and $\mathrm{Args}(o, A_2) \neq \emptyset$. Let i be a natural number. Suppose $i \in \mathrm{dom}\,\mathrm{Arity}(o)$. Let x be an element of A_1, $\pi_i \mathrm{Arity}(o)$. Then $h(\pi_i \mathrm{Arity}(o))(x) \in (\text{the sorts of } A_2)(\pi_i \mathrm{Arity}(o))$.

(9) Let S be a non empty non void many sorted signature, and let o be an operation symbol of S, and let i be a natural number. Suppose $i \in \mathrm{dom}\,\mathrm{Arity}(o)$. Let A_1, A_2 be algebras over S, and let h be a many sorted function from A_1 into A_2, and let a, b be elements of $\mathrm{Args}(o, A_1)$. Suppose $a \in \mathrm{Args}(o, A_1)$ and $h\#a \in \mathrm{Args}(o, A_2)$. Let f, g_1, g_2 be functions. Suppose $f = a$ and $g_1 = h\#a$ and $g_2 = h\#b$. Let x be an element of A_1, $\pi_i \mathrm{Arity}(o)$. If $b = f +\cdot (i, x)$, then $g_2(i) = h(\pi_i \mathrm{Arity}(o))(x)$ and $h\#b = g_1 +\cdot (i, g_2(i))$.

Let S be a non empty non void many sorted signature, let o be an operation symbol of S, let i be a natural number, let A be an algebra over S, and let a be a function. The functor $o_i^A(a, -)$ yields a function and is defined by the conditions (Def. 4).

(Def. 4) (i) $\mathrm{dom}(o_i^A(a, -)) = (\text{the sorts of } A)(\pi_i \mathrm{Arity}(o))$, and

(ii) for every set x such that $x \in (\text{the sorts of } A)(\pi_i \mathrm{Arity}(o))$ holds $o_i^A(a, -)(x) = (\mathrm{Den}(o, A))(a +\cdot (i, x))$.

One can prove the following proposition

(10) Let S be a non empty non void many sorted signature, and let o be an operation symbol of S, and let i be a natural number. Suppose $i \in \mathrm{dom}\,\mathrm{Arity}(o)$. Let A be a feasible algebra over S and let a be a function. Suppose $a \in \mathrm{Args}(o, A)$. Then $o_i^A(a, -)$ is a function from (the sorts of A)($\pi_i \mathrm{Arity}(o)$) into (the sorts of A)(the result sort of o).

Let S be a non empty non void many sorted signature, let s_1, s_2 be sort symbols of S, let A be an algebra over S, and let f be a function. We say that f is an elementary translation in A from s_1 into s_2 if and only if the condition (Def. 5) is satisfied.

(Def. 5) There exists an operation symbol o of S such that

(i) the result sort of $o = s_2$, and

(ii) there exists a natural number i such that $i \in \mathrm{dom}\,\mathrm{Arity}(o)$ and $\pi_i \mathrm{Arity}(o) = s_1$ and there exists a function a such that $a \in \mathrm{Args}(o, A)$ and $f = o_i^A(a, -)$.

One can prove the following propositions:

(11) Let S be a non empty non void many sorted signature, and let s_1, s_2 be sort symbols of S, and let A be a feasible algebra over S, and let f be a function. Suppose f is an elementary translation in A from s_1 into s_2.

Then

(i) f is a function from (the sorts of A)(s_1) into (the sorts of A)(s_2),

(ii) (the sorts of A)(s_1) $\neq \emptyset$, and

(iii) (the sorts of A)(s_2) $\neq \emptyset$.

(12) Let S be a non empty non void many sorted signature, and let s_1, s_2 be sort symbols of S, and let A be an algebra over S, and let f be a function. If f is an elementary translation in A from s_1 into s_2, then $\langle s_1, s_2\rangle \in \mathrm{TranslRel}(S)$.

(13) Let S be a non empty non void many sorted signature, and let s_1, s_2 be sort symbols of S, and let A be a non-empty algebra over S. If $\langle s_1, s_2\rangle \in \mathrm{TranslRel}(S)$, then there exists function which is an elementary translation in A from s_1 into s_2.

(14) Let S be a non empty non void many sorted signature, and let A be a feasible algebra over S, and let s_1, s_2 be sort symbols of S. Suppose TranslRel(S) reduces s_1 to s_2. Let q be a reduction sequence w.r.t. TranslRel(S) and let p be a function yielding finite sequence. Suppose that

(i) $\operatorname{len} q = \operatorname{len} p + 1$,

(ii) $s_1 = q(1)$,

(iii) $s_2 = q(\operatorname{len} q)$, and

(iv) for every natural number i and for every function f and for all sort symbols s_1, s_2 of S such that $i \in \operatorname{dom} p$ and $f = p(i)$ and $s_1 = q(i)$ and $s_2 = q(i+1)$ holds f is an elementary translation in A from s_1 into s_2.

Then

(v) $\mathrm{compose}_{(\text{the sorts of } A)(s_1)}\, p$ is a function from (the sorts of A)(s_1) into (the sorts of A)(s_2), and

(vi) if $p \neq \emptyset$, then (the sorts of A)(s_1) $\neq \emptyset$ and (the sorts of A)(s_2) $\neq \emptyset$.

Let S be a non empty non void many sorted signature, let A be a non-empty algebra over S, and let s_1, s_2 be sort symbols of S. Let us assume that TranslRel(S) reduces s_1 to s_2. A function from (the sorts of A)(s_1) into (the sorts of A)(s_2) is called a translation in A from s_1 into s_2 if it satisfies the condition (Def. 6).

(Def. 6) There exists a reduction sequence q w.r.t. TranslRel(S) and there exists a function yielding finite sequence p such that

(i) it $= \mathrm{compose}_{(\text{the sorts of } A)(s_1)}\, p$,

(ii) $\operatorname{len} q = \operatorname{len} p + 1$,

(iii) $s_1 = q(1)$,

(iv) $s_2 = q(\operatorname{len} q)$, and

(v) for every natural number i and for every function f and for all sort symbols s_1, s_2 of S such that $i \in \operatorname{dom} p$ and $f = p(i)$ and $s_1 = q(i)$ and $s_2 = q(i+1)$ holds f is an elementary translation in A from s_1 into s_2.

We now state the proposition

(15) Let S be a non empty non void many sorted signature, and let A be a non-empty algebra over S, and let s_1, s_2 be sort symbols of S. Sup-

pose TranslRel(S) reduces s_1 to s_2. Let q be a reduction sequence w.r.t. TranslRel(S) and let p be a function yielding finite sequence. Suppose that

(i) $\operatorname{len} q = \operatorname{len} p + 1$,

(ii) $s_1 = q(1)$,

(iii) $s_2 = q(\operatorname{len} q)$, and

(iv) for every natural number i and for every function f and for all sort symbols s_1, s_2 of S such that $i \in \operatorname{dom} p$ and $f = p(i)$ and $s_1 = q(i)$ and $s_2 = q(i+1)$ holds f is an elementary translation in A from s_1 into s_2.

Then $\mathrm{compose}_{(\text{the sorts of } A)(s_1)}\, p$ is a translation in A from s_1 into s_2.

In the sequel A is a non-empty algebra over S.

The following propositions are true:

(16) For every sort symbol s of S holds $\mathrm{id}_{(\text{the sorts of } A)(s)}$ is a translation in A from s into s

(17) Let s_1, s_2 be sort symbols of S and let f be a function. Suppose f is an elementary translation in A from s_1 into s_2. Then TranslRel(S) reduces s_1 to s_2 and f is a translation in A from s_1 into s_2.

(18) Let s_1, s_2, s_3 be sort symbols of S. Suppose TranslRel(S) reduces s_1 to s_2 and TranslRel(S) reduces s_2 to s_3. Let t_1 be a translation in A from s_1 into s_2 and let t_2 be a translation in A from s_2 into s_3. Then $t_2 \cdot t_1$ is a translation in A from s_1 into s_3.

(19) Let s_1, s_2, s_3 be sort symbols of S. Suppose TranslRel(S) reduces s_1 to s_2. Let t be a translation in A from s_1 into s_2 and let f be a function. Suppose f is an elementary translation in A from s_2 into s_3. Then $f \cdot t$ is a translation in A from s_1 into s_3.

(20) Let s_1, s_2, s_3 be sort symbols of S. Suppose TranslRel(S) reduces s_2 to s_3. Let f be a function. Suppose f is an elementary translation in A from s_1 into s_2. Let t be a translation in A from s_2 into s_3. Then $t \cdot f$ is a translation in A from s_1 into s_3

The scheme *TranslationInd* concerns a non empty non void many sorted signature $\mathcal{A}$, a non-empty algebra $\mathcal{B}$ over $\mathcal{A}$, and a ternary predicate $\mathcal{P}$, and states that:

Let s_1, s_2 be sort symbols of $\mathcal{A}$. Suppose TranslRel($\mathcal{A}$) reduces s_1 to s_2. Let t be a translation in $\mathcal{B}$ from s_1 into s_2. Then $\mathcal{P}[t, s_1, s_2]$

provided the parameters meet the following requirements:

- For every sort symbol s of $\mathcal{A}$ holds $\mathcal{P}[\mathrm{id}_{(\text{the sorts of } \mathcal{B})(s)}, s, s]$,
- Let s_1, s_2, s_3 be sort symbols of $\mathcal{A}$. Suppose TranslRel($\mathcal{A}$) reduces s_1 to s_2. Let t be a translation in $\mathcal{B}$ from s_1 into s_2. Suppose $\mathcal{P}[t, s_1, s_2]$. Let f be a function. If f is an elementary translation in $\mathcal{B}$ from s_2 into s_3, then $\mathcal{P}[f \cdot t, s_1, s_3]$.

The following propositions are true:

(21) Let A_1, A_2 be non-empty algebras over S and let h be a many sorted function from A_1 into A_2. Suppose h is a homomorphism of A_1 into A_2

Let o be an operation symbol of S and let i be a natural number. Suppose $i \in \operatorname{dom} \operatorname{Arity}(o)$. Let a be an element of $\operatorname{Args}(o, A_1)$. Then $h(\text{the result sort of } o) \cdot o_i^{A_1}(a, -) = o_i^{A_2}(h\#a, -) \cdot h(\pi_i \operatorname{Arity}(o))$.

(22) Let h be an endomorphism of A, and let o be an operation symbol of S, and let i be a natural number. Suppose $i \in \operatorname{dom} \operatorname{Arity}(o)$. Let a be an element of $\operatorname{Args}(o, A)$. Then $h(\text{the result sort of } o) \cdot o_i^{A}(a, -) = o_i^{A}(h\#a, -) \cdot h(\pi_i \operatorname{Arity}(o))$.

(23) Let A_1, A_2 be non-empty algebras over S and let h be a many sorted function from A_1 into A_2. Suppose h is a homomorphism of A_1 into A_2 Let s_1, s_2 be sort symbols of S and let t be a function. Suppose t is an elementary translation in A_1 from s_1 into s_2. Then there exists a function T from (the sorts of A_2)(s_1) into (the sorts of A_2)(s_2) such that T is an elementary translation in A_2 from s_1 into s_2 and $T \cdot h(s_1) = h(s_2) \cdot t$.

(24) Let h be an endomorphism of A, and let s_1, s_2 be sort symbols of S, and let t be a function. Suppose t is an elementary translation in A from s_1 into s_2. Then there exists a function T from (the sorts of A)(s_1) into (the sorts of A)(s_2) such that T is an elementary translation in A from s_1 into s_2 and $T \cdot h(s_1) = h(s_2) \cdot t$.

(25) Let A_1, A_2 be non-empty algebras over S and let h be a many sorted function from A_1 into A_2. Suppose h is a homomorphism of A_1 into A_2 Let s_1, s_2 be sort symbols of S. Suppose TranslRel(S) reduces s_1 to s_2. Let t be a translation in A_1 from s_1 into s_2. Then there exists a translation T in A_2 from s_1 into s_2 such that $T \cdot h(s_1) = h(s_2) \cdot t$.

(26) Let h be an endomorphism of A and let s_1, s_2 be sort symbols of S. Suppose TranslRel(S) reduces s_1 to s_2. Let t be a translation in A from s_1 into s_2. Then there exists a translation T in A from s_1 into s_2 such that $T \cdot h(s_1) = h(s_2) \cdot t$.

2. Compatibility, invariantness, and stability

Let S be a non empty non void many sorted signature, let A be an algebra over S, and let R be a many sorted relation of A. We say that R is compatible if and only if the condition (Def. 7) is satisfied.

(Def. 7) Let o be an operation symbol of S and let a, b be functions. Suppose $a \in \operatorname{Args}(o, A)$ and $b \in \operatorname{Args}(o, A)$ and for every natural number n such that $n \in \operatorname{dom} \operatorname{Arity}(o)$ holds $\langle a(n), b(n) \rangle \in R(\pi_n \operatorname{Arity}(o))$. Then $\langle (\operatorname{Den}(o, A))(a), (\operatorname{Den}(o, A))(b) \rangle \in R(\text{the result sort of } o)$.

We say that R is invariant if and only if the condition (Def. 8) is satisfied.

(Def. 8) Let s_1, s_2 be sort symbols of S and let t be a function. Suppose t is an elementary translation in A from s_1 into s_2. Let a, b be sets. If $\langle a, b \rangle \in R(s_1)$, then $\langle t(a), t(b) \rangle \in R(s_2)$.

We say that R is stable if and only if the condition (Def. 9) is satisfied.

(Def. 9) Let h be an endomorphism of A, and let s be a sort symbol of S, and let a, b be sets. If $\langle a, b\rangle \in R(s)$, then $\langle h(s)(a), h(s)(b)\rangle \in R(s)$.

The following propositions are true:

(27) Let R be an equivalence many sorted relation of A. Then R is compatible if and only if R is a congruence of A.

(28) Let R be a many sorted relation of A. Then R is invariant if and only if for all sort symbols s_1, s_2 of S such that $\mathrm{TranslRel}(S)$ reduces s_1 to s_2 and for every translation f in A from s_1 into s_2 and for all sets a, b such that $\langle a, b\rangle \in R(s_1)$ holds $\langle f(a), f(b)\rangle \in R(s_2)$.

Let S be a non empty non void many sorted signature and let A be a non-empty algebra over S. Note that every equivalence many sorted relation of A which is invariant is also compatible and every equivalence many sorted relation of A which is compatible is also invariant.

Let X be a non empty set. Note that $\triangle_X$ is non empty.

Now we present two schemes. The scheme *MSRExistence* deals with a non empty set $\mathcal{A}$, a non-empty many sorted set $\mathcal{B}$ indexed by $\mathcal{A}$, and a ternary predicate $\mathcal{P}$, and states that:

There exists a many sorted relation R of $\mathcal{B}$ such that for every element i of $\mathcal{A}$ and for all elements a, b of $\mathcal{B}(i)$ holds $\langle a, b\rangle \in R(i)$ if and only if $\mathcal{P}[i, a, b]$

for all values of the parameters.

The scheme *MSRLambdaU* deals with a set $\mathcal{A}$, a many sorted set $\mathcal{B}$ indexed by $\mathcal{A}$, and a unary functor $\mathcal{F}$ yielding a set, and states that:

(i) There exists a many sorted relation R of $\mathcal{B}$ such that for every set i such that $i \in \mathcal{A}$ holds $R(i) = \mathcal{F}(i)$, and

(ii) for all many sorted relations R_1, R_2 of $\mathcal{B}$ such that for every set i such that $i \in \mathcal{A}$ holds $R_1(i) = \mathcal{F}(i)$ and for every set i such that $i \in \mathcal{A}$ holds $R_2(i) = \mathcal{F}(i)$ holds $R_1 = R_2$

provided the parameters meet the following requirement:

- For every set i such that $i \in \mathcal{A}$ holds $\mathcal{F}(i)$ is a relation between $\mathcal{B}(i)$ and $\mathcal{B}(i)$.

Let I be a set and let A be a many sorted set indexed by I. The functor $\triangle_A^I$ yielding a many sorted relation of A is defined by:

(Def. 10) For every set i such that $i \in I$ holds $(\triangle_A^I)(i) = \triangle_{A(i)}$.

Let S be a non empty non void many sorted signature and let A be a non-empty algebra over S. One can verify that every many sorted relation of A which is equivalence is also non-empty.

Let S be a non empty non void many sorted signature and let A be a non-empty algebra over S. Observe that there exists a many sorted relation of A which is invariant stable and equivalence.

3. Invariant, stable, and invariant stable closure

In the sequel S will denote a non empty non void many sorted signature, A will denote a non-empty algebra over S, and R will denote a many sorted relation of the sorts of A.

The scheme *MSRelCl* concerns a non empty non void many sorted signature $\mathcal{A}$, a non-empty algebra $\mathcal{B}$ over $\mathcal{A}$, many sorted relations $\mathcal{Q}$, $\mathcal{D}$ of $\mathcal{B}$, a unary predicate $\mathcal{Q}$, and a ternary predicate $\mathcal{P}$, and states that:

$\mathcal{Q}[\mathcal{D}]$ and $\mathcal{Q} \subseteq \mathcal{D}$ and for every many sorted relation P of $\mathcal{B}$ such that $\mathcal{Q}[P]$ and $\mathcal{Q} \subseteq P$ holds $\mathcal{D} \subseteq P$

provided the following requirements are met:

- Let R be a many sorted relation of $\mathcal{B}$. Then $\mathcal{Q}[R]$ if and only if for all sort symbols s_1, s_2 of $\mathcal{A}$ and for every function f from (the sorts of $\mathcal{B}$)(s_1) into (the sorts of $\mathcal{B}$)(s_2) such that $\mathcal{P}[f, s_1, s_2]$ and for all sets a, b such that $\langle a, b\rangle \in R(s_1)$ holds $\langle f(a), f(b)\rangle \in R(s_2)$,
- Let s_1, s_2, s_3 be sort symbols of $\mathcal{A}$, and let f_1 be a function from (the sorts of $\mathcal{B}$)(s_1) into (the sorts of $\mathcal{B}$)(s_2), and let f_2 be a function from (the sorts of $\mathcal{B}$)(s_2) into (the sorts of $\mathcal{B}$)(s_3). If $\mathcal{P}[f_1, s_1, s_2]$ and $\mathcal{P}[f_2, s_2, s_3]$, then $\mathcal{P}[f_2 \cdot f_1, s_1, s_3]$,
- For every sort symbol s of $\mathcal{A}$ holds $\mathcal{P}[\mathrm{id}_{(\text{the sorts of } \mathcal{B})(s)}, s, s]$,
- Let s be a sort symbol of $\mathcal{A}$ and let a, b be element of $\mathcal{B}$, s. Then $\langle a, b\rangle \in \mathcal{D}(s)$ if and only if there exists a sort symbol s' of $\mathcal{A}$ and there exists a function f from (the sorts of $\mathcal{B}$)(s') into (the sorts of $\mathcal{B}$)(s) and there exist element x, y of $\mathcal{B}$, s' such that $\mathcal{P}[f, s', s]$ and $\langle x, y\rangle \in \mathcal{Q}(s')$ and $a = f(x)$ and $b = f(y)$.

Let S be a non empty non void many sorted signature, let A be a non-empty algebra over S, and let R be a many sorted relation of the sorts of A. The functor $\mathrm{InvCl}(R)$ is an invariant many sorted relation of A and is defined as follows:

(Def. 11) $R \subseteq \mathrm{InvCl}(R)$ and for every invariant many sorted relation Q of A such that $R \subseteq Q$ holds $\mathrm{InvCl}(R) \subseteq Q$.

The following propositions are true:

(29) Let R be a many sorted relation of the sorts of A, and let s be a sort symbol of S, and let a, b be element of A, s. Then $\langle a, b\rangle \in (\mathrm{InvCl}(R))(s)$ if and only if there exists a sort symbol s' of S and there exist element x, y of A, s' and there exists a translation t in A from s' into s such that $\mathrm{TranslRel}(S)$ reduces s' to s and $\langle x, y\rangle \in R(s')$ and $a = t(x)$ and $b = t(y)$.

(30) For every stable many sorted relation R of A holds $\mathrm{InvCl}(R)$ is stable.

Let S be a non empty non void many sorted signature, let A be a non-empty algebra over S, and let R be a many sorted relation of the sorts of A. The functor $\mathrm{StabCl}(R)$ is a stable many sorted relation of A and is defined by:

(Def. 12) $R \subseteq \mathrm{StabCl}(R)$ and for every stable many sorted relation Q of A such that $R \subseteq Q$ holds $\mathrm{StabCl}(R) \subseteq Q$.

