

---

# Post's thesis and wrong Janov-Mucnik's statement in multi-valued logic

Maydim A. Malkov

Russian Research Center for Artificial Intelligence, Moscow, Russia

**Email address:**

mamalkov@gmail.com

**To cite this article:**

Maydim A. Malkov. Post's thesis and wrong Janov-Mucnik's statement in multi-valued logic. *Pure and Applied Mathematics Journal*.

Vol. 4, No. 4, 2015, pp. 172-177. doi: 10.11648/j.pamj.20150404.16

---

**Abstract:** Post stated that multi-valued logic has no principle difference with respect to two-valued logic. But Janov and Mucnik stated that multi-valued logic has essentially difference with respect to two-valued logic. We show that Post's thesis is well but Janov-Mucnik's statement is wrong.

**Keywords:** Multi-Valued Logic, Post Algebra, Fictitious Closed Sets of Functions, Classification of Function

---

## 1. Introduction

Multi-valued logic attracts the intense attention because of the connection with computer technology.

Post ([1], 1921) stated that multi-valued logic has no principle difference with respect to two-valued logic. In particular, he proved that every predicate of multi-valued logic can be presented by logical formula with connectives  $\vee$  and  $\neg$ , as in 2-valued logic. But Janov and Mučnik ([2], 1959) stated that multi-valued logic has essentially difference with respect to two-valued logic because there are continuum of closed sets of functions. But these closed sets are fictitious objects of multi-valued logic. It is natural that the number of fictitious object is continuum since almost every theory has infinitely more fictitious objects with respect to essential objects. And multi-valued logic has countable number of essential closed sets. Two-valued logic has countable number of essential closed sets, too.

This paper is devoted to confirm Post's thesis.

## 2. Conjunctive (CNF) and Disjunctive (DNF) Normal Forms in Multi-Valued Logic

These normal forms use connectives  $\neg$ ,  $\vee$  and  $\wedge$ .

Post gave definitions of the connectives. He produced them by computable functions:  $\bar{x} = x + 1 \pmod{k}$ ,  $x_1 \vee x_2 = \max(x_1, x_2)$ , and  $x_1 \wedge x_2 = \min(x_1, x_2)$ .

These definitions hold in 2-valued logic too.

Definitions of CNF and DNF in multi-valued logic differ from definitions in 2-valued logic only by number of negations: CNF is conjunction of disjuncts of literals, DNF is a disjunction of conjuncts of literals, literals are logical variables with or without negations.

If we have a table (a sequence of lines) of a function then we can build full CNF and full DNF corresponding to the function. And we build full DNF by using the next rule, which holds as in 2-valued logic as in multi-valued logic:

- we create a sequence of lines of the table such that the lines have value  $k-1$  in function-column,
- we delete  $k-1$  in lines of the sequence,
- we replace value  $j$  of  $i$ -th variable in every line of the sequence by  $\bar{x}_i^{-k-1-j}$ ,
- these lines connected by  $\vee$  create full DNF.

The similar rule exists for full CNF too.

Reduced DNFs have no superfluous literals with respect to full DNFs, shortest DNFs have no superfluous conjuncts. These DNFs are built in multi-valued logic as in 2-valued logic.

Instead of DNF Post used expansion of a function in 2-valued logic:  $(DNF_1, DNF_0)$ , where  $DNF_1$  is DNF of a function and  $DNF_0$  is DNF of negation of the function. In  $k$ -valued logic an expansion of a function is  $(DNF_{k-1}, DNF_{k-2}, \dots, DNF_0)$ . Here  $DNF_{k-1}$  is DNF of a function,  $DNF_{k-2}$  is DNF of negation of the function,  $DNF_0$  is DNF of  $k-1$ -times negation of the function.

Post used the expansion to build classification of Boolean functions. And we can use the expansion to build classification of functions of multi-valued logic.

We have similar results for CNFs.  
This confirms Post's thesis.

### 3. Classification of Functions

#### 3.1. Preiterative Algebra

Now there are 3 algebras for classification of functions: Post (or preiterative), Mal'cev (or iterative) and clone algebras.

