Computer Science Journal of Moldova, vol.**23**, no.3(69), 2015

Set-theoretic Analysis of Nominative Data*

Volodymyr G. Skobelev, Ievgen Ivanov, Mykola Nikitchenko

Abstract

In the paper we investigate the notion of nominative data that can be considered as a general mathematical model of data used in computing systems. The main attention is paid to flat nominative data called nominative sets. The structure of the partially-ordered set of nominative sets is investigated in terms of set theory, lattice theory, and algebraic systems theory. To achieve this aim the correct transferring of basic set-theoretic operations to nominative sets is proposed. We investigate a lower semilattice of nominative sets in terms of lower and upper cones, closed and maximal closed intervals of nominative sets. The obtained results can be used in formal software development.

Keywords: nominative set, nominative data, set theory, lattice theory, algebraic system, lower semilattice, lower and upper cones, closed intervals.

1 Introduction

The significance of the problem of elaborating the theory of programming and linking it with software development practice was recognized by many researchers [1–6], and in particular, it was mentioned as one of the grand challenges in computing by T. Hoare in his influential talk "The Verifying Compiler: a Grand Challenge for computing research of the 21st century" [7]: *"To build the link between the theory of programming and the products of software engineering practice is still a grand challenge for scientific research in computing; the development*

* This work was partially supported by the research project No. 11BF015-02 "Formal specifications and methods of development of reliable software systems", Taras Shevchenko National University of Kyiv, Ukraine.

of a verifying compiler is an essential tool and target for this research; it will make the results of the research available to software engineers of the future, and so contribute to the quality and reliability of all the programs that they produce."

Currently there exist various approaches that try to deal with this global problem [2–6,8–11], each of which has its own methodology. This paper advocates the so-called *composition-nominative approach* to program formalization [12–14]. The starting point of this approach is the view of software as a data processor that must deal with many forms of data used in computing systems (e.g. arrays, lists, dictionaries, tables, trees, etc.). Thus for solving the mentioned grand problem, firstly one needs to develop a unified, adequate, and tractable theoretical model of data that can serve as a basis for building adequate semantic models of programming language constructs and programs.

The unified data model proposed in the composition-nominative approach is called *nominative data* [12, 15, 16] and is based on the name-value relation. In the simplest case one can view a nominative data as a collection of associations between names and values that can be denoted as $[name_1 \mapsto value_1, name_2 \mapsto value_2, ...]$, or, more formally, as a partial function from the set of all possible names to the set of all possible values.

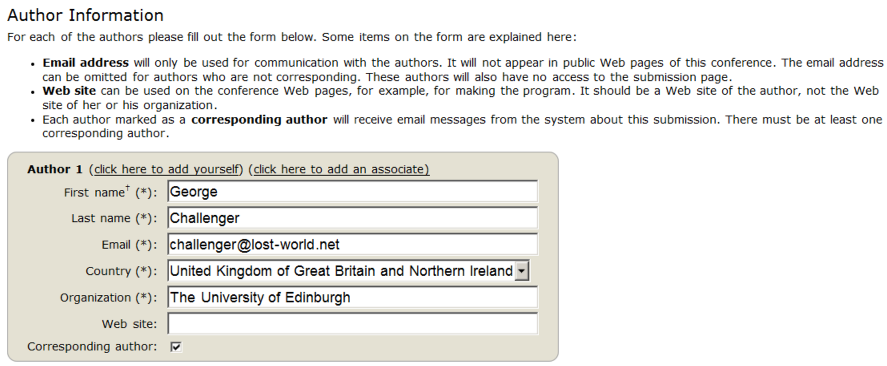
The following simple example illustrates this notion. In most web applications (e.g. online reservation systems, online stores, search engines, etc.) the primary method of obtaining information from a user is based on web forms. A typical web form consists of several named mandatory and optional fields, e.g. Fig. 1.

A natural mathematical model of a user-supplied data in this case is a partial function that maps field names to the corresponding filled values, assuming that this function is undefined on all unfilled fields. This partial function is a nominative data of a particular type called a *nominative set*. For example, for a web form with mandatory fields $FirstName$, $LastName$, $Email$, $Country$, $Organization$ and an optional field $WebSite$, a data instance provided by the user can be modeled as a nominative set (partial function)

$$d : \{FirstName, LastName, Email, Country,$$

$$Organization, WebSite\} \to A,$$

where A denotes the set of all possible field values (strings).


Author Information

For each of the authors please fill out the form below. Some items on the form are explained here:

- **Email address** will only be used for communication with the authors. It will not appear in public Web pages of this conference. The email address can be omitted for authors who are not corresponding. These authors will also have no access to the submission page.
- **Web site** can be used on the conference Web pages, for example, for making the program. It should be a Web site of the author, not the Web site of her or his organization.
- Each author marked as a **corresponding author** will receive email messages from the system about this submission. There must be at least one corresponding author.

Author 1 (click here to add yourself) (click here to add an associate)

First name† (*): George
Last name (*): Challenger
Email (*): challenger@lost-world.net
Country (*): United Kingdom of Great Britain and Northern Ireland
Organization (*): The University of Edinburgh
Web site:
Corresponding author:

Figure 1. A screenshot of the EasyChair conference management system [17] (`http://www.easychair.org`). An example of a web form with mandatory and optional fields.

If $d(WebSite)$ is undefined, this means that the user left the corresponding field unfilled. The values of such a function can be conveniently specified using a notation of the form

$$[FieldName1 \mapsto FieldValue1, FieldName2 \mapsto FieldValue2, ...],$$

e.g.,

$$[FirstName \mapsto \mathit{George}, LastName \mapsto \mathit{Challenger},$$
$$\mathit{Email} \mapsto \mathit{challenger@lost\text{-}world.net}, Country \mapsto \mathit{the}\ UK,$$
$$Organization \mapsto \mathit{The\ University\ of\ Edinburgh}].$$

Clearly, development of the composition-nominative approach requires refinement of the idea of data as name-value relations. Very basic questions concerning the nature of name-value relations already give hints concerning the possible directions of this refinement:

- Are the names unstructured (simple) or structured (complex), e.g. strings in a certain alphabet ?