We now state two propositions:

(31) Let R be a many sorted relation of the sorts of A, and let s be a sort symbol of S, and let a, b be element of A, s. Then $\langle a, b\rangle \in (\mathrm{StabCl}(R))(s)$ if and only if there exist element x, y of A, s and there exists an endomorphism h of A such that $\langle x, y\rangle \in R(s)$ and $a = h(s)(x)$ and $b = h(s)(y)$.

(32) InvCl(StabCl(R)) is stable.

Let S be a non empty non void many sorted signature, let A be a non-empty algebra over S, and let R be a many sorted relation of the sorts of A. The functor TRS(R) is an invariant stable many sorted relation of A and is defined by:

(Def. 13) $R \subseteq \mathrm{TRS}(R)$ and for every invariant stable many sorted relation Q of A such that $R \subseteq Q$ holds $\mathrm{TRS}(R) \subseteq Q$.

Let S be a non empty non void many sorted signature, let A be a non-empty algebra over S, and let R be a non-empty many sorted relation of A. One can check the following observations:

* InvCl(R) is non-empty,
* StabCl(R) is non-empty, and
* TRS(R) is non-empty.

We now state several propositions:

(33) For every invariant many sorted relation R of A holds $\mathrm{InvCl}(R) = R$.

(34) For every stable many sorted relation R of A holds $\mathrm{StabCl}(R) = R$.

(35) For every invariant stable many sorted relation R of A holds $\mathrm{TRS}(R) = R$.

(36) $\mathrm{StabCl}(R) \subseteq \mathrm{TRS}(R)$ and $\mathrm{InvCl}(R) \subseteq \mathrm{TRS}(R)$ and $\mathrm{StabCl}(\mathrm{InvCl}(R)) \subseteq \mathrm{TRS}(R)$.

(37) $\mathrm{InvCl}(\mathrm{StabCl}(R)) = \mathrm{TRS}(R)$.

(38) Let R be a many sorted relation of the sorts of A, and let s be a sort symbol of S, and let a, b be element of A, s. Then $\langle a, b\rangle \in (\mathrm{TRS}(R))(s)$ if and only if there exists a sort symbol s' of S such that TranslRel(S) reduces s' to s and there exist element l, r of A, s' and there exists an endomorphism h of A and there exists a translation t in A from s' into s such that $\langle l, r\rangle \in R(s')$ and $a = t(h(s')(l))$ and $b = t(h(s')(r))$.

4. Equational theory

One can prove the following propositions:

(39) Let A be a set and let R, E be binary relations on A. Suppose that for all sets a, b such that $a \in A$ and $b \in A$ holds $\langle a, b\rangle \in E$ iff a and b are convertible w.r.t. R. Then E is equivalence relation-like.

(40) Let A be a set, and let R be a binary relation on A, and let E be an equivalence relation of A. Suppose $R \subseteq E$. Let a, b be sets. If $a \in A$ and $b \in A$ and a and b are convertible w.r.t. R, then $\langle a, b \rangle \in E$.

(41) Let A be a non empty set, and let R be a binary relation on A, and let a, b be elements of A. Then $\langle a, b \rangle \in \mathrm{EqCl}(R)$ if and only if a and b are convertible w.r.t. R.

(42) Let S be a non empty set, and let A be a non-empty many sorted set indexed by S and let R be a many sorted relation of A, and let s be an element of S, and let a, b be elements of $A(s)$. Then $\langle a, b \rangle \in (\mathrm{EqCl}(R))(s)$ if and only if a and b are convertible w.r.t. $R(s)$.

Let S be a non empty non void many sorted signature and let A be a non-empty algebra over S. An equational theory of A is a stable invariant equivalence many sorted relation of A. Let R be a many sorted relation of A. The functor $\mathrm{EqCl}(R, A)$ yielding an equivalence many sorted relation of A is defined as follows:

(Def. 14) $\mathrm{EqCl}(R, A) = \mathrm{EqCl}(R)$.

We now state four propositions:

(43) For every many sorted relation R of A holds $R \subseteq \mathrm{EqCl}(R, A)$.

(44) Let R be a many sorted relation of A and let E be an equivalence many sorted relation of A. If $R \subseteq E$, then $\mathrm{EqCl}(R, A) \subseteq E$.

(45) Let R be a stable many sorted relation of A, and let s be a sort symbol of S, and let a, b be element of A, s. Suppose a and b are convertible w.r.t. $R(s)$. Let h be an endomorphism of A. Then $h(s)(a)$ and $h(s)(b)$ are convertible w.r.t. $R(s)$.

(46) For every stable many sorted relation R of A holds $\mathrm{EqCl}(R, A)$ is stable.

Let us consider S, A and let R be a stable many sorted relation of A. Note that $\mathrm{EqCl}(R, A)$ is stable.

We now state two propositions:

(47) Let R be an invariant many sorted relation of A, and let s_1, s_2 be sort symbols of S, and let a, b be element of A, s_1. Suppose a and b are convertible w.r.t. $R(s_1)$. Let t be a function. Suppose t is an elementary translation in A from s_1 into s_2. Then $t(a)$ and $t(b)$ are convertible w.r.t. $R(s_2)$.

(48) For every invariant many sorted relation R of A holds $\mathrm{EqCl}(R, A)$ is invariant.

Let us consider S, A and let R be an invariant many sorted relation of A. One can check that $\mathrm{EqCl}(R, A)$ is invariant.

Next we state three propositions:

(49) Let S be a non empty set, and let A be a non-empty many sorted set indexed by S, and let R, E be many sorted relations of A. Suppose that for every element s of S and for all elements a, b of $A(s)$ holds $\langle a, b \rangle \in E(s)$ iff a and b are convertible w.r.t. $R(s)$. Then E is equivalence.

(50) Let R, E be many sorted relations of A. Suppose that for every sort symbol s of S and for all element a, b of A, s holds $\langle a, b\rangle \in E(s)$ iff a and b are convertible w.r.t. $(\mathrm{TRS}(R))(s)$. Then E is an equational theory of A.

(51) Let S be a non empty set, and let A be a non-empty many sorted set indexed by S and let R be a many sorted relation of A, and let E be an equivalence many sorted relation of A. Suppose $R \subseteq E$. Let s be an element of S and let a, b be elements of $A(s)$. If a and b are convertible w.r.t. $R(s)$, then $\langle a, b\rangle \in E(s)$.

Let S be a non empty non void many sorted signature, let A be a non-empty algebra over S, and let R be a many sorted relation of the sorts of A. The functor $\mathrm{EqTh}(R)$ is an equational theory of A and is defined by:

(Def. 15) $R \subseteq \mathrm{EqTh}(R)$ and for every equational theory Q of A such that $R \subseteq Q$ holds $\mathrm{EqTh}(R) \subseteq Q$.

Next we state three propositions:

(52) For every many sorted relation R of A holds $\mathrm{EqCl}(R, A) \subseteq \mathrm{EqTh}(R)$ and $\mathrm{InvCl}(R) \subseteq \mathrm{EqTh}(R)$ and $\mathrm{StabCl}(R) \subseteq \mathrm{EqTh}(R)$ and $\mathrm{TRS}(R) \subseteq \mathrm{EqTh}(R)$.

(53) Let R be a many sorted relation of A, and let s be a sort symbol of S, and let a, b be element of A, s. Then $\langle a, b\rangle \in (\mathrm{EqTh}(R))(s)$ if and only if a and b are convertible w.r.t. $(\mathrm{TRS}(R))(s)$.

(54) For every many sorted relation R of A holds $\mathrm{EqTh}(R) = \mathrm{EqCl}(\mathrm{TRS}(R), A)$.

References

[1] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[2] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[3] Grzegorz Bancerek. Reduction relations. *Formalized Mathematics*, 5(**4**):469–478, 1996.
[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. 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. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.
[9] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[10] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[11] Krzysztof Hryniewiecki. Relations of tolerance. *Formalized Mathematics*, 2(**1**):105–109, 1991.
[12] Małgorzata Korolkiewicz. Homomorphisms of many sorted algebras. *Formalized Mathematics*, 5(**1**):61–65, 1996.
[13] Małgorzata Korolkiewicz. Many sorted quotient algebra. *Formalized Mathematics*, 5(**1**):79–84, 1996.

[14] Jarosław Kotowicz, Beata Madras, and Małgorzata Korolkiewicz. Basic notation of universal algebra. *Formalized Mathematics*, 3(**2**):251–253, 1992.
[15] Beata Madras. Product of family of universal algebras. *Formalized Mathematics*, 4(**1**):103–108, 1993.
[16] Robert Milewski. Lattice of congruences in many sorted algebra. *Formalized Mathematics*, 5(**4**):479–483, 1996.
[17] Konrad Raczkowski and Paweł Sadowski. Equivalence relations and classes of abstraction. *Formalized Mathematics*, 1(**3**):441–444, 1990.
[18] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[19] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[20] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[21] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[22] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[23] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[24] Wolfgang Wechler. *Universal Algebra for Computer Scientists*. Volume 25 of *EATCS Monographs on TCS*, Springer–Verlag, 1992.
[25] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[26] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received February 9, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

More on the Lattice of Many Sorted Equivalence Relations

Robert Milewski
Warsaw University
Białystok

MML Identifier: MSUALG_7.

The notation and terminology used here are introduced in the following papers: [26], [28], [7], [2], [10], [27], [29], [30], [23], [5], [6], [21], [20], [4], [25], [31], [1], [8], [9], [17], [11], [24], [3], [15], [16], [18], [22], [19], [12], [14], and [13].

1. Lattice of Many Sorted Equivalence Relations is Complete

For simplicity we adopt the following convention: I will be a non empty set, M will be a many sorted set indexed by I, x will be arbitrary, and r_1, r_2 will be real numbers.

We now state several propositions:

(1) For every set X holds $x \in$ the carrier of EqRelLatt(X) iff x is an equivalence relation of X.

(2) id_M is an equivalence relation of M.

(3) $[\![M, M]\!]$ is an equivalence relation of M.

(4) $\perp_{\mathrm{EqRelLatt}(M)} = \mathrm{id}_M$.

(5) $\top_{\mathrm{EqRelLatt}(M)} = [\![M, M]\!]$.

Let us consider I, M. Note that EqRelLatt(M) is bounded.

One can prove the following propositions:

(6) Every subset of the carrier of EqRelLatt(M) is a family of many sorted subsets of $[\![M, M]\!]$.

(7) Let a, b be elements of the carrier of EqRelLatt(M) and let A, B be equivalence relations of M. If $a = A$ and $b = B$, then $a \sqsubseteq b$ iff $A \subseteq B$.

ISSN 1426–2630

(8) Let X be a subset of the carrier of EqRelLatt(M) and let X_1 be a family of many sorted subsets of $[\![M, M]\!]$. Suppose $X_1 = X$. Let a, b be equivalence relations of M. If $a = \bigcap |{:}X_1{:}|$ and $b \in X$, then $a \subseteq b$.

(9) Let X be a subset of the carrier of EqRelLatt(M) and let X_1 be a family of many sorted subsets of $[\![M, M]\!]$. If $X_1 = X$ and X is non empty, then $\bigcap |{:}X_1{:}|$ is an equivalence relation of M.

Let L be a non empty lattice structure. Let us observe that L is complete if and only if the condition (Def. 1) is satisfied.

(Def. 1) Let X be a subset of the carrier of L. Then there exists an element a of the carrier of L such that $X \sqsubseteq a$ and for every element b of the carrier of L such that $X \sqsubseteq b$ holds $a \sqsubseteq b$.

Next we state the proposition

(10) EqRelLatt(M) is complete.

Let us consider I, M. Observe that EqRelLatt(M) is complete.

We now state the proposition

(11) Let X be a subset of the carrier of EqRelLatt(M) and let X_1 be a family of many sorted subsets of $[\![M, M]\!]$. Suppose $X_1 = X$ and X is non empty. Let a, b be equivalence relations of M. If $a = \bigcap |{:}X_1{:}|$ and $b = \bigsqcap_{\mathrm{EqRelLatt}(M)} X$, then $a = b$.

2. Sublattices inheriting SUP's and INF's

Let L be a lattice and let I_1 be a sublattice of L. We say that I_1 is $\bigsqcap$-inheriting if and only if:

(Def. 2) For every subset X of the carrier of I_1 holds $\bigsqcap_L X \in$ the carrier of I_1.

We say that I_1 is $\bigsqcup$-inheriting if and only if:

(Def. 3) For every subset X of the carrier of I_1 holds $\bigsqcup_L X \in$ the carrier of I_1.

The following propositions are true:

(12) Let L be a lattice, and let L' be a sublattice of L, and let a, b be elements of the carrier of L, and let a', b' be elements of the carrier of L'. If $a = a'$ and $b = b'$, then $a \sqcup b = a' \sqcup b'$ and $a \sqcap b = a' \sqcap b'$.

(13) Let L be a lattice, and let L' be a sublattice of L, and let X be a subset of the carrier of L', and let a be an element of the carrier of L, and let a' be an element of the carrier of L'. If $a = a'$, then $a \sqsubseteq X$ iff $a' \sqsubseteq X$.

(14) Let L be a lattice, and let L' be a sublattice of L, and let X be a subset of the carrier of L', and let a be an element of the carrier of L, and let a' be an element of the carrier of L'. If $a = a'$, then $X \sqsubseteq a$ iff $X \sqsubseteq a'$.

(15) Let L be a complete lattice and let L' be a sublattice of L. If L' is $\bigsqcap$-inheriting, then L' is complete.

(16) Let L be a complete lattice and let L' be a sublattice of L. If L' is $\bigsqcup$-inheriting, then L' is complete.

Let L be a complete lattice. Note that there exists a sublattice of L which is complete.

Let L be a complete lattice. One can verify that every sublattice of L which is $\bigsqcap$-inheriting is also complete and every sublattice of L which is $\bigsqcup$-inheriting is also complete.

Next we state four propositions:

(17) Let L be a complete lattice and let L' be a sublattice of L. Suppose L' is $\bigsqcap$-inheriting. Let A' be a subset of the carrier of L'. Then $\bigsqcap_L A' = \bigsqcap_{L'} A'$.

(18) Let L be a complete lattice and let L' be a sublattice of L. Suppose L' is $\bigsqcup$-inheriting. Let A' be a subset of the carrier of L'. Then $\bigsqcup_L A' = \bigsqcup_{L'} A'$.

(19) Let L be a complete lattice and let L' be a sublattice of L. Suppose L' is $\bigsqcap$-inheriting. Let A be a subset of the carrier of L and let A' be a subset of the carrier of L'. If $A = A'$, then $\bigsqcap A = \bigsqcap A'$.

(20) Let L be a complete lattice and let L' be a sublattice of L. Suppose L' is $\bigsqcup$-inheriting. Let A be a subset of the carrier of L and let A' be a subset of the carrier of L'. If $A = A'$, then $\bigsqcup A = \bigsqcup A'$.

3. Segment of Real Numbers as a Complete Lattice

Let us consider r_1, r_2. Let us assume that $r_1 \leq r_2$. The functor $\mathrm{RealSubLatt}(r_1, r_2)$ yields a strict lattice and is defined by the conditions (Def. 4).

(Def. 4) (i) The carrier of $\mathrm{RealSubLatt}(r_1, r_2) = [r_1, r_2]$,

(ii) the join operation of $\mathrm{RealSubLatt}(r_1, r_2) = \max_{\mathbb{R}} \restriction ([\!: [r_1, r_2], [r_1, r_2] :\!]$ **qua** set), and

(iii) the meet operation of $\mathrm{RealSubLatt}(r_1, r_2) = \min_{\mathbb{R}} \restriction ([\!: [r_1, r_2], [r_1, r_2] :\!]$ **qua** set).

One can prove the following propositions:

(21) For all r_1, r_2 such that $r_1 \leq r_2$ holds $\mathrm{RealSubLatt}(r_1, r_2)$ is complete.

(22) There exists sublattice of $\mathrm{RealSubLatt}(0, 1)$ which is $\bigsqcup$-inheriting and non $\bigsqcap$-inheriting.

(23) There exists a complete lattice L such that there exists sublattice of L which is $\bigsqcup$-inheriting and non $\bigsqcap$-inheriting.

(24) There exists sublattice of $\mathrm{RealSubLatt}(0, 1)$ which is $\bigsqcap$-inheriting and non $\bigsqcup$-inheriting.

(25) There exists a complete lattice L such that there exists sublattice of L which is $\bigsqcap$-inheriting and non $\bigsqcup$-inheriting.

References

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

[2] Józef Białas. Infimum and supremum of the set of real numbers. Measure theory. *Formalized Mathematics*, 2(**1**):163–171, 1991.

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

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

[9] Marek Chmur. The lattice of real numbers. The lattice of real functions. *Formalized Mathematics*, 1(**4**):681–684, 1990.

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

[11] Andrzej Iwaniuk. On the lattice of subspaces of a vector space. *Formalized Mathematics*, 5(**3**):305–308, 1996.

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

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

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

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

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

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

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

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

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

[21] Konrad Raczkowski and Paweł Sadowski. Topological properties of subsets in real numbers. *Formalized Mathematics*, 1(**4**):777–780, 1990.

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

[23] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.

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

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

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

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

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

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

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

[31] Stanisław Żukowski. Introduction to lattice theory. *Formalized Mathematics*, 1(**1**):215–222, 1990.

Received February 9, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

Modifying Addresses of Instructions of $\mathbf{SCM}_{\mathrm{FSA}}$

Andrzej Trybulec
Warsaw University
Białystok

Yatsuka Nakamura
Shinshu University
Nagano

MML Identifier: SCMFSA_4.

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

1. Preliminaries

Let N be a non empty set with non empty elements and let S be an AMI over N. One can check that every finite partial state of S is finite.

Let N be a non empty set with non empty elements and let S be an AMI over N. One can verify that there exists a finite partial state of S which is programmed.

Next we state the proposition

(1) Let N be a non empty set with non empty elements, and 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 N be a non empty set with non empty elements, let S be a definite AMI over N, and let I, J be programmed finite partial states of S. Then $I{+}{\cdot}J$ is a programmed finite partial state of S.

Next we state the proposition

(2) Let N be a non empty set with non empty elements, and let S be a definite AMI over N, and let f be a function from the instructions of S into the instructions of S, and let s be a programmed finite partial state of S. Then $\operatorname{dom}(f \cdot s) = \operatorname{dom} s$.

ISSN 1426–2630

2. Incrementing and decrementing the instruction locations

In the sequel i, k, l, m, n, p will denote natural numbers.

Let l_1 be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$ and let k be a natural number. The functor $l_1 + k$ yielding an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$ is defined by:

(Def. 1) There exists a natural number m such that $l_1 = \mathrm{insloc}(m)$ and $l_1 + k = \mathrm{insloc}(m + k)$.

The functor $l_1 -' k$ yields an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$ and is defined by:

(Def. 2) There exists a natural number m such that $l_1 = \mathrm{insloc}(m)$ and $l_1 -' k = \mathrm{insloc}(m -' k)$.

We now state two propositions:

(3) For every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ and for all m, n holds $(l + m) + n = l + (m + n)$.

(4) For every instruction-location l_1 of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every natural number k holds $(l_1 + k) -' k = l_1$.

In the sequel L will be an instruction-location of $\mathbf{SCM}$ and I will be an instruction of $\mathbf{SCM}$.

The following three propositions are true:

(5) For every instruction-location l of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every L such that $L = l$ holds $l + k = L + k$.

(6) For all instructions-locations l_2, l_3 of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every natural number k holds $\mathrm{Start\text{-}At}(l_2 + k) = \mathrm{Start\text{-}At}(l_3 + k)$ iff $\mathrm{Start\text{-}At}(l_2) = \mathrm{Start\text{-}At}(l_3)$.

(7) For all instructions-locations l_2, l_3 of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every natural number k such that $\mathrm{Start\text{-}At}(l_2) = \mathrm{Start\text{-}At}(l_3)$ holds $\mathrm{Start\text{-}At}(l_2 -' k) = \mathrm{Start\text{-}At}(l_3 -' k)$.

3. Incrementing addresses

Let i be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ and let k be a natural number. The functor $\mathrm{IncAddr}(i, k)$ yielding an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$ is defined as follows:

(Def. 3) (i) There exists an instruction I of $\mathbf{SCM}$ such that $I = i$ and $\mathrm{IncAddr}(i, k) = \mathrm{IncAddr}(I, k)$ if $\mathrm{InsCode}(i) \in \{6, 7, 8\}$,

(ii) $\mathrm{IncAddr}(i, k) = i$, otherwise.

