The System of Set Theory for Operating with Essences in the Objects Intellectual Environment

Dmytro Terletskyi
Department of Information Systems, Faculty of Cybernetics
Taras Shevchenko National University of Kyiv
Glushkov Avenue 4d, Kyiv, Ukraine 03680
Email: dmytro.terletskyi@gmail.com

***Abstract*—This paper contains analysis of the basic concepts of set theory and consideration of the process of sets and multisets creation. The author proposes a version of constructive sets theory (CST). The main features of CST are opportunities for working with sets of objects, in particular sets creation, searching of objects, their comparison and classification, creation of new objects prototypes. These opportunities can be implemented using some basic operations on objects, which are given in CST.**

***Keywords*—set, multiset.**

I. Introduction

The term “set” is central in set theory and one of the most important for mathematics in general, apart from this, such concept has important and global importance in the human life. We use sets in our mental activity during perception, analysis, comparison, retrieval, classification every day. We create the set consciously or unconsciously, operate them and apply to them a variety of operations, including set-theoretic. But the questions about the origin of specific sets are emerging while analysing the definition of this term which is given in [2]. We can conclude that the “new” set can be obtained by set-theoretic operations over “existing” sets, basing on the foundations of classical set theory, which are given in [1], and it is really so. However, the questions about origin of these so-called “existing” sets, their number, their types and so on do not disappear. There are questions about the possibility of automating the process of creating sets for machines unaided; automating the classification and identification of elements of the set; the automatic generation of sets that belong to a certain class. I.e. there is a question about the possibility of practical implementation for machines ability to operate with such basic category of human thought as a “set”.

II. The Proposed Approach

Some expansions and formalizations of classical set theory are proposed considering all its features. In particular, the new level – the level of objects is introduced. It means that objects are components for creating sets. It extends the classical theory and serves as a constructive explanation for the existence of the sets. The concept of the class of objects that gives a way to formalize the classification of the objects themselves is introduced also. The author proposes a constructive sets theory (CST) which is a new version of the set theory, in which the notion of “singleton set” and “empty set” are rejected. CST is similar to alternative sets theory which proposed in [5] and it is based on concept of infinity by Brouwer. Also the author offers some operations on objects within the framework of CST, and shows the examples of their practical application.

III. The Basic Concepts of CST

We know that each set consists of elements, which are forming it. Everything, phenomena of our imagination of our world can be the elements of the set. It is convenient for us to call them objects. Let us consider such object as “natural number”. It is clear that every natural number must be integer and positive. These are characteristic properties of natural numbers. It is obvious, that 2 is really a natural number, but -3 and 8.29, for example, are not natural numbers. We can conclude, that each object has certain properties, which define it as some essence while analysing this fact. After all, we can formulate the definition of “object”.

Object A – is a carrier of some properties and attributes $P(A) = (p_1(A), \dots, p_n(A))$, where $P(A)$ is a vector of characteristic properties (specification) of object A, which defines it as an essence. We will denote arbitrary object as follows:

$$A = A/P(A) = A/(p_1(A), \dots, p_n(A)).$$

Clearly, that every object has number of properties, that is why we will define it like dimension of the object, and we will denote it like $D(A)$, where A is an object.

Objects A and B are homogeneous objects, if and only if, when they have the same dimension and equivalent specification. If objects A and B have different dimension and not equivalent specification, they are inhomogeneous.

In general, we can divide the objects on concrete and abstract, and does not matter when or how each particular object was created. It is material implementation of its abstract image – a prototype. This prototype is essentially an abstract specification for creation the future real objects. Besides properties of objects, operations (methods) that can be applied to objects, considering the features of their specifications, should also be allocated. We often use specification and signature of object without object, in programming [6], and we define it as an object type or as a class of object.

Class of objects is an abstract specification of the certain quantity of objects, and some operations which we can apply

TABLE I
ACCORDANCE OF OBJECTS 3, 2.75, −16, 4, −7.48 TO SPECIFICATION OF THE CLASS R

$p_j(A_i)/A_i$	3	2.75	−16	4	−7.48
Integer number	1	0	1	1	0
Natural number	1	0	0	1	0
Fractional number	0	1	0	0	1
Negative number	0	0	1	0	1
Pair number	0	0	1	1	1

to them. We will denote arbitrary class of object as follows:

$$T = (P(T), F(A)) = \\ = ((p_1(T), \dots, p_n(T)), (f_1(T), \dots, f_k(T))).$$

Where T is a name of the class, $P(T)$ is an abstract specification, $F(T)$ are certain operations (methods) which can be applied to objects of the class T considering the peculiarities of their specifications.

Class of objects is homogeneous (inhomogeneous) class, if and only if, the creation of homogeneous (inhomogeneous) objects based on its specification is possible.