- Are the values unstructured (simple) or structured (complex), e.g. can values be nominative data themselves ?

- Is only direct naming possible (i.e. values cannot be names) ? Or indirect naming is also allowed (values can be names) ?

Different answers to these questions lead to different types of nominative data (TND1–TND8) [15] illustrated in Fig. 2.

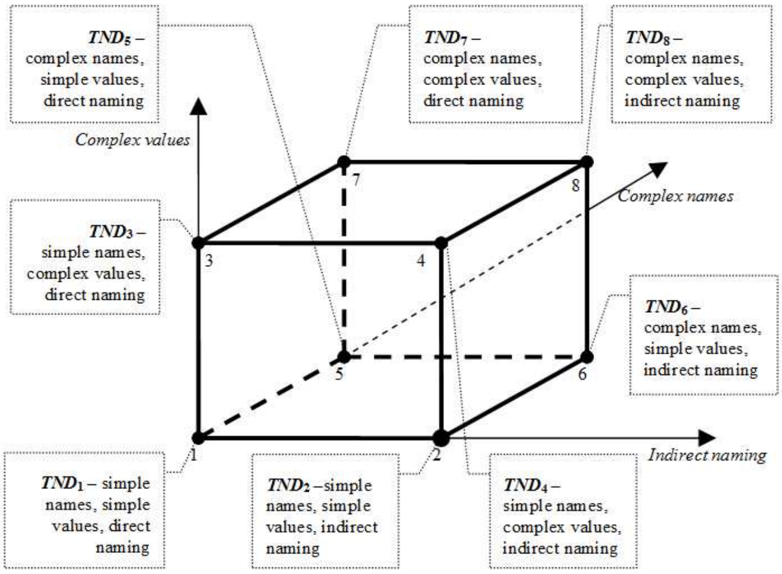


Figure 2. Types of nominative data

Although the idea behind nominative data is intuitively clear, the absence of unique answers to even such basic questions makes the process of their formalization and application to program semantics and the problems of software specification and verification non-trivial. In fact, one has to consider and investigate many formalizations of nominative data and study their suitability for different classes of problems. Another complication is that algebraic structures that arise from sets of nominative data of different types turn out to be different from structures traditionally considered in algebra and computer science (e.g. rings, lattices, boolean algebras, etc.) and currently remain virtually unstudied.

To deal effectively with complexity of data and program formalization, composition-nominative approach proposes the following principles [12]:

- *Development principle* (from abstract to concrete): the process of development of program notions must start from abstract understanding and proceed to more concrete considerations.

- *Principle of integrity of intensional and extensional aspects*: program notions should be presented in the integrity of their intensional and extensional aspects, but the intensional aspects play a leading role.

- *Principle of priority of semantics over syntax*: semantic and syntactical aspects of programs should be first studied separately, and then in their integrity in which semantic aspects prevail over syntactical aspects.

- *Compositionality principle*: programs are constructed from simpler programs using operations called compositions which represent semantics of programming language constructs.

- *Nominativity principle*: naming relations are the basic ones in constructing data and programs.

These principles are applied for constructing a hierarchy of program models of various levels of abstraction and generality with the general aim of providing a mathematical basis for development of formal methods of analysis and synthesis of reliable software systems.

Above mentioned principles are applied to program formalization as follows:

- Data in computing systems are formalized as specific classes of nominative data. A set of nominative data of a particular type together with the basic operations on these data forms an algebra called a data algebra.

- Programs that operate on data are formalized as partial functions that map nominative data to nominative data, also called (bi-) nominative functions.
- Program combination operators (e.g. sequential execution, branching, cycle, etc.) are formalized as operations (also called compositions) that map (bi-)nominative functions to (bi-)nominative function. A set of programs (modeled as nominative functions) that can be obtained from basic operations on data using compositions together with compositions forms an algebra (program algebra) that represents compositional semantics of a programming language. Proving program properties is done by proving certain facts in a program algebra.

In this paper we will study the basic type of nominative data TND1, or data with unstructured names and unstructured values, also called nominative sets. In particular, we will investigate rich algebraic structures that arise from it.

This paper is organized as follows: in Section 2 we give rigorous definitions of nominative sets and other associated notions; in Section 3 we define the basic operations on nominative sets by analogy with set-theoretic operations and study algebraic systems that arise from these definitions; in Section 4 we investigate a partial ordering on nominative sets and the associated poset using lattice theory [18–20]; in Section 5 we give conclusions.

2 Basic notions

Let V and A be non-empty finite or countable sets of names and data respectively. The set $\mathfrak{F}_{V,A}$ of all V-nominative sets over A is the set of all (possibly, partial) mappings from V to A.

If $|A| = 1$, then $\mathfrak{F}_{V,A}$ can be considered as the set $\mathcal{B}(V)$ of all subsets of the set V, while if $|V| = 1$, then $\mathfrak{F}_{V,A}$ can be considered as the set consisting of the empty set and all 1-element subsets of the set A. Thus in what follows it is supposed that $|V| \geq 2$ and $|A| \geq 2$.

We will deal with the set $\mathfrak{G}_{V,A} = \{graph(f) | f \in \mathfrak{F}_{V,A}\}$, where $graph(f) = \{(v,a) \in \mathrm{Dom} f \times \mathrm{Val} f | f(v) = a\}$.

