FORMALIZED MATHEMATICS
Vol. 25, No. 3, Pages 205–216, 2017
DOI: 10.1515/forma-2017-0020

DE GRUYTER OPEN
degruyter.com/view/j/forma

Simple-Named Complex-Valued Nominative Data – Definition and Basic Operations

Ievgen Ivanov
Taras Shevchenko National University
Kyiv, Ukraine

Mykola Nikitchenko
Taras Shevchenko National University
Kyiv, Ukraine

Andrii Kryvolap
Taras Shevchenko National University
Kyiv, Ukraine

Artur Korniłowicz
Institute of Informatics
University of Białystok
Poland

Summary. In this paper we give a formal definition of the notion of nominative data with simple names and complex values [15, 16, 19] and formal definitions of the basic operations on such data, including naming, denaming and overlapping, following the work [19].

The notion of nominative data plays an important role in the composition-nominative approach to program formalization [15, 16] which is a development of composition programming [18]. Both approaches are compared in [14].

The composition-nominative approach considers mathematical models of computer software and data on various levels of abstraction and generality and provides mathematical tools for reasoning about their properties. In particular, nominative data are mathematical models of data which are stored and processed in computer systems. The composition-nominative approach considers different types [14, 19] of nominative data, but all of them are based on the name-value relation. One powerful type of nominative data, which is suitable for representing many kinds of data commonly used in programming like lists, multidimensional arrays, trees, tables, etc. is the type of nominative data with simple (abstract) names and complex (structured) values. The set of nominative data of given type together with a number of basic operations on them like naming, denaming and overlapping [19] form an algebra which is called *data algebra*.

ISSN 1426–2630(Print), 1898-9934(Online)

In the composition-nominative approach computer programs which process data are modeled as partial functions which map nominative data from the carrier of a given data algebra (input data) to nominative data (output data). Such functions are also called *binominative functions*. Programs which evaluate conditions are modeled as partial predicates on nominative data (nominative predicates). Programming language constructs like sequential execution, branching, cycle, etc. which construct programs from the existing programs are modeled as operations which take binominative functions and predicates and produce binominative functions. Such operations are called *compositions*. A set of binominative functions and a set of predicates together with appropriate compositions form an algebra which is called *program algebra*. This algebra serves as a semantic model of a programming language.

For functions over nominative data a special computability called abstract computability is introduces and complete classes of computable functions are specified [16].

For reasoning about properties of programs modeled as binominative functions a Floyd-Hoare style logic [1, 2] is introduced and applied [12, 13, 8, 11, 9, 10]. One advantage of this approach to reasoning about programs is that it naturally handles programs which process complex data structures (which can be quite straightforwardly represented as nominative data). Also, unlike classical Floyd-Hoare logic, the mentioned logic allows reasoning about assertions which include partial pre- and post-conditions [11].

Besides modeling data processed by programs, nominative data can be also applied to modeling data processed by signal processing systems in the context of the mathematical systems theory [4, 6, 7, 5, 3].

MSC: 68Q60 03B70 03B35

Keywords: program semantics; software verification; nominative data

MML identifier: `NOMIN_1`, version: `8.1.06 5.44.1305`

1. Preliminaries

From now on a, a_1, a_2, v, v_1, v_2, x denote objects, V, A denote sets, m, n denote natural numbers, and S, S_1, S_2 denote finite sequences.

Now we state the propositions:

(1) Let us consider a finite sequence f. If $n \in \operatorname{dom} f$, then $(\langle x \rangle \frown f)(n+1) = f(n)$.

(2) Let us consider a function f. Suppose $\operatorname{dom} f = \mathbb{N}$. Then $f \restriction \operatorname{Seg} n$ is a finite sequence.

(3) Let us consider finite sequences f, g. Then

(i) $\operatorname{dom} f \subseteq \operatorname{dom} g$, or

(ii) $\operatorname{dom} g \subseteq \operatorname{dom} f$.

Let f, g be finite sequences. One can check that $f+\cdot g$ is finite sequence-like.

Let f_1, f_2 be functions. Note that $f_2 \cup f_1 \restriction (\operatorname{dom} f_1 \setminus \operatorname{dom} f_2)$ is function-like.

Let f, g be functions and x, y be objects. We say that $f(x) \cong g(y)$ if and only if

(Def. 1) ($x \in \operatorname{dom} f$ iff $y \in \operatorname{dom} g$) and if $x \in \operatorname{dom} f$, then $f(x) = g(y)$.

2. Definition of Simple-Named Complex-Valued Nominative Data

Let us consider V and A.

