

Standard Ordering of Instruction Locations

Andrzej Trybulec
University of Białystok

Piotr Rudnicki
University of Alberta

Artur Korniłowicz
University of Białystok

MML Identifier: AMISTD_1.

WWW: http://mizar.org/JFM/Vol12/amistd_1.html

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

1. PRELIMINARIES

For simplicity, we adopt the following rules: x, X are sets, D is a non empty set, n is a natural number, and z is a natural number.

Next we state two propositions:

- (1) For every real number r holds $\max\{r\} = r$.
- (2) $\max\{n\} = n$.

Let us observe that there exists a finite sequence which is non trivial.

Next we state the proposition

- (3) For every trivial finite sequence f of elements of D holds f is empty or there exists an element x of D such that $f = \langle x \rangle$.

Let us note that every binary relation has non empty elements.

We now state the proposition

- (4) id_X is bijective.

Let A be a finite set and let B be a set. One can verify that $A \mapsto B$ is finite.

Let x, y be sets. Observe that $x \mapsto y$ is trivial.

2. RESTRICTED CONCATENATION

Let f_1 be a non empty finite sequence and let f_2 be a finite sequence. Observe that $f_1 \smile f_2$ is non empty.

Next we state several propositions:

- (5) Let f_1 be a non empty finite sequence of elements of D and f_2 be a finite sequence of elements of D . Then $(f_1 \smile f_2)_1 = (f_1)_1$.
- (6) Let f_1 be a finite sequence of elements of D and f_2 be a non trivial finite sequence of elements of D . Then $(f_1 \smile f_2)_{\text{len}(f_1 \smile f_2)} = (f_2)_{\text{len} f_2}$.

- (7) For every finite sequence f holds $f \frown \emptyset = f$.
- (8) For every finite sequence f holds $f \frown \langle x \rangle = f$.
- (9) For all finite sequences f_1, f_2 of elements of D such that $1 \leq n$ and $n \leq \text{len } f_1$ holds $(f_1 \frown f_2)_n = (f_1)_n$.
- (10) For all finite sequences f_1, f_2 of elements of D such that $1 \leq n$ and $n < \text{len } f_2$ holds $(f_1 \frown f_2)_{\text{len } f_1 + n} = (f_2)_{n+1}$.

3. AMI-STRUCT

For simplicity, we adopt the following convention: N denotes a set with non empty elements, S denotes an IC-Ins-separated definite non empty non void AMI over N , i denotes an element of the instructions of S , l, l_1, l_2, l_3 denote instruction-locations of S , and s denotes a state of S .

The following proposition is true

- (11) Let S be a definite non empty non void AMI over N , I be an element of the instructions of S , and s be a state of S . Then $s \dashv \cdot ((\text{the instruction locations of } S) \dashv \rightarrow I)$ is a state of S .

Let N be a set and let S be an AMI over N . One can check that every finite partial state of S which is empty is also programmed.

Let N be a set and let S be an AMI over N . One can verify that there exists a finite partial state of S which is empty.

Let N be a set with non empty elements and let S be an IC-Ins-separated definite non empty non void AMI over N . Note that there exists a finite partial state of S which is non empty, trivial, and programmed.

Let N be a set with non empty elements, let S be a non void AMI over N , let i be an element of the instructions of S , and let s be a state of S . Observe that $(\text{the execution of } S)(i)(s)$ is function-like and relation-like.

Let N be a set with non empty elements and let S be a steady-programmed IC-Ins-separated definite non empty non void AMI over N . One can verify that there exists a finite partial state of S which is non empty, trivial, autonomic, and programmed.

Next we state two propositions:

- (12) Let S be a steady-programmed IC-Ins-separated definite non empty non void AMI over N , i_1 be an instruction-location of S , and I be an element of the instructions of S . Then $i_1 \dashv \rightarrow I$ is autonomic.
- (13) Every steady-programmed IC-Ins-separated definite non empty non void AMI over N is programmable.

Let N be a set with non empty elements. Observe that every IC-Ins-separated definite non empty non void AMI over N which is steady-programmed is also programmable.

Let N be a set with non empty elements, let S be a non empty non void AMI over N , and let T be an instruction type of S . We say that T is jump-only if and only if the condition (Def. 3) is satisfied.

(Def. 3)¹ Let s be a state of S , o be an object of S , and I be an instruction of S . If $\text{InsCode}(I) = T$ and $o \neq \mathbf{IC}_S$, then $(\text{Exec}(I, s))(o) = s(o)$.