We now state a number of propositions:

(8) For every natural number k holds $\mathrm{IncAddr}(\mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}, k) = \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

(9) For every natural number k and for all integer locations a, b holds $\mathrm{IncAddr}(a{:=}b, k) = a{:=}b$.

(10) For every natural number k and for all integer locations a, b holds $\mathrm{IncAddr}(\mathrm{AddTo}(a,b),k) = \mathrm{AddTo}(a,b)$.

(11) For every natural number k and for all integer locations a, b holds $\mathrm{IncAddr}(\mathrm{SubFrom}(a,b),k) = \mathrm{SubFrom}(a,b)$.

(12) For every natural number k and for all integer locations a, b holds $\mathrm{IncAddr}(\mathrm{MultBy}(a,b),k) = \mathrm{MultBy}(a,b)$.

(13) For every natural number k and for all integer locations a, b holds $\mathrm{IncAddr}(\mathrm{Divide}(a,b),k) = \mathrm{Divide}(a,b)$.

(14) For every natural number k and for every instruction-location l_1 of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{IncAddr}(\mathrm{goto}\ l_1,k) = \mathrm{goto}\ (l_1+k)$.

(15) Let k be a natural number, and let l_1 be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$, and let a be an integer location. Then $\mathrm{IncAddr}(\mathbf{if}\ a = 0\ \mathbf{goto}\ l_1,k) = \mathbf{if}\ a=0\ \mathbf{goto}\ l_1+k$.

(16) Let k be a natural number, and let l_1 be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$, and let a be an integer location. Then $\mathrm{IncAddr}(\mathbf{if}\ a > 0\ \mathbf{goto}\ l_1,k) = \mathbf{if}\ a>0\ \mathbf{goto}\ l_1+k$.

(17) Let k be a natural number, and let a, b be integer locations, and let f be a finite sequence location. Then $\mathrm{IncAddr}(b{:=}f_a,k) = b{:=}f_a$.

(18) Let k be a natural number, and let a, b be integer locations, and let f be a finite sequence location. Then $\mathrm{IncAddr}(f_a{:=}b,k) = f_a{:=}b$.

(19) Let k be a natural number, and let a be an integer location, and let f be a finite sequence location. Then $\mathrm{IncAddr}(a{:=}\mathrm{len}f,k) = a{:=}\mathrm{len}f$.

(20) Let k be a natural number, and let a be an integer location, and let f be a finite sequence location. Then $\mathrm{IncAddr}(f{:=}\langle\underbrace{0,\ldots,0}_{a}\rangle,k) = f{:=}\langle\underbrace{0,\ldots,0}_{a}\rangle$.

(21) For every instruction i of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every I such that $i = I$ holds $\mathrm{IncAddr}(i,k) = \mathrm{IncAddr}(I,k)$.

(22) For every instruction I of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every natural number k holds $\mathrm{InsCode}(\mathrm{IncAddr}(I,k)) = \mathrm{InsCode}(I)$.

Let I_1 be a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. We say that I_1 is initial if and only if:

(Def. 4) For all m, n such that $\mathrm{insloc}(n) \in \mathrm{dom}\, I_1$ and $m < n$ holds $\mathrm{insloc}(m) \in \mathrm{dom}\, I_1$.

The finite partial state $\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}}$ of $\mathbf{SCM}_{\mathrm{FSA}}$ is defined as follows:

(Def. 5) $\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}} = \mathrm{insloc}(0) \mapsto \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}$.

Let us note that $\mathrm{Stop}_{\mathrm{SCM}_{\mathrm{FSA}}}$ is non empty initial and programmed.

One can verify that there exists a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ which is initial programmed and non empty.

Let f be a function and let g be a finite function. Note that $f \cdot g$ is finite.

Let N be a non empty set with non empty elements, let S be a definite AMI over N, let s be a programmed finite partial state of S, and let f be a function from the instructions of S into the instructions of S. Then $f \cdot s$ is a programmed finite partial state of S.

In the sequel i will denote an instruction of $\mathbf{SCM}_{\rm FSA}$.

The following proposition is true

(23) $\mathrm{IncAddr}(\mathrm{IncAddr}(i, m), n) = \mathrm{IncAddr}(i, m + n)$.

4. Incremeting addresses in a finite partial state

Let p be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ and let k be a natural number. The functor $\mathrm{IncAddr}(p, k)$ yielding a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ is defined by:

(Def. 6) $\mathrm{dom}\,\mathrm{IncAddr}(p, k) = \mathrm{dom}\,p$ and for every m such that $\mathrm{insloc}(m) \in \mathrm{dom}\,p$ holds $(\mathrm{IncAddr}(p, k))(\mathrm{insloc}(m)) = \mathrm{IncAddr}(\pi_{\mathrm{insloc}(m)}p, k)$.

The following propositions are true:

(24) Let p be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$, and let k be a natural number, and let l be an instruction-location of $\mathbf{SCM}_{\rm FSA}$. If $l \in \mathrm{dom}\,p$, then $(\mathrm{IncAddr}(p, k))(l) = \mathrm{IncAddr}(\pi_l p, k)$.

(25) For all programmed finite partial states I, J of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{IncAddr}(I {+\cdot} J, n) = \mathrm{IncAddr}(I, n) {+\cdot} \mathrm{IncAddr}(J, n)$.

(26) Let f be a function from the instructions of $\mathbf{SCM}_{\rm FSA}$ into the instructions of $\mathbf{SCM}_{\rm FSA}$. Suppose $f = \mathrm{id}_{(\text{the instructions of } \mathbf{SCM}_{\rm FSA})} {+\cdot} (\mathbf{halt}_{\mathbf{SCM}_{\rm FSA}} \dot{\longmapsto} i)$. Let s be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$. Then $\mathrm{IncAddr}(f \cdot s, n) = (\mathrm{id}_{(\text{the instructions of } \mathbf{SCM}_{\rm FSA})} {+\cdot} (\mathbf{halt}_{\mathbf{SCM}_{\rm FSA}} \dot{\longmapsto} \mathrm{IncAddr}(i, n))) \cdot \mathrm{IncAddr}(s, n)$.

(27) For every programmed finite partial state I of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{IncAddr}(\mathrm{IncAddr}(I, m), n) = \mathrm{IncAddr}(I, m + n)$.

(28) For every state s of $\mathbf{SCM}_{\rm FSA}$ holds $\mathrm{Exec}(\mathrm{IncAddr}(\mathrm{CurInstr}(s), k), s {+\cdot} \text{Start-At}(\mathbf{IC}_s + k)) = \mathrm{Following}(s) {+\cdot} \text{Start-At}(\mathbf{IC}_{\mathrm{Following}(s)} + k)$.

(29) Let I_2 be an instruction of $\mathbf{SCM}_{\rm FSA}$, and let s be a state of $\mathbf{SCM}_{\rm FSA}$, and let p be a finite partial state of $\mathbf{SCM}_{\rm FSA}$, and let i, j, k be natural numbers. If $\mathbf{IC}_s = \mathrm{insloc}(j + k)$, then $\mathrm{Exec}(I_2, s {+\cdot} \text{Start-At}(\mathbf{IC}_s -' k)) = \mathrm{Exec}(\mathrm{IncAddr}(I_2, k), s) {+\cdot} \text{Start-At}(\mathbf{IC}_{\mathrm{Exec}(\mathrm{IncAddr}(I_2, k), s)} -' k)$.

5. Shifting the finite partial state

Let p be a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ and let k be a natural number. The functor $\mathrm{Shift}(p, k)$ yields a programmed finite partial state of $\mathbf{SCM}_{\rm FSA}$ and is defined as follows:

(Def. 7) $\mathrm{dom}\,\mathrm{Shift}(p,k) = \{\mathrm{insloc}(m+k) : \mathrm{insloc}(m) \in \mathrm{dom}\,p\}$ and for every m such that $\mathrm{insloc}(m) \in \mathrm{dom}\,p$ holds $(\mathrm{Shift}(p,k))(\mathrm{insloc}(m+k)) = p(\mathrm{insloc}(m))$.

The following propositions are true:

(30) Let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$, and let k be a natural number, and let p be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. If $l \in \mathrm{dom}\,p$, then $(\mathrm{Shift}(p,k))(l+k) = p(l)$.

(31) Let p be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let k be a natural number. Then $\mathrm{dom}\,\mathrm{Shift}(p,k) = \{i_1 + k : i_1$ ranges over instructions-locations of $\mathbf{SCM}_{\mathrm{FSA}}$, $i_1 \in \mathrm{dom}\,p\}$.

(32) For every programmed finite partial state I of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{Shift}(\mathrm{Shift}(I,m),n) = \mathrm{Shift}(I,m+n)$.

(33) Let s be a programmed finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let f be a function from the instructions of $\mathbf{SCM}_{\mathrm{FSA}}$ into the instructions of $\mathbf{SCM}_{\mathrm{FSA}}$, and given n. Then $\mathrm{Shift}(f \cdot s, n) = f \cdot \mathrm{Shift}(s,n)$.

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

(35) For all natural numbers i, j and for every programmed finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{Shift}(\mathrm{IncAddr}(p,i),j) = \mathrm{IncAddr}(\mathrm{Shift}(p,j),i)$.

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

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

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

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

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

[8] 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.

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

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

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

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

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

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

[15] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[16] Yasushi Tanaka. Relocatability. *Formalized Mathematics*, 5(**1**):103–108, 1996.
[17] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[18] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[19] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Formalized Mathematics*, 4(**1**):51–56, 1993.
[20] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\rm FSA}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[21] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[22] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[23] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[24] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received February 14, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

The Correspondence Between Monotonic Many Sorted Signatures and Well-Founded Graphs. Part I [1]

Czesław Byliński
Warsaw University
Białystok

Piotr Rudnicki
University of Alberta
Edmonton

Summary. We prove a number of auxiliary facts about graphs, mainly about vertex sequences of chains and oriented chains. Then we define a graph to be *well-founded* if for each vertex in the graph the length of oriented chains ending at the vertex is bounded. A *well-founded* graph does not have directed cycles or infinite descending chains. In the second part of the article we prove some auxiliary facts about free algebras and locally-finite algebras.

MML Identifier: `MSSCYC_1`.

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

1. Some properties of graphs

The following proposition is true

(1) For every finite function f such that for every set x such that $x \in \operatorname{dom} f$ holds $f(x)$ is finite holds $\prod f$ is finite.

In the sequel G will denote a graph and m, n will denote natural numbers.

Let G be a graph. Let us note that the chain of G can be characterized by the following (equivalent) condition:

(Def. 1) It is a finite sequence of elements of the edges of G and there exists finite sequence of elements of the vertices of G which is vertex sequence of it.

[1]This work was partially supported by NSERC Grant OGP9207.

ISSN 1426–2630

One can prove the following proposition

(2) For all finite sequences p, q such that $1 \leq n$ and $n \leq \operatorname{len} p$ holds $\langle p(1), \ldots, p(n) \rangle = \langle (p \frown q)(1), \ldots, (p \frown q)(n) \rangle$.

Let G be a graph and let I_1 be a chain of G. We introduce I_1 is directed as a synonym of I_1 is oriented.

Let G be a graph and let I_1 be a chain of G. We say that I_1 is cyclic if and only if:

(Def. 2) There exists a finite sequence p of elements of the vertices of G such that p is vertex sequence of I_1 and $p(1) = p(\operatorname{len} p)$.

Let I_1 be a graph. We say that I_1 is empty if and only if:

(Def. 3) The edges of I_1 is empty.

One can verify that there exists a graph which is empty.

Next we state the proposition

(3) For every graph G holds rng (the source of G) $\cup$ rng (the target of G) $\subseteq$ the vertices of G.

Let us observe that there exists a graph which is finite simple connected non empty and strict.

Let G be a non empty graph. Note that the edges of G is non empty.

We now state two propositions:

(4) Let e be arbitrary. Suppose $e \in$ the edges of G. Let s, t be elements of the vertices of G. Suppose $s =$ (the source of G)(e) and $t =$ (the target of G)(e). Then $\langle s, t \rangle$ is vertex sequence of $\langle e \rangle$.

(5) For arbitrary e such that $e \in$ the edges of G holds $\langle e \rangle$ is a directed chain of G.

In the sequel G is a non empty graph.

Let us consider G. Observe that there exists a chain of G which is directed non empty and path-like.

The following propositions are true:

(6) Let c be a chain of G and let p be a finite sequence of elements of the vertices of G. If c is cyclic and p is vertex sequence of c, then $p(1) = p(\operatorname{len} p)$.

(7) Let G be a graph and let e be arbitrary. Suppose $e \in$ the edges of G. Let f_1 be a directed chain of G. If $f_1 = \langle e \rangle$, then vertex-seq$(f_1) = \langle$(the source of G)(e), (the target of G)$(e)\rangle$.

(8) For every finite sequence f holds $\operatorname{len}\langle f(m), \ldots, f(n) \rangle \leq \operatorname{len} f$.

(9) For every directed chain c of G such that $1 \leq m$ and $m \leq n$ and $n \leq \operatorname{len} c$ holds $\langle c(m), \ldots, c(n) \rangle$ is a directed chain of G.

(10) For every non empty directed chain o_1 of G holds len vertex-seq$(o_1) = \operatorname{len} o_1 + 1$.

Let us consider G and let o_1 be a directed non empty chain of G. Observe that vertex-seq(o_1) is non empty.

One can prove the following propositions:

(11) Let o_1 be a directed non empty chain of G and given n. Suppose $1 \leq n$ and $n \leq \operatorname{len} o_1$. Then $(\text{vertex-seq}(o_1))(n) = (\text{the source of } G)(o_1(n))$ and $(\text{vertex-seq}(o_1))(n+1) = (\text{the target of } G)(o_1(n))$.

(12) For every non empty finite sequence f such that $1 \leq m$ and $m \leq n$ and $n \leq \operatorname{len} f$ holds $\langle f(m), \ldots, f(n) \rangle$ is non empty.

(13) For all directed chains c, c_1 of G such that $1 \leq m$ and $m \leq n$ and $n \leq \operatorname{len} c$ and $c_1 = \langle c(m), \ldots, c(n) \rangle$ holds $\text{vertex-seq}(c_1) = \langle (\text{vertex-seq}(c))(m), \ldots, (\text{vertex-seq}(c))(n+1) \rangle$.

(14) For every directed non empty chain o_1 of G holds $(\text{vertex-seq}(o_1))(\operatorname{len} o_1 + 1) = (\text{the target of } G)(o_1(\operatorname{len} o_1))$.

(15) For all directed non empty chains c_1, c_2 of G holds $(\text{vertex-seq}(c_1))(\operatorname{len} c_1 + 1) = (\text{vertex-seq}(c_2))(1)$ iff $c_1 \frown c_2$ is a directed non empty chain of G.

(16) For all directed non empty chains c, c_1, c_2 of G such that $c = c_1 \frown c_2$ holds $(\text{vertex-seq}(c))(1) = (\text{vertex-seq}(c_1))(1)$ and $(\text{vertex-seq}(c))(\operatorname{len} c + 1) = (\text{vertex-seq}(c_2))(\operatorname{len} c_2 + 1)$.

(17) For every directed non empty chain o_1 of G such that o_1 is cyclic holds $(\text{vertex-seq}(o_1))(1) = (\text{vertex-seq}(o_1))(\operatorname{len} o_1 + 1)$.

(18) Let c be a directed non empty chain of G. Suppose c is cyclic. Given n. Then there exists a directed chain c_3 of G such that $\operatorname{len} c_3 = n$ and $c_3 \frown c$ is a directed non empty chain of G.

Let I_1 be a graph. We say that I_1 is directed cycle-less if and only if:

(Def. 4) For every directed chain d_1 of I_1 such that d_1 is non empty holds d_1 is non cyclic.

We introduce I_1 has directed cycle as an antonym of I_1 is directed cycle-less.

Let us mention that every graph which is empty is also directed cycle-less.

Let I_1 be a graph. We say that I_1 is well-founded if and only if the condition (Def. 5) is satisfied.

(Def. 5) Let v be an element of the vertices of I_1. Then there exists n such that for every directed chain c of I_1 if c is non empty and $(\text{vertex-seq}(c))(\operatorname{len} c + 1) = v$, then $\operatorname{len} c \leq n$.

Let G be an empty graph. Note that every chain of G is empty.

One can check that every graph which is empty is also well-founded.

Let us observe that every graph which is non well-founded is also non empty.

One can check that there exists a graph which is well-founded.

Let us note that every graph which is well-founded is also directed cycle-less.

Let us note that there exists a graph which is non well-founded.

One can verify that there exists a graph which is directed cycle-less.

We now state the proposition

(19) For every decorated tree t and for every node p of t and for every natural number k holds $p \restriction k$ is a node of t.

2. Some properties of many sorted algebras

Next we state two propositions:

(20) 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, and let t be a term of S over X. Suppose t is not root. Then there exists an operation symbol o of S such that $t(\varepsilon) = \langle o, \text{the carrier of } S\rangle$.

(21) Let S be a non void non empty many sorted signature, and let A be an algebra over S, and let G be a generator set of A, and let B be a subset of A. If $G \subseteq B$, then B is a generator set of A.

Let S be a non void non empty many sorted signature and let A be a finitely-generated non-empty algebra over S. Note that there exists a generator set of A which is non-empty and locally-finite.

One can prove the following two propositions:

(22) Let S be a non void non empty many sorted signature, and let A be a non-empty algebra over S, and let X be a non-empty generator set of A. Then there exists many sorted function from $\mathrm{Free}(X)$ into A which is an epimorphism of $\mathrm{Free}(X)$ onto A

(23) Let S be a non void non empty many sorted signature, and let A be a non-empty algebra over S, and let X be a non-empty generator set of A. If A is non locally-finite, then $\mathrm{Free}(X)$ is non locally-finite.

Let S be a non void non empty many sorted signature, let X be a non-empty locally-finite many sorted set indexed by the carrier of S, and let v be a sort symbol of S. One can check that $\mathrm{FreeGenerator}(v, X)$ is finite.

One can prove the following propositions:

(24) Let S be a non void non empty many sorted signature, and let X be a non-empty locally-finite many sorted set indexed by the carrier of S, and let v be a sort symbol of S. Then $\mathrm{FreeGenerator}(v, X)$ is finite.

(25) Let S be a non void non empty many sorted signature, and let A be a non-empty algebra over S, and let o be an operation symbol of S. If (the arity of $S)(o) = \varepsilon$, then $\mathrm{dom}\,\mathrm{Den}(o, A) = \{\varepsilon\}$.

Let I_1 be a non void non empty many sorted signature. We say that I_1 is finitely operated if and only if:

(Def. 6) For every sort symbol s of I_1 holds $\{o : o$ ranges over operation symbols of I_1, the result sort of $o = s\}$ is finite.

Next we state three propositions:

(26) Let S be a non void non empty many sorted signature, and let A be a non-empty algebra over S, and let v be a sort symbol of S. If S is finitely operated, then $\mathrm{Constants}(A, v)$ is finite.

(27) 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, and let v be a sort

symbol of S Then $\{t : t$ ranges over elements of (the sorts of $\mathrm{Free}(X))(v)$, $\mathrm{depth}(t) = 0\} = \mathrm{FreeGenerator}(v, X) \cup \mathrm{Constants}(\mathrm{Free}(X), v)$.

(28) 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, and let v, v_1 be sort symbols of S, and let o be an operation symbol of S, and let t be an element of (the sorts of $\mathrm{Free}(X))(v)$, and let a be an argument sequence of $\mathrm{Sym}(o, X)$, and let k be a natural number, and let a_1 be an element of (the sorts of $\mathrm{Free}(X))(v_1)$. If $t = \langle o$, the carrier of $S\rangle$-tree(a) and $k \in \mathrm{dom}\, a$ and $a_1 = a(k)$, then $\mathrm{depth}(a_1) < \mathrm{depth}(t)$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.
[2] Grzegorz Bancerek. Cartesian product of functions. *Formalized Mathematics*, 2(**4**):547–552, 1991.
[3] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[4] Grzegorz Bancerek. Introduction to trees. *Formalized Mathematics*, 1(**2**):421–427, 1990.
[5] Grzegorz Bancerek. Joining of decorated trees. *Formalized Mathematics*, 4(**1**):77–82, 1993.
[6] Grzegorz Bancerek. König's lemma. *Formalized Mathematics*, 2(**3**):397–402, 1991.
[7] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[8] Grzegorz Bancerek. Sets and functions of trees and joining operations of trees. *Formalized Mathematics*, 3(**2**):195–204, 1992.
[9] Grzegorz Bancerek. Subtrees. *Formalized Mathematics*, 5(**2**):185–190, 1996.
[10] Grzegorz Bancerek. Terms over many sorted universal algebra. *Formalized Mathematics*, 5(**2**):191–198, 1996.
[11] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[12] Grzegorz Bancerek and Piotr Rudnicki. On defining functions on trees. *Formalized Mathematics*, 4(**1**):91–101, 1993.
[13] Ewa Burakowska. Subalgebras of many sorted algebra. Lattice of subalgebras. *Formalized Mathematics*, 5(**1**):47–54, 1996.
[14] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[15] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[16] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.
[17] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[18] Patricia L. Carlson and Grzegorz Bancerek. Context-free grammar - part 1. *Formalized Mathematics*, 2(**5**):683–687, 1991.
[19] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[20] 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.
[21] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[22] Krzysztof Hryniewiecki. Graphs. *Formalized Mathematics*, 2(**3**):365–370, 1991.
[23] Małgorzata Korolkiewicz. Homomorphisms of many sorted algebras. *Formalized Mathematics*, 5(**1**):61–65, 1996.
[24] Yatsuka Nakamura and Piotr Rudnicki. Vertex sequences induced by chains. *Formalized Mathematics*, 5(**3**):297–304, 1996.
[25] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, II. *Formalized Mathematics*, 5(**2**):215–220, 1996.