The following partial ordering can be defined on the set $\mathfrak{F}_{V,A}$:

$$f_1 \preceq f_2 \Leftrightarrow graph(f_1) \subseteq graph(f_2) \;\; (f_1, f_2 \in \mathfrak{F}_{V,A}). \tag{1}$$

The least element of the poset $\mathfrak{F}_{V,A}$ is the V-nominative set $0_{V,A}$ with empty domain, while the set of all maximal elements of the poset $\mathfrak{F}_{V,A}$ is the set $\mathfrak{F}_{V,A}^{(ttl)}$ of all total V-nominative sets.

Since $|A| \geq 2$, for any set of names V ($|V| \geq 2$) the poset $(\mathfrak{F}_{V,A}, \preceq)$ does not have the largest element. Thus this poset is not isomorphic to any Boolean algebra.

We write $f_1 \prec f_2$ ($f_1, f_2 \in \mathfrak{F}_{V,A}$) if and only if $f_1 \preceq f_2$ and $f_1 \neq f_2$. By "$\succeq$" (or, respectively, by "$\succ$") we denote the relation that is an inverse of the relation " $\preceq$ " (or, respectively, of the relation "$\prec$").

3 Algebra of nominative sets

In this section we will transfer the basic set-theoretic operations to the set $\mathfrak{F}_{V,A}$ with the purpose of providing correct operations on V-nominative sets over A with perspectives of application in automation of software development and analysis.

The unary set-theoretic operation of complement of a set cannot be transferred to $\mathfrak{F}_{V,A}$ since this set does not contain the largest element.

Let us transfer binary set-theoretic operations to the set $\mathfrak{F}_{V,A}$.

There are no difficulties with transferring the set-theoretic operations of intersection of two sets "$\cap$" and the difference of two sets "$\backslash$" to the set $\mathfrak{F}_{V,A}$. Indeed, for any $f_1, f_2, f \in \mathfrak{F}_{V,A}$ we can define:

$$f_1 \cap f_2 = f \Leftrightarrow graph(f_1) \cap graph(f_2) = graph(f), \tag{2}$$
$$f_1 \backslash f_2 = f \Leftrightarrow graph(f_1) \backslash graph(f_2) = graph(f).$$

Obviously, for any $f_1, f_2 \in \mathfrak{F}_{V,A}$ the following formulas hold:

$$\mathrm{Dom}(f_1 \cap f_2) \subseteq \mathrm{Dom} f_1 \cap \mathrm{Dom} f_2,$$

$$f_1|_X \cap f_2|_Y = (f_1 \cap f_2)|_{X\cap Y} \ (X, Y \subseteq V),$$

$$\mathrm{Dom} f_1 \backslash \mathrm{Dom} f_2 \subseteq \mathrm{Dom}(f_1 \backslash f_2) \subseteq \mathrm{Dom} f_1.$$

The following two propositions are true:

Proposition 1. *Algebraic system* $(\mathfrak{F}_{V,A}, \cap)$ *is a commutative semi-group without neutral element, but with zero element which is the* V*-nominative set* $0_{V,A}$ *with empty domain.*

Proposition 2. *Algebraic system* $(\mathfrak{F}_{V,A}, \backslash)$ *is a non-commutative non-associative magma in which the* V*-nominative set* $0_{V,A}$ *with the empty domain is both the right identity element and the left zero element.*

A different situation occurs with transferring of the operations of the union of two sets "$\cup$" and of the symmetric difference of two sets " $\oplus$ " to the set $\mathfrak{F}_{V,A}$. Indeed, for any $f_1, f_2 \in \mathfrak{F}_{V,A}$ we get:

$$graph(f_1) \cup graph(f_2) \in \mathfrak{G}_{V,A} \Leftrightarrow f_1|_{\mathrm{Dom} f_1 \cap \mathrm{Dom} f_2} = f_2|_{\mathrm{Dom} f_1 \cap \mathrm{Dom} f_2},$$

$$graph(f_1) \oplus graph(f_2) \in \mathfrak{G}_{V,A} \Leftrightarrow f_1|_{\mathrm{Dom} f_1 \cap \mathrm{Dom} f_2} = f_2|_{\mathrm{Dom} f_1 \cap \mathrm{Dom} f_2}.$$

Thus the formulas

$$f_1 \cup f_2 = f \Leftrightarrow graph(f_1) \cup graph(f_2) = graph(f) \ \ (f_1, f_2, f \in \mathfrak{F}_{V,A}),$$

$$f_1 \oplus f_2 = f \Leftrightarrow graph(f_1) \oplus graph(f_2) = graph(f) \ \ (f_1, f_2, f \in \mathfrak{F}_{V,A})$$

can define only partial operations on the set $\mathfrak{F}_{V,A}$.

In order to avoid such a situation we transfer the operations "$\cup$" and "$\oplus$" to the set $\mathfrak{F}_{V,A}$ as follows: for any $f_1, f_2, f \in \mathfrak{F}_{V,A}$ we define:

$$f_1 \triangleright f_2 = f \Leftrightarrow graph(f_1) \cup graph(f_2|_{\mathrm{Dom} f_2 \backslash \mathrm{Dom} f_1}) = graph(f)$$

and

$$f_1 \boxplus f_2 = f \Leftrightarrow$$

$$\Leftrightarrow graph(f_1|_{\mathrm{Dom} f_1 \backslash \mathrm{Dom} f_2}) \cup graph(f_2|_{\mathrm{Dom} f_2 \backslash \mathrm{Dom} f_1}) = graph(f).$$

It is worth noting that these two operations are intended to join together any two V-nominative sets over A. The operation " $\triangleright$ " is called overlapping (of the second nominative set by the first one), the operation " $\boxplus$ " can be called the exclusive compound. The following proposition is true:

Proposition 3. *For any $f_1, f_2, f_3 \in \mathfrak{F}_{V,A}$ the following formulas hold:*
(i) $\mathrm{Dom}(f_1 \triangleright f_2) = \mathrm{Dom} f_1 \cup \mathrm{Dom} f_2$;
(ii) $f_1 \preceq f_1 \triangleright f_2$;
(iii) $f_1 \preceq f_2 \Rightarrow f_1 \triangleright f_2 = f_2 \triangleright f_1 = f_2$;
(iv) $(f_1 \triangleright f_2) \cap f_3 \preceq (f_1 \cap f_6) \triangleright (f_2 \cap f_3)$;
(v) $f_3 \cap (f_1 \triangleright f_2) \preceq (f_1 \cap f_3) \triangleright (f_2 \cap f_3)$;
(vi) $(f_1 \cap f_2) \triangleright f_3 \succeq (f_1 \triangleright f_3) \cap (f_2 \triangleright f_3)$;
(vii) $f_1 \triangleright (f_2 \cap f_3) = (f_1 \triangleright f_2) \cap (f_1 \triangleright f_3)$.