Let N be a set with non empty elements, let S be a non empty non void AMI over N , and let I be an instruction of S . We say that I is jump-only if and only if:

(Def. 4) $\text{InsCode}(I)$ is jump-only.

Let us consider N, S, l and let i be an element of the instructions of S . The functor $\text{NIC}(i, l)$ yielding a subset of the instruction locations of S is defined as follows:

¹ The definitions (Def. 1) and (Def. 2) have been removed.

(Def. 5) $\text{NIC}(i, l) = \{\mathbf{IC}_{\text{Following}(s)} : \mathbf{IC}_s = l \wedge s(l) = i\}$.

Let N be a set with non empty elements, let S be a realistic IC-Ins-separated definite non empty non void AMI over N , let i be an element of the instructions of S , and let l be an instruction-location of S . Observe that $\text{NIC}(i, l)$ is non empty.

Let us consider N, S, i . The functor $\text{JUMP}(i)$ yields a subset of the instruction locations of S and is defined as follows:

(Def. 6) $\text{JUMP}(i) = \bigcap \{\text{NIC}(i, l)\}$.

Let us consider N, S, l . The functor $\text{SUCC}(l)$ yielding a subset of the instruction locations of S is defined by:

(Def. 7) $\text{SUCC}(l) = \bigcup \{\text{NIC}(i, l) \setminus \text{JUMP}(i)\}$.

Next we state two propositions:

- (14) Let i be an element of the instructions of S . Suppose the instruction locations of S are non trivial and for every instruction-location l of S holds $\text{NIC}(i, l) = \{l\}$. Then $\text{JUMP}(i)$ is empty.
- (15) Let S be a realistic IC-Ins-separated definite non empty non void AMI over N , i_1 be an instruction-location of S , and i be an instruction of S . If i is halting, then $\text{NIC}(i, i_1) = \{i_1\}$.

4. ORDERING OF INSTRUCTION LOCATIONS

Let us consider N, S, l_1, l_2 . The predicate $l_1 \leq l_2$ is defined by the condition (Def. 8).

(Def. 8) There exists a non empty finite sequence f of elements of the instruction locations of S such that $f_1 = l_1$ and $f_{\text{len } f} = l_2$ and for every n such that $1 \leq n$ and $n < \text{len } f$ holds $f_{n+1} \in \text{SUCC}(f_n)$.

Let us note that the predicate $l_1 \leq l_2$ is reflexive.

The following proposition is true

- (16) If $l_1 \leq l_2$ and $l_2 \leq l_3$, then $l_1 \leq l_3$.

Let us consider N, S . We say that S is InsLoc-antisymmetric if and only if:

(Def. 9) For all l_1, l_2 such that $l_1 \leq l_2$ and $l_2 \leq l_1$ holds $l_1 = l_2$.

Let us consider N, S . We say that S is standard if and only if the condition (Def. 10) is satisfied.

(Def. 10) There exists a function f from \mathbb{N} into the instruction locations of S such that f is bijective and for all natural numbers m, n holds $m \leq n$ iff $f(m) \leq f(n)$.

We now state three propositions:

(17) Let f_1, f_2 be functions from \mathbb{N} into the instruction locations of S . Suppose that

- (i) f_1 is bijective,
- (ii) for all natural numbers m, n holds $m \leq n$ iff $f_1(m) \leq f_1(n)$,
- (iii) f_2 is bijective, and
- (iv) for all natural numbers m, n holds $m \leq n$ iff $f_2(m) \leq f_2(n)$.

Then $f_1 = f_2$.

(18) Let f be a function from \mathbb{N} into the instruction locations of S . Suppose f is bijective. Then the following statements are equivalent

- (i) for all natural numbers m, n holds $m \leq n$ iff $f(m) \leq f(n)$,
- (ii) for every natural number k holds $f(k+1) \in \text{SUCC}(f(k))$ and for every natural number j such that $f(j) \in \text{SUCC}(f(k))$ holds $k \leq j$.

(19) S is standard if and only if there exists a function f from \mathbb{N} into the instruction locations of S such that f is bijective and for every natural number k holds $f(k+1) \in \text{SUCC}(f(k))$ and for every natural number j such that $f(j) \in \text{SUCC}(f(k))$ holds $k \leq j$.

5. STANDARD TRIVIAL COMPUTER

Let N be a set with non empty elements. The functor $\text{STC}(N)$ yielding a strict AMI over N is defined by the conditions (Def. 11).