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

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

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

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

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

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

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

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

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

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

Received February 14, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

Relocability for $\mathbf{SCM}_{\mathrm{FSA}}$

Andrzej Trybulec
Warsaw University
Białystok

Yatsuka Nakamura
Shinshu University
Nagano

MML Identifier: `SCMFSA_5`.

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

1. Relocability

In this paper j, k will denote natural numbers.

Let p be a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let k be a natural number. The functor $\mathrm{Relocated}(p,k)$ yields a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and is defined as follows:

(Def. 1) $\mathrm{Relocated}(p,k) = \mathrm{Start\text{-}At}(\mathbf{IC}_p + k) {+\cdot}\, \mathrm{IncAddr}(\mathrm{Shift}(\mathrm{ProgramPart}(p), k), k) {+\cdot}\, \mathrm{DataPart}(p)$.

We now state a number of propositions:

(1) For every finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every natural number k holds $\mathrm{DataPart}(\mathrm{Relocated}(p,k)) = \mathrm{DataPart}(p)$.

(2) For every finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every natural number k holds $\mathrm{ProgramPart}(\mathrm{Relocated}(p,k)) = \mathrm{IncAddr}(\mathrm{Shift}(\mathrm{ProgramPart}(p), k), k)$.

(3) For every finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ holds $\mathrm{dom}\,\mathrm{ProgramPart}(\mathrm{Relocated}(p,k)) = \{\mathrm{insloc}(j+k) : \mathrm{insloc}(j) \in \mathrm{dom}\,\mathrm{ProgramPart}(p)\}$.

(4) Let p be a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let k be a natural number, and let l be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$. Then $l \in \mathrm{dom}\,p$ if and only if $l+k \in \mathrm{dom}\,\mathrm{Relocated}(p,k)$.

(5) For every finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every natural number k holds $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\,\mathrm{Relocated}(p,k)$.

ISSN 1426–2630

(6) For every finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every natural number k holds $\mathbf{IC}_{\mathrm{Relocated}(p,k)} = \mathbf{IC}_p + k$.

(7) Let p be a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let k be a natural number, and let l_1 be an instruction-location of $\mathbf{SCM}_{\mathrm{FSA}}$, and let I be an instruction of $\mathbf{SCM}_{\mathrm{FSA}}$. If $l_1 \in \mathrm{dom}\,\mathrm{ProgramPart}(p)$ and $I = p(l_1)$, then $\mathrm{IncAddr}(I, k) = (\mathrm{Relocated}(p, k))(l_1 + k)$.

(8) For every finite partial state p of $\mathbf{SCM}_{\mathrm{FSA}}$ and for every natural number k holds $\mathrm{Start\text{-}At}(\mathbf{IC}_p + k) \subseteq \mathrm{Relocated}(p, k)$.

(9) Let s be a data-only finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let p be a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let k be a natural number. If $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\,p$, then $\mathrm{Relocated}(p{+\cdot}s, k) = \mathrm{Relocated}(p, k){+\cdot}s$.

(10) Let k be a natural number, and let p be an autonomic finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let s_1, s_2 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. If $p \subseteq s_1$ and $\mathrm{Relocated}(p, k) \subseteq s_2$, then $p \subseteq s_1{+\cdot}s_2 \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations})$.

2. Main Theorems of Relocability

We now state several propositions:

(11) Let k be a natural number and let p be an autonomic finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\,p$. Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s$. Let i be a natural number. Then $(\mathrm{Computation}(s{+\cdot}\,\mathrm{Relocated}(p, k)))(i) = (\mathrm{Computation}(s))(i){+\cdot}\,\mathrm{Start\text{-}At}(\mathbf{IC}_{(\mathrm{Computation}(s))(i)} + k){+\cdot}\,\mathrm{ProgramPart}(\mathrm{Relocated}(p, k))$.

(12) Let k be a natural number, and let p be an autonomic finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$, and let s_1, s_2, s_3 be states of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\,p$ and $p \subseteq s_1$ and $\mathrm{Relocated}(p, k) \subseteq s_2$ and $s_3 = s_1{+\cdot}s_2 \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations})$. Let i be a natural number. Then $\mathbf{IC}_{(\mathrm{Computation}(s_1))(i)} + k = \mathbf{IC}_{(\mathrm{Computation}(s_2))(i)}$ and $\mathrm{IncAddr}(\mathrm{CurInstr}((\mathrm{Computation}(s_1))(i)), k) = \mathrm{CurInstr}((\mathrm{Computation}(s_2))(i))$ and $(\mathrm{Computation}(s_1))(i) \restriction \mathrm{dom}\,\mathrm{DataPart}(p) = (\mathrm{Computation}(s_2))(i) \restriction \mathrm{dom}\,\mathrm{DataPart}(\mathrm{Relocated}(p, k))$ and $(\mathrm{Computation}(s_3))(i) \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations}) = (\mathrm{Computation}(s_2))(i) \restriction (\mathrm{Int\text{-}Locations} \cup \mathrm{FinSeq\text{-}Locations})$.

(13) Let p be an autonomic finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ and let k be a natural number. If $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\,p$, then p is halting iff $\mathrm{Relocated}(p, k)$ is halting.

(14) Let k be a natural number and let p be an autonomic finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\,p$. Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $\mathrm{Relocated}(p, k) \subseteq$

s. Let i be a natural number. Then $(\mathrm{Computation}(s))(i) = (\mathrm{Computation}(s{+\cdot}p))(i){+\cdot}\,\mathrm{Start\text{-}At}(\mathbf{IC}_{(\mathrm{Computation}(s+\cdot p))(i)} + k){+\cdot}s \restriction \mathrm{dom}\,\mathrm{ProgramPart}(p){+\cdot}\,\mathrm{ProgramPart}(\mathrm{Relocated}(p,k))$.

(15) Let k be a natural number and let p be a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\,p$. Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $p \subseteq s$ and $\mathrm{Relocated}(p,k)$ is autonomic. Let i be a natural number. Then $(\mathrm{Computation}(s))(i) = (\mathrm{Computation}(s{+\cdot}\,\mathrm{Relocated}(p,k)))(i){+\cdot}\,\mathrm{Start\text{-}At}(\mathbf{IC}_{(\mathrm{Computation}(s+\cdot\,\mathrm{Relocated}(p,k)))(i)} -' k){+\cdot}s \restriction \mathrm{dom}\,\mathrm{ProgramPart}(\mathrm{Relocated}(p,k)){+\cdot}\,\mathrm{ProgramPart}(p)$.

(16) Let p be a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\,p$. Let k be a natural number. Then p is autonomic if and only if $\mathrm{Relocated}(p,k)$ is autonomic.

(17) Let p be a halting autonomic finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. If $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\,p$, then for every natural number k holds $\mathrm{DataPart}(\mathrm{Result}(p)) = \mathrm{DataPart}(\mathrm{Result}(\mathrm{Relocated}(p,k)))$.

(18) Let F be a data-only partial function from $\mathrm{FinPartSt}(\mathbf{SCM}_{\mathrm{FSA}})$ to $\mathrm{FinPartSt}(\mathbf{SCM}_{\mathrm{FSA}})$ and let p be a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $\mathbf{IC}_{\mathbf{SCM}_{\mathrm{FSA}}} \in \mathrm{dom}\,p$. Let k be a natural number. Then p computes F if and only if $\mathrm{Relocated}(p,k)$ computes F.

References

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

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

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

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

[5] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 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. The modification of a function by a function and the iteration of the composition of a function. *Formalized Mathematics*, 1(**3**):521–527, 1990.

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

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

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

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

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

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

[16] Yasushi Tanaka. On the decomposition of the states of SCM. *Formalized Mathematics*, 5(**1**):1–8, 1996.
[17] Yasushi Tanaka. Relocatability. *Formalized Mathematics*, 5(**1**):103–108, 1996.
[18] Andrzej Trybulec. Enumerated sets. *Formalized Mathematics*, 1(**1**):25–34, 1990.
[19] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[20] Andrzej Trybulec and Yatsuka Nakamura. Computation in $\mathbf{SCM}_{\mathrm{FSA}}$. *Formalized Mathematics*, 5(**4**):537–542, 1996.
[21] Andrzej Trybulec and Yatsuka Nakamura. Modifying addresses of instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. *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 Piotr Rudnicki. The $\mathbf{SCM}_{\mathrm{FSA}}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[24] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[25] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[26] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[27] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received February 22, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

More on the Lattice of Congruences in Many Sorted Algebra

Robert Milewski
Warsaw University
Białystok

MML Identifier: MSUALG_8.

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

1. More on the Lattice of Equivalence Relations

For simplicity we follow a convention: Y denotes a set, I denotes a non empty set, M denotes a many sorted set indexed by I, x, y are arbitrary, k denotes a natural number, p denotes a finite sequence, S denotes a non void non empty many sorted signature, and A denotes a non-empty algebra over S.

The following proposition is true

(1) For every natural number n and for every finite sequence p holds $1 \leq n$ and $n < \operatorname{len} p$ iff $n \in \operatorname{dom} p$ and $n+1 \in \operatorname{dom} p$.

The scheme *NonUniqSeqEx* concerns a natural number $\mathcal{A}$ and a binary predicate $\mathcal{P}$, and states that:

There exists p such that $\operatorname{dom} p = \operatorname{Seg} \mathcal{A}$ and for every k such that $k \in \operatorname{Seg} \mathcal{A}$ holds $\mathcal{P}[k, p(k)]$

provided the following requirement is met:

- For every k such that $k \in \operatorname{Seg} \mathcal{A}$ there exists x such that $\mathcal{P}[k, x]$.

The following three propositions are true:

(2) Let a, b be elements of the carrier of $\operatorname{EqRelLatt}(Y)$ and let A, B be equivalence relations of Y. If $a = A$ and $b = B$, then $a \sqsubseteq b$ iff $A \subseteq B$.

(3) $\perp_{\operatorname{EqRelLatt}(Y)} = \triangle_Y$.

ISSN 1426–2630

(4) $\top_{\mathrm{EqRelLatt}(Y)} = \nabla_Y$.

Let us consider Y. Note that $\mathrm{EqRelLatt}(Y)$ is bounded.

Next we state the proposition

(5) $\mathrm{EqRelLatt}(Y)$ is complete.

Let us consider Y. One can check that $\mathrm{EqRelLatt}(Y)$ is complete.

The following propositions are true:

(6) For every set Y and for every subset X of the carrier of $\mathrm{EqRelLatt}(Y)$ holds $\bigcup X$ is a binary relation on Y.

(7) For every set Y and for every subset X of the carrier of $\mathrm{EqRelLatt}(Y)$ holds $\bigcup X \subseteq \bigsqcup X$.

(8) Let Y be a set, and let X be a subset of the carrier of $\mathrm{EqRelLatt}(Y)$, and let R be a binary relation on Y. If $R = \bigcup X$, then $\bigsqcup X = \mathrm{EqCl}(R)$.

(9) Let Y be a set, and let X be a subset of the carrier of $\mathrm{EqRelLatt}(Y)$, and let R be a binary relation. If $R = \bigcup X$, then $R = R^{\smile}$.

(10) Let Y be a set and let X be a subset of the carrier of $\mathrm{EqRelLatt}(Y)$. Suppose $x \in Y$ and $y \in Y$. Then $\langle x, y\rangle \in \bigsqcup X$ if and only if there exists a finite sequence f such that $1 \leq \mathrm{len}\, f$ and $x = f(1)$ and $y = f(\mathrm{len}\, f)$ and for every natural number i such that $1 \leq i$ and $i < \mathrm{len}\, f$ holds $\langle f(i), f(i+1)\rangle \in \bigcup X$.

2. Lattice of Congruences in Many Sorted Algebra as Sublattice of Lattice of Many Sorted Equivalence Relations Inherited Sup's and Inf's

The following proposition is true

(11) For every subset B of the carrier of $\mathrm{CongrLatt}(A)$ holds $\bigsqcap_{\mathrm{EqRelLatt}(\text{the sorts of } A)} B$ is a congruence of A.

Let us consider S, A and let E be an element of the carrier of EqRelLatt(the sorts of A). The functor $\mathrm{CongrCl}(E)$ yields a congruence of A and is defined by the condition (Def. 1).

(Def. 1) $\mathrm{CongrCl}(E) = \bigsqcap_{\mathrm{EqRelLatt}(\text{the sorts of } A)}\{x : x$ ranges over elements of the carrier of EqRelLatt(the sorts of A), x is a congruence of $A \ \wedge \ E \sqsubseteq x\}$.

Let us consider S, A and let X be a subset of the carrier of EqRelLatt(the sorts of A). The functor $\mathrm{CongrCl}(X)$ yields a congruence of A and is defined by the condition (Def. 2).

(Def. 2) $\mathrm{CongrCl}(X) = \bigsqcap_{\mathrm{EqRelLatt}(\text{the sorts of } A)}\{x : x$ ranges over elements of the carrier of EqRelLatt(the sorts of A), x is a congruence of $A \ \wedge \ X \sqsubseteq x\}$.

The following propositions are true:

(12) For every element C of the carrier of EqRelLatt(the sorts of A) such that C is a congruence of A holds $\mathrm{CongrCl}(C) = C$.

(13) For every subset X of the carrier of EqRelLatt(the sorts of A) holds CongrCl($\bigsqcup_{\text{EqRelLatt(the sorts of }A)} X$) = CongrCl($X$).

(14) Let B_1, B_2 be subsets of the carrier of CongrLatt(A) and let C_1, C_2 be congruences of A. Suppose $C_1 = \bigsqcup_{\text{EqRelLatt(the sorts of }A)} B_1$ and $C_2 = \bigsqcup_{\text{EqRelLatt(the sorts of }A)} B_2$. Then $C_1 \sqcup C_2 = \bigsqcup_{\text{EqRelLatt(the sorts of }A)}(B_1 \cup B_2)$.

(15) Let X be a subset of the carrier of CongrLatt(A). Then $\bigsqcup_{\text{EqRelLatt(the sorts of }A)} X = \bigsqcup_{\text{EqRelLatt(the sorts of }A)}\{\bigsqcup_{\text{EqRelLatt(the sorts of }A)} X_0 : X_0$ ranges over subsets of the carrier of EqRelLatt(the sorts of A), X_0 is a finite subset of $X\}$.

(16) Let i be an element of I and let e be an equivalence relation of $M(i)$. Then there exists an equivalence relation E of M such that $E(i) = e$ and for every element j of I such that $j \neq i$ holds $E(j) = \nabla_{M(j)}$.

Let I be a non empty set, let M be a many sorted set indexed by I, let i be an element of I, and let X be a subset of the carrier of EqRelLatt(M). Then $\pi_i X$ is a subset of the carrier of EqRelLatt($M(i)$) and it can be characterized by the condition:

(Def. 3) $x \in \pi_i X$ iff there exists an equivalence relation E_1 of M such that $x = E_1(i)$ and $E_1 \in X$.

We introduce EqRelSet(X, i) as a synonym of $\pi_i X$.

Next we state four propositions:

(17) Let i be an element of the carrier of S, and let X be a subset of the carrier of EqRelLatt(the sorts of A), and let B be an equivalence relation of the sorts of A. If $B = \bigsqcup X$, then $B(i) = \bigsqcup_{\text{EqRelLatt((the sorts of }A)(i))} \text{EqRelSet}(X, i)$.

(18) For every subset X of the carrier of CongrLatt(A) holds $\bigsqcup_{\text{EqRelLatt(the sorts of }A)} X$ is a congruence of A.

(19) CongrLatt(A) is $\bigsqcap$-inheriting.

(20) CongrLatt(A) is $\bigsqcup$-inheriting.

Let us consider S, A. Observe that CongrLatt(A) is $\bigsqcap$-inheriting and $\bigsqcup$-inheriting.

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 fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[4] Grzegorz Bancerek. König's theorem. *Formalized Mathematics*, 1(**3**):589–593, 1990.
[5] Grzegorz Bancerek. Reduction relations. *Formalized Mathematics*, 5(**4**):469–478, 1996.
[6] Grzegorz Bancerek. Translations, endomorphisms, and stable equational theories. *Formalized Mathematics*, 5(**4**):553–564, 1996.
[7] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[8] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(**1**):175–180, 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. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[12] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[13] Artur Korniłowicz. Certain facts about families of subsets of many sorted sets. *Formalized Mathematics*, 5(**3**):451–456, 1996.
[14] Artur Korniłowicz. On the closure operator and the closure system of many sorted sets. *Formalized Mathematics*, 5(**4**):543–551, 1996.
[15] Małgorzata Korolkiewicz. Many sorted quotient algebra. *Formalized Mathematics*, 5(**1**):79–84, 1996.
[16] Beata Madras. Products of many sorted algebras. *Formalized Mathematics*, 5(**1**):55–60, 1996.
[17] Robert Milewski. Lattice of congruences in many sorted algebra. *Formalized Mathematics*, 5(**4**):479–483, 1996.
[18] Robert Milewski. More on the lattice of many sorted equivalence relations. *Formalized Mathematics*, 5(**4**):565–569, 1996.
[19] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[20] Konrad Raczkowski and Paweł Sadowski. Equivalence relations and classes of abstraction. *Formalized Mathematics*, 1(**3**):441–444, 1990.
[21] Alexander Yu. Shibakov and Andrzej Trybulec. The Cantor set. *Formalized Mathematics*, 5(**2**):233–236, 1996.
[22] Andrzej Trybulec. Domains and their Cartesian products. *Formalized Mathematics*, 1(**1**):115–122, 1990.
[23] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[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. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[27] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[28] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.
[29] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.
[30] Stanisław Żukowski. Introduction to lattice theory. *Formalized Mathematics*, 1(**1**):215–222, 1990.

Received March 6, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

The Correspondence Between Monotonic Many Sorted Signatures and Well-Founded Graphs. Part II [1]

Czesław Byliński
Warsaw University
Białystok

Piotr Rudnicki
University of Alberta
Edmonton

Summary. The graph induced by a many sorted signature is defined as follows: the vertices are the symbols of sorts, and if a sort s is an argument of an operation with result sort t, then a directed edge $[s,t]$ is in the graph. The key lemma states relationship between the depth of elements of a free many sorted algebra over a signature and the length of directed chains in the graph induced by the signature. Then we prove that a monotonic many sorted signature (every finitely-generated algebra over it is locally-finite) induces a *well-founded* graph. The converse holds with an additional assumption that the signature is finitely operated, i.e. there is only a finite number of operations with the given result sort.

MML Identifier: MSSCYC_2.

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

In this paper n will be a natural number.

Let S be a many sorted signature. The functor InducedEdges(S) yields a set and is defined by the condition (Def. 1).

(Def. 1) Let x be a set. Then $x \in$ InducedEdges(S) if and only if there exist sets o_1, v such that $x = \langle o_1, v \rangle$ and $o_1 \in$ the operation symbols of S and $v \in$ the carrier of S and there exists a natural number n and there exists an element a_1 of (the carrier of S)* such that (the arity of S)$(o_1) = a_1$ and $n \in \operatorname{dom} a_1$ and $a_1(n) = v$.

Next we state the proposition

[1]This work was partially supported by NSERC Grant OGP9207.

ISSN 1426–2630

(1) For every many sorted signature S holds InducedEdges$(S) \subseteq [\!:$the operation symbols of S, the carrier of $S\,:\!]$.