It is not difficult to give examples showing that there may be strict inequalities in the formulas (ii), (iv)-(vi).

The following theorem is true:

Theorem 1. *The algebraic system $(\mathfrak{F}_{V,A}, \triangleright)$ is a non-commutative monoid with neutral element which is the V-nominative set $0_{V,A}$ with the empty domain.*

Proposition 1 and Theorem 1 imply that the algebraic system $(\mathfrak{F}_{V,A}, \triangleright, \cap)$ differs from well-known algebraic systems with two binary operations (i.e. a field, a ring, a semi-ring, etc.). Thus the properties of the set of all valid formulas in the algebraic system $(\mathfrak{F}_{V,A}, \triangleright, \cap)$ can substantially differ from the properties of the sets of all valid formulas in standard algebraic systems with two binary operations. The following proposition is true:

Proposition 4. *The algebraic system $(\mathfrak{F}_{V,A}, \boxplus)$ is a commutative semigroup with the neutral element which is the V-nominative set $0_{V,A}$ with empty domain.*

Since $(f_1 \boxplus f_2) \cap f_3 = (f_1 \cap f_3) \boxplus (f_2 \cap f_3)$ for any $f_1, f_2, f_3 \in \mathfrak{F}_{V,A}$, Propositions 1 and 4 imply that the following theorem is true:

Theorem 2. *The algebraic system $(\mathfrak{F}_{V,A}, \cap, \boxplus)$ is a semiring.*

Thus we have defined an algebraic system $(\mathfrak{F}_{V,A}, \mathfrak{O}_{V,A}, \mathfrak{R}_{V,A})$, where $\mathfrak{F}_{V,A}$ is the base, $\mathfrak{O}_{V,A} = \{\cap, \setminus, \triangleright, \boxplus\}$ is the set of operations and $\mathfrak{R}_{V,A} = \{=, \preceq\}$ is the set of relations.

It is worth noting that since the operations $\cap$ and $\boxplus$ are associative and can be naturally extended to any finite (consisting of at least two elements) or infinite sequence of elements of the set $\mathfrak{F}_{V,A}$, so that the notations of the form $\cap_{i\in I} f_i$ and $\boxplus_{i\in I} f_i$ do not cause any misunderstanding.

4 Analysis of the poset $(\mathfrak{F}_{V,A}, \preceq)$ in terms of lattice theory

The formulas (1) and (2) imply that the poset $(\mathfrak{F}_{V,A}, \preceq)$ is a lower semilattice such that $\inf\{f_1, f_2\} = f_1 \cap f_2$ $(f_1, f_2 \in \mathfrak{F}_{V,A})$. Thus, all basic set-theoretic structures defined on lower semilattices can be transferred to the poset $(\mathfrak{F}_{V,A}, \preceq)$. Let us analyze these structures.

For any non-empty set $S \subseteq \mathfrak{F}_{V,A}$ its lower and upper cones are defined, respectively, using the identities

$$S^{\bigtriangledown} = \{f \in \mathfrak{F}_{V,A} | (\forall f_1 \in S)(f \preceq f_1)\},$$

$$S^{\bigtriangleup} = \{f \in \mathfrak{F}_{V,A} | (\forall f_1 \in S)(f \succeq f_1)\}.$$

Lower cones of non-empty subsets of the poset $(\mathfrak{F}_{V,A}, \preceq)$ can be characterized via the following three propositions:

Proposition 5. *For any non-empty subset $S \subseteq \mathfrak{F}_{V,A}$:*

1) the least element of the lower cone $S^{\bigtriangledown}$ is the V-nominative set $0_{V,A}$ with empty domain;

2) the largest element of the lower cone $S^{\bigtriangledown}$ is $\cap_{f\in S} f$.

Proposition 6. *For any non-empty subsets $S_1, S_2 \subseteq \mathfrak{F}_{V,A}$ the following formulas hold:*

(i) $S_1 \subseteq S_2 \Rightarrow S_1^{\bigtriangledown} \supseteq S_2^{\bigtriangledown}$;

(ii) $S_1 \subset S_2 \& \cap_{f_2 \in S_2 \setminus S_1} f_2 \prec \cap_{f_1 \in S_1} f_1 \Rightarrow S_1^{\bigtriangledown} \supset S_2^{\bigtriangledown}$;

(iii) $S_1 \cup S_2 \subseteq \mathfrak{F}_{V,A} \Rightarrow (S_1 \cup S_2)^{\bigtriangledown} = S_1^{\bigtriangledown} \cap S_2^{\bigtriangledown}$.