(Def. 11) The carrier of $\text{STC}(N) = \mathbb{N} \cup \{\mathbb{N}\}$ and the instruction counter of $\text{STC}(N) = \mathbb{N}$ and the instruction locations of $\text{STC}(N) = \mathbb{N}$ and the instruction codes of $\text{STC}(N) = \{0, 1\}$ and the instructions of $\text{STC}(N) = \{\langle 0, 0 \rangle, \langle 1, 0 \rangle\}$ and the object kind of $\text{STC}(N) = (\mathbb{N} \mapsto \{\langle 1, 0 \rangle, \langle 0, 0 \rangle\}) + \cdot (\{\mathbb{N}\} \mapsto \mathbb{N})$ and there exists a function f from \prod (the object kind of $\text{STC}(N)$) into \prod (the object kind of $\text{STC}(N)$) such that for every element s of \prod (the object kind of $\text{STC}(N)$) holds $f(s) = s + \cdot (\{\mathbb{N}\} \mapsto \text{succ}.s(\mathbb{N}))$ and the execution of $\text{STC}(N) = (\{\langle 1, 0 \rangle\} \mapsto f) + \cdot (\{\langle 0, 0 \rangle\} \mapsto \text{id}_{\prod(\text{the object kind of } \text{STC}(N))})$.

Let N be a set with non empty elements. Note that the instruction locations of $\text{STC}(N)$ is infinite.

Let N be a set with non empty elements. One can verify that $\text{STC}(N)$ is non empty and non void.

Let N be a set with non empty elements. One can check that $\text{STC}(N)$ is IC-Ins-separated, definite, realistic, steady-programmed, and data-oriented.

One can prove the following propositions:

- (20) For every instruction i of $\text{STC}(N)$ such that $\text{InsCode}(i) = 0$ holds i is halting.
- (21) For every instruction i of $\text{STC}(N)$ such that $\text{InsCode}(i) = 1$ holds i is non halting.
- (22) For every element i of the instructions of $\text{STC}(N)$ holds $\text{InsCode}(i) = 1$ or $\text{InsCode}(i) = 0$.
- (23) Every instruction of $\text{STC}(N)$ is jump-only.
- (24) For every instruction-location l of $\text{STC}(N)$ such that $l = z$ holds $\text{SUCC}(l) = \{z, z + 1\}$.

Let N be a set with non empty elements. One can verify that $\text{STC}(N)$ is standard.

Let N be a set with non empty elements. One can check that $\text{STC}(N)$ is halting.

Let N be a set with non empty elements. Note that there exists an IC-Ins-separated definite non empty non void AMI over N which is standard, halting, realistic, steady-programmed, and programmable.

In the sequel T denotes a standard IC-Ins-separated definite non empty non void AMI over N .

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N , and let k be a natural number. The functor $\text{il}_S(k)$ yielding an instruction-location of S is defined by the condition (Def. 12).

(Def. 12) There exists a function f from \mathbb{N} into the instruction locations of S such that f is bijective and for all natural numbers m, n holds $m \leq n$ iff $f(m) \leq f(n)$ and $\text{il}_S(k) = f(k)$.

We now state two propositions:

- (25) For all natural numbers k_1, k_2 such that $\text{il}_T(k_1) = \text{il}_T(k_2)$ holds $k_1 = k_2$.
- (26) For every instruction-location l of T there exists a natural number k such that $l = \text{il}_T(k)$.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N , and let l be an instruction-location of S . The functor $\text{locnum}(l)$ yielding a natural number is defined by:

(Def. 13) $\text{il}_S(\text{locnum}(l)) = l$.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N , and let l be an instruction-location of S . Then $\text{locnum}(l)$ is a natural number.

Next we state four propositions:

- (27) For all instruction-locations l_1, l_2 of T such that $\text{locnum}(l_1) = \text{locnum}(l_2)$ holds $l_1 = l_2$.

- (28) For all natural numbers k_1, k_2 holds $il_T(k_1) \leq il_T(k_2)$ iff $k_1 \leq k_2$.
- (29) For all instruction-locations l_1, l_2 of T holds $locnum(l_1) \leq locnum(l_2)$ iff $l_1 \leq l_2$.
- (30) If S is standard, then S is InsLoc-antisymmetric.