A nominative set of V and A is a partial function from V to A. Let us note that there exists a nominative set of V and A which is finite.

A nominative data with simple names from V and simple values from A is a finite nominative set of V and A. The functor $\mathrm{ND_{SS}}(V, A)$ yielding a set is defined by the term

(Def. 2) the set of all d where d is a nominative data with simple names from V and simple values from A.

Let us note that $\mathrm{ND_{SS}}(V, A)$ is non empty.

Now we state the propositions:

(4) If $x \in \mathrm{ND_{SS}}(V, A)$, then x is a nominative data with simple names from V and simple values from A.

(5) $\mathrm{ND_{SS}}(V, A) \subseteq V \dot{\rightarrow} A$.

(6) $\emptyset \in \mathrm{ND_{SS}}(V, A)$.

(7) Let us consider sets A, B. If $A \subseteq B$, then $\mathrm{ND_{SS}}(V, A) \subseteq \mathrm{ND_{SS}}(V, B)$.

(8) Let us consider finite functions f, g. Suppose $f \approx g$ and $f, g \in \mathrm{ND_{SS}}(V, A)$. Then $f \cup g \in \mathrm{ND_{SS}}(V, A)$. The theorem is a consequence of (4).

Let us consider V and A. The functor $\mathrm{FND_{SC}}(V, A)$ yielding a function is defined by

(Def. 3) $\operatorname{dom} \mathit{it} = \mathbb{N}$ and $\mathit{it}(0) = A$ and for every natural number n, $\mathit{it}(n+1) = \mathrm{ND_{SS}}(V, A \cup \mathit{it}(n))$.

Now we state the propositions:

(9) $(\mathrm{FND_{SC}}(V, A))(1) = \mathrm{ND_{SS}}(V, A)$.

(10) $(\mathrm{FND_{SC}}(V, A))(2) = \mathrm{ND_{SS}}(V, A \cup \mathrm{ND_{SS}}(V, A))$. The theorem is a consequence of (9).

(11) $A \subseteq \bigcup \mathrm{FND_{SC}}(V, A)$.

(12) If $1 \leqslant n$, then $\emptyset \in (\mathrm{FND_{SC}}(V, A))(n)$. The theorem is a consequence of (6).

Let us consider V, A, and n. One can check that $\mathrm{FND_{SC}}(V,A){\restriction}\operatorname{Seg} n$ is finite sequence-like.

Now we state the proposition:

(13) If $m \neq 0$ and $m \leqslant n$, then $(\mathrm{FND_{SC}}(V,A))(m) \subseteq (\mathrm{FND_{SC}}(V,A))(n)$.
PROOF: Set $S = \mathrm{FND_{SC}}(V,A)$. Define $\mathcal{P}[\text{natural number}] \equiv$ if $m \leqslant \$_1$, then $S(m) \subseteq S(\$_1)$. For every natural number k such that $\mathcal{P}[k]$ holds $\mathcal{P}[k+1]$. For every natural number k, $\mathcal{P}[k]$. □

Let us consider V and A. Let S be a finite sequence. We say that S is a rank sequence if and only if

(Def. 4) $S(1) = \mathrm{ND_{SS}}(V,A)$ and for every natural number n such that $n, n+1 \in \operatorname{dom} S$ holds $S(n+1) = \mathrm{ND_{SS}}(V, A \cup S(n))$.

Now we state the propositions:

(14) If S is a rank sequence, then $S \neq \emptyset$.

(15) If S is a rank sequence and $S_1 \subseteq S$ and $S_1 \neq \emptyset$, then S_1 is a rank sequence.

(16) If S is a rank sequence and $n \in \operatorname{dom} S$, then $S{\restriction}n$ is a rank sequence. The theorem is a consequence of (15).

(17) If S is a rank sequence, then $S \frown \langle \mathrm{ND_{SS}}(V, A \cup S(\operatorname{len} S)) \rangle$ is a rank sequence.

(18) If $1 \leqslant n$, then $\mathrm{FND_{SC}}(V,A){\restriction}\operatorname{Seg} n$ is a rank sequence. The theorem is a consequence of (9).

(19) If S is a rank sequence and $n \in \operatorname{dom} S$, then $S(n) = (\mathrm{FND_{SC}}(V,A))(n)$.
PROOF: Set $F = \mathrm{FND_{SC}}(V,A)$. Define $\mathcal{P}[\text{natural number}] \equiv$ if $\$_1 \in \operatorname{dom} S$, then $S(\$_1) = F(\$_1)$. For every n such that $\mathcal{P}[n]$ holds $\mathcal{P}[n+1]$. For every n, $\mathcal{P}[n]$. □

(20) If S is a rank sequence, then $S = \mathrm{FND_{SC}}(V,A){\restriction}\operatorname{dom} S$. The theorem is a consequence of (19).