Let us define type Int in the programming language C++, using the concept of homogeneous class. Let class Int has next specification

$$P(Int) = (p_1(Int), p_2(Int)),$$

where property $p_1(Int)$ is "integer numbers", and property $p_2(Int)$ is "numbers not more than 2147336147 or not less than −2147336148". Obviously that, all the numbers which have properties $p_1(Int)$ and $p_2(Int)$ are objects of class Int.

Now let us define the following set of operations for class Int

$$F(Int) = (f_1(Int), f_2(Int), f_3(Int), f_4(Int)),$$

where $f_1(Int)$ ="+", $f_2(Int)$ ="−", $f_3(Int)$ ="·", and $f_4(Int)$ ="÷".

Let us define class of real numbers R, using the concept of inhomogeneous class. Let class R has next specification

$$P(R) = (p_1(R), p_2(R), p_3(R), p_4(R), p_5(R)),$$

where property $p_1(R)$ is "integer numbers", $p_2(R)$ is "natural numbers", $p_3(R)$ is "fractional numbers", $p_4(R)$ is "negative numbers", and $p_5(R)$ is "pair numbers". Let us consider the numbers: 3, 2.75, −16, 4, −7.48. They are inhomogeneous objects. It is evident from their comparison. All this objects are objects of class R and they must comply specification $P(R)$. It is clear that each of these numbers is appropriate to specifications $P(R)$ in its own way (see Table I).

Clearly that all these objects are objects of class R, because

$$P(3), P(2.75), P(-16), P(4), P(-7.48) \subset P(R).$$

And now, we can define concept of set of objects, using the concepts of object and class of objects.

The set of objects S is a union that can be created using one of the following schemes:

1) $O_1 \cup \dots \cup O_n = S/T_S$,
2) $S_1 \cup \dots \cup S_m \cup O_1 \cup \dots \cup O_n = S/T_S$,
3) $S_1 \cup \dots \cup S_m = S/T_S$,

where O_i, $i = \overline{1,n}$ are some objects, and S_j, $j = \overline{1,m}$ are some sets of objects, herewith O_i and S_j can have different classes.

IV. Basic Operations of CST

We introduce some operations for objects using the concept of object and object class, in particular union, intersection, difference, symmetrical difference and cloning operations.

Set of objects S (scheme 1) is the union $\cup$ into a single unit not less than two arbitrary objects of arbitrary classes, that is

$$S = (A_1/T_1) \cup \dots \cup (A_n/T_n) = \{A_1, \dots, A_n\}/T_S,$$

and $T_1 \cup \dots \cup T_n = T_S$, where T_S is the class of new set, its specification has next structure

$$P(T_S) = (Core(T_S), pr_1(A_1), \dots, pr_n(A_n)),$$

where $Core(T_S)$ is the core of class T_S specification, which includes properties similar for specifications

$$P(A_1), \dots, P(A_n),$$

and where $pr_1(A_1), \dots, pr_n(A_n)$ are projections of objects $A_1, \dots, A_n$, which consist of properties typical only for these objects.

Set of objects S (scheme 2) is the union $\cup$ into a single unit not less than one arbitrary set of objects $S_1 = B_1, \dots, B_m$, and not les then one arbitrary object of arbitrary classes, that is

$$S = (S_1/T_{S_1}) \cup \dots \cup (S_m/T_{S_m}) \cup (A_1/T_{A_1}) \cup \dots \cup \\ \cup (A_n/T_{A_n}) = \{B_1, \dots, B_m, A_1, \dots, A_n\}/T_S$$

and $T_{S_1} \cup \dots \cup T_{S_m} \cup T_{A_1} \cup \dots \cup T_{A_n} = T_S$, where T_S is the class of new set, its specification has the next structure

$$P(T_S) = (Core(T_S), pr_1(B_1), \dots, pr_m(B_m), \\ pr_1(A_1), \dots, pr_n(A_n)),$$

where $Core(T_S)$ is the core of class T_S specification, which includes properties similar for specifications

$$P(B_1), \dots, P(B_m), P(A_1), \dots, P(A_n),$$

and $pr_1(B_1), \dots, pr_m(B_m), pr_1(A_1), \dots, pr_n(A_n)$ are projections of objects $B_1, \dots, B_m, A_1, \dots, A_n$, which consist of properties typical only for these objects.