Let us consider N . Note that every IC-Ins-separated definite non empty non void AMI over N which is standard is also InsLoc-antisymmetric.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N , let f be an instruction-location of S , and let k be a natural number. The functor $f + k$ yields an instruction-location of S and is defined by:

(Def. 14) $f + k = il_S(locnum(f) + k)$.

Next we state three propositions:

- (31) For every instruction-location f of T holds $f + 0 = f$.
- (32) For all instruction-locations f, g of T such that $f + z = g + z$ holds $f = g$.
- (33) For every instruction-location f of T holds $locnum(f) + z = locnum(f + z)$.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N , and let f be an instruction-location of S . The functor NextLoc f yielding an instruction-location of S is defined by:

(Def. 15) NextLoc $f = f + 1$.

The following propositions are true:

- (34) For every instruction-location f of T holds NextLoc $f = il_T(locnum(f) + 1)$.
- (35) For every instruction-location f of T holds $f \neq \text{NextLoc } f$.
- (36) For all instruction-locations f, g of T such that NextLoc $f = \text{NextLoc } g$ holds $f = g$.
- (37) $il_{STC(N)}(z) = z$.
- (38) For every instruction i of $STC(N)$ and for every state s of $STC(N)$ such that $\text{InsCode}(i) = 1$ holds $(\text{Exec}(i, s))(\mathbf{IC}_{STC(N)}) = \text{NextLoc } \mathbf{IC}_s$.
- (39) Let l be an instruction-location of $STC(N)$ and i be an element of the instructions of $STC(N)$. If $\text{InsCode}(i) = 1$, then $\text{NIC}(i, l) = \{\text{NextLoc } l\}$.
- (40) For every instruction-location l of $STC(N)$ holds $\text{SUCC}(l) = \{l, \text{NextLoc } l\}$.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N , and let i be an instruction of S . We say that i is sequential if and only if:

(Def. 16) For every state s of S holds $(\text{Exec}(i, s))(\mathbf{IC}_S) = \text{NextLoc } \mathbf{IC}_s$.

Next we state two propositions:

- (41) Let S be a standard realistic IC-Ins-separated definite non empty non void AMI over N , i_1 be an instruction-location of S , and i be an instruction of S . If i is sequential, then $\text{NIC}(i, i_1) = \{\text{NextLoc } i_1\}$.
- (42) Let S be a realistic standard IC-Ins-separated definite non empty non void AMI over N and i be an instruction of S . If i is sequential, then i is non halting.

Let us consider N and let S be a realistic standard IC-Ins-separated definite non empty non void AMI over N . Observe that every instruction of S which is sequential is also non halting and every instruction of S which is halting is also non sequential.

One can prove the following proposition

- (43) For every instruction i of T such that $\text{JUMP}(i)$ is non empty holds i is non sequential.

6. CLOSEDNESS OF FINITE PARTIAL STATES

Let N be a set with non empty elements, let S be an IC-Ins-separated definite non empty non void AMI over N , and let F be a finite partial state of S . We say that F is closed if and only if:

(Def. 17) For every instruction-location l of S such that $l \in \text{dom } F$ holds $\text{NIC}(\pi_l F, l) \subseteq \text{dom } F$.

We say that F is really-closed if and only if:

(Def. 18) For every state s of S such that $F \subseteq s$ and $\mathbf{IC}_s \in \text{dom } F$ and for every natural number k holds $\mathbf{IC}_{(\text{Computation}(s))(k)} \in \text{dom } F$.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N , and let F be a finite partial state of S . We say that F is para-closed if and only if:

(Def. 19) For every state s of S such that $F \subseteq s$ and $\mathbf{IC}_s = \text{il}_S(0)$ and for every natural number k holds $\mathbf{IC}_{(\text{Computation}(s))(k)} \in \text{dom } F$.

One can prove the following propositions:

- (44) Let S be a standard steady-programmed IC-Ins-separated definite non empty non void AMI over N and F be a finite partial state of S . If F is really-closed and $\text{il}_S(0) \in \text{dom } F$, then F is para-closed.
- (45) Let S be an IC-Ins-separated definite steady-programmed non empty non void AMI over N and F be a finite partial state of S . If F is closed, then F is really-closed.

Let N be a set with non empty elements and let S be an IC-Ins-separated definite steady-programmed non empty non void AMI over N . Note that every finite partial state of S which is closed is also really-closed.

Next we state the proposition

- (46) For every standard realistic halting IC-Ins-separated definite non empty non void AMI S over N holds $\text{il}_S(0) \vdash \mathbf{halt}_S$ is closed.