(21) If S_1 is a rank sequence and S_2 is a rank sequence, then $S_1 \approx S_2$.
PROOF: Define $\mathcal{P}[\text{natural number}] \equiv$ if $\$_1 \in \operatorname{dom} S_1 \cap \operatorname{dom} S_2$, then $S_1(\$_1) = S_2(\$_1)$. $\mathcal{P}[0]$. For every n such that $\mathcal{P}[n]$ holds $\mathcal{P}[n+1]$. For every n, $\mathcal{P}[n]$. □

(22) If S_1 is a rank sequence and S_2 is a rank sequence, then $S_1 \subseteq S_2$ or $S_2 \subseteq S_1$. The theorem is a consequence of (20) and (3).

(23) If S_1 is a rank sequence and S_2 is a rank sequence, then $S_1 {+\cdot} S_2 = S_1$ or $S_1 {+\cdot} S_2 = S_2$. The theorem is a consequence of (21) and (3).

(24) If S_1 is a rank sequence and S_2 is a rank sequence, then $S_1 {+\cdot} S_2$ is a rank sequence.

(25) If S is a rank sequence and $m \leqslant n$ and $n \in \operatorname{dom} S$, then $S(m) \subseteq S(n)$. The theorem is a consequence of (19) and (13).

(26) Let us consider a finite sequence F. Suppose F is a rank sequence. Then there exists a finite sequence S such that

(i) $\operatorname{len} S = 1 + \operatorname{len} F$, and

(ii) S is a rank sequence, and

(iii) for every natural number n such that $n \in \operatorname{dom} S$ holds $S(n) = \mathrm{ND_{SS}}(V, A \cup (\langle A \rangle \frown F)(n))$.

Proof: Set $G = \langle A \rangle ^\frown F$. Define $\mathcal{F}(\text{object}) = \mathrm{ND_{SS}}(V, A \cup G(\$_1))$. Consider S being a finite sequence such that $\operatorname{len} S = \operatorname{len} G$ and for every natural number n such that $n \in \operatorname{dom} S$ holds $S(n) = \mathcal{F}(n)$. For every natural number n such that $n \in \operatorname{dom} F$ holds $G(n+1) = F(n)$. S is a rank sequence by (1), [17, (20)]. □

(27) $\langle \mathrm{ND_{SS}}(V, A) \rangle$ is a rank sequence.

(28) $\langle \mathrm{ND_{SS}}(V, A), \mathrm{ND_{SS}}(V, A \cup \mathrm{ND_{SS}}(V, A)) \rangle$ is a rank sequence. The theorem is a consequence of (27) and (17).

(29) $\langle \mathrm{ND_{SS}}(V, A), \mathrm{ND_{SS}}(V, A \cup \mathrm{ND_{SS}}(V, A)), \mathrm{ND_{SS}}(V, A \cup \mathrm{ND_{SS}}(V, A \cup \mathrm{ND_{SS}}(V, A))) \rangle$ is a rank sequence. The theorem is a consequence of (17) and (28).

Let us consider V and A.

A non-atomic nominative data of V and A is a function and is defined by

(Def. 5) there exists a finite sequence S such that S is a rank sequence and $it \in \bigcup S$.

From now on D, D_1, D_2 denote non-atomic nominative data of V and A.

Now we state the propositions:

(30) $\emptyset$ is a non-atomic nominative data of V and A. The theorem is a consequence of (27).

(31) $D \in \bigcup \mathrm{FND_{SC}}(V, A)$.

(32) Let us consider a set d. If $d \subseteq D$, then d is a non-atomic nominative data of V and A. The theorem is a consequence of (4).

(33) There exists a natural number n such that D is a nominative data with simple names from V and simple values from $A \cup (\mathrm{FND_{SC}}(V, A))(n)$. The theorem is a consequence of (19) and (4).

Let us consider V and A. Note that every non-atomic nominative data of V and A is finite.

Now we state the propositions:

(34) If $D_1 \approx D_2$ and S is a rank sequence and $D_1, D_2 \in S(m)$, then $D_1 \cup D_2 \in S(m)$. The theorem is a consequence of (4) and (8).

(35) If $D_1 \approx D_2$ and S_2 is a rank sequence and $S_1 \subseteq S_2$ and $D_1 \in \bigcup S_1$ and $D_2 \in \bigcup S_2$, then $D_1 \cup D_2 \in \bigcup S_2$. The theorem is a consequence of (25) and (34).