Let S be a many sorted signature. The functor InducedSource(S) yields a function from InducedEdges(S) into the carrier of S and is defined as follows:

(Def. 2) For every set e such that $e \in$ InducedEdges(S) holds (InducedSource(S))$(e) = e_{\mathbf{2}}$.

The functor InducedTarget(S) yielding a function from InducedEdges(S) into the carrier of S is defined by:

(Def. 3) For every set e such that $e \in$ InducedEdges(S) holds (InducedTarget(S))$(e) =$ (the result sort of S)$(e_{\mathbf{1}})$.

Let S be a non empty many sorted signature. The functor InducedGraph(S) yields a graph and is defined by:

(Def. 4) InducedGraph$(S) = \langle$the carrier of S, InducedEdges(S), InducedSource(S), InducedTarget$(S)\rangle$.

One can prove the following propositions:

(2) 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, and let v be a sort symbol of S, and given n. Suppose $1 \leq n$. Then there exists an element t of (the sorts of Free(X))(v) such that depth$(t) = n$ if and only if there exists a directed chain c of InducedGraph(S) such that len $c = n$ and (vertex-seq(c))(len $c + 1) = v$.

(3) For every void non empty many sorted signature S holds S is monotonic iff InducedGraph(S) is well-founded.

(4) For every non void non empty many sorted signature S such that S is monotonic holds InducedGraph(S) is well-founded.

(5) Let S be a non void non empty many sorted signature and let X be a non-empty locally-finite many sorted set indexed by the carrier of S. Suppose S is finitely operated. Let n be a natural number and let v be a sort symbol of S. Then $\{t : t$ ranges over elements of (the sorts of Free(X))(v), depth$(t) \leq n\}$ is finite.

(6) Let S be a non void non empty many sorted signature. If S is finitely operated and InducedGraph(S) is well-founded, then S is monotonic.

References

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

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

[3] Grzegorz Bancerek. Introduction to trees. *Formalized Mathematics*, 1(**2**):421–427, 1990.

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

[5] Grzegorz Bancerek. König's lemma. *Formalized Mathematics*, 2(**3**):397–402, 1991.

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

[7] Grzegorz Bancerek. Sets and functions of trees and joining operations of trees. *Formalized Mathematics*, 3(**2**):195–204, 1992.
[8] Grzegorz Bancerek. Subtrees. *Formalized Mathematics*, 5(**2**):185–190, 1996.
[9] Grzegorz Bancerek. Terms over many sorted universal algebra. *Formalized Mathematics*, 5(**2**):191–198, 1996.
[10] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[11] Ewa Burakowska. Subalgebras of many sorted algebra. Lattice of subalgebras. *Formalized Mathematics*, 5(**1**):47–54, 1996.
[12] Czesław Byliński. Functions and their basic properties. *Formalized Mathematics*, 1(**1**):55–65, 1990.
[13] Czesław Byliński. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[14] Czesław Byliński. Partial functions. *Formalized Mathematics*, 1(**2**):357–367, 1990.
[15] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[16] Czesław Byliński and Piotr Rudnicki. The correspondence between monotonic many sorted signatures and well-founded graphs. Part I. *Formalized Mathematics*, 5(**4**):577–582, 1996.
[17] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[18] 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.
[19] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[20] Krzysztof Hryniewiecki. Graphs. *Formalized Mathematics*, 2(**3**):365–370, 1991.
[21] Jarosław Kotowicz. Functions and finite sequences of real numbers. *Formalized Mathematics*, 3(**2**):275–278, 1992.
[22] Yatsuka Nakamura and Piotr Rudnicki. Vertex sequences induced by chains. *Formalized Mathematics*, 5(**3**):297–304, 1996.
[23] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Introduction to circuits, I. *Formalized Mathematics*, 5(**2**):227–232, 1996.
[24] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, I. *Formalized Mathematics*, 5(**2**):167–172, 1996.
[25] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, II. *Formalized Mathematics*, 5(**2**):215–220, 1996.
[26] Andrzej Nędzusiak. σ-fields and probability. *Formalized Mathematics*, 1(**2**):401–407, 1990.
[27] Beata Perkowska. Free many sorted universal algebra. *Formalized Mathematics*, 5(**1**):67–74, 1996.
[28] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[29] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[30] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[31] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[32] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[33] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[34] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received April 10, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

Functors for Alternative Categories

Andrzej Trybulec
Warsaw University
Białystok

Summary. An attempt to define the concept of a functor covering both cases (covariant and contravariant) resulted in a structure consisting of two fields: the object map and the morphism map, the first one mapping the Cartesian squares of the set of objects rather than the set of objects. We start with an auxiliary notion of *bifunction*, i.e. a function mapping the Cartesian square of a set A into the Cartesian square of a set B. A *bifunction* f is said to be *covariant* if there is a function g from A into B that f is the Cartesian square of g and f is *contravariant* if there is a function g such that $f(o_1, o_2) = \langle g(o_2), g(o_1)\rangle$. The term *transformation*, another auxiliary notion, might be misleading. It is not related to natural transformations. A transformation from a many sorted set A indexed by I into a many sorted set B indexed by J w.r.t. a function f from I into J is a (many sorted) function from A into $B \cdot f$. Eventually, the morphism map of a functor from C_1 into C_2 is a transformation from the arrows of the category C_1 into the composition of the object map of the functor and the arrows of C_2.

Several kinds of functor structures have been defined: one-to-one, faithful, onto, full and id-preserving. We were pressed to split property that the composition be preserved into two: comp-preserving (for covariant functors) and comp-reversing (for contravariant functors). We defined also some operation on functors, e.g. the composition, the inverse functor. In the last section it is defined what is meant that two categories are isomorphic (anti-isomorphic).

MML Identifier: `FUNCTOR0`.

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

1. Preliminaries

The scheme *ValOnPair* concerns a non empty set $\mathcal{A}$, a function $\mathcal{B}$, elements $\mathcal{C}$, $\mathcal{D}$ of $\mathcal{A}$, a binary functor $\mathcal{F}$ yielding arbitrary, and a binary predicate $\mathcal{P}$, and states that:

ISSN 1426–2630

$\mathcal{B}(\mathcal{C},\, \mathcal{D}) = \mathcal{F}(\mathcal{C}, \mathcal{D})$

provided the following conditions are met:

- $\mathcal{B} = \{\langle\langle o,\, o'\rangle,\, \mathcal{F}(o, o')\rangle : o$ ranges over elements of $\mathcal{A}$, o' ranges over elements of $\mathcal{A}$, $\mathcal{P}[o, o']\}$,
- $\mathcal{P}[\mathcal{C}, \mathcal{D}]$.

One can prove the following propositions:

(1) For every set A holds $\emptyset$ is a function from A into $\emptyset$.

(2) For every set A and for every function f from A into $\emptyset$ holds $f = \emptyset$.

(3) For every set I and for every many sorted set M indexed by I holds $M \cdot \mathrm{id}_I = M$.

Let f be an empty function. Note that $\curvearrowleft f$ is empty. Let g be a function. One can verify that $[\!:f,\, g\,:\!]$ is empty and $[\!:g,\, f\,:\!]$ is empty.

The following propositions are true:

(4) For every set A and for every function f holds $f^{\circ}(\mathrm{id}_A) = (\curvearrowleft f)^{\circ}(\mathrm{id}_A)$.

(5) For all sets X, Y and for every function f from X into Y holds f is onto iff $[\!:f,\, f\,:\!]$ is onto.

Let I_1 be a set and let f, g be many sorted functions of I_1. Then $g \circ f$ is a many sorted function of I_1.

Let f be a function yielding function. One can verify that $\curvearrowleft f$ is function yielding.

One can prove the following propositions:

(6) For all sets A, B and for arbitrary a holds $\curvearrowleft([\!:A,\, B\,:\!] \longmapsto a) = [\!:B,\, A\,:\!] \longmapsto a$.

(7) For all functions f, g such that f is one-to-one and g is one-to-one holds $[\!:f,\, g\,:\!]^{-1} = [\!:f^{-1},\, g^{-1}\,:\!]$.

(8) For every function f such that $[\!:f,\, f\,:\!]$ is one-to-one holds f is one-to-one.

(9) For every function f such that f is one-to-one holds $\curvearrowleft f$ is one-to-one.

(10) For all functions f, g such that $\curvearrowleft[\!:f,\, g\,:\!]$ is one-to-one holds $[\!:g,\, f\,:\!]$ is one-to-one.

(11) For all functions f, g such that f is one-to-one and g is one-to-one holds $(\curvearrowleft[\!:f,\, g\,:\!])^{-1} = \curvearrowleft([\!:g,\, f\,:\!]^{-1})$.

(12) For all sets A, B and for every function f from A into B such that f is onto holds $\mathrm{id}_B \subseteq [\!:f,\, f\,:\!]^{\circ}(\mathrm{id}_A)$.

(13) For all function yielding functions F, G and for every function f holds $(G \circ F) \cdot f = (G \cdot f) \circ (F \cdot f)$.

Let A, B, C be sets and let f be a function from $[\!:A,\, B\,:\!]$ into C. Then $\curvearrowleft f$ is a function from $[\!:B,\, A\,:\!]$ into C.

Next we state two propositions:

(14) For all sets A, B, C and for every function f from $[\!:A,\, B\,:\!]$ into C such that $\curvearrowleft f$ is onto holds f is onto.

(15) For every set A and for every non empty set B and for every function f from A into B holds $[\!:f, f:\!]^{\circ}(\mathrm{id}_A) \subseteq \mathrm{id}_B$.

2. Functions Between Cartesian Squares

Let A, B be sets.

(Def. 1) A function from $[\!: A, A :\!]$ into $[\!: B, B :\!]$ is called a bifunction from A into B.

Let A, B be sets and let f be a bifunction from A into B. We say that f is precovariant if and only if:

(Def. 2) There exists a function g from A into B such that $f = [\!: g, g :\!]$.

We say that f is precontravariant if and only if:

(Def. 3) There exists a function g from A into B such that $f = \curvearrowleft[\!: g, g :\!]$.

The following proposition is true

(16) Let A be a set, and let B be a non empty set, and let b be an element of B, and let f be a bifunction from A into B. If $f = [\!: A, A :\!] \longmapsto \langle b, b \rangle$, then f is precovariant and precontravariant.

Let A, B be sets. Note that there exists a bifunction from A into B which is precovariant and precontravariant.

Next we state the proposition

(17) Let A, B be non empty sets and let f be a precovariant precontravariant bifunction from A into B. Then there exists an element b of B such that $f = [\!: A, A :\!] \longmapsto \langle b, b \rangle$.

3. Unary Transformations

Let I_1, I_2 be sets, let f be a function from I_1 into I_2, let A be a many sorted set indexed by I_1, and let B be a many sorted set indexed by I_2. A many sorted set indexed by I_1 is called a f-transformation from A to B if:

(Def. 4) (i) There exists a non empty set I_2' and there exists a many sorted set B' indexed by I_2' and there exists a function f' from I_1 into I_2' such that $f = f'$ and $B = B'$ and it is a many sorted function from A into $B' \cdot f'$ if $I_2 \neq \emptyset$,

(ii) it $= \emptyset_{(I_1)}$, otherwise.

Let I_1 be a set, let I_2 be a non empty set, let f be a function from I_1 into I_2, let A be a many sorted set indexed by I_1, and let B be a many sorted set indexed by I_2. Let us note that the f-transformation from A to B can be characterized by the following (equivalent) condition:

(Def. 5) It is a many sorted function from A into $B \cdot f$.

Let I_1, I_2 be sets, let f be a function from I_1 into I_2, let A be a many sorted set indexed by I_1, and let B be a many sorted set indexed by I_2. Note that every f-transformation from A to B is function yielding.

We now state the proposition

(18) Let I_1 be a set, and let I_2, I_3 be non empty sets, and let f be a function from I_1 into I_2, and let g be a function from I_2 into I_3, and let B be a many sorted set indexed by I_2 and let C be a many sorted set indexed by I_3 and let G be a g-transformation from B to C. Then $G \cdot f$ is a $g \cdot f$-transformation from $B \cdot f$ to C.

Let I_1 be a set, let I_2 be a non empty set, let f be a function from I_1 into I_2, let A be a many sorted set indexed by $[\!: I_1, I_1 :\!]$, let B be a many sorted set indexed by $[\!: I_2, I_2 :\!]$, and let F be a $[\!: f, f :\!]$-transformation from A to B. Then $\curvearrowleft F$ is a $[\!: f, f :\!]$-transformation from $\curvearrowleft A$ to $\curvearrowleft B$.

One can prove the following two propositions:

(19) Let I_1, I_2 be non empty sets, and let A be a many sorted set indexed by I_1 and let B be a many sorted set indexed by I_2 and let o be an element of I_2. Suppose $B(o) \neq \emptyset$. Let m be an element of $B(o)$ and let f be a function from I_1 into I_2. Suppose $f = I_1 \longmapsto o$. Then $\{\langle o', A(o') \longmapsto m\rangle : o'$ ranges over elements of $I_1\}$ is a f-transformation from A to B.

(20) Let I_1 be a set, and let I_2, I_3 be non empty sets, and let f be a function from I_1 into I_2, and let g be a function from I_2 into I_3, and let A be a many sorted set indexed by I_1 and let B be a many sorted set indexed by I_2 and let C be a many sorted set indexed by I_3 and let F be a f-transformation from A to B, and let G be a $g \cdot f$-transformation from $B \cdot f$ to C. Suppose that for arbitrary i_1 such that $i_1 \in I_1$ and $(B \cdot f)(i_1) = \emptyset$ holds $A(i_1) = \emptyset$ or $(C \cdot (g \cdot f))(i_1) = \emptyset$. Then $G \circ (F$ **qua** function yielding function) is a $g \cdot f$-transformation from A to C.

4. Functors

Let C_1, C_2 be 1-sorted structures. We introduce bimap structures from C_1 into C_2 which are systems

$\langle$ an object map $\rangle$,

where the object map is a bifunction from the carrier of C_1 into the carrier of C_2.

Let C_1, C_2 be non empty graphs, let F be a bimap structure from C_1 into C_2, and let o be an object of C_1. The functor $F(o)$ yields an object of C_2 and is defined as follows:

(Def. 6) $F(o) = (\text{the object map of } F)(o, o)_{\mathbf{1}}$.

Let A, B be 1-sorted structures and let F be a bimap structure from A into B. We say that F is one-to-one if and only if:

(Def. 7) The object map of F is one-to-one.

We say that F is onto if and only if:

(Def. 8) The object map of F is onto.

We say that F is reflexive if and only if:

(Def. 9) (The object map of F)$^\circ(\mathrm{id}_{(\text{the carrier of } A)}) \subseteq \mathrm{id}_{(\text{the carrier of } B)}$.

We say that F is coreflexive if and only if:

(Def. 10) $\mathrm{id}_{(\text{the carrier of } B)} \subseteq$ (the object map of F)$^\circ(\mathrm{id}_{(\text{the carrier of } A)})$.

Let A, B be non empty graphs and let F be a bimap structure from A into B. Let us observe that F is reflexive if and only if:

(Def. 11) For every object o of A holds (the object map of F)$(o, o) = \langle F(o), F(o)\rangle$.

We now state the proposition

(21) Let A, B be reflexive non empty graphs and let F be a bimap structure from A into B. Suppose F is coreflexive. Let o be an object of B. Then there exists an object o' of A such that $F(o') = o$.

Let C_1, C_2 be non empty graphs and let F be a bimap structure from C_1 into C_2. We say that F is feasible if and only if:

(Def. 12) For all objects o_1, o_2 of C_1 such that $\langle o_1, o_2\rangle \neq \emptyset$ holds (the arrows of C_2)((the object map of F)$(o_1, o_2)) \neq \emptyset$.

Let C_1, C_2 be graphs. We introduce functor structures from C_1 to C_2 which are extensions of bimap structure from C_1 into C_2 and are systems

⟨ an object map, a morphism map ⟩,

where the object map is a bifunction from the carrier of C_1 into the carrier of C_2 and the morphism map is a the object map-transformation from the arrows of C_1 to the arrows of C_2.

Let C_1, C_2 be 1-sorted structures and let I_4 be a bimap structure from C_1 into C_2. We say that I_4 is precovariant if and only if:

(Def. 13) The object map of I_4 is precovariant.

We say that I_4 is precontravariant if and only if:

(Def. 14) The object map of I_4 is precontravariant.

Let C_1, C_2 be graphs. One can verify that there exists a functor structure from C_1 to C_2 which is precovariant and there exists a functor structure from C_1 to C_2 which is precontravariant.

Let C_1, C_2 be graphs, let F be a functor structure from C_1 to C_2, and let o_1, o_2 be objects of C_1. The functor $\text{Morph-Map}_F(o_1, o_2)$ is defined as follows:

(Def. 15) $\text{Morph-Map}_F(o_1, o_2) =$ (the morphism map of F)(o_1, o_2).

Let C_1, C_2 be graphs, let F be a functor structure from C_1 to C_2, and let o_1, o_2 be objects of C_1. Observe that $\text{Morph-Map}_F(o_1, o_2)$ is relation-like and function-like.

Let C_1, C_2 be non empty graphs, let F be a precovariant functor structure from C_1 to C_2, and let o_1, o_2 be objects of C_1. Then $\text{Morph-Map}_F(o_1, o_2)$ is a function from $\langle o_1, o_2\rangle$ into $\langle F(o_1), F(o_2)\rangle$.

Let C_1, C_2 be non empty graphs, let F be a precovariant functor structure from C_1 to C_2, and let o_1, o_2 be objects of C_1. Let us assume that $\langle o_1, o_2\rangle \neq \emptyset$ and $\langle F(o_1), F(o_2)\rangle \neq \emptyset$. Let m be a morphism from o_1 to o_2. The functor $F(m)$ yielding a morphism from $F(o_1)$ to $F(o_2)$ is defined as follows:

(Def. 16) $F(m) = (\text{Morph-Map}_F(o_1, o_2))(m)$.

Let C_1, C_2 be non empty graphs, let F be a precontravariant functor structure from C_1 to C_2, and let o_1, o_2 be objects of C_1. Then $\text{Morph-Map}_F(o_1, o_2)$ is a function from $\langle o_1, o_2\rangle$ into $\langle F(o_2), F(o_1)\rangle$.

Let C_1, C_2 be non empty graphs, let F be a precontravariant functor structure from C_1 to C_2, and let o_1, o_2 be objects of C_1. Let us assume that $\langle o_1, o_2\rangle \neq \emptyset$ and $\langle F(o_2), F(o_1)\rangle \neq \emptyset$. Let m be a morphism from o_1 to o_2. The functor $F(m)$ yielding a morphism from $F(o_2)$ to $F(o_1)$ is defined as follows:

(Def. 17) $F(m) = (\text{Morph-Map}_F(o_1, o_2))(m)$.

Let C_1, C_2 be non empty graphs and let o be an object of C_2. Let us assume that $\langle o, o\rangle \neq \emptyset$. Let m be a morphism from o to o. The functor $C_1 \longmapsto m$ yields a strict functor structure from C_1 to C_2 and is defined by the conditions (Def. 18).

(Def. 18) (i) The object map of $C_1 \longmapsto m = [\!:$ the carrier of C_1, the carrier of $C_1 :\!] \longmapsto \langle o, o\rangle$, and

(ii) the morphism map of $C_1 \longmapsto m = \{\langle\langle o_1, o_2\rangle, (\langle o_1, o_2\rangle) \longmapsto m\rangle : o_1$ ranges over objects of C_1, o_2 ranges over objects of $C_1\}$.

We now state the proposition

(22) Let C_1, C_2 be non empty graphs and let o_2 be an object of C_2. Suppose $\langle o_2, o_2\rangle \neq \emptyset$. Let m be a morphism from o_2 to o_2 and let o_1 be an object of C_1. Then $(C_1 \longmapsto m)(o_1) = o_2$.

Let C_1 be a non empty graph, let C_2 be a non empty reflexive graph, let o be an object of C_2, and let m be a morphism from o to o. One can verify that $C_1 \longmapsto m$ is precovariant precontravariant and feasible.

Let C_1 be a non empty graph and let C_2 be a non empty reflexive graph. One can check that there exists a functor structure from C_1 to C_2 which is feasible precovariant and precontravariant.

The following proposition is true

(23) Let C_1, C_2 be non empty graphs, and let F be a precovariant functor structure from C_1 to C_2, and let o_1, o_2 be objects of C_1 Then (the object map of $F)(o_1, o_2) = \langle F(o_1), F(o_2)\rangle$.