Let N be a set with non empty elements, let S be an IC-Ins-separated definite non empty non void AMI over N , and let F be a finite partial state of S . We say that F is lower if and only if the condition (Def. 20) is satisfied.

(Def. 20) Let l be an instruction-location of S . Suppose $l \in \text{dom } F$. Let m be an instruction-location of S . If $m \leq l$, then $m \in \text{dom } F$.

We now state the proposition

- (47) Every empty finite partial state of S is lower.

Let N be a set with non empty elements and let S be an IC-Ins-separated definite non empty non void AMI over N . Observe that every finite partial state of S which is empty is also lower.

We now state the proposition

- (48) For every element i of the instructions of T holds $\text{il}_T(0) \vdash i$ is lower.

Let N be a set with non empty elements and let S be a standard IC-Ins-separated definite non empty non void AMI over N . One can check that there exists a finite partial state of S which is lower, non empty, trivial, and programmed.

The following two propositions are true:

- (49) For every lower non empty programmed finite partial state F of T holds $\text{il}_T(0) \in \text{dom } F$.
- (50) For every lower programmed finite partial state P of T holds $z < \text{card } P$ iff $\text{il}_T(z) \in \text{dom } P$.

Let N be a set with non empty elements, let S be a standard IC-Ins-separated definite non empty non void AMI over N , and let F be a non empty programmed finite partial state of S . The functor $\text{LastLoc } F$ yields an instruction-location of S and is defined by the condition (Def. 21).

(Def. 21) There exists a finite non empty natural-membered set M such that $M = \{\text{locnum}(l); l \text{ ranges over elements of the instruction locations of } S: l \in \text{dom } F\}$ and $\text{LastLoc } F = \text{il}_S(\max M)$.

The following propositions are true:

- (51) For every non empty programmed finite partial state F of T holds $\text{LastLoc } F \in \text{dom } F$.
- (52) For all non empty programmed finite partial states F, G of T such that $F \subseteq G$ holds $\text{LastLoc } F \leq \text{LastLoc } G$.
- (53) Let F be a non empty programmed finite partial state of T and l be an instruction-location of T . If $l \in \text{dom } F$, then $l \leq \text{LastLoc } F$.
- (54) Let F be a lower non empty programmed finite partial state of T and G be a non empty programmed finite partial state of T . If $F \subseteq G$ and $\text{LastLoc } F = \text{LastLoc } G$, then $F = G$.
- (55) For every lower non empty programmed finite partial state F of T holds $\text{LastLoc } F = \text{il}_T(\text{card } F - 1)$.

Let N be a set with non empty elements and let S be a standard steady-programmed IC-Ins-separated definite non empty non void AMI over N . One can verify that every finite partial state of S which is really-closed, lower, non empty, and programmed is also para-closed.

Let N be a set with non empty elements, let S be a standard halting IC-Ins-separated definite non empty non void AMI over N , and let F be a non empty programmed finite partial state of S . We say that F is halt-ending if and only if:

(Def. 22) $F(\text{LastLoc } F) = \mathbf{halt}_S$.

We say that F is unique-halt if and only if:

(Def. 23) For every instruction-location f of S such that $F(f) = \mathbf{halt}_S$ and $f \in \text{dom } F$ holds $f = \text{LastLoc } F$.

Let N be a set with non empty elements and let S be a standard halting IC-Ins-separated definite non empty non void AMI over N . Observe that there exists a lower non empty programmed finite partial state of S which is halt-ending, unique-halt, and trivial.

Let N be a set with non empty elements and let S be a standard halting realistic IC-Ins-separated definite non empty non void AMI over N . Note that there exists a finite partial state of S which is trivial, closed, lower, non empty, and programmed.

Let N be a set with non empty elements and let S be a standard halting realistic IC-Ins-separated definite non empty non void AMI over N . One can verify that there exists a lower non empty programmed finite partial state of S which is halt-ending, unique-halt, trivial, and closed.

Let N be a set with non empty elements and let S be a standard halting realistic steady-programmed IC-Ins-separated definite non empty non void AMI over N . Note that there exists a lower non empty programmed finite partial state of S which is halt-ending, unique-halt, autonomic, trivial, and closed.

Let N be a set with non empty elements and let S be a standard halting IC-Ins-separated definite non empty non void AMI over N . A pre-Macro of S is a halt-ending unique-halt lower non empty programmed finite partial state of S .