Mal'cev gave mathematically precise definitions of preiterative and iterative algebras. By Mal'cev, preiterative algebra has 4 primitive operations: cyclic permutation of variables of a function, permutation of two variables, identification of two variables and substitution a function into a function (for more details see below). Iterative algebra has one more operation that adds fictitious variables in a function.

Unfortunately, preiterative algebra is ignored by contemporary researches. For example, the monograph [3] used only iterative and clone algebras. Only Rosenberg displayed an interest for Mal'cev's results. He devoted one of his works [4] to Mal'cev. He marked "Preiterative sets are slightly more general than clones". In reality, the preiterative algebra is more general than the algebra of clones (Rosenberg used "sets" instead of "algebra"). But Rosenberg continues to use only iterative and clone algebras since the algebras are more simple to get new results.

Mal'cev called iterative algebra *Post iterative algebra*. But Post did not use the algebra. Therefore we call it Mal'cev algebra.

Mal'cev algebra is a part of Post algebra. And clone algebra is a part of Mal'cev algebra.

Definition The Post algebra  $P_k$  is

$$P_k = (P_k; \zeta, \tau, \triangleright, *)$$

where  $P_k$  is the set of all  $k$ -valued everywhere defined functions with the domain  $\{0, \dots, k-1\}$ . And  $\zeta, \tau, \triangleright, *$  are primitive operations (primitives) over these functions.

In accordance with the standard definition of algebras,  $P_k$  is the universe of the algebra,  $\zeta, \tau, \triangleright$  and  $*$  are the fundamental operations of the algebra. Only they generate closed sets of functions.

These fundamental operations are primitives.

Primitives are those operations that build all other operations. The formal definitions of primitives were given by Mal'cev [5]. We have generalized the definitions.

By informally, the primitive  $\zeta$  is the cyclic operation: a function  $f(x_1, \dots, x_n)$  becomes the function  $f(x_n, x_1, \dots, x_{n-1})$ . The primitive  $\tau$  is permutation operation: a function  $f(x_1, \dots, x_n)$  becomes the function

$f(x_n, x_2, \dots, x_{n-1}, x_1)$ . The primitive  $\triangleright$  is identification operation: a function  $f(x_1, \dots, x_n)$  becomes the function  $f(x_1, \dots, x_{n-1}, x_1)$ . The primitive  $*$  is a substitution operation: a function  $f_1(x_1, \dots, x_{n_1})$  and a function  $f_2(x_1, \dots, x_{n_2})$  become the function  $f_1(f_2(x_1, \dots, x_{n_2}), x_{n_2+1}, \dots, x_{n_1+n_2-1})$ .

Sometimes the operation of adding a fictitious variable is considered as primitive. It was shown [6] that this operation is not primitive since fictitious functions add fictitious variables to a function by using the substitution operation, but fictitious functions can be generated by essential functions.

Algebras with  $k=1$  and  $k=0$  exist too. Using the formula  $k^{k^n}$  for number of functions of arity  $n$ , we find that the universe is empty at  $k=0$  and that the universe contains only functions of arity 1 at  $k=1$ . Algebra with  $k=\omega$  exists too, and it has countable number of functions for every  $n$ .

Again Post's thesis is confirmed.

#### 3.2. Family of Functions and First Equivalence Relation

We will prove that there are 6 levels of classification of functions. The same levels of classifications exist in 2-valued logic. This confirms Post's thesis.

A family is a set of functions with the same type of diagonals. We will call families of functions with the same type of diagonals briefly families.

Families were introduced by Post. There are 4 families in two-valued logic:  $\alpha, \beta, \gamma, \delta$ . Post's classification of function and of their closed sets is based on the families.

Families of functions exist for all  $k \geq 1$  including  $k=\omega$ . The number of these families is  $2^k$ .

The family  $\alpha$  is defined standardly:  $f(x, \dots, x) = x$ . The family  $\delta$  is defined standardly, too:  $f(x, \dots, x) \neq x$ . Definitions of the other families are special in  $k$ -valued logics since the logic has  $k$  constants. But the definitions are an extension of Post's definitions and they hold for 2-valued logic too.