Proposition 7. *For any $f_1, f_2 \in \mathfrak{F}_{V,A}$ the following formulas hold:*
(i) $\{f_1\}^{\triangledown} \neq \{f_2\}^{\triangledown} \Leftrightarrow f_1 \neq f_2$;
(ii) $\{f_1\}^{\triangledown} \subseteq \{f_2\}^{\triangledown} \Leftrightarrow f_1 \preceq f_2$;
(iii) $\{f_1 \cap f_2\}^{\triangledown} = \{f_1\}^{\triangledown} \cap \{f_2\}^{\triangledown}$;
(iv) $f_2 \not\preceq f_1 \Rightarrow \{f_1 \triangleright f_2\}^{\triangledown} \supseteq \{f_1\}^{\triangledown} \cap \{f_2 \backslash f_1\}^{\triangledown}$;
(v) $f_1|_{\mathrm{Dom} f_1 \cap \mathrm{Dom} f_2} = f_2|_{\mathrm{Dom} f_1 \cap \mathrm{Dom} f_2} \Rightarrow$
$\Rightarrow \{f_1 \triangleright f_2\}^{\triangledown} = \{f_1\}^{\triangledown} \cap \{f_2\}^{\triangledown}$.

Upper cones of non-empty subsets of the poset $(\mathfrak{F}_{V,A}, \preceq)$ can be characterized in the following way:

for any 1-element subset $S = \{f\}$ ($f \in \mathfrak{F}_{V,A}$) the following inequality holds: $S^{\triangle} \neq \emptyset$ (since $f \in \{f\}^{\triangle}$ for any $f \in \mathfrak{F}_{V,A}$).

It is worth to note that $\{0_{V,A}\}^{\triangle} = \mathfrak{F}_{V,A}$. The following proposition is true:

Proposition 8. *For any ($f_1, f_2 \in \mathfrak{F}_{V,A}$) the following formulas hold:*
(i) $\{f_1\}^{\triangle} \neq \{f_2\}^{\triangle} \Leftrightarrow f_1 \neq f_2$;
(ii) $\{f_1\}^{\triangle} \subseteq \{f_2\}^{\triangle} \Leftrightarrow f_1 \preceq f_2$.

The next example illustrates that there exist subsets $S \subseteq \mathfrak{F}_{V,A}$ ($|S| \geq 2$), such that $S^{\triangle} = \emptyset$.

Example 1. *Let $v \in V$ and $a_1, a_2 \in A$ ($a_1 \neq a_2$) be fixed elements. We set $S = \{f_1, f_2\}$, where $f_1, f_2 \in \mathfrak{F}_{V,A}$ are V-nominative sets over A such that $Dom f_1 = Dom f_2 = \{v\}$, $f_1(v) = a_1$ and $f_2(v) = a_2$.*

The formula (1) implies that there does not exist any V-nominative set $f \in \mathfrak{F}_{V,A}$, such that $f_1 \preceq f$ and $f_2 \preceq f$. Thus, $S^{\triangle} = \emptyset$.

Now we extract subsets $S \subseteq \mathfrak{F}_{V,A}$ such that $S^{\triangle} \neq \emptyset$.

We will say that elements $f_1, f_2 \in \mathfrak{F}_{V,A}$ are *compatible*, if the identity $f_1|_{\mathrm{Dom} f_1 \cap \mathrm{Dom} f_2} = f_2|_{\mathrm{Dom} f_1 \cap \mathrm{Dom} f_2}$ holds. It is evident that if elements $f_1, f_2 \in \mathfrak{F}_{V,A}$ are compatible, then the following identity holds

$$graph(f_1 \triangleright f_2) = graph(f_1) \cup graph(f_2).$$

Thus we get that for any compatible elements $f_1, f_2 \in \mathfrak{F}_{V,A}$ the following identities hold: $f_1 \triangleright f_2 = f_2 \triangleright f_1 = f_1 \cup f_2$ and $\{f_1 \cup f_2\}^{\triangledown} = \{f_1\}^{\triangledown} \cap \{f_2\}^{\triangledown}$.

A non-empty subset $S \subseteq \mathfrak{F}_{V,A}$ will be called *compatible*, if its elements are pairwise compatible. We denote $\mathfrak{S}^{cmp}_{V,A}$ the set of all compatible subsets of the set $\mathfrak{F}_{V,A}$. The following theorem is true:

Theorem 3. *For any set $\mathfrak{F}_{V,A}$ the following formula holds:*

$$(\forall S \subseteq \mathfrak{F}_{V,A})(S \neq \emptyset \Rightarrow (S^{\triangle} \neq \emptyset \Leftrightarrow S \in \mathfrak{S}^{cmp}_{V,A})).$$

Upper cones of elements of the set $\mathfrak{S}^{cmp}_{V,A}$ can be characterized in the following way:

Proposition 9. *For any $S \in \mathfrak{S}^{cmp}_{V,A}$ the following formulas hold:*
(i) g.l.b.$(S^{\triangle}) = f \Leftrightarrow graph(f) = \cup_{f' \in S}\, graph(f')$;
(ii) $(\forall S_1, S_2 \subseteq S)(\emptyset \neq S_1 \subseteq S_2 \Rightarrow S_1^{\triangle} \supseteq S_2^{\triangle})$;
(iii) $(\forall S_1, S_2 \subseteq S)(\emptyset \neq S_1 \subset S_2\&$
$\&\cup_{f_1 \in S_1} graph(f_1) \subset \cup_{f_2 \in S_2 \setminus S_1} graph(f_2) \Rightarrow S_1^{\triangle} \supset S_2^{\triangle})$;
(iv) $(\forall S_1, S_2 \subseteq S)(S_1 \neq \emptyset \& S_2 \neq \emptyset \Rightarrow S_1 \cup S_2 = S_1^{\triangle} \cap S_2^{\triangle})$.