Set of objects S (scheme 3) is the union $\cup$ into a single unit not less than two arbitrary sets of objects of arbitrary classes, that is

$$S = (S_1/T_{S_1}) \cup (S_2/T_{S_2}) \cup \dots \cup (S_m/T_{S_m}) = \\ = \{A_1, \dots, A_n, B_1, \dots, B_m, \dots, Z_1, \dots, Z_k\}/T_S$$

and $T_{S_1} \cup \dots \cup T_{S_m} \cup T_{A_1} \cup \dots \cup T_{A_n} = T_S$, where T_S is the class of new set, its specification has next structure

$$P(T_S) = (Core(T_S), pr_1(A_1), \dots, pr_n(A_n), \\ pr_1(B_1), \dots, pr_m(B_m), \dots, pr_1(Z_1), \dots, pr_k(Z_k)),$$

where $Core(T_S)$ is the core of class T_S specification, which includes properties similar for specifications $P(A_i), P(B_j), \ldots, P(Z_t)$, where $i = \overline{1,n}$, $j = \overline{1,m}$, $t = \overline{1,k}$, and $pr_i(A_i), pr_j(B_j), \ldots, pr_k(Z_k)$ are projections of objects $A_1, \ldots, A_n, B_1, \ldots, B_m, \ldots, Z_1, \ldots, Z_k$, which consist of properties typical only for these objects.

The new object (clone) A_i, $i = \overline{1,n}$ will be the result of cloning operation applied to the object A. It will have the same specification and will belong to the same class as object A, that is $P(A) \equiv P(A_i)$ and $T_A \equiv T_{A_i}$.

$$Clone(A) = A_i/(p_1(A), \ldots, p_n(A)),$$

where A_i is i-th clone of object A.

The intersection $\cap$ of two arbitrary objects A and B, which have specifications $P(A)$, $P(B)$ and belong to the classes $T(A_1)$ and $T(A_2)$ is an abstract object of class T_A, which specification is defined as follows

$$P(T_A) = \begin{cases} p_{i_1}(A_1), & Eq(p_{i_1}(A_1), p_{i_2}(A_2)) = 1; \\ p_i(T_A), & Eq(p_{i_1}(A_1), p_{i_2}(A_2)) = k \in (0,1); \\ break, & i_1 \vee i_2 > \min(D(A_1), D(A_2)). \end{cases}$$

The difference $\setminus$ of two arbitrary objects A and B, which have specifications $P(A)$, $P(B)$ and belong to the classes $T(A_1)$ and $T(A_2)$ is an abstract object of class T_A, which specification is defined as follows

$$P(T_A) = \begin{cases} p_{i_1}(A_1), & Eq(p_{i_1}(A_1), p_{i_2}(A_2)) = 0; \\ p_i(T_A), & Eq(p_{i_1}(A_1), p_{i_2}(A_2)) = k \in (0,1); \\ p_{i_1}(A_1), & i_1 > i_2; \\ break, & i_1 > D(A_1). \end{cases}$$

The symmetrical difference $\div$ of two arbitrary objects A and B, which have specifications $P(A)$, $P(B)$ and belong to the classes $T(A_1)$ and $T(A_2)$ is an abstract object of class T_A, which specification is defined as follows

$$P(T_A) = \begin{cases} p_{i_1}(A_1), p_{i_2}(A_2), & Eq(p_{i_1}(A_1), p_{i_2}(A_2)) = 0; \\ p_i(T_A), & Eq(p_{i_1}(A_1), p_{i_2}(A_2)) = k; \\ p_{i_1}(A_1), & i_1 > D(A_2); \\ p_{i_2}(A_2), & i_2 > D(A_1); \\ break, & \left[\begin{array}{l} i_1 > \max(D(A_1), D(A_2)); \\ i_2 > \max(D(A_1), D(A_2)); \end{array}\right. \end{cases}$$

where $k \in (0,1)$.

And now, let us consider examples which will show some practical applications of operations on objects, which were described earlier (see results in Table II).

Let us consider two objects A_1 and A_2. Let the object A_1 is a word of English language "function", and object A_2 is a word of English language "fact". Let us describe their specification $P(A_i)$, $i = \overline{1,5}$, where $p_1(A_i)$ is a number of letters in the word, $p_2(A_i)$ is a number of vowel letters in the word, $p_3(A_i)$ is a number of consonant letters in the word, $p_4(A_i)$ is plural or singular, $p_5(A_i)$ is a set of letters that make up the word.

TABLE II
THE RESULTS OF OPERATIONS ON OBJECTS A_1 AND A_2

	p_1	p_2	p_3	p_4	p_5
A_1	8	3	5	s.	{f,u,n,c,t,i,o,n}
A_2	4	1	3	s.	{f,a,c,t}
$A_1 \cup A_2$	(8,4)	(3,1)	(5,3)	s.	({f,u,n,c,t,i,o,n},{f,a,c,t})
$A_1 \cap A_2$	4	1	3	×	{f,c,t}
$A_1 \setminus A_2$	4	1	3	×	{u,n,i,o,n}
$A_2 \setminus A_1$	×	×	×	×	{a}
$A_1 \div A_2$	4	1	3	×	{u,n,i,o,n,a}

V. CREATION OF MULTISETS

Now we know how to create sets, it is clear from definition of set of objects, but situation with multiset is not so obvious. We should consider this issue in detail.