Let C_1, C_2 be non empty graphs and let F be a precovariant functor structure from C_1 to C_2. Let us observe that F is feasible if and only if:

(Def. 19) For all objects o_1, o_2 of C_1 such that $\langle o_1, o_2\rangle \neq \emptyset$ holds $\langle F(o_1), F(o_2)\rangle \neq \emptyset$.

One can prove the following proposition

(24) Let C_1, C_2 be non empty graphs, and let F be a precontravariant functor structure from C_1 to C_2, and let o_1, o_2 be objects of C_1 Then (the object map of $F)(o_1, o_2) = \langle F(o_2), F(o_1)\rangle$.

Let C_1, C_2 be non empty graphs and let F be a precontravariant functor structure from C_1 to C_2. Let us observe that F is feasible if and only if:

(Def. 20) For all objects o_1, o_2 of C_1 such that $\langle o_1, o_2 \rangle \neq \emptyset$ holds $\langle F(o_2), F(o_1) \rangle \neq \emptyset$.

Let C_1, C_2 be graphs and let F be a functor structure from C_1 to C_2. Observe that the morphism map of F is function yielding.

Let us note that there exists a category structure which is non empty and reflexive.

Let C_1, C_2 be non empty category structures with units and let F be a functor structure from C_1 to C_2. We say that F is id-preserving if and only if:

(Def. 21) For every object o of C_1 holds $(\text{Morph-Map}_F(o, o))(\text{id}_o) = \text{id}_{F(o)}$.

We now state the proposition

(25) Let C_1, C_2 be non empty graphs and let o_2 be an object of C_2. Suppose $\langle o_2, o_2 \rangle \neq \emptyset$. Let m be a morphism from o_2 to o_2, and let o, o' be objects of C_1 and let f be a morphism from o to o'. If $\langle o, o' \rangle \neq \emptyset$, then $(\text{Morph-Map}_{C_1 \longmapsto m}(o, o'))(f) = m$.

One can check that every non empty category structure which has units is reflexive.

Let C_1, C_2 be non empty category structures with units and let o_2 be an object of C_2. Note that $C_1 \longmapsto \text{id}_{(o_2)}$ is id-preserving.

Let C_1 be a non empty graph, let C_2 be a non empty reflexive graph, let o_2 be an object of C_2, and let m be a morphism from o_2 to o_2. Observe that $C_1 \longmapsto m$ is reflexive.

Let C_1 be a non empty graph and let C_2 be a non empty reflexive graph. Observe that there exists a functor structure from C_1 to C_2 which is feasible and reflexive.

Let C_1, C_2 be non empty category structures with units. Note that there exists a functor structure from C_1 to C_2 which is id-preserving feasible reflexive and strict.

Let C_1, C_2 be non empty category structures and let F be a functor structure from C_1 to C_2. We say that F is comp-preserving if and only if the condition (Def. 22) is satisfied.

(Def. 22) Let o_1, o_2, o_3 be objects of C_1 Suppose $\langle o_1, o_2 \rangle \neq \emptyset$ and $\langle o_2, o_3 \rangle \neq \emptyset$. Let f be a morphism from o_1 to o_2 and let g be a morphism from o_2 to o_3. Then there exists a morphism f' from $F(o_1)$ to $F(o_2)$ and there exists a morphism g' from $F(o_2)$ to $F(o_3)$ such that $f' = (\text{Morph-Map}_F(o_1, o_2))(f)$ and $g' = (\text{Morph-Map}_F(o_2, o_3))(g)$ and $(\text{Morph-Map}_F(o_1, o_3))(g \cdot f) = g' \cdot f'$.

Let C_1, C_2 be non empty category structures and let F be a functor structure from C_1 to C_2. We say that F is comp-reversing if and only if the condition (Def. 23) is satisfied.

(Def. 23) Let o_1, o_2, o_3 be objects of C_1 Suppose $\langle o_1, o_2 \rangle \neq \emptyset$ and $\langle o_2, o_3 \rangle \neq \emptyset$. Let f be a morphism from o_1 to o_2 and let g be a morphism from o_2 to o_3.

Then there exists a morphism f' from $F(o_2)$ to $F(o_1)$ and there exists a morphism g' from $F(o_3)$ to $F(o_2)$ such that $f' = (\text{Morph-Map}_F(o_1, o_2))(f)$ and $g' = (\text{Morph-Map}_F(o_2, o_3))(g)$ and $(\text{Morph-Map}_F(o_1, o_3))(g \cdot f) = f' \cdot g'$.

Let C_1 be a non empty transitive category structure, let C_2 be a non empty reflexive category structure, and let F be a precovariant feasible functor structure from C_1 to C_2. Let us observe that F is comp-preserving if and only if the condition (Def. 24) is satisfied.

(Def. 24) Let o_1, o_2, o_3 be objects of C_1 Suppose $\langle o_1, o_2 \rangle \neq \emptyset$ and $\langle o_2, o_3 \rangle \neq \emptyset$. Let f be a morphism from o_1 to o_2 and let g be a morphism from o_2 to o_3. Then $F(g \cdot f) = F(g) \cdot F(f)$.

Let C_1 be a non empty transitive category structure, let C_2 be a non empty reflexive category structure, and let F be a precontravariant feasible functor structure from C_1 to C_2. Let us observe that F is comp-reversing if and only if the condition (Def. 25) is satisfied.

(Def. 25) Let o_1, o_2, o_3 be objects of C_1 Suppose $\langle o_1, o_2 \rangle \neq \emptyset$ and $\langle o_2, o_3 \rangle \neq \emptyset$. Let f be a morphism from o_1 to o_2 and let g be a morphism from o_2 to o_3. Then $F(g \cdot f) = F(f) \cdot F(g)$.

The following two propositions are true:

(26) Let C_1 be a non empty graph, and let C_2 be a non empty reflexive graph, and let o_2 be an object of C_2, and let m be a morphism from o_2 to o_2, and let F be a precovariant feasible functor structure from C_1 to C_2. Suppose $F = C_1 \longmapsto m$. Let o, o' be objects of C_1 and let f be a morphism from o to o'. If $\langle o, o' \rangle \neq \emptyset$, then $F(f) = m$.

(27) Let C_1 be a non empty graph, and let C_2 be a non empty reflexive graph, and let o_2 be an object of C_2, and let m be a morphism from o_2 to o_2, and let o, o' be objects of C_1 and let f be a morphism from o to o'. If $\langle o, o' \rangle \neq \emptyset$, then $(C_1 \longmapsto m)(f) = m$.

Let C_1 be a non empty transitive category structure, let C_2 be a non empty category structure with units, and let o be an object of C_2. Note that $C_1 \longmapsto \text{id}_o$ is comp-preserving and comp-reversing.

Let C_1 be a transitive non empty category structure with units and let C_2 be a non empty category structure with units. A functor structure from C_1 to C_2 is said to be a functor from C_1 to C_2 if:

(Def. 26) It is feasible and id-preserving but it is precovariant and comp-preserving or it is precontravariant and comp-reversing.

Let C_1 be a transitive non empty category structure with units, let C_2 be a non empty category structure with units, and let F be a functor from C_1 to C_2. We say that F is covariant if and only if:

(Def. 27) F is precovariant and comp-preserving.

We say that F is contravariant if and only if:

(Def. 28) F is precontravariant and comp-reversing.

Let A be a category structure and let B be a substructure of A. The functor $\overset{B}{\hookrightarrow}$ yields a strict functor structure from B to A and is defined by the conditions (Def. 29).

(Def. 29) (i) The object map of $\overset{B}{\hookrightarrow} = \mathrm{id}_{[\text{the carrier of } B, \text{ the carrier of } B]}$, and

(ii) the morphism map of $\overset{B}{\hookrightarrow} = \mathrm{id}_{(\text{the arrows of } B)}$.

Let A be a graph. The functor id_A yielding a strict functor structure from A to A is defined by the conditions (Def. 30).

(Def. 30) (i) The object map of $\mathrm{id}_A = \mathrm{id}_{[\text{the carrier of } A, \text{ the carrier of } A]}$, and

(ii) the morphism map of $\mathrm{id}_A = \mathrm{id}_{(\text{the arrows of } A)}$.

Let A be a category structure and let B be a substructure of A. Note that $\overset{B}{\hookrightarrow}$ is precovariant.

One can prove the following propositions:

(28) Let A be a non empty category structure, and let B be a non empty substructure of A, and let o be an object of B. Then $(\overset{B}{\hookrightarrow})(o) = o$.

(29) Let A be a non empty category structure, and let B be a non empty substructure of A, and let o_1, o_2 be objects of B Then $\langle o_1, o_2 \rangle \subseteq \langle (\overset{B}{\hookrightarrow})(o_1), (\overset{B}{\hookrightarrow})(o_2) \rangle$.

Let A be a non empty category structure and let B be a non empty substructure of A. Observe that $\overset{B}{\hookrightarrow}$ is feasible.

Let A, B be graphs and let F be a functor structure from A to B. We say that F is faithful if and only if:

(Def. 31) The morphism map of F is "1-1".

Let A, B be graphs and let F be a functor structure from A to B. We say that F is full if and only if the condition (Def. 32) is satisfied.

(Def. 32) There exists a many sorted set B' indexed by $[$the carrier of A, the carrier of $A]$ and there exists a many sorted function f from the arrows of A into B' such that $B' =$ (the arrows of B) $\cdot$ (the object map of F) and $f =$ the morphism map of F and f is onto.

Let A be a graph, let B be a non empty graph, and let F be a functor structure from A to B. Let us observe that F is full if and only if the condition (Def. 33) is satisfied.

(Def. 33) There exists a many sorted function f from the arrows of A into (the arrows of B) $\cdot$ (the object map of F) such that $f =$ the morphism map of F and f is onto.

Let A, B be graphs and let F be a functor structure from A to B. We say that F is injective if and only if:

(Def. 34) F is one-to-one and faithful.

We say that F is surjective if and only if:

(Def. 35) F is full and onto.

Let A, B be graphs and let F be a functor structure from A to B. We say that F is bijective if and only if:

(Def. 36) F is injective and surjective.

Let A, B be transitive non empty category structures with units. One can check that there exists a functor from A to B which is strict covariant contravariant and feasible.

The following two propositions are true:

(30) For every non empty graph A and for every object o of A holds $\mathrm{id}_A(o) = o$.

(31) Let A be a non empty graph and let o_1, o_2 be objects of A If $\langle o_1, o_2 \rangle \neq \emptyset$, then for every morphism m from o_1 to o_2 holds $(\text{Morph-Map}_{\mathrm{id}_A}(o_1, o_2))(m) = m$.

Let A be a non empty graph. Note that id_A is feasible and precovariant.

Let A be a non empty graph. Note that there exists a functor structure from A to A which is precovariant and feasible.

One can prove the following proposition

(32) Let A be a non empty graph and let o_1, o_2 be objects of A Suppose $\langle o_1, o_2 \rangle \neq \emptyset$. Let F be a precovariant feasible functor structure from A to A. If $F = \mathrm{id}_A$, then for every morphism m from o_1 to o_2 holds $F(m) = m$.

Let A be a transitive non empty category structure with units. One can check that id_A is id-preserving and comp-preserving.

Let A be a transitive non empty category structure with units. Then id_A is a strict covariant functor from A to A.

Let A be a graph. One can verify that id_A is bijective.

5. The Composition of Functors

Let C_1 be a non empty graph, let C_2, C_3 be non empty reflexive graphs, let F be a feasible functor structure from C_1 to C_2, and let G be a functor structure from C_2 to C_3. The functor $G \cdot F$ yielding a strict functor structure from C_1 to C_3 is defined by the conditions (Def. 37).

(Def. 37) (i) The object map of $G \cdot F =$ (the object map of G) $\cdot$ (the object map of F), and

(ii) the morphism map of $G \cdot F =$ ((the morphism map of G) $\cdot$ (the object map of F)) $\circ$ (the morphism map of F).

Let C_1 be a non empty graph, let C_2, C_3 be non empty reflexive graphs, let F be a precovariant feasible functor structure from C_1 to C_2, and let G be a precovariant functor structure from C_2 to C_3. Observe that $G \cdot F$ is precovariant.

Let C_1 be a non empty graph, let C_2, C_3 be non empty reflexive graphs, let F be a precontravariant feasible functor structure from C_1 to C_2, and let G be a precovariant functor structure from C_2 to C_3. Observe that $G \cdot F$ is precontravariant.

Let C_1 be a non empty graph, let C_2, C_3 be non empty reflexive graphs, let F be a precovariant feasible functor structure from C_1 to C_2, and let G be a precontravariant functor structure from C_2 to C_3. Observe that $G \cdot F$ is precontravariant.

Let C_1 be a non empty graph, let C_2, C_3 be non empty reflexive graphs, let F be a precontravariant feasible functor structure from C_1 to C_2, and let G be a precontravariant functor structure from C_2 to C_3. Observe that $G \cdot F$ is precovariant.

Let C_1 be a non empty graph, let C_2, C_3 be non empty reflexive graphs, let F be a feasible functor structure from C_1 to C_2, and let G be a feasible functor structure from C_2 to C_3. Note that $G \cdot F$ is feasible.

The following three propositions are true:

(33) Let C_1 be a non empty graph, and let C_2, C_3, C_4 be non empty reflexive graphs, and let F be a feasible functor structure from C_1 to C_2, and let G be a feasible functor structure from C_2 to C_3, and let H be a functor structure from C_3 to C_4. Then $(H \cdot G) \cdot F = H \cdot (G \cdot F)$.

(34) Let C_1 be a non empty category structure, and let C_2, C_3 be non empty reflexive category structures, and let F be a feasible reflexive functor structure from C_1 to C_2, and let G be a functor structure from C_2 to C_3, and let o be an object of C_1. Then $(G \cdot F)(o) = G(F(o))$.

(35) Let C_1 be a non empty graph, and let C_2, C_3 be non empty reflexive graphs, and let F be a feasible reflexive functor structure from C_1 to C_2, and let G be a functor structure from C_2 to C_3, and let o be an object of C_1. Then $\text{Morph-Map}_{G \cdot F}(o, o) = \text{Morph-Map}_G(F(o), F(o)) \cdot \text{Morph-Map}_F(o, o)$.

Let C_1, C_2, C_3 be non empty category structures with units, let F be an id-preserving feasible reflexive functor structure from C_1 to C_2, and let G be an id-preserving functor structure from C_2 to C_3. Note that $G \cdot F$ is id-preserving.

Let A, C be non empty reflexive category structures, let B be a non empty substructure of A, and let F be a functor structure from A to C. The functor $F \restriction B$ yielding a functor structure from B to C is defined as follows:

(Def. 38) $F \restriction B = F \cdot (\overset{B}{\hookrightarrow})$.

6. The Inverse Functor

Let A, B be non empty graphs and let F be a functor structure from A to B. Let us assume that F is bijective. The functor F^{-1} yielding a strict functor structure from B to A is defined by the conditions (Def. 39).

(Def. 39) (i) The object map of F^{-1} = (the object map of $F)^{-1}$, and

(ii) there exists a many sorted function f from the arrows of A into (the arrows of B) $\cdot$ (the object map of F) such that f = the morphism map of F and the morphism map of $F^{-1} = f^{-1} \cdot$ (the object map of $F)^{-1}$.

One can prove the following propositions:

(36) Let A, B be transitive non empty category structures with units and let F be a feasible functor structure from A to B. If F is bijective, then F^{-1} is bijective and feasible.

(37) Let A, B be transitive non empty category structures with units and let F be a feasible reflexive functor structure from A to B If F is bijective and coreflexive, then F^{-1} is reflexive.

(38) Let A, B be transitive non empty category structures with units and let F be a feasible reflexive id-preserving functor structure from A to B If F is bijective and coreflexive, then F^{-1} is id-preserving.

(39) Let A, B be transitive non empty category structures with units and let F be a feasible functor structure from A to B If F is bijective and precovariant, then F^{-1} is precovariant.

(40) Let A, B be transitive non empty category structures with units and let F be a feasible functor structure from A to B If F is bijective and precontravariant, then F^{-1} is precontravariant.

(41) Let A, B be transitive non empty category structures with units and let F be a feasible reflexive functor structure from A to B Suppose F is bijective coreflexive and precovariant. Let o_1, o_2 be objects of B and let m be a morphism from o_1 to o_2. If $\langle o_1, o_2 \rangle \neq \emptyset$, then $(\text{Morph-Map}_F(F^{-1}(o_1), F^{-1}(o_2)))((\text{Morph-Map}_{F^{-1}}(o_1, o_2))(m)) = m$.

(42) Let A, B be transitive non empty category structures with units and let F be a feasible reflexive functor structure from A to B Suppose F is bijective coreflexive and precontravariant. Let o_1, o_2 be objects of B and let m be a morphism from o_1 to o_2. If $\langle o_1, o_2 \rangle \neq \emptyset$, then $(\text{Morph-Map}_F(F^{-1}(o_2), F^{-1}(o_1)))((\text{Morph-Map}_{F^{-1}}(o_1, o_2))(m)) = m$.

(43) Let A, B be transitive non empty category structures with units and let F be a feasible reflexive functor structure from A to B Suppose F is bijective comp-preserving precovariant and coreflexive. Then F^{-1} is comp-preserving.

(44) Let A, B be transitive non empty category structures with units and let F be a feasible reflexive functor structure from A to B Suppose F is bijective comp-reversing precontravariant and coreflexive. Then F^{-1} is comp-reversing.

Let C_1 be a 1-sorted structure and let C_2 be a non empty 1-sorted structure. One can verify that every bimap structure from C_1 into C_2 which is precovariant is also reflexive.

Let C_1 be a 1-sorted structure and let C_2 be a non empty 1-sorted structure. One can verify that every bimap structure from C_1 into C_2 which is precontravariant is also reflexive.

Next we state two propositions:

(45) Let C_1, C_2 be 1-sorted structures and let M be a bimap structure from C_1 into C_2. If M is precovariant and onto, then M is coreflexive.

(46) Let C_1, C_2 be 1-sorted structures and let M be a bimap structure from C_1 into C_2. If M is precontravariant and onto, then M is coreflexive.

Let C_1 be a transitive non empty category structure with units and let C_2 be a non empty category structure with units. Note that every functor from C_1

to C_2 which is covariant is also reflexive.

Let C_1 be a transitive non empty category structure with units and let C_2 be a non empty category structure with units. One can verify that every functor from C_1 to C_2 which is contravariant is also reflexive.

The following propositions are true:

(47) Let C_1 be a transitive non empty category structure with units, and let C_2 be a non empty category structure with units, and let F be a functor from C_1 to C_2. If F is covariant and onto, then F is coreflexive.

(48) Let C_1 be a transitive non empty category structure with units, and let C_2 be a non empty category structure with units, and let F be a functor from C_1 to C_2. If F is contravariant and onto, then F is coreflexive.

(49) Let A, B be transitive non empty category structures with units and let F be a covariant functor from A to B. Suppose F is bijective. Then there exists a functor G from B to A such that $G = F^{-1}$ and G is bijective and covariant.

(50) Let A, B be transitive non empty category structures with units and let F be a contravariant functor from A to B. Suppose F is bijective. Then there exists a functor G from B to A such that $G = F^{-1}$ and G is bijective and contravariant.

Let A, B be transitive non empty category structures with units. We say that A and B are isomorphic if and only if:

(Def. 40) There exists functor from A to B which is bijective and covariant.

Let us observe that this predicate is reflexive and symmetric. We say that A, B are anti-isomorphic if and only if:

(Def. 41) There exists functor from A to B which is bijective and contravariant.

Let us note that the predicate introduced above is symmetric.

References

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

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

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

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

[5] 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.

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

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

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

[9] Yozo Toda. The formalization of simple graphs. *Formalized Mathematics*, 5(**1**):137–144, 1996.

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

[11] Andrzej Trybulec. Categories without uniqueness of **cod** and **dom**. *Formalized Mathematics*, 5(**2**):259–267, 1996.
[12] Andrzej Trybulec. Examples of category structures. *Formalized Mathematics*, 5(**4**):493–500, 1996.
[13] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[14] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[15] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[16] Andrzej Trybulec. Tuples, projections and Cartesian products. *Formalized Mathematics*, 1(**1**):97–105, 1990.
[17] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[18] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received April 24, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

Basic Properties of Functor Structures

Claus Zinn
University of Erlangen–Nürnberg

Wolfgang Jaksch
University of Erlangen–Nürnberg

Summary. This article presents some theorems about functor structures. We start with some basic lemmata concerning the composition of functor structures. Then, two theorems about the restriction operator are formulated. Later, we show two theorems stating that the properties 'full' and 'faithful' of functor structures which are equivalent to the 'onto' and 'one-to-one' properties of their morphmaps, respectively. Furthermore, we prove some theorems about the inversion of functor structures.