In the poset $(\mathfrak{F}_{V,A}, \preceq)$ any two elements $f_1, f_2 \in \mathfrak{F}_{V,A}$ such that $f_1 \preceq f_2$ define a closed interval

$$[f_1, f_2] = \{f \in \mathfrak{F}_{V,A} | f_1 \preceq f \preceq f_2\}.$$

It is evident that

$$[f_1, f_2] \in \mathfrak{S}^{cmp}_{V,A}\ (f_1, f_2 \in \mathfrak{F}_{V,A}, f_1 \preceq f_2).$$

The following theorem is true:

Theorem 4. *The algebraic system* $([f_1, f_2], \{\cup, \cap\})$ $(f_1, f_2 \in \mathfrak{F}_{V,A}; f_1 \preceq f_2)$ *is a complete distributive lattice.*

On any closed interval

$$[f_1, f_2]\ (f_1, f_2 \in \mathfrak{F}_{V,A}, f_1 \preceq f_2)$$

the following unary operation $\mathsf{C}_{[f_1,f_2]}$ can be defined:

$$\mathsf{C}_{[f_1,f_2]}(f) = f' \Leftrightarrow graph(f') = graph(f_2) \backslash graph(f) \cup graph(f_1).$$

The following theorem is true:

Theorem 5. *The algebraic system*

$$([f_1, f_2], \{\cup, \cap, \mathsf{C}_{[f_1,f_2]}\}) \ \ (f_1, f_2 \in \mathfrak{F}_{V,A}; f_1 \preceq f_2)$$

is a Boolean algebra.

Proof. By Theorem 4 the lattice $([f_1, f_2], \cup, \cap)$ is distributive.

From the definition of $\mathsf{C}_{[f_1,f_2]}(f)$ it follows that for each $f \in [f_1, f_2]$

$$graph(f) \cup graph\mathsf{C}_{[f_1,f_2]}(f) =$$

$$= graph(f) \cup (graph(f_2) \backslash graph(f) \cup graph(f_1)) =$$

$$= (graph(f) \cup graph(f_2) \backslash graph(f)) \cup graph(f_1) =$$

$$= graph(f_2) \cup graph(f_1) = graph(f_2),$$

i.e. $f \cup \mathsf{C}_{[f_1,f_2]}(f) = f_2$, and also,

$$graph(f) \cap graph\mathsf{C}_{[f_1,f_2]}(f) =$$

$$= graph(f) \cap (graph(f_2) \backslash graph(f)) \cup graph(f_1)) =$$

$$= (graph(f) \cap (graph(f_2) \backslash graph(f))) \cup (graphf \cap graph(f_1)) =$$

$$= \emptyset \cup graph(f_1) = graph(f_1),$$

i.e. $f \cap \mathsf{C}_{[f_1,f_2]}(f) = f_1$.

Since $f \cup \mathsf{C}_{[f_1,f_2]}(f) = f_2$ and $f \cap \mathsf{C}_{[f_1,f_2]}(f) = f_1$, the element $\mathsf{C}_{[f_1,f_2]}(f) \in [f_1, f_2]$ is a relative complement of the element $f \in [f_1, f_2]$ in the interval $[f_1, f_2]$.

By the definition, a distributive lattice with a relative complement is a Boolean algebra. □

We will say that a mapping $\varphi : \mathfrak{F}_{V,A} \to \mathfrak{F}_{V,A}$ is isotonic on some set $S \subseteq \mathfrak{F}_{V,A}$ $(S \neq \emptyset)$, if the inequality $\varphi(f_1) \preceq \varphi(f_2)$ holds for all $f_1, f_2 \in S$, such that $f_1 \preceq f_2$. It is evident that if $\varphi : \mathfrak{F}_{V,A} \to \mathfrak{F}_{V,A}$ is any mapping isotonic onto some closed interval

$$[f_1, f_2], (f_1, f_2 \in \mathfrak{F}_{V,A}; f_1 \preceq f_2)$$

and the inclusion $\mathrm{Val}\varphi|_{[f_1,f_2]} \subseteq [f_1, f_2]$ holds, then the mapping $\varphi|_{[f_1,f_2]}$ has at least one fixed point.

Let $f_1^{(i)}, f_2^{(i)} \in \mathfrak{F}_{V,A}$ $(i = 1, 2)$ be elements such that $f_1^{(i)} \preceq f_2^{(i)}$. The closed intervals $[f_1^{(1)}, f_2^{(1)}]$ and $[f_1^{(2)}, f_2^{(2)}]$ are *isomorphic*, if there exists a mapping $\varphi : \mathfrak{F}_{V,A} \to \mathfrak{F}_{V,A}$ such that $\varphi|_{[f_1^{(1)},f_2^{(1)}]}$ is bijection of $[f_1^{(1)}, f_2^{(1)}]$ onto $[f_1^{(2)}, f_2^{(2)}]$ for which the identities

$$\varphi|_{[f_1^{(1)},f_2^{(1)}]}(f' \cup f'') = \varphi|_{[f_1^{(1)},f_2^{(1)}]}(f') \cup \varphi|_{[f_1^{(1)},f_2^{(1)}]}(f'')$$

and

$$\varphi|_{[f_1^{(1)},f_2^{(1)}]}(f' \cap f'') = \varphi|_{[f_1^{(1)},f_2^{(1)}]}(f') \cap \varphi|_{[f_1^{(1)},f_2^{(1)}]}(f'')$$

hold for any $f', f'' \in [f_1^{(1)}, f_2^{(1)}]$.

It is evident that if closed intervals $[f_1^{(1)}, f_2^{(1)}]$ and $[f_1^{(2)}, f_2^{(2)}]$ are isomorphic, then the algebraic systems $([f_1^{(1)}, f_2^{(1)}], \{\cup, \cap\})$ and $([f_1^{(2)}, f_2^{(2)}], \{\cup, \cap\})$, as well as Boolean algebras

$$([f_1^{(1)}, f_2^{(1)}], \{\cup, \cap, \mathsf{C}_{[f_1^{(1)},f_2^{(1)}]}\}),$$

$$([f_1^{(2)}, f_2^{(2)}], \{\cup, \cap, \mathsf{C}_{[f_1^{(2)},f_2^{(2)}]}\})$$

are isomorphic.