Let A is an object, which has specification

$$P(A) = (p_1(A), \ldots, p_n(A)),$$

and belongs to the class T_A. Using cloning operation, we can create the clones of object A, for example

$$Clone(A) = (A_1/p_1(A), \ldots, p_n(A))/T_A;$$
$$Clone(A) = (A_2/p_1(A), \ldots, p_n(A))/T_A;$$
$$Clone(A) = (A_3/p_1(A), \ldots, p_n(A))/T_A.$$

Clearly, that object A and all its clones A_1, A_2, A_3 are homogeneous objects. After this, we can apply union operation for object A and all its clones and in such a way we will create multiset S, i.e.

$$S = (A/T_A) \cup (A_1/T_A) \cup (A_2/T_A) \cup (A_3/T_A) = \\ = \{A, A_1, A_2, A_3\}/T_S.$$

Thus when we have created multiset S, we also have created a new class T_S with the following specification

$$P(T_S) = P(T_A) = (p_1(A), \ldots, p_n(A)).$$

Clearly, that S is a homogeneous multiset, because

$$A \equiv A_1 \equiv A_2 \equiv A_3.$$

Considering this fact we can conclude that

$$S = A \bigcup_{i=1}^{n} (Clone_i(A)).$$

Let A and B are objects, which have different specifications $P(A)$, $P(B)$, and belong to the class T_A and T_B respectively. Using cloning operation, we can create the clones of object A and object B, for example

$$Clone(A) = (A_1/p_1(A), \ldots, p_n(A))/T_A;$$
$$Clone(A) = (A_2/p_1(A), \ldots, p_n(A))/T_A;$$
$$Clone(B) = (B_1/p_1(B), \ldots, p_m(B))/T_B;$$
$$Clone(B) = (B_2/p_1(B), \ldots, p_m(B))/T_B.$$

Clearly, that object A and all its clones A_1, A_2 are homogeneous objects. Also we have similar situation in case of object B and all its clones B_1, B_2. After this, we can apply union operation for objects A, B and all their clones and in such a way we will create multiset S, i.e.

$$S = (A/T_A) \cup (A_1/T_A) \cup (A_2/T_A) \cup (B/T_B) \cup$$
$$\cup (B_1/T_B) \cup (B_2/T_B) = \{A, A_1, A_2, B, B_1, B_2\}/T_S.$$

Thus when we have created multiset S, we also have created a new class T_S with the following specification

$$P(T_S) = (Core(T_S), pr(A), pr(B)).$$

Clearly, that S is a inhomogeneous multiset, because

$$A \equiv A_1 \equiv A_2 \text{ and } B \equiv B_1 \equiv B_2,$$

however A and B are objects of different classes. Considering this fact we can conclude that

$$S = \bigcup_{i=1}^{n} \left(A_i \bigcup_{j=1}^{n_i} (Clone_j(A_i)) \right).$$

VI. CONCLUSIONS

In this paper the concept of object, class of object, sets of objects, multisets of objects and 3 schemes for sets creation and 1 scheme for multisets creation are proposed. The methods of implementation each of those schemes also are offered in this paper. These methods are constructive and allow not only creating (generating) the sets and multisets, but also classifying them. It allows to consider the problem of object classification and object identification in another way.

Also, this paper contains some examples of practical application of operations on objects and some examples of sets and multisets creation.

CST is very interesting version of set theory, which differs from other systems of sets theory, such as Cantors set theory, Russells type theory, Quines new foundations and set theory of mathematical logic, axiomatic systems of set theory by ZermeloFraenkel and Von NeumannBernaysGdel [3], [4], [5]. It uses some ideas of these theories, but in the same time many concepts and things are impossible in CST which is very perspective, but it needs further research.

The results that presented in this paper can be applied in the areas such as expert systems and systems of artificial intelligence, particularly in the design and development of intelligent information systems, pattern recognition and theory of decision making.

REFERENCES

[1] R. R. Stoll, *Sets, Logic and Axiomatic Theories.* 2nd edt., San Francisco, CA, USA: W.H. Freeman & Co Ltd, 1975.

[2] J. W. Dauben, *Georg Cantor: His Mathematics and Philosophy of the Infinite.* Princeton, NJ, USA: Harvard UniversityvPress, 1979.

[3] H. Wang and R. McNaughton, *Les Systémes Axiomatiques de la Théorie des Ensembles.* Paris, France: Gauthier-Villars, 1953.

[4] A. A. Fraenkel and Ye. Bar-Hillel, *Foundations of set theory.* Amsterdam, Holland: North-Holland Publishing Company, 1958.

[5] P. Vopenka, *Mathematics in the alternative set theory.* Leipzig, Germany: B.G. Teubner, 1979.

[6] B. Stroustrup, *The C++ Programming Language: Special Edition.* USA: Addison-Wesley Professional, 2000.