We denote all families by  $\{j_1, \dots, j_m\}$ , where  $0 \leq m \leq k-1$ . And every function of a family has the diagonal  $f(x, \dots, x)$  equal to  $x$  if  $x = j_1 \vee \dots \vee x = j_m$ , and not equal to  $x$  if  $x \neq j_1 \wedge \dots \wedge x \neq j_m$ , i.e., if only  $x = j_1 \vee \dots \vee x = j_m$ . We say that a function of a family  $\{j_1, \dots, j_m\}$  preserves values  $j_1, \dots, j_m$ .

Therefore we denote  $\alpha$  family by  $\{0, 1, \dots, k-1\}$  and  $\delta$  family by  $\{\}$ . A constant  $j$  belongs to family  $\{j\}$  since a diagonal of a constant is this constant. The  $\beta$  family of two-valued logic is denoted by  $\{1\}$  and the  $\gamma$

family is denoted by  $\{0\}$ . Functions of the families preserve 1 and 0 respectively.

We call functions of  $\alpha$  or of  $\delta$  families  $\alpha$  or  $\delta$  functions.

The next theorem is useful.

**Theorem** Any function of family  $X = \{j_1, \dots, j_m\}$  generates functions of family  $Y \supseteq X$ .

*Proof.* The operations of permutations and identification do not change a family of a function. The operation of substitution can change a family of functions.

Let functions  $f_1, f_2$  be of the same family  $X$ ,  $f$  be the substitution of  $f_2$  into  $f_1$ , and denotation of family of  $f$  be  $Y$ .

Then  $f(j, \dots, j) = f_1(f_2(j, \dots, j), j, \dots, j) = j$  for all  $j \in X$ . In this case  $j \in Y$  for all  $j \in X$ , therefore  $Y \supseteq X$ . If  $f(j, \dots, j) \neq j$  for all  $j \notin X$  then  $Y \supseteq X$  since, by definition,  $f_1(j, \dots, j) \neq j$  and  $f_2(j, \dots, j) \neq j$  for all  $j \in X$ . But if  $f(j, \dots, j) = j$  for some  $j \notin X$  then this  $j \in Y$ . In this case  $Y \supset X$ . The proof is finished.

**Corollary** A function of the family  $\alpha$  generates only function of this family. A function of the family  $\delta$  can generate functions of any family.

*Proof.* The denotation of family  $\alpha$  is the set  $X = \{0, \dots, k-1\}$  and there is no family with denotation  $Y$  such that  $Y \supset X$ . The family  $\delta$  has the denotation  $X = \{\}$  and  $Y \supseteq X$  for any family with denotation  $Y$ .

The Webb function and all other generators of  $P_k$  (usually they are named Sheffer functions) can belong only to the family  $\delta$  for all  $k$  including  $k = \omega$ .

Families realize a natural classification of functions since they are disjoint. This means that there is an equivalence relation.

**Definition** Two functions are equivalent if they belong to the same family.

Families realize the first level of the function classification. There are 6 levels of the classification.

### 3.3. Types of Functions and Second Equivalence Relation

In 2-valued logic Post defined 2 conditions  $a^\mu$  and  $A^\mu$  for functions. In  $k$ -valued logic there are  $k$  conditions  $a_0^\mu, \dots, a_{k-1}^\mu$ . And this holds for all  $k$  including  $k = \omega$ .

In the next definition we call tuple  $l$  a line of a table of a function with value  $l$  of the function. A tuple  $l$  contains only values of variables of a function since the value of the function is pointed out by the subscript. And we call a column a part of a column in a table such that the part belongs to all tuple  $l$  of a function.

**Definition** A function satisfies a condition

- $a_l^\mu$  ( $0 \leq \mu < \omega$ ) if every of  $\mu$  tuples  $l$  in the table of the function has a column with value  $l$  of variables and if there exist  $\mu + 1$  tuples  $l$  which have no column with value  $l$  of variables,
- $a_l^{\omega+m}$  ( $0 \leq m < \omega$ ) if all tuples  $l$  have a column with value  $l$ , the number of tuples  $l$  equals  $m$ .