MML Identifier: `FUNCTOR1`.

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

1. Definitions

In this paper X, Y denote sets and Z denotes a non empty set.

Let us mention that there exists a non empty category structure which is transitive and reflexive and has units.

Let A be a non empty reflexive category structure. One can verify that there exists a substructure of A which is non empty and reflexive.

Let C_1, C_2 be non empty reflexive category structures, let F be a feasible functor structure from C_1 to C_2, and let A be a non empty reflexive substructure of C_1. Observe that $F \restriction A$ is feasible.

2. Theorems about sets and functions

We now state four propositions:

ISSN 1426–2630

(1) For every set X holds id_X is onto.

(2) Let A be a non empty set, and let B, C be non empty subsets of A and let D be a non empty subset of B. If $C = D$, then $\overset{C}{\hookrightarrow} = (\overset{B}{\hookrightarrow}) \cdot (\overset{D}{\hookrightarrow})$.

(3) For every function f from X into Y such that f is bijective holds f^{-1} is a function from Y into X.

(4) Let f be a function from X into Y and let g be a function from Y into Z. Suppose f is bijective and g is bijective. Then there exists a function h from X into Z such that $h = g \cdot f$ and h is bijective.

3. Theorems about the composition of functor structures

The following propositions are true:

(5) Let A be a non empty reflexive category structure, and let B be a non empty reflexive substructure of A, and let C be a non empty substructure of A, and let D be a non empty substructure of B. If $C = D$, then $\overset{C}{\hookrightarrow} = (\overset{B}{\hookrightarrow}) \cdot (\overset{D}{\hookrightarrow})$.

(6) Let A, B be non empty category structures and let F be a functor structure from A to B. Suppose F is bijective. Then the object map of F is bijective and the morphism map of F is "1-1".

(7) Let C_1 be a non empty graph, and let C_2, C_3 be non empty reflexive graphs, and let F be a feasible functor structure from C_1 to C_2, and let G be a functor structure from C_2 to C_3. If F is one-to-one and G is one-to-one, then $G \cdot F$ is one-to-one.

(8) Let C_1 be a non empty graph, and let C_2, C_3 be non empty reflexive graphs, and let F be a feasible functor structure from C_1 to C_2, and let G be a functor structure from C_2 to C_3 If F is faithful and G is faithful, then $G \cdot F$ is faithful.

(9) Let C_1 be a non empty graph, and let C_2, C_3 be non empty reflexive graphs, and let F be a feasible functor structure from C_1 to C_2, and let G be a functor structure from C_2 to C_3 If F is onto and G is onto, then $G \cdot F$ is onto.

(10) Let C_1 be a non empty graph, and let C_2, C_3 be non empty reflexive graphs, and let F be a feasible functor structure from C_1 to C_2, and let G be a functor structure from C_2 to C_3 If F is full and G is full, then $G \cdot F$ is full.

(11) Let C_1 be a non empty graph, and let C_2, C_3 be non empty reflexive graphs, and let F be a feasible functor structure from C_1 to C_2, and let G be a functor structure from C_2 to C_3 If F is injective and G is injective, then $G \cdot F$ is injective.

(12) Let C_1 be a non empty graph, and let C_2, C_3 be non empty reflexive graphs, and let F be a feasible functor structure from C_1 to C_2, and let G

be a functor structure from C_2 to C_3 If F is surjective and G is surjective, then $G \cdot F$ is surjective.

(13) Let C_1 be a non empty graph, and let C_2, C_3 be non empty reflexive graphs, and let F be a feasible functor structure from C_1 to C_2, and let G be a functor structure from C_2 to C_3 If F is bijective and G is bijective, then $G \cdot F$ is bijective.

4. Theorems about the restriction and inclusion operator

We now state three propositions:

(14) Let A, I be non empty reflexive category structures, and let B be a non empty reflexive substructure of A, and let C be a non empty substructure of A, and let D be a non empty substructure of B. Suppose $C = D$. Let F be a functor structure from A to I. Then $F \restriction C = F \restriction B \restriction D$.

(15) Let C_1, C_2, C_3 be non empty reflexive category structures, and let F be a feasible functor structure from C_1 to C_2, and let G be a functor structure from C_2 to C_3 and let A be a non empty reflexive substructure of C_1. Then $(G \cdot F) \restriction A = G \cdot (F \restriction A)$.

(17)[1] Let A be a non empty category structure and let B be a non empty substructure of A. Then B is full if and only if $\overset{B}{\hookrightarrow}$ is full.

5. Theorems about 'full' and 'faithful' functor structures

Next we state two propositions:

(18) Let C_1, C_2 be non empty category structures and let F be a precovariant functor structure from C_1 to C_2. Then F is full if and only if for all objects o_1, o_2 of C_1 holds $\text{Morph-Map}_F(o_1, o_2)$ is onto.

(19) Let C_1, C_2 be non empty category structures and let F be a precovariant functor structure from C_1 to C_2. Then F is faithful if and only if for all objects o_1, o_2 of C_1 holds $\text{Morph-Map}_F(o_1, o_2)$ is one-to-one.

6. Theorems about the inversion of functor structures

One can prove the following propositions:

(20) For every transitive non empty category structure A with units holds $(\text{id}_A)^{-1} = \text{id}_A$.

[1]The proposition (16) has been removed.

(21) Let A, B be transitive reflexive non empty category structures with units. Suppose A and B are isomorphic. Let F be a strict feasible functor structure from A to B. Suppose F is bijective. Let G be a strict feasible functor structure from B to A. If $G = F^{-1}$, then $F \cdot G = \mathrm{id}_B$.

(22) Let A, B be transitive reflexive non empty category structures with units. Suppose A and B are isomorphic. Let F be a strict feasible functor structure from A to B. If F is bijective, then $F^{-1} \cdot F = \mathrm{id}_A$.

(23) Let A, B be transitive reflexive non empty category structures with units. Suppose A and B are isomorphic. Let F be a strict feasible functor structure from A to B. If F is bijective, then $(F^{-1})^{-1} = F$.

(24) Let A, B, C be transitive reflexive non empty category structures with units, and let G be a strict feasible functor structure from A to B, and let F be a strict feasible functor structure from B to C, and let G_1 be a strict feasible functor structure from B to A, and let F_1 be a strict feasible functor structure from C to B. Suppose F is bijective and G is bijective and F_1 is bijective and G_1 is bijective and $G_1 = G^{-1}$ and $F_1 = F^{-1}$. Then $(F \cdot G)^{-1} = G_1 \cdot F_1$.

Acknowledgments

This article has been written during the four week internship of the authors in Białystok in order to get familiar with the MIZAR system. We would like to thank Andrzej Trybulec and the members of the MIZAR group for their invitation and their instructive support.

References

[1] Ewa Burakowska. Subalgebras of many sorted algebra. Lattice of subalgebras. *Formalized Mathematics*, 5(**1**):47–54, 1996.
[2] Czesław Byliński. Basic functions and operations on functions. *Formalized Mathematics*, 1(**1**):245–254, 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. Functions from a set to a set. *Formalized Mathematics*, 1(**1**):153–164, 1990.
[6] Czesław Byliński. Some basic properties of sets. *Formalized Mathematics*, 1(**1**):47–53, 1990.
[7] Artur Korniłowicz. On the group of automorphisms of universal algebra & many sorted algebra. *Formalized Mathematics*, 5(**2**):221–226, 1996.
[8] Małgorzata Korolkiewicz. Homomorphisms of many sorted algebras. *Formalized Mathematics*, 5(**1**):61–65, 1996.
[9] Beata Madras. Product of family of universal algebras. *Formalized Mathematics*, 4(**1**):103–108, 1993.
[10] Yozo Toda. The formalization of simple graphs. *Formalized Mathematics*, 5(**1**):137–144, 1996.
[11] Andrzej Trybulec. Categories without uniqueness of **cod** and **dom**. *Formalized Mathematics*, 5(**2**):259–267, 1996.
[12] Andrzej Trybulec. Examples of category structures. *Formalized Mathematics*, 5(**4**):493–500, 1996.

[13] Andrzej Trybulec. Functors for alternative categories. *Formalized Mathematics*, 5(**4**):595–608, 1996.
[14] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[15] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[16] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[17] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[18] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received April 24, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

Some Multi Instructions Defined by Sequence of Instructions of $\mathbf{SCM}_{\rm FSA}$

Noriko Asamoto
Ochanomizu University
Tokyo

MML Identifier: `SCMFSA_7`.

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

In this paper m will be a natural number.

Let us note that every finite partial state of $\mathbf{SCM}_{\rm FSA}$ is finite.

Let p be a finite sequence and let x, y be arbitrary. Note that $p +\cdot (x, y)$ is finite sequence-like.

Let i be an integer. Then $|i|$ is a natural number.

Let D be a set. Note that D^* is non empty.

The following four propositions are true:

(1) For every natural number k holds $|k| = k$.

(2) For all natural numbers a, b, c such that $a \geq c$ and $b \geq c$ and $a -' c = b -' c$ holds $a = b$.

(3) For all natural numbers a, b such that $a \geq b$ holds $a -' b = a - b$.

(4) For all integers a, b such that $a < b$ holds $a \leq b - 1$.

The scheme *CardMono"* concerns a set $\mathcal{A}$, a non empty set $\mathcal{B}$, and a unary functor $\mathcal{F}$ yielding arbitrary, and states that:

$\mathcal{A} \approx \{\mathcal{F}(d) : d$ ranges over elements of $\mathcal{B}$, $d \in \mathcal{A}\}$

provided the parameters satisfy the following conditions:

- $\mathcal{A} \subseteq \mathcal{B}$,
- For all elements d_1, d_2 of $\mathcal{B}$ such that $d_1 \in \mathcal{A}$ and $d_2 \in \mathcal{A}$ and $\mathcal{F}(d_1) = \mathcal{F}(d_2)$ holds $d_1 = d_2$.

One can prove the following propositions:

(5) For all finite sequences p_1, p_2, q such that $p_1 \subseteq q$ and $p_2 \subseteq q$ and $\operatorname{len} p_1 = \operatorname{len} p_2$ holds $p_1 = p_2$.

ISSN 1426–2630

(6) For all finite sequences p, q such that $p \frown q = p$ holds $q = \varepsilon$.

(7) For every finite sequence p and for arbitrary x holds $\mathrm{len}(p \frown \langle x \rangle) = \mathrm{len}\, p + 1$.

(8) For all finite sequences p, q such that $p \subseteq q$ holds $\mathrm{len}\, p \leq \mathrm{len}\, q$.

(9) For all finite sequences p, q and for every natural number i such that $1 \leq i$ and $i \leq \mathrm{len}\, p$ holds $(p \frown q)(i) = p(i)$.

(10) For all finite sequences p, q and for every natural number i such that $1 \leq i$ and $i \leq \mathrm{len}\, q$ holds $(p \frown q)(\mathrm{len}\, p + i) = q(i)$.

(11) For every finite sequence p and for every natural number i holds $i \in \mathrm{dom}\, p$ iff $1 \leq i$ and $i \leq \mathrm{len}\, p$.

(12) For every finite sequence p such that $p \neq \varepsilon$ holds $\mathrm{len}\, p \in \mathrm{dom}\, p$.

(13) For every set D holds $\mathrm{Flat}(\varepsilon_{D^*}) = \varepsilon_D$.

(14) For every set D and for all finite sequences F, G of elements of D^* holds $\mathrm{Flat}(F \frown G) = \mathrm{Flat}(F) \frown \mathrm{Flat}(G)$.

(15) For every set D and for all elements p, q of D^* holds $\mathrm{Flat}(\langle p, q \rangle) = p \frown q$.

(16) For every set D and for all elements p, q, r of D^* holds $\mathrm{Flat}(\langle p, q, r \rangle) = p \frown q \frown r$.

(17) Let D be a non empty set and let p, q be finite sequences of elements of D. If $p \subseteq q$, then there exists a finite sequence p' of elements of D such that $p \frown p' = q$.

(18) Let D be a non empty set, and let p, q be finite sequences of elements of D, and let i be a natural number. If $p \subseteq q$ and $1 \leq i$ and $i \leq \mathrm{len}\, p$, then $q(i) = p(i)$.

(19) For every set D and for all finite sequences F, G of elements of D^* such that $F \subseteq G$ holds $\mathrm{Flat}(F) \subseteq \mathrm{Flat}(G)$.

(20) For every finite sequence p holds $p \restriction \mathrm{Seg}\, 0 = \varepsilon$.

(21) For all finite sequences f, g holds $f \restriction \mathrm{Seg}\, 0 = g \restriction \mathrm{Seg}\, 0$.

(22) For every non empty set D and for every element x of D holds $\langle x \rangle$ is a finite sequence of elements of D.

(23) Let D be a set and let p, q be finite sequences of elements of D. Then $p \frown q$ is a finite sequence of elements of D.

Let f be a finite sequence of elements of the instructions of $\mathbf{SCM}_{\mathrm{FSA}}$. The functor $\mathrm{Load}(f)$ yielding a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ is defined by:

(Def. 1) $\mathrm{dom}\,\mathrm{Load}(f) = \{\mathrm{insloc}(m -' 1) : m \in \mathrm{dom}\, f\}$ and for every natural number k such that $\mathrm{insloc}(k) \in \mathrm{dom}\,\mathrm{Load}(f)$ holds $(\mathrm{Load}(f))(\mathrm{insloc}(k)) = \pi_{k+1} f$.

The following propositions are true:

(24) Let f be a finite sequence of elements of the instructions of $\mathbf{SCM}_{\mathrm{FSA}}$ and let k be a natural number. Then $\mathrm{dom}\,\mathrm{Load}(f) = \{\mathrm{insloc}(m -' 1) : m \in \mathrm{dom}\, f\}$.

(25) For every finite sequence f of elements of the instructions of $\mathbf{SCM}_{\rm FSA}$ holds $\operatorname{card}\operatorname{Load}(f) = \operatorname{len} f$.

(26) Let p be a finite sequence of elements of the instructions of $\mathbf{SCM}_{\rm FSA}$ and let k be a natural number. Then $\operatorname{insloc}(k) \in \operatorname{dom}\operatorname{Load}(p)$ if and only if $k+1 \in \operatorname{dom} p$.

(27) For all natural numbers k, n holds $k < n$ iff $0 < k+1$ and $k+1 \leq n$.

(28) For all natural numbers k, n holds $k < n$ iff $1 \leq k+1$ and $k+1 \leq n$.

(29) Let p be a finite sequence of elements of the instructions of $\mathbf{SCM}_{\rm FSA}$ and let k be a natural number. Then $\operatorname{insloc}(k) \in \operatorname{dom}\operatorname{Load}(p)$ if and only if $k < \operatorname{len} p$.

(30) For every non empty finite sequence f of elements of the instructions of $\mathbf{SCM}_{\rm FSA}$ holds $1 \in \operatorname{dom} f$ and $\operatorname{insloc}(0) \in \operatorname{dom}\operatorname{Load}(f)$.

(31) For all finite sequences p, q of elements of the instructions of $\mathbf{SCM}_{\rm FSA}$ holds $\operatorname{Load}(p) \subseteq \operatorname{Load}(p \frown q)$.

(32) For all finite sequences p, q of elements of the instructions of $\mathbf{SCM}_{\rm FSA}$ such that $p \subseteq q$ holds $\operatorname{Load}(p) \subseteq \operatorname{Load}(q)$.

Let a be an integer location and let k be an integer. The functor $a{:=}k$ yields a finite partial state of $\mathbf{SCM}_{\rm FSA}$ and is defined as follows:

(Def. 2) (i) There exists a natural number k_1 such that $k_1 + 1 = k$ and $a{:=}k = \operatorname{Load}(\langle a{:=}\operatorname{intloc}(0)\rangle \frown (k_1 \mapsto \operatorname{AddTo}(a, \operatorname{intloc}(0))) \frown \langle \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}\rangle)$ if $k > 0$,

(ii) there exists a natural number k_1 such that $k_1 + k = 1$ and $a{:=}k = \operatorname{Load}(\langle a{:=}\operatorname{intloc}(0)\rangle \frown (k_1 \mapsto \operatorname{SubFrom}(a, \operatorname{intloc}(0))) \frown \langle \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}\rangle)$, otherwise.

Let a be an integer location and let k be an integer. The functor $\operatorname{aSeq}(a, k)$ yielding a finite sequence of elements of the instructions of $\mathbf{SCM}_{\rm FSA}$ is defined by:

(Def. 3) (i) There exists a natural number k_1 such that $k_1 + 1 = k$ and $\operatorname{aSeq}(a, k) = \langle a{:=}\operatorname{intloc}(0)\rangle \frown (k_1 \mapsto \operatorname{AddTo}(a, \operatorname{intloc}(0)))$ if $k > 0$,

(ii) there exists a natural number k_1 such that $k_1 + k = 1$ and $\operatorname{aSeq}(a, k) = \langle a{:=}\operatorname{intloc}(0)\rangle \frown (k_1 \mapsto \operatorname{SubFrom}(a, \operatorname{intloc}(0)))$, otherwise.

One can prove the following proposition

(33) For every integer location a and for every integer k holds $a{:=}k = \operatorname{Load}((\operatorname{aSeq}(a, k)) \frown \langle \mathbf{halt}_{\mathbf{SCM}_{\rm FSA}}\rangle)$.

Let f be a finite sequence location and let p be a finite sequence of elements of $\mathbb{Z}$. The functor $\operatorname{aSeq}(f, p)$ yields a finite sequence of elements of the instructions of $\mathbf{SCM}_{\rm FSA}$ and is defined by the condition (Def. 4).

(Def. 4) There exists a finite sequence p_3 of elements of (the instructions of $\mathbf{SCM}_{\rm FSA})^*$ such that

(i) $\operatorname{len} p_3 = \operatorname{len} p$,

(ii) for every natural number k such that $1 \leq k$ and $k \leq \operatorname{len} p$ there exists an integer i such that $i = p(k)$ and $p_3(k) = (\operatorname{aSeq}(\operatorname{intloc}(1), k)) \frown$

aSeq(intloc(2), i) $\frown$ $\langle f_{\mathrm{intloc}(1)}{:=}\,\mathrm{intloc}(2)\rangle$, and

(iii) aSeq$(f, p) = \mathrm{Flat}(p_3)$.

Let f be a finite sequence location and let p be a finite sequence of elements of $\mathbb{Z}$ The functor $f{:=}p$ yielding a finite partial state of $\mathbf{SCM}_{\mathrm{FSA}}$ is defined by:

(Def. 5) $f{:=}p = \mathrm{Load}((\mathrm{aSeq}(\mathrm{intloc}(1), \mathrm{len}\, p)) \frown \langle f{:=}\langle\underbrace{0, \ldots, 0}_{\mathrm{intloc}(1)}\rangle\rangle \frown \mathrm{aSeq}(f, p) \frown \langle \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}\rangle)$.

Next we state several propositions:

(34) For every integer location a holds $a{:=}1 = \mathrm{Load}(\langle a{:=}\,\mathrm{intloc}(0)\rangle \frown \langle \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}\rangle)$.

(35) For every integer location a holds $a{:=}0 = \mathrm{Load}(\langle a{:=}\,\mathrm{intloc}(0)\rangle \frown \langle \mathrm{SubFrom}(a, \mathrm{intloc}(0))\rangle \frown \langle \mathbf{halt}_{\mathbf{SCM}_{\mathrm{FSA}}}\rangle)$.

(36) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $s(\mathrm{intloc}(0)) = 1$. Let c_0 be a natural number. Suppose $\mathbf{IC}_s = \mathrm{insloc}(c_0)$. Let a be an integer location and let k be an integer. Suppose $a \neq \mathrm{intloc}(0)$ and for every natural number c such that $c \in \mathrm{dom}\,\mathrm{aSeq}(a, k)$ holds $(\mathrm{aSeq}(a, k))(c) = s(\mathrm{insloc}((c_0{+}c){-}'1))$. Then

(i) for every natural number i such that $i \leq \mathrm{len}\,\mathrm{aSeq}(a, k)$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(i)} = \mathrm{insloc}(c_0 + i)$ and for every integer location b such that $b \neq a$ holds $(\mathrm{Computation}(s))(i)(b) = s(b)$ and for every finite sequence location f holds $(\mathrm{Computation}(s))(i)(f) = s(f)$, and

(ii) $(\mathrm{Computation}(s))(\mathrm{len}\,\mathrm{aSeq}(a, k))(a) = k$.