The following theorem is true:

Theorem 6. *A closed interval* $[f_1, f_2]$ $(f_1, f_2 \in \mathfrak{F}_{V,A}; f_1 \preceq f_2)$ *is isomorphic to the closed interval* $[0_{V,A}, f_2 \backslash f_1]$.

Proof. Let $\varphi : \mathfrak{F}_{V,A} \to \mathfrak{F}_{V,A}$ be any mapping such that $\varphi(f) = f \backslash f_1$ holds for all $f \in [f_1, f_2]$.

Then $\varphi(f_1) = f_1 \backslash f_1 = 0_{V,A}$, $\varphi(f_2) = f_2 \backslash f_1$, $0_{V,A} \preceq \varphi(f) \preceq f_2 \backslash f_1$ for all $f \in [f_1, f_2]$, i.e. φ maps the interval $[f_1, f_2]$ onto the interval $[0_{V,A}, f_2 \backslash f_1]$.

If $f' \neq f''$ $(f', f'' \in [f_1, f_2])$, then

$$\varphi(f') = f_2 \backslash f' \neq f_2 \backslash f'' = \varphi(f'').$$

Thus, $\varphi|_{[f_1,f_2]}$ is a bijection from $[f_1, f_2]$ onto the interval $[0_{V,A}, f_2\backslash f_1]$.

Since $f_1 \preceq f'$ $f_1 \preceq f''$ for any elements $f', f'' \in [f_1, f_2]$, we have

$$\varphi(f' \cup f'') = (f' \cup f'')\backslash f_1 = f'\backslash f_1 \cup f''\backslash f_1 = \varphi(f') \cup \varphi(f'')$$

and

$$\varphi(f' \cap f'') = (f' \cap f'')\backslash f_1 = f'\backslash f_1 \cap f''\backslash f_1 = \varphi(f') \cap \varphi(f'').$$

Thus $\varphi|_{[f_1,f_2]}$ is an isomorphism from $[f_1, f_2]$ onto $[0_{V,A}, f_2\backslash f_1]$.

□

We will say that a closed interval $[0_{V,A}, f]$ is *maximal* in the poset $(\mathfrak{F}_{V,A}, \preceq)$, if $f \in \mathfrak{F}^{(ttl)}_{V,A}$. The following theorem is true:

Theorem 7. *Any two maximal closed intervals in the poset* $(\mathfrak{F}_{V,A}, \preceq)$ *are isomorphic.*

Proof. Let us fix any elements $f^{(1)}, f^{(2)} \in \mathfrak{F}^{(ttl)}_{V,A}$ $(f^{(1)} \neq f^{(2)})$ and consider maximal intervals $[0_{V,A}, f^{(1)}]$ and $[0_{V,A}, f^{(2)}]$. Let $g = f^{(2)}\backslash f^{(1)}$. Let us define a mapping $\varphi : \mathfrak{F}_{V,A} \to \mathfrak{F}_{V,A}$ as follows:

$$\varphi(f) = g|_{\mathrm{Dom} f} \triangleright f \;\; (f \in \mathfrak{F}_{V,A}).$$

From this it follows that

$$\varphi(0_{V,A}) = g|_{\mathrm{Dom} 0_{V,A}} \triangleright 0_{V,A} = g|_{\emptyset} \triangleright 0_{V,A} = 0_{V,A} \triangleright 0_{V,A} = 0_{V,A},$$

$$\varphi(f^{(1)}) = g|_{\mathrm{Dom} f^{(1)}} \triangleright f^{(1)} = g|_V \triangleright f^{(1)} = g \triangleright f^{(1)} = f^{(2)},$$

and also, $0_{V,A} \preceq \varphi(f) \preceq f^{(2)}$ for all $f \in [0_{V,A}, f^{(1)}]$, i.e. φ maps the interval $[0_{V,A}, f^{(1)}]$ onto the interval $[0_{V,A}, f^{(2)}]$.

Since for any elements $f', f'' \in [0_{V,A}, f^{(1)}]$ $(f' \neq f'')$ the inequality $\mathrm{Dom} f' \neq \mathrm{Dom} f''$ holds, at least one of the inequalities

$$\mathrm{Dom} g \cap \mathrm{Dom} f' \neq \mathrm{Dom} g \cap \mathrm{Dom} f''$$

or

$$\mathrm{Dom} f'\backslash \mathrm{Dom} g \neq \mathrm{Dom} f''\backslash \mathrm{Dom} g$$

holds. From this it follows that if $f' \neq f''$ ($f', f'' \in [0_{V,A}, f^{(1)}]$), then $\varphi(f') \neq \varphi(f'')$.

Thus $\varphi|_{[0_{V,A},f^{(1)}]}$ is a bijection from the interval $[0_{V,A}, f^{(1)}]$ onto the interval $[0_{V,A}, f^{(2)}]$.

For each $f', f'' \in [0_{V,A}, f^{(1)}]$ we have

$$\varphi(f' \cup f'') = g|_{\mathrm{Dom}(f' \cup f'')} \rhd (f' \cup f'') = g|_{\mathrm{Dom}(f' \cup f'')} \rhd f' \cup g|_{\mathrm{Dom}(f' \cup f'')} \rhd f'' =$$

$$= g|_{\mathrm{Dom} f'} \rhd f' \cup g|_{\mathrm{Dom} f''} \rhd f'' = \varphi(f') \cup \varphi(f'').$$

From the definition of φ and Proposition 3(vii) it follows that for each $f', f'' \in [0_{V,A}, f^{(1)}]$,

$$\varphi(f' \cap f'') = g|_{\mathrm{Dom}(f' \cap f'')} \rhd (f' \cap f'') = g|_{\mathrm{Dom}(f' \cap f'')} \rhd f' \cap g|_{\mathrm{Dom}(f' \cap f'')} \rhd f'' =$$

$$= g|_{\mathrm{Dom} f'} \rhd f' \cup g|_{\mathrm{Dom} f''} \rhd f'' = \varphi(f') \cup \varphi(f'').$$

Thus $\varphi|_{[f_1,f_2]}$ is an isomorphism from the interval $[0_{V,A}, f^{(1)}]$ onto the interval $[0_{V,A}, f^{(2)}]$. □

Thus the poset $(\mathfrak{F}_{V,A}, \preceq)$ is a union of the set of overlapping isomorphic maximal closed intervals. At the same time, mappings defining the isomorphism of two intervals differ significantly from each other. Moreover, the structure of the family of these mappings is sufficiently complicated. These circumstances, largely cause high internal complexity of various structures defined on the poset $(\mathfrak{F}_{V,A}, \preceq)$.