If a function does not satisfy a condition of the first part of definition then the function satisfies a condition of the second part of definition. Indeed, let a function do not satisfy  $a^\mu$ . Then all tuples  $l$  are absent or  $\mu$  tuples  $l$  exist but  $\mu + 1$  tuples  $l$  are absent. In the former case the function satisfies the condition  $a_l^\omega$ . In the latter case the function satisfies the condition  $a_l^{\omega+\mu}$ .

So, a function satisfies the condition  $a_l^\omega$  if tuples  $l$  are absent. If tuples  $l$  exist but the value  $l$  is absent in some tuple  $l$ , then the function satisfies condition  $a_l^0$ . A constant  $j$  satisfies  $a_l^\omega$  if  $l \neq j$  (since the constant is not equal  $l$ ) and satisfies  $a_j^0$  (since the constant equals  $l=j$ ).

The conditions and a name of a family form a type.

**Definition** The type of a function is  $(family, a_0^{\mu_0}, \dots, a_{k-1}^{\mu_{k-1}})$ .

And the type of a constant  $j$  is  $(\{j\}, a_0^\omega, \dots, a_{j-1}^\omega, a_j^0, a_{j+1}^\omega, \dots, a_{k-1}^\omega)$ .

Every function belongs only to one type.

**Definition** Two functions are equivalent by the second equivalence relation if they have the same type.

The first equivalence relation divides the set of all functions into families. The second equivalence relation divides families into subfamilies with the same type.

### 3.4. Classes of Compositions of Function

Closed sets of functions are called sometimes classes. This is not well since classes must be disjoint. But a closed set can contain other closed sets, therefore these closed sets are disjoint.

A closed set becomes empty or not empty after removing closed sets contained into it. Now we use only closed sets that become non-empty after removing their closed sets.

**Definition** A class of function's compositions is a non-empty part of a closed set remaining after removing all other closed sets from the closed set.

Further we call classes of function's compositions briefly classes. A name of a class is a name of a closed set containing the class.

**Theorem** Classes are disjoint.

*Proof.* Any function generates a closed set. Every function of a class generates its closed set  $X$ . This function cannot belong to a class of a closed set contained in  $X$  since, by definition, the function does not belong to any closed set contained in  $X$ . And the function cannot belong to a class of

another closed set since the function generates only the closed set  $X$ .

So any function of a class of a closed set is the member of one membered basis of the closed set. We must take one of these functions as a representative of bases. For that we introduce an order of functions.

Definition A function  $f_1$  is greater than  $f_2$  if the arity of  $f_1$  is greater than the arity of  $f_2$ . If the arity of  $f_1$  equals the arity of  $f_2$  then  $f_1 > f_2$  whenever  $j_1 > j_2$ , where  $j_i$  is the ordinal number of  $f_i$ : the ordinal is the contents of the column with values of the function at reading the column from top to down.

Definition The minimal function of a class of a closed set is called the generator of the closed set.

So the representative of bases of a closed set is its generator.

### 3.5. Third Equivalence Relation

We must prove that all functions of a class have the same type. Before we must prove one lemma. And, for simplification, we will say that a function has  $\mu$  if it satisfies a condition  $a_i^\mu$ .

Lemma All functions of a class belong to the same family.

Proof. By the theorem of the previous subsection,  $\alpha$  functions generate only  $\alpha$  functions. Therefore a class containing an  $\alpha$  function contains only  $\alpha$  functions and its closed set contains only  $\alpha$  functions, too. By the theorem, functions of the family  $\{0, \dots, k-2\}$  can generate only functions of the family and  $\alpha$  functions. A class containing a function of family  $\{0, \dots, k-2\}$  contains only function of the family since  $\alpha$  functions do not belong to the class. By the theorem, functions of a family  $Y$  generate functions of the family  $Y$  and of a family  $X \subset Y$ , but functions of families  $X$  and  $Y$  belong to different classes. This means that a class containing a function of a family  $Y$  contains functions only of the family  $Y$ .