(36) If $D_1 \approx D_2$, then $D_1 \cup D_2$ is a non-atomic nominative data of V and A. The theorem is a consequence of (22) and (35).

(37) If $D_1 \approx D_2$, then $D_1 {+\cdot} D_2$ is a non-atomic nominative data of V and A. The theorem is a consequence of (36).

Let us consider V and A. A nominative data with simple names from V and complex values from A is a set and is defined by

(Def. 6) *it* $\in A$ or *it* is a non-atomic nominative data of V and A.

The functor $\mathrm{ND}_{\mathrm{SC}}(V, A)$ yielding a set is defined by the term

(Def. 7) the set of all D where D is a nominative data with simple names from V and complex values from A.

Let us observe that $\mathrm{ND}_{\mathrm{SC}}(V, A)$ is non empty. Now we state the propositions:

(38) $\emptyset \in \mathrm{ND}_{\mathrm{SC}}(V, A)$. The theorem is a consequence of (30).

(39) If $x \in \mathrm{ND}_{\mathrm{SC}}(V, A)$, then x is a nominative data with simple names from V and complex values from A.

(40) $\mathrm{ND}_{\mathrm{SC}}(V, A) = \bigcup \mathrm{FND}_{\mathrm{SC}}(V, A)$. The theorem is a consequence of (39), (11), (31), (4), and (18).

(41) $D \in \mathrm{ND}_{\mathrm{SC}}(V, A)$.

(42) If $D \notin A$, then $D \in \mathrm{ND}_{\mathrm{SC}}(V, A) \setminus A$. The theorem is a consequence of (41).

(43) If $x \in \mathrm{ND}_{\mathrm{SC}}(V, A) \setminus A$, then x is a non-atomic nominative data of V and A.

(44) Let us consider a nominative data D with simple names from V and complex values from A. Then $D \in \bigcup \mathrm{FND}_{\mathrm{SC}}(V, A)$. The theorem is a consequence of (11) and (31).

3. Examples of Simple-Named Complex-Valued Nominative Data

Let us consider v and a. The functor $\mathrm{ND}(v, a)$ yielding a set is defined by the term

(Def. 8) $v \dot{\longmapsto} a$.

Observe that $\mathrm{ND}(v, a)$ is function-like and relation-like.

Now we state the propositions:

(45) If $v \in V$ and $a \in A$, then $\mathrm{ND}(v, a) \in \mathrm{ND}_{\mathrm{SS}}(V, A)$.

(46) If $v \in V$ and $a \in A$, then $\mathrm{ND}(v, a)$ is a non-atomic nominative data of V and A. The theorem is a consequence of (27) and (45).

Let V, A be non empty sets, v be an element of V, and a be an element of A. Observe that the functor $\mathrm{ND}(v, a)$ yields a non-atomic nominative data of V and A. Now we state the proposition:

(47) If $v \in V$ and $a \in A$, then $\mathrm{ND}(v, a)$ is a nominative data with simple names from V and complex values from A. The theorem is a consequence of (46).

Let us consider v, v_1, and a_1. The functor $\mathrm{ND}(v, v_1, a_1)$ yielding a set is defined by the term

(Def. 9) $v \dot{\longmapsto} (v_1 \dot{\longmapsto} a_1)$.

Note that $\mathrm{ND}(v, v_1, a_1)$ is function-like and relation-like.

Now we state the propositions:

(48) If $\{v, v_1\} \subseteq V$ and $a_1 \in A$, then $\mathrm{ND}(v, v_1, a_1) \in \mathrm{ND}_{\mathrm{SS}}(V, A \cup \mathrm{ND}_{\mathrm{SS}}(V, A))$.

(49) If $\{v, v_1\} \subseteq V$ and $a_1 \in A$, then $\mathrm{ND}(v, v_1, a_1)$ is a non-atomic nominative data of V and A. The theorem is a consequence of (28) and (48).

Let V, A be non empty sets, v, v_1 be elements of V, and a be an element of A. Let us note that the functor $\mathrm{ND}(v, v_1, a)$ yields a non-atomic nominative data of V and A. Now we state the proposition:

(50) If $\{v, v_1\} \subseteq V$ and $a_1 \in A$, then $\mathrm{ND}(v, v_1, a_1)$ is a nominative data with simple names from V and complex values from A. The theorem is a consequence of (49).

Let us consider v, v_1, a, and a_1. The functor $\mathrm{ND}(v, v_1, a, a_1)$ yielding a set is defined by the term

(Def. 10) $[v \longmapsto a, v_1 \longmapsto a_1]$.