(37) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $\mathbf{IC}_s = \mathrm{insloc}(0)$ and $s(\mathrm{intloc}(0)) = 1$. Let a be an integer location and let k be an integer. Suppose $\mathrm{Load}(\mathrm{aSeq}(a, k)) \subseteq s$ and $a \neq \mathrm{intloc}(0)$. Then

(i) for every natural number i such that $i \leq \mathrm{len}\,\mathrm{aSeq}(a, k)$ holds $\mathbf{IC}_{(\mathrm{Computation}(s))(i)} = \mathrm{insloc}(i)$ and for every integer location b such that $b \neq a$ holds $(\mathrm{Computation}(s))(i)(b) = s(b)$ and for every finite sequence location f holds $(\mathrm{Computation}(s))(i)(f) = s(f)$, and

(ii) $(\mathrm{Computation}(s))(\mathrm{len}\,\mathrm{aSeq}(a, k))(a) = k$.

(38) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $\mathbf{IC}_s = \mathrm{insloc}(0)$ and $s(\mathrm{intloc}(0)) = 1$. Let a be an integer location and let k be an integer. Suppose $a{:=}k \subseteq s$ and $a \neq \mathrm{intloc}(0)$. Then

(i) s is halting,

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

(iii) for every integer location b such that $b \neq a$ holds $(\mathrm{Result}(s))(b) = s(b)$, and

(iv) for every finite sequence location f holds $(\mathrm{Result}(s))(f) = s(f)$.

(39) Let s be a state of $\mathbf{SCM}_{\mathrm{FSA}}$. Suppose $\mathbf{IC}_s = \mathrm{insloc}(0)$ and $s(\mathrm{intloc}(0)) = 1$. Let f be a finite sequence location and let p be a finite sequence of elements of $\mathbb{Z}$. Suppose $f{:=}p \subseteq s$. Then

(i) s is halting,

(ii) $(\mathrm{Result}(s))(f) = p$,

(iii) for every integer location b such that $b \neq \text{intloc}(1)$ and $b \neq \text{intloc}(2)$ holds $(\text{Result}(s))(b) = s(b)$, and

(iv) for every finite sequence location g such that $g \neq f$ holds $(\text{Result}(s))(g) = s(g)$.

References

[1] Grzegorz Bancerek. Cardinal numbers. *Formalized Mathematics*, 1(**2**):377–382, 1990.
[2] Grzegorz Bancerek. The fundamental properties of natural numbers. *Formalized Mathematics*, 1(**1**):41–46, 1990.
[3] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[4] Grzegorz Bancerek and Piotr Rudnicki. On defining functions on trees. *Formalized Mathematics*, 4(**1**):91–101, 1993.
[5] Grzegorz Bancerek and Andrzej Trybulec. Miscellaneous facts about functions. *Formalized Mathematics*, 5(**4**):485–492, 1996.
[6] Czesław Byliński. Binary operations. *Formalized Mathematics*, 1(**1**):175–180, 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] Agata Darmochwał. Finite sets. *Formalized Mathematics*, 1(**1**):165–167, 1990.
[10] Krzysztof Hryniewiecki. Basic properties of real numbers. *Formalized Mathematics*, 1(**1**):35–40, 1990.
[11] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[12] Yatsuka Nakamura and Andrzej Trybulec. On a mathematical model of programs. *Formalized Mathematics*, 3(**2**):241–250, 1992.
[13] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Formalized Mathematics*, 4(**1**):83–86, 1993.
[14] Jan Popiołek. Some properties of functions modul and signum. *Formalized Mathematics*, 1(**2**):263–264, 1990.
[15] Andrzej Trybulec. Semilattice operations on finite subsets. *Formalized Mathematics*, 1(**2**):369–376, 1990.
[16] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[17] Andrzej Trybulec, Yatsuka Nakamura, and Piotr Rudnicki. The $\mathbf{SCM}_{\mathrm{FSA}}$ computer. *Formalized Mathematics*, 5(**4**):519–528, 1996.
[18] Michał J. Trybulec. Integers. *Formalized Mathematics*, 1(**3**):501–505, 1990.
[19] Wojciech A. Trybulec. Binary operations on finite sequences. *Formalized Mathematics*, 1(**5**):979–981, 1990.
[20] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[21] Zinaida Trybulec and Halina Święczkowska. Boolean properties of sets. *Formalized Mathematics*, 1(**1**):17–23, 1990.
[22] Edmund Woronowicz. Relations and their basic properties. *Formalized Mathematics*, 1(**1**):73–83, 1990.

Received April 24, 1996

FORMALIZED MATHEMATICS
Volume 5, Number 4, 1996
Warsaw University - Białystok

More on Products of Many Sorted Algebras

Mariusz Giero
Warsaw University
Białystok

Summary. This article is continuation of an article defining products of many sorted algebras [12]. Some properties of notions such as commute, Frege, Args() are shown in this article. Notions of constant of operations in many sorted algebras and projection of products of family of many sorted algebras are defined. There is also introduced the notion of class of family of many sorted algebras. The main theorem states that product of family of many sorted algebras and product of class of family of many sorted algebras are isomorphic.

MML Identifier: PRALG_3.

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

1. Preliminaries

For simplicity we adopt the following convention: I denotes a non empty set, J denotes a many sorted set indexed by I, S denotes a non void non empty many sorted signature, i denotes an element of I, c denotes a set, A denotes an algebra family of I over S, E_1 denotes an equivalence relation of I, U_0, U_1, U_2 denote algebras over S, s denotes a sort symbol of S, o denotes an operation symbol of S, and f denotes a function.

Let I be a set, let us consider S, and let A_1 be an algebra family of I over S. One can verify that $\prod A_1$ is non-empty.

Let I be a non empty set and let E_1 be an equivalence relation of I. Note that $\mathrm{Classes}\, E_1$ is non empty.

Let I be a set. Then id_I is a many sorted set indexed by I.

ISSN 1426–2630

Let us consider I, E_1. Note that $\mathrm{Classes}\, E_1$ has non empty elements.

Let X be a set with non empty elements. Then id_X is a non-empty many sorted set indexed by X.

Next we state several propositions:

(1) For all functions f, F and for every set A such that $f \in \prod F$ holds $f \restriction A \in \prod(F \restriction A)$.

(2) Let A be an algebra family of I over S, and let s be a sort symbol of S, and let a be a non empty subset of I, and let A_2 be an algebra family of a over S. If $A \restriction a = A_2$, then $\mathrm{Carrier}(A_2, s) = \mathrm{Carrier}(A, s) \restriction a$.

(3) Let i be a set, and let I be a non empty set, and let E_1 be an equivalence relation of I, and let c_1, c_2 be elements of $\mathrm{Classes}\, E_1$. If $i \in c_1$ and $i \in c_2$, then $c_1 = c_2$.

(4) For all sets X, Y and for every function f such that $f \in Y^X$ holds $\mathrm{dom}\, f = X$ and $\mathrm{rng}\, f \subseteq Y$.

(5) Let D be a non empty set, and let F be a many sorted function of D, and let C be a functional non empty set with common domain. Suppose $C = \mathrm{rng}\, F$. Let d be an element of D and let e be a set. If $d \in \mathrm{dom}\, F$ and $e \in \mathrm{DOM}(C)$, then $F(d)(e) = (\mathrm{commute}(F))(e)(d)$.

2. Constants of Many Sorted Algebras

Let us consider S, U_0 and let o be an operation symbol of S. The functor $\mathrm{const}(o, U_0)$ is defined by:

(Def. 1) $\mathrm{const}(o, U_0) = (\mathrm{Den}(o, U_0))(\varepsilon)$.

Next we state four propositions:

(6) If $\mathrm{Arity}(o) = \varepsilon$ and $\mathrm{Result}(o, U_0) \neq \emptyset$, then $\mathrm{const}(o, U_0) \in \mathrm{Result}(o, U_0)$.

(7) Suppose (the sorts of U_0)$(s) \neq \emptyset$. Then $\mathrm{Constants}(U_0, s) = \{\mathrm{const}(o, U_0) : o$ ranges over elements of the operation symbols of S, the result sort of $o = s \wedge \mathrm{Arity}(o) = \varepsilon\}$.

(8) If $\mathrm{Arity}(o) = \varepsilon$, then $(\mathrm{commute}(\mathrm{OPER}(A)))(o) \in ((\bigcup\{\mathrm{Result}(o, A(i')) :$ i' ranges over elements of $I\})^{\{\square\}})^I$.

(9) If $\mathrm{Arity}(o) = \varepsilon$, then $\mathrm{const}(o, \prod A) \in (\bigcup\{\mathrm{Result}(o, A(i')) : i'$ ranges over elements of $I\})^I$.

Let us consider S, I, o, A. Observe that $\mathrm{const}(o, \prod A)$ is relation-like and function-like.

One can prove the following three propositions:

(10) For every operation symbol o of S such that $\mathrm{Arity}(o) = \varepsilon$ holds $(\mathrm{const}(o, \prod A))(i) = \mathrm{const}(o, A(i))$.

(11) If $\mathrm{Arity}(o) = \varepsilon$ and $\mathrm{dom}\, f = I$ and for every element i of I holds $f(i) = \mathrm{const}(o, A(i))$, then $f = \mathrm{const}(o, \prod A)$.

(12) Let e be an element of $\mathrm{Args}(o, U_1)$. Suppose $e = \varepsilon$ and $\mathrm{Arity}(o) = \varepsilon$ and $\mathrm{Args}(o, U_1) \neq \emptyset$ and $\mathrm{Args}(o, U_2) \neq \emptyset$. Let F be a many sorted function from U_1 into U_2. Then $F\#e = \varepsilon$.

3. Properties of Arguments of Operations in Many Sorted Algebras

Next we state a number of propositions:

(13) Let U_1, U_2 be non-empty algebras over S, and let F be a many sorted function from U_1 into U_2, and let x be an element of $\mathrm{Args}(o, U_1)$. Then $x \in \prod(\mathrm{dom}_\kappa(F \cdot \mathrm{Arity}(o))(\kappa))$.

(14) Let U_1, U_2 be non-empty algebras over S, and let F be a many sorted function from U_1 into U_2, and let x be an element of $\mathrm{Args}(o, U_1)$, and let n be a set. If $n \in \mathrm{dom}\,\mathrm{Arity}(o)$, then $(F\#x)(n) = F(\pi_n \mathrm{Arity}(o))(x(n))$.

(15) Let x be an element of $\mathrm{Args}(o, \prod A)$. Then $x \in ((\bigcup\{(\text{the sorts of } A(i'))(s') : i' \text{ ranges over elements of } I,\ s' \text{ ranges over elements of the carrier of } S\})^I)^{\mathrm{dom}\,\mathrm{Arity}(o)}$.

(16) For every element x of $\mathrm{Args}(o, \prod A)$ and for every set n such that $n \in \mathrm{dom}\,\mathrm{Arity}(o)$ holds $x(n) \in \prod \mathrm{Carrier}(A, \pi_n \mathrm{Arity}(o))$.

(17) Let i be an element of I and let n be a set. Suppose $n \in \mathrm{dom}\,\mathrm{Arity}(o)$. Let s be a sort symbol of S. Suppose $s = \mathrm{Arity}(o)(n)$. Let y be an element of $\mathrm{Args}(o, \prod A)$ and let g be a function. If $g = y(n)$, then $g(i) \in$ (the sorts of $A(i))(s)$.

(18) For every element y of $\mathrm{Args}(o, \prod A)$ such that $\mathrm{Arity}(o) \neq \varepsilon$ holds $\mathrm{commute}(y) \in \prod(\mathrm{dom}_\kappa A(o)(\kappa))$.

(19) For every element y of $\mathrm{Args}(o, \prod A)$ such that $\mathrm{Arity}(o) \neq \varepsilon$ holds $y \in \mathrm{dom}\,\blacksquare\mathrm{commute}(\mathrm{Frege}(A(o)))$.

(20) Given I, S, A, o and let s be a sort symbol of S. Suppose $s =$ the result sort of o. Let x be an element of $\mathrm{Args}(o, \prod A)$. Then $(\mathrm{Den}(o, \prod A))(x) \in \prod \mathrm{Carrier}(A, s)$.

(21) Given I, S, A, i and let o be an operation symbol of S. Suppose $\mathrm{Arity}(o) \neq \varepsilon$. Let U_1 be a non-empty algebra over S, and let x be an element of $\mathrm{Args}(o, \prod A)$, and let F be a many sorted function from $\prod A$ into U_1. Then $(\mathrm{commute}(x))(i)$ is an element of $\mathrm{Args}(o, A(i))$.

(22) Given I, S, A, i, o, and let x be an element of $\mathrm{Args}(o, \prod A)$, and let n be a set. If $n \in \mathrm{dom}\,\mathrm{Arity}(o)$, then for every function f such that $f = x(n)$ holds $(\mathrm{commute}(x))(i)(n) = f(i)$.

(23) Let o be an operation symbol of S. Suppose $\mathrm{Arity}(o) \neq \emptyset$. Let y be an element of $\mathrm{Args}(o, \prod A)$, and let i' be an element of I, and let g be a function. If $g = (\mathrm{Den}(o, \prod A))(y)$, then $g(i') = (\mathrm{Den}(o, A(i')))((\mathrm{commute}(y))(i'))$.

4. The Projection of Family of Many Sorted Algebras

Let f be a function and let x be a set. The functor $\mathrm{proj}(f,x)$ yields a function and is defined as follows:

(Def. 2) $\mathrm{dom}\,\mathrm{proj}(f,x) = \prod f$ and for every function y such that $y \in \mathrm{dom}\,\mathrm{proj}(f,x)$ holds $(\mathrm{proj}(f,x))(y) = y(x)$.

Let us consider I, S, let A be an algebra family of I over S, and let i be an element of I. The functor $\mathrm{proj}(A,i)$ yielding a many sorted function from $\prod A$ into $A(i)$ is defined by:

(Def. 3) For every element s of the carrier of S holds $(\mathrm{proj}(A,i))(s) = \mathrm{proj}(\mathrm{Carrier}(A,s),i)$.

Next we state several propositions:

(24) For every element x of $\mathrm{Args}(o,\prod A)$ such that $\mathrm{Args}(o,\prod A) \neq \varepsilon$ and $\mathrm{Arity}(o) \neq \emptyset$ and for every element i of I holds $\mathrm{proj}(A,i)\#x = (\mathrm{commute}(x))(i)$.

(25) For every element i of I and for every algebra family A of I over S holds $\mathrm{proj}(A,i)$ is a homomorphism of $\prod A$ into $A(i)$.

(26) Let U_1 be a non-empty algebra over S and let F be a many sorted function of I. Suppose that for every element i of I there exists a many sorted function F_1 from U_1 into $A(i)$ such that $F_1 = F(i)$ and F_1 is a homomorphism of U_1 into $A(i)$ Then $F \in (\{F(i')(s_1) : s_1$ ranges over sort symbols of S, i' ranges over elements of $I\}^{\text{the carrier of } S})^I$ and $(\mathrm{commute}(F))(s)(i) = F(i)(s)$.

(27) Let U_1 be a non-empty algebra over S and let F be a many sorted function of I. Suppose that for every element i of I there exists a many sorted function F_1 from U_1 into $A(i)$ such that $F_1 = F(i)$ and F_1 is a homomorphism of U_1 into $A(i)$ Then $(\mathrm{commute}(F))(s) \in ((\bigcup\{(\text{the sorts of } A(i'))(s_1) : i'$ ranges over elements of I, s_1 ranges over sort symbols of $S\})^{(\text{the sorts of } U_1)(s)})^I$.

(28) Let U_1 be a non-empty algebra over S and let F be a many sorted function of I. Suppose that for every element i of I there exists a many sorted function F_1 from U_1 into $A(i)$ such that $F_1 = F(i)$ and F_1 is a homomorphism of U_1 into $A(i)$ Let F' be a many sorted function from U_1 into $A(i)$. Suppose $F' = F(i)$. Let x be a set. Suppose $x \in (\text{the sorts of } U_1)(s)$. Let f be a function. If $f = (\mathrm{commute}((\mathrm{commute}(F))(s)))(x)$, then $f(i) = F'(s)(x)$.

(29) Let U_1 be a non-empty algebra over S and let F be a many sorted function of I. Suppose that for every element i of I there exists a many sorted function F_1 from U_1 into $A(i)$ such that $F_1 = F(i)$ and F_1 is a homomorphism of U_1 into $A(i)$ Let x be a set. If $x \in (\text{the sorts of } U_1)(s)$, then $(\mathrm{commute}((\mathrm{commute}(F))(s)))(x) \in \prod \mathrm{Carrier}(A,s)$.

(30) Let U_1 be a non-empty algebra over S and let F be a many sorted function of I. Suppose that for every element i of I there exists a many sorted function F_1 from U_1 into $A(i)$ such that $F_1 = F(i)$ and F_1 is a homomorphism of U_1 into $A(i)$ Then there exists a many sorted function H from U_1 into $\prod A$ such that H is a homomorphism of U_1 into $\prod A$ and for every element i of I holds $\mathrm{proj}(A, i) \circ H = F(i)$.

5. The Class of Family of Many Sorted Algebras

Let us consider I, J, S. A many sorted set indexed by I is said to be a MSAlgebra-Class of S, J if:

(Def. 4) For every set i such that $i \in I$ holds $\mathrm{it}(i)$ is an algebra family of $J(i)$ over S.

Let us consider I, S, A, E_1. The functor $\frac{A}{E_1}$ yields a MSAlgebra-Class of S, $\mathrm{id}_{\mathrm{Classes}\, E_1}$ and is defined by:

(Def. 5) For every c such that $c \in \mathrm{Classes}\, E_1$ holds $(\frac{A}{E_1})(c) = A \restriction c$.

Let us consider I, S, let J be a non-empty many sorted set indexed by I, and let C be a MSAlgebra-Class of S, J. The functor $\prod C$ yields an algebra family of I over S and is defined by the condition (Def. 6).

(Def. 6) Given i. Suppose $i \in I$. Then there exists a non empty set J_1 and there exists an algebra family C_1 of J_1 over S such that $J_1 = J(i)$ and $C_1 = C(i)$ and $(\prod C)(i) = \prod C_1$.

We now state the proposition

(31) Let A be an algebra family of I over S and let E_1 be an equivalence relation of I. Then $\prod A$ and $\prod \prod (\frac{A}{E_1})$ are isomorphic.

Acknowledgments

I would like to thank Professor A.Trybulec for his help in preparation of the article.

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 and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Formalized Mathematics*, 1(**1**):107–114, 1990.
[5] Ewa Burakowska. Subalgebras of many sorted algebra. Lattice of subalgebras. *Formalized Mathematics*, 5(**1**):47–54, 1996.
[6] Czesław Byliński. A classical first order language. *Formalized Mathematics*, 1(**4**):669–676, 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] Małgorzata Korolkiewicz. Homomorphisms of many sorted algebras. *Formalized Mathematics*, 5(**1**):61–65, 1996.
[11] Beata Madras. Product of family of universal algebras. *Formalized Mathematics*, 4(**1**):103–108, 1993.
[12] Beata Madras. Products of many sorted algebras. *Formalized Mathematics*, 5(**1**):55–60, 1996.
[13] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Formalized Mathematics*, 3(**2**):151–160, 1992.
[14] Beata Padlewska. Families of sets. *Formalized Mathematics*, 1(**1**):147–152, 1990.
[15] Konrad Raczkowski and Paweł Sadowski. Equivalence relations and classes of abstraction. *Formalized Mathematics*, 1(**3**):441–444, 1990.
[16] Andrzej Trybulec. Binary operations applied to functions. *Formalized Mathematics*, 1(**2**):329–334, 1990.
[17] Andrzej Trybulec. Function domains and Frænkel operator. *Formalized Mathematics*, 1(**3**):495–500, 1990.
[18] Andrzej Trybulec. Many sorted algebras. *Formalized Mathematics*, 5(**1**):37–42, 1996.
[19] Andrzej Trybulec. Many-sorted sets. *Formalized Mathematics*, 4(**1**):15–22, 1993.
[20] Andrzej Trybulec. Tarski Grothendieck set theory. *Formalized Mathematics*, 1(**1**):9–11, 1990.
[21] Wojciech A. Trybulec. Pigeon hole principle. *Formalized Mathematics*, 1(**3**):575–579, 1990.
[22] Zinaida Trybulec 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.
[24] Edmund Woronowicz. Relations defined on sets. *Formalized Mathematics*, 1(**1**):181–186, 1990.

Received April 29, 1996

alalalalalal alalalalalalal

Index of MML Identifiers

628

Contents

Formaliz. Math. 5 (**4**)

Continued on inside back cover