Theorem All functions of a class have the same type.

Proof. It is enough to prove that the functions satisfy the same conditions. And the conditions are  $a_i^\mu \wedge 0 \leq \mu < \omega$  or  $a_i^{\omega+m} \wedge 0 \leq m < \omega$ . At first we will prove two preliminary statements and then we will prove the theorem.

- Let a function  $f_0$  of a class be the generator of the class and satisfy a condition  $a_i^{\mu_0} \wedge 0 \leq \mu_0 < \omega$  for some  $l$ . We must prove the statement that  $f_0$  generates the class of functions  $f$  with  $\mu \geq \mu_0$ . For that we will use the induction rule. If  $\mu_0 = 0$  then  $f_0$  generates functions with  $\mu \geq \mu_0$ . The statement holds in this case. Let all functions of a class have  $\mu \geq n$  if the class is generated by  $f_0$  with  $\mu_0 = n$  and let now  $f_0$  have  $\mu_0 = n+1$ . Then  $f_0$  generates class of

functions with  $\mu \geq n+1$  since functions with  $\mu < n+1$  belong to the other classes. The statement is proved.

- Let a function  $f_0$  of a class be the generator of the class and satisfy a condition  $a_i^{\omega+m_0} \wedge 0 \leq m_0 < \omega$  for some  $l$ . We must prove the statement that  $f_0$  generates the class of functions  $f$  with  $m \geq m_0$ . For that we will use the induction rule. If  $m_0 = 0$  then  $f_0 \neq l$  and all functions of the class generated by  $f_0$  have  $m = 0$ . The statement holds in this case. Let all functions of a class have  $m \geq n$  if the class is generated by  $f_0$  with  $m_0 = n$  and let now  $f_0$  have  $m_0 = n+1$ . Then  $f_0$  generates the class of functions with  $m \geq n+1$  since functions with  $m < n+1$  belong to the other classes. The statement is proved.
- All functions of a class have the same  $m$ .

Indeed, let a function  $f_1$  have  $m_1$  and let a function  $f_2$  have  $m_2 > m_1$ . Then the function  $f_2$  cannot generate the function  $f_1$ . Hence the functions cannot belong to the same class. Only functions with the same  $m$  can belong to the same class.

This means that classes create the third level of the classification.

Definition Two functions are equivalent by the third equivalence relation if they belong to the same class.

The third equivalence relation continues the classification of functions: the set of functions is divided into families, every family has subfamilies, every subfamily has classes.

### 3.6. Functions with the Same Range and Fourth Equivalence Relation

Classes of compositions have subclasses of functions with the same range.

Indeed, if a function has a range with one member then the function is a constant or the function is fictitious. Hence, in this case a closed set contains only the constant or contains the function, a constant generated by the function, and all fictitious functions of the same range.

A class of a closed set containing only a constant has only the constant. The class of closed sets containing a constant and its fictitious functions has only the fictitious functions.

If a function has a range with two members then the function can generate functions that have ranges with one or two members. The functions, which have ranges with two members, form a class, since the functions, which have ranges with one member, belong to the other classes.

And so on.

The next equivalence relation continues the classification of functions.

Definition Two functions are equivalent by the fourth

equivalence relation if they have the same range.

This is the fourth level of the classification.

We call a range containing not all values and all subsets of the range a *partial range*. There are partial complete generators of all functions with the same partial range.

*Definition A complete generator is a generator of all functions. Complete generators usually are called Sheffer functions. A partial complete generator is a generator of all functions that have the same partial range.*

If  $k = 3$  then there are 60 partial generators for every partial range with 2 values [7] (there are 3 such partial ranges:  $\{0,1\}$ ,  $\{0,2\}$ ,  $\{1,2\}$ ). And there are 3 774 complete generators [9].

If  $k = 4$  then every of the subclasses has 13 920 partial complete generators for every partial range with 2 values and has 5 494 944 partial complete generators for every partial range with 3 values [8]. There are 942 897 552 complete generators [8].