Let us note that $\mathrm{ND}(v, v_1, a, a_1)$ is function-like and relation-like.

Now we state the propositions:

(51) If $\{v, v_1\} \subseteq V$ and $\{a, a_1\} \subseteq A$, then $\mathrm{ND}(v, v_1, a, a_1) \in \mathrm{ND}_{\mathrm{SS}}(V, A)$. The theorem is a consequence of (45) and (8).

(52) If $\{v, v_1\} \subseteq V$ and $\{a, a_1\} \subseteq A$, then $\mathrm{ND}(v, v_1, a, a_1)$ is a non-atomic nominative data of V and A. The theorem is a consequence of (27) and (51).

Let V, A be non empty sets, v, v_1 be elements of V, and a, a_1 be elements of A. Let us observe that the functor $\mathrm{ND}(v, v_1, a, a_1)$ yields a non-atomic nominative data of V and A. Now we state the proposition:

(53) Suppose $\{v, v_1\} \subseteq V$ and $\{a, a_1\} \subseteq A$. Then $\mathrm{ND}(v, v_1, a, a_1)$ is a nominative data with simple names from V and complex values from A. The theorem is a consequence of (52).

Let us consider v, v_1, v_2, a, and a_1. The functor $\mathrm{ND}(v, v_1, v_2, a, a_1)$ yielding a set is defined by the term

(Def. 11) $[v \longmapsto a, v_1 \longmapsto v_2 \dot{\longmapsto} a_1]$.

Let us note that $\mathrm{ND}(v, v_1, v_2, a, a_1)$ is function-like and relation-like.

Now we state the propositions:

(54) Suppose $\{v, v_1, v_2\} \subseteq V$ and $\{a, a_1\} \subseteq A$. Then $\mathrm{ND}(v, v_1, v_2, a, a_1) \in \mathrm{ND}_{\mathrm{SS}}(V, A \cup \mathrm{ND}_{\mathrm{SS}}(V, A))$.
Proof: Set $g = \mathrm{ND}(v, v_1, v_2, a, a_1)$. $\mathrm{rng}\, g \subseteq A \cup \mathrm{ND}_{\mathrm{SS}}(V, A)$. $\square$

(55) If $\{v, v_1, v_2\} \subseteq V$ and $\{a, a_1\} \subseteq A$, then $\mathrm{ND}(v, v_1, v_2, a, a_1)$ is a non-atomic nominative data of V and A. The theorem is a consequence of (54) and (28).

Let V, A be non empty sets, v, v_1, v_2 be elements of V, and a, a_1 be elements of A. One can check that the functor $\mathrm{ND}(v, v_1, v_2, a, a_1)$ yields a non-atomic nominative data of V and A. Now we state the propositions:

(56) Suppose $\{v, v_1, v_2\} \subseteq V$ and $\{a, a_1\} \subseteq A$. Then $\mathrm{ND}(v, v_1, v_2, a, a_1)$ is a nominative data with simple names from V and complex values from A. The theorem is a consequence of (55).

(57) $\langle x \rangle$ is a non-atomic nominative data of $\{1\}$ and $\{x\}$.
Proof: $\langle x \rangle \in \mathrm{ND}_{\mathrm{SS}}(\{1\}, \{x\})$. $\square$

4. Operations on Simple-Named Complex-Valued Nominative Data

Let us consider V, A, v, and D. Assume $v \in \mathrm{dom}\, D$. The functor $v \Rightarrow_a D$ yielding a nominative data with simple names from V and complex values from A is defined by the term

(Def. 12) $D(v)$.

Let v, D be objects. Assume D is a nominative data with simple names from V and complex values from A. Assume $v \in V$. The functor $\Rightarrow v(D)$ yielding a non-atomic nominative data of V and A is defined by the term

(Def. 13) $v \dot{\longmapsto} D$.

Let a be an object and f be a V-valued finite sequence. Assume $\mathrm{len}\, f > 0$. The functor $\Rightarrow(V, A, f, a)$ yielding a finite sequence is defined by

(Def. 14) $\mathrm{len}\, it = \mathrm{len}\, f$ and $it(1) = \Rightarrow(f(\mathrm{len}\, f))(a)$ and for every natural number n such that $1 \leqslant n < \mathrm{len}\, it$ holds $it(n+1) = \Rightarrow(f(\mathrm{len}\, f - n))(it(n))$.