5 Conclusions

In the paper a mathematical (algebraic, in essence) formalism intended for investigating the structure of nominative data sets has been proposed. It forms a part of theoretical foundations for unified development of formal methods for automated software design and verification. In this context investigation of algebras of programs over nominative data is essential.

In the given paper we restricted ourselves to basic (flat) nominative data called nominative sets. Nominative sets adequately represent such commonly used data structures as arrays, records, and dictionaries. Hierarchical types of data can naturally represent a much larger set of data structures used in programming, including multidimensional arrays, lists, trees, algebraic data types, etc. The details of such representation are given in [16]. These types of nominative data induce sufficiently rich program algebras. We plan to investigate such algebras in future papers.

References

[1] J. Woodcock, P.G. Larsen, J. Bicarregui, J. Fitzgerald. *Formal methods: practice and experience.* ACM Computing Surveys, No 4, 2009, pp. 1–36.

[2] R. Floyd. *Assigning meanings to programs.* In: Proceedings of the American Mathematical Society Symposia on Applied Mathematics. Volume 19, pp. 19–31, 1967.

[3] C.A.R. Hoare. *An axiomatic basis for computer programming.* Communications of the ACM, pp. 576–580, 1969.

[4] M.A. Jackson. *Principles of program design.* Academic Press, London, 1975.

[5] J. Backus. *Can programming be liberated from the von Neumann style? A functional style and its algebra of programs.* Communications of the ACM, Vol. 21, 1978, pp. 613–641.

[6] C.A.R. Hoare, J. Misra, G. Leavens, N. Shankar. *The verified software initiative: A manifesto.* ACM Computing Surveys. Volume 41, Issue 4, 2009.

[7] T. Hoare. *The verifying compiler: A grand challenge for computing research.* Gresham College, 18/03/2004, Barnard's Inn Hall, 2004. `http://www.cs.ox.ac.uk/files/6187/Grand.pdf`

[8] H.R. Nielson, F. Nielson. *Semantics with applications – a formal introduction.* Wiley professional computing. Wiley, 1992.

[9] D. Sannella, A. Tarlecki. *Foundations of Algebraic Specification and Formal Software Development*. Monographs in Theoretical Computer Science. An EATCS Series. Springer, 2012.

[10] A.V. Lamsweerde. *Requirements engineering: from system goals to UML models to software specifications.* Gran Bretaa, Inglaterra, ISBN: 978-470-01270-3, 2009.

[11] V.N. Redko. *Backgrounds of compositional programming.* Programming, No. 3, 1979, pp. 3–13. [in Russian]

[12] N.S. Nikitchenko. *A Composition-nominative approach to program semantics.* Technical Report IT-TR 1998-020, Technical University of Denmark, ISSN 1396-1608, 1998.

[13] M.S. Nikitchenko, S.S. Shkilnjak. *Mathematical logic and algorithms theory*. Kiev National University Press, 2008 [in Ukrainian].

[14] M.S. Nikitchenko, S.S. Shkilnjak. *Applied logic*. Kiev National University Press, 2013 [in Ukrainian].

[15] M. Nikitchenko, Ie. Ivanov. *Programming with nominative data.* In: Proceedings of CSE'2010 International Scientific Conference on Computer Science and Engineering, September 20–22, 2010, Kosice, Slovakia, pp. 30–39, 2010.

[16] V.G. Skobelev, M. Nikitchenko, Ie. Ivanov. *On algebraic properties of nominative data and functions*. Communications in Computer and Information Science (CCIS), Springer International Publishing, Volume 469, pp. 117–138, 2014.

[17] A. Voronkov. *EasyChair*. WWV 2010. 6th International Workshop on Automated Specification and Verification of Web Systems (Laura Kovacs and Temur Kutsia eds.). EasyChair Proceedings in Computing. Volume 18, 2013.
`http://www.easychair.org/publications/paper/EasyChair`

[18] D.A. Davey, H.A. Priestly. *Introduction to lattices and order*. Cambridge University Press, 2002.

[19] G. Gratzer. *Lattice theory: foundation.* Springer, Basel AG, 2011.

[20] R. Lidl, G. Pills. *Applied abstract algebra.* Springer, NY, 1998.

Volodymyr G. Skobelev, Ievgen Ivanov,
Mykola Nikitchenko

Received October 19, 2015

Volodymyr G. Skobelev
V.M. Glushkov Institute of Cybernetics of NAS of Ukraine
40 Glushkova ave., Kyiv, Ukraine, 03187
Phone: +38 063 431 86 05
E–mail: skobelevvg@mail.ru

Ievgen Ivanov
Taras Shevchenko National University of Kyiv
01601, Kyiv, Volodymyrska st, 60
Phone: +38044 2590519
E–mail: ivanov.eugen@gmail.com

Mykola Nikitchenko
Taras Shevchenko National University of Kyiv
01601, Kyiv, Volodymyrska st, 60
Phone: +38044 2590519
E–mail: nikitchenko@unicyb.kiev.ua