Let N be a set with non empty elements and let S be a standard realistic halting IC-Ins-separated definite non empty non void AMI over N . Note that there exists a pre-Macro of S which is closed.

REFERENCES

- [1] Grzegorz Bancerek. Cardinal numbers. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Voll1/card_1.html.
- [2] Grzegorz Bancerek. The fundamental properties of natural numbers. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Voll1/nat_1.html.

- [3] Grzegorz Bancerek. The ordinal numbers. *Journal of Formalized Mathematics*, 1, 1989. <http://mizar.org/JFM/Vol1/ordinal1.html>.
- [4] Grzegorz Bancerek. Sequences of ordinal numbers. *Journal of Formalized Mathematics*, 1, 1989. <http://mizar.org/JFM/Vol1/ordinal2.html>.
- [5] Grzegorz Bancerek. König's theorem. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/card_3.html.
- [6] Grzegorz Bancerek and Krzysztof Hryniewiecki. Segments of natural numbers and finite sequences. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/finseq_1.html.
- [7] Józef Białas. Group and field definitions. *Journal of Formalized Mathematics*, 1, 1989. <http://mizar.org/JFM/Vol1/realset1.html>.
- [8] Czesław Byliński. Functions and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/funct_1.html.
- [9] Czesław Byliński. Functions from a set to a set. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/funct_2.html.
- [10] Czesław Byliński. Partial functions. *Journal of Formalized Mathematics*, 1, 1989. <http://mizar.org/JFM/Vol1/partfun1.html>.
- [11] Czesław Byliński. Some basic properties of sets. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/zfmisc_1.html.
- [12] Czesław Byliński. A classical first order language. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/cqc_lang.html.
- [13] Czesław Byliński. The modification of a function by a function and the iteration of the composition of a function. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/funct_4.html.
- [14] Agata Darmochwał. Finite sets. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/finset_1.html.
- [15] Yatsuka Nakamura and Piotr Rudnicki. Vertex sequences induced by chains. *Journal of Formalized Mathematics*, 7, 1995. http://mizar.org/JFM/Vol7/graph_2.html.
- [16] Yatsuka Nakamura, Piotr Rudnicki, Andrzej Trybulec, and Pauline N. Kawamoto. Preliminaries to circuits, I. *Journal of Formalized Mathematics*, 6, 1994. http://mizar.org/JFM/Vol6/pre_circ.html.
- [17] Yatsuka Nakamura and Andrzej Trybulec. A mathematical model of CPU. *Journal of Formalized Mathematics*, 4, 1992. http://mizar.org/JFM/Vol4/ami_1.html.
- [18] Takaya Nishiyama and Yasuho Mizuhara. Binary arithmetics. *Journal of Formalized Mathematics*, 5, 1993. <http://mizar.org/JFM/Vol5/binarith.html>.
- [19] Beata Padlewska. Families of sets. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/setfam_1.html.
- [20] Yasushi Tanaka. On the decomposition of the states of SCM. *Journal of Formalized Mathematics*, 5, 1993. http://mizar.org/JFM/Vol5/ami_5.html.
- [21] Andrzej Trybulec. Domains and their Cartesian products. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/domain_1.html.
- [22] Andrzej Trybulec. Tarski Grothendieck set theory. *Journal of Formalized Mathematics*, Axiomatics, 1989. <http://mizar.org/JFM/Axiomatics/tarski.html>.
- [23] Andrzej Trybulec. On the sets inhabited by numbers. *Journal of Formalized Mathematics*, 15, 2003. <http://mizar.org/JFM/Vol16/membered.html>.
- [24] Andrzej Trybulec. Subsets of real numbers. *Journal of Formalized Mathematics*, Addenda, 2003. <http://mizar.org/JFM/Addenda/numbers.html>.
- [25] Andrzej Trybulec and Yatsuka Nakamura. Some remarks on the simple concrete model of computer. *Journal of Formalized Mathematics*, 5, 1993. http://mizar.org/JFM/Vol5/ami_3.html.
- [26] Wojciech A. Trybulec. Pigeon hole principle. *Journal of Formalized Mathematics*, 2, 1990. http://mizar.org/JFM/Vol2/finseq_4.html.
- [27] Zinaida Trybulec. Properties of subsets. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/subset_1.html.
- [28] Edmund Woronowicz. Relations and their basic properties. *Journal of Formalized Mathematics*, 1, 1989. http://mizar.org/JFM/Vol1/relat_1.html.

Received April 14, 2000

Published January 2, 2004