Now we state the proposition:

(58) Let us consider a V-valued finite sequence f. Suppose $1 \leqslant n \leqslant \mathrm{len}\, f$. Then $(\Rightarrow(V, A, f, a))(n)$ is a non-atomic nominative data of V and A.

Let us consider V and A. Let f be a V-valued finite sequence and a be an object. The functor $\Rightarrow f(a)$ yielding a set is defined by the term

(Def. 15) $(\Rightarrow(V, A, f, a))(\text{len} \Rightarrow(V, A, f, a))$.

Now we state the propositions:

(59) Let us consider a V-valued finite sequence f. Suppose $\text{len} f > 0$. Then $\Rightarrow f(a)$ is a non-atomic nominative data of V and A. The theorem is a consequence of (58).

(60) Let us consider a non empty set V, and an element v of V. Then $\Rightarrow\langle v\rangle(a) = \Rightarrow v(a)$.

(61) Let us consider a non empty set V, and elements v_1, v_2 of V. Suppose a is a nominative data with simple names from V and complex values from A. Then $\Rightarrow\langle v_1, v_2\rangle(a) = v_1 \dot{\longmapsto} (v_2 \dot{\longmapsto} a)$. The theorem is a consequence of (58).

(62) Let us consider a nominative data D with simple names from V and complex values from A. If $v \in V$, then $v \Rightarrow_a \Rightarrow v(D) = D$.

(63) If $v \in \text{dom}\, D$, then $\Rightarrow v(v \Rightarrow_a D) = v \dot{\longmapsto} D(v)$. The theorem is a consequence of (33).

Let us consider V and A. Let d_1, d_2 be objects. Assume d_1 is a nominative data with simple names from V and complex values from A and d_2 is a nominative data with simple names from V and complex values from A.

The functor $d_1 \nabla_a d_2$ yielding a nominative data with simple names from V and complex values from A is defined by

(Def. 16) (i) there exist functions f_1, f_2 such that $f_1 = d_1$ and $f_2 = d_2$ and $it = f_2 \cup f_1 \restriction (\text{dom}\, f_1 \setminus \text{dom}\, f_2)$, **if** $d_1 \notin A$ and $d_2 \notin A$,

(ii) $it = d_2$, **otherwise**.

Let d_1, d_2, v be objects.

The functor $d_1 \nabla_a^v d_2$ yielding a nominative data with simple names from V and complex values from A is defined by the term

(Def. 17) $d_1 \nabla_a (\Rightarrow v(d_2))$.

Now we state the propositions:

(64) If $D_1 \notin A$ and $D_2 \notin A$, then $D_1 \nabla_a D_2 = D_2 \cup D_1 \restriction (\text{dom}\, D_1 \setminus \text{dom}\, D_2)$.

(65) If $D_1 \notin A$ and $D_2 \notin A$ and $\text{dom}\, D_1 \subseteq \text{dom}\, D_2$, then $D_1 \nabla_a D_2 = D_2$. The theorem is a consequence of (64).

(66) If $D \notin A$, then $D \nabla_a D = D$. The theorem is a consequence of (65).

(67) Suppose $v \in V$ and $v \dot{\longmapsto} a_1 \notin A$ and $v \dot{\longmapsto} a_2 \notin A$ and a_1 is a nominative data with simple names from V and complex values from A and a_2 is a nominative data with simple names from V and complex values from A. Then $(v \dot{\longmapsto} a_1) \nabla_a (v \dot{\longmapsto} a_2) = v \dot{\longmapsto} a_2$. The theorem is a consequence of (65).

(68) Suppose $\{v_1, v_2\} \subseteq V$ and $v_1 \neq v_2$ and $v_1 \dot{\longmapsto} a_1 \notin A$ and $v_2 \dot{\longmapsto} a_2 \notin A$ and a_1 is a nominative data with simple names from V and complex values from A and a_2 is a nominative data with simple names from V and complex values from A. Then $(v_1 \dot{\longmapsto} a_1) \nabla_a (v_2 \dot{\longmapsto} a_2) = [v_2 \longmapsto a_2, v_1 \longmapsto a_1]$. The theorem is a consequence of (64).

(69) Suppose $\{v, v_1, v_2\} \subseteq V$ and $v \neq v_1$ and $a_2 \in A$ and $v_1 \dot{\longmapsto} a_1 \notin A$ and $v \dot{\longmapsto} (v_2 \dot{\longmapsto} a_2) \notin A$ and a_1 is a nominative data with simple names from V and complex values from A. Then $(v_1 \dot{\longmapsto} a_1) \nabla_a^v (v_2 \dot{\longmapsto} a_2) = [v \longmapsto v_2 \dot{\longmapsto} a_2, v_1 \longmapsto a_1]$. The theorem is a consequence of (47) and (68).