### 3.7. Fictitious Functions and Fifth Equivalence Relation

If a subclass has a fictitious function of two or more arity then the subclass contains infinite number of fictitious functions since every essential function has infinite number of fictitious functions by adding fictitious variables to the essential function.

Fictitious functions are superfluous. They are absent in applied algebras. But in abstract algebras fictitious functions are used for construction other functions. For example, the construction of recursive functions uses fictitious functions although the functions can be constructed without fictitious functions.

The next equivalence relation continues the classification of functions.

*Definition Two functions are equivalent by the fifth equivalence relation if they are both essential or they are both fictitious.*

We use the relation to construct classes of the fifth level. The classes are subclasses of classes of the previous level. The classes of the fifth level allow us to isolate fictitious functions from essential. There is a bijection between all classes of essential functions of the preiterative algebra and all classes of the iterative algebra. A pair of classes of the algebras becomes equal if we remove fictitious functions in the iterative algebra. So all properties of functions in the iterative algebra can be found in the preiterative algebra.

### 3.8. Functions with Renumbered Variables and Sixth Equivalence Relation

The other superfluous functions are functions with renumbered variables since a numeration of variables of a new constructed function is subjective. A renumeration of variables of a function does not change the essence of the function but changes the structure of table of the function. Hence the table becomes new and we have a new function that is superfluous. So we must take only one of functions with renumbered variables. For that we take the minimal

function from the functions. We call the function *non-renumbered* and we call the other functions *renumbered*.

We continue to classify functions.

*Definition Two functions are equivalent by the sixth equivalence relation if they both are non-renumbered or if they both are renumbered.*

This relation allows us to construct classes of the sixth level. The classes are subclasses of classes of fifth level. And we isolate non-renumbered functions from renumbered.

One non-renumbered function has a finite number of its renumbered. But the number is very big at large  $k$ .

So we have constructed 6 levels of classification of functions. Two-valued logic has the same 6 levels [6]. This confirms Post's thesis.

## 4. Classification of Closed Sets and Fictitious Closed Sets

### 4.1. Classification of Closed Sets

There are two theories: the theory of  $k$ -valued functions and the theory of closed sets of  $k$ -valued functions.

The main problem of any theory is classification of its objects. The next problem is to find fictitious (useless) objects and to remove them.

We have constructed classification of objects of the first theory, they are functions. We have found that fictitious objects are fictitious functions or renumbered functions. And we have isolated the fictitious objects.

Now we construct the classification of objects of the second theory. These objects are closed sets.

Again we use natural classification of objects: every object belongs only to one class and classes are disjoint. For that we use the number of members contained in minimal bases of closed sets.

Every infinite closed set has infinite number of bases. We use the minimal basis: a basis is minimal if it has the least number of members. If there are several bases with the number of members then we use the basis which maximal member is minimal among the other bases (we do not use basis with minimal member among the other bases since a basis with the minimal member can have very big maximal member). If we have several such bases then we use the pre-maximal members and so on.

We introduce the next classification of closed sets.

*Definition The class  $S_0$  of closed sets contains closed sets without a basis. The class  $S_m$  of closed sets contains closed sets with  $m$  membered minimal basis. The class  $S_\infty$  of closed sets contains closed sets only with infinite bases.*

This classification covers all closed sets and every closed set belongs to only one of classes. This means that classes of closed sets are disjoint. Below we call classes of closed sets briefly classes.

There are 3 classes if  $k=2$ :  $S_1$ ,  $S_2$ , and  $S_3$  [10]. And there are all classes if  $k \geq 3$  [11].

Each class  $S_m$  with finite  $m$  has countable closed sets. The class  $S_\omega$  has continuum closed sets [11].

A member of  $S_0$  is a closed set that is upper limit of the sequence of closed sets of  $S_1$  such that any member of the sequence contains the previous closed set [11].

Any member of  $S_1$  is a closed set that does not contain other closed sets or contains other closed sets but it becomes not empty after removing the sets. By the previous section,  $S_1$  contains all classes of functions.

Any member of  $S_m$  is a union of  $m$  closed sets of  $S_1$ . This member becomes empty after removing all of the closed sets from it.