Let us consider V, A, and v. The functor $v \Rightarrow_a$ yielding a partial function from $\mathrm{ND_{SC}}(V, A)$ to $\mathrm{ND_{SC}}(V, A)$ is defined by

(Def. 18) $\mathrm{dom}\, it = \{d$, where d is a non-atomic nominative data of V and $A : v \in \mathrm{dom}\, d\}$ and for every non-atomic nominative data D of V and A such that $D \in \mathrm{dom}\, it$ holds $it(D) = v \Rightarrow_a D$.

The functor $\Rightarrow v$ yielding a function from $\mathrm{ND_{SC}}(V, A)$ into $\mathrm{ND_{SC}}(V, A)$ is defined by

(Def. 19) for every nominative data D with simple names from V and complex values from A, $it(D) = \Rightarrow v(D)$.

The functor ∇_a^v yielding a partial function from $\mathrm{ND_{SC}}(V, A) \times \mathrm{ND_{SC}}(V, A)$ to $\mathrm{ND_{SC}}(V, A)$ is defined by

(Def. 20) $\mathrm{dom}\, it = (\mathrm{ND_{SC}}(V, A) \setminus A) \times \mathrm{ND_{SC}}(V, A)$ and for every non-atomic nominative data d_1 of V and A and for every object d_2 such that $d_1 \notin A$ and $d_2 \in \mathrm{ND_{SC}}(V, A)$ holds $it(\langle d_1, d_2 \rangle) = d_1 \nabla_a^v d_2$.

References

[1] R.W. Floyd. Assigning meanings to programs. *Mathematical aspects of computer science*, 19(19–32), 1967.

[2] C.A.R. Hoare. An axiomatic basis for computer programming. *Commun. ACM*, 12(10): 576–580, 1969.

[3] Ievgen Ivanov. On the underapproximation of reach sets of abstract continuous-time systems. In Erika Ábrahám and Sergiy Bogomolov, editors, *Proceedings 3rd International Workshop on Symbolic and Numerical Methods for Reachability Analysis, SNR@ETAPS 2017, Uppsala, Sweden, 22nd April 2017*, volume 247 of *EPTCS*, pages 46–51, 2017. doi:10.4204/EPTCS.247.4.

[4] Ievgen Ivanov. On representations of abstract systems with partial inputs and outputs. In T. V. Gopal, Manindra Agrawal, Angsheng Li, and S. Barry Cooper, editors, *Theory and Applications of Models of Computation – 11th Annual Conference, TAMC 2014, Chennai, India, April 11–13, 2014. Proceedings*, volume 8402 of *Lecture Notes in Computer Science*, pages 104–123. Springer, 2014. ISBN 978-3-319-06088-0. doi:10.1007/978-3-319-06089-7_8.

[5] Ievgen Ivanov. On local characterization of global timed bisimulation for abstract continuous-time systems. In Ichiro Hasuo, editor, *Coalgebraic Methods in Computer Science – 13th IFIP WG 1.3 International Workshop, CMCS 2016, Colocated with ETAPS 2016, Eindhoven, The Netherlands, April 2–3, 2016, Revised Selected Papers*, volume

9608 of *Lecture Notes in Computer Science*, pages 216–234. Springer, 2016. ISBN 978-3-319-40369-4. doi:10.1007/978-3-319-40370-0_13.

[6] Ievgen Ivanov, Mykola Nikitchenko, and Uri Abraham. On a decidable formal theory for abstract continuous-time dynamical systems. In Vadim Ermolayev, Heinrich C. Mayr, Mykola Nikitchenko, Aleksander Spivakovsky, and Grygoriy Zholtkevych, editors, *Information and Communication Technologies in Education, Research, and Industrial Applications*, volume 469 of *Communications in Computer and Information Science*, pages 78–99. Springer International Publishing, 2014. ISBN 978-3-319-13205-1. doi:10.1007/978-3-319-13206-8_4.

[7] Ievgen Ivanov, Mykola Nikitchenko, and Uri Abraham. Event-based proof of the mutual exclusion property of Peterson's algorithm. *Formalized Mathematics*, 23(4):325–331, 2015. doi:10.1515/forma-2015-0026.

[8] Ievgen Ivanov, Mykola Nikitchenko, and Volodymyr G. Skobelev. Proving properties of programs on hierarchical nominative data. *The Computer Science Journal of Moldova*, 24(3):371–398, 2016.

[9] Artur Kornilowicz, Andrii Kryvolap, Mykola Nikitchenko, and Ievgen Ivanov. Formalization of the algebra of nominative data in Mizar. In Maria Ganzha, Leszek A. Maciaszek, and Marcin Paprzycki, editors, *Proceedings of the 2017 Federated Conference on Computer Science and Information Systems, FedCSIS 2017, Prague, Czech Republic, September 3–6, 2017.*, pages 237–244, 2017. ISBN 978-83-946253-7-5. doi:10.15439/2017F301.

[10] Artur Kornilowicz, Andrii Kryvolap, Mykola Nikitchenko, and Ievgen Ivanov. Formalization of the nominative algorithmic algebra in Mizar. In Leszek Borzemski, Jerzy Świątek, and Zofia Wilimowska, editors, *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology – ISAT 2017 – Part II, Szklarska Poręba, Poland, September 17–19, 2017*, volume 656 of *Advances in Intelligent Systems and Computing*, pages 176–186. Springer, 2017. ISBN 978-3-319-67228-1. doi:10.1007/978-3-319-67229-8_16.

[11] Artur Korniłowicz, Andrii Kryvolap, Mykola Nikitchenko, and Ievgen Ivanov. An approach to formalization of an extension of Floyd-Hoare logic. In Vadim Ermolayev, Nick Bassiliades, Hans-Georg Fill, Vitaliy Yakovyna, Heinrich C. Mayr, Vyacheslav Kharchenko, Vladimir Peschanenko, Mariya Shyshkina, Mykola Nikitchenko, and Aleksander Spivakovsky, editors, *Proceedings of the 13th International Conference on ICT in Education, Research and Industrial Applications. Integration, Harmonization and Knowledge Transfer, Kyiv, Ukraine, May 15–18, 2017*, volume 1844 of *CEUR Workshop Proceedings*, pages 504–523. CEUR-WS.org, 2017.

[12] Andrii Kryvolap, Mykola Nikitchenko, and Wolfgang Schreiner. *Extending Floyd-Hoare Logic for Partial Pre- and Postconditions*, pages 355–378. Springer International Publishing, 2013. ISBN 978-3-319-03998-5. doi:10.1007/978-3-319-03998-5_18.

[13] Mykola Nikitchenko and Andrii Kryvolap. Properties of inference systems for Floyd-Hoare logic with partial predicates. *Acta Electrotechnica et Informatica*, 13(4):70–78, 2013. doi:10.15546/aeei-2013-0052.

[14] Mykola S. Nikitchenko. Composition-nominative aspects of address programming. *Cybernetics and Systems Analysis*, 45(864), 2009. doi:10.1007/s10559-009-9159-4. (Translated from Kibernetika i Sistemnyi Analiz, No. 6, pp. 24–35, November–December 2009).

[15] Nikolaj S. Nikitchenko. A composition nominative approach to program semantics. Technical Report IT-TR 1998-020, Department of Information Technology, Technical University of Denmark, 1998.

[16] N.S. Nikitchenko. Abstract computability of non-deterministic programs over various data structures. In Zamulin A.V. Bjorner D., Broy M., editor, *Perspectives of System Informatics: 4th International Andrei Ershov Memorial Conference, PSI 2001*, volume 2244 of *Lecture Notes in Computer Science*, pages 468–481. Springer, Berlin, Heidelberg, 2001. doi:10.1007/3-540-45575-2_45.

[17] Robin Nittka. Conway's games and some of their basic properties. *Formalized Mathematics*, 19(**2**):73–81, 2011. doi:10.2478/v10037-011-0013-6.

[18] V.N. Red'ko. Backgrounds of compositional programming. *Programming [in Russian]*, (3):3–13, 1979.

[19] Volodymyr G. Skobelev, Mykola Nikitchenko, and Ievgen Ivanov. On algebraic properties of nominative data and functions. In Vadim Ermolayev, Heinrich C. Mayr, Mykola Nikitchenko, Aleksander Spivakovsky, and Grygoriy Zholtkevych, editors, *Information and*

Communication Technologies in Education, Research, and Industrial Applications – 10th International Conference, ICTERI 2014, Kherson, Ukraine, June 9–12, 2014, Revised Selected Papers, volume 469 of *Communications in Computer and Information Science*, pages 117–138. Springer, 2014. ISBN 978-3-319-13205-1. doi:10.1007/978-3-319-13206-8_6.

Received August 30, 2017

The English version of this volume of Formalized Mathematics was financed under agreement 548/P-DUN/2016 with the funds from the Polish Minister of Science and Higher Education for the dissemination of science.