Any member of  $S_\omega$  is a closed set containing infinite number of closed sets of  $S_1$  such that the union of any pair of the closed sets is a member of  $S_2$ .

#### 4.2. Fictitious Closed Sets

Every function generates some closed set. The other closed sets are fictitious (useless)<sup>i</sup>. Only the class  $S_1$  does not contain fictitious closed sets. We have used its closed sets to construct the classification of functions. Every essential closed sets of the classification become a class of functions after removing the other closed sets contained in the closed set. Fictitious closed sets become empty after removing closed sets contained in it. Therefore fictitious closed sets are useless for the classification of functions like fictitious variables are useless for calculation of functions.

Only the class  $S_1$  contains the 6 levels of the more deep classification, the other classes have only one level. But we have constructed the more deep classification of fictitious closed sets for  $k=2$  [10], and this classification has demonstrated useless of the construction.

Any theory must give a classification of all theory's objects including fictitious. But it is enough to construct only the first level of the classification of fictitious objects since fictitious objects need not in a more deep classification.

By Janov and Mu ĉ nik [2], multi-valued logic is essentially different with respect to two-valued logic since we have continuum of closed sets of functions. But except closed sets of  $S_1$ , all closed sets are fictitious, i.e. useless and superfluous. Closed sets of  $S_1$  are countable.

Fictitious objects are in any theory and their number, as a rule, is infinity more with respect to the number of essential objects. In the theory of functions we have infinite number of fictitious functions for every essential function (by adding fictitious variables). So it is natural that the number of fictitious closed sets is continuum but the number of

essential closed sets is countable.

So, we have shown that conjunctive and disjunctive normal forms are the same as in 2-valued logic as in multi-valued logic. We have shown that Post algebra holds for all logics including  $\omega$ -valued logic. We have shown that there are 6 levels of classification of functions, they are the same as in 2-valued logic as in multi-valued logic. We have built the classification of closed sets of functions for all logics. There are essential and fictitious closed sets, the number of essential closed sets is countable (in all logics) and the number of fictitious sets is continuum. But fictitious closed sets are useless and usually are excluded. All this means that Post's thesis is well and Janov-Mu ĉ nik's statement is wrong.

---

## References

- [1] E. L. Post, Introduction to a general theory of elementary propositions. Amer. J. Math. (1921) 43:4, 163-185.
- [2] Ju. I. Janov, A. A. Mu ĉ nik, On existence of  $k$ -valued closed classes without basis. Doklady SSSR (1959) 127:1, 44-46 (Russian).
- [3] D. Lau, Function algebras on finite sets, Springer, Berlin, 2006.
- [4] I. Rosenberg, Mal'cev algebras for universal algebra terms, Algebraic Logic and Universal Algebra in Computer Science, proceedings of conference (1988), 195-208.
- [5] A. I. Mal'cev, Iterative Post algebras, NGU, Novosibirsk, 1976 (in Russian).
- [6] M. A. Malkov, Algebra of logic and Post algebra (theory of two-valued functions), Mathematical Logic, Moscow, 2012 (in Russian).
- [7] M. A. Malkov, Complete generators in 3-valued logic and wrong Wheeler's results, Int. J. of Math. and its Applications (2014) 2:4, 25-28.
- [8] M. A. Malkov, Complete generators in 4-valued logic and Rousseau's results, Int. J. of Math. and its Applications (2014) 2:4, 49-57.
- [9] Norman M. Martin, The Sheffer functions of 3-valued logic, J. of Symb. logic (1954) 19:1, 45-51.
- [10] M. A. Malkov, Classification of Boolean functions and their closed sets, Sop Transactions on Applied Math. (2014) 1:2, 172-193.
- [11] M. A. Malkov, Classification of closed sets of functions in multi-valued logic, Sop Transactions on Applied Math. (2014) 1:3, 96-105.

---

<sup>i</sup> Do not confuse fictitious closed sets and fictitious functions. There are fictitious closed sets of non-fictitious functions and non-fictitious closed sets of fictitious functions.