UDC 519.683.8

PRIMITIVE PROGRAMING ALGEBRA: GENERAL APPROFCH TO A PROBLEM OF FUNCTIONAL COMPLETENESS

P.O. YAHANOV, D.I. REDKO, I.V. REDKO, T.L. ZAKHARCHENKO

The goal of the research is development of scientific foundations of programming problems solutions genesis. Investigations carried out are based on algebraic research methods of programs and compositional programming methods. Basis of the last ones consists of program algebras with special classes of functions as carriers, and compositions that represent abstractions from program synthesis tools as operations. Problems of completeness in classes of computable functions that took one of the most important places in programming problems are well defined and solved in the context of program algebras. Universal method for the problem of completeness solution in primitive program algebras (PPA) on different classes of computable functions proposed in the article. Results achieved are presented as series of original statements, lemmas and theorems. The results can be applied in algebraic characteristics research of different computable functions classes in problems of programming language semantics formalization

INTRODUCTION

Today's posture in IT field and, particularly, in programming, considerably defined by process of more and more vast it's penetration in all aspects of human's life. Naturally, with every step taken in that direction, requirements made to quality of product produced and effectiveness of its production are constantly increasing. Despite of impressive and speaking for theirselves results achieved with programming activity (PA) today, it becomes more obvious that the results in majority are extensive, so sustainment of this tendency becomes more problematic and impossible in foreseeable future. The reason is typical for nowadays understanding of PA, particularly, programming, its excessive simplicity, which is not corresponding level of complexity of problematic indicated.

As for programming, simplicity of its understanding led to the fact, that, mainly, attention paid to results of programming without consideration of processes, which made that results possible. It makes process of programming problems solution too subjective, regarding intuitive principles of paramount importance. These facts are not allowing us seriously discuss problems of software quality management, effectiveness of its production and preservation of investments. An avalanche-like increase of number of such facts stimulated discussions about development of crisis in programming, depression in IT industry etc [1–3]. Now, not a crisis of the field should be discussed itself, but crisis of its ways of development! Statements, made above, one more time demonstrate that contemporary programming, and the overall field can not effectively develop now exclusively on objectively-intuitive basis, which is the source of different concepts of PA. Long ago, problems of the field became so significant and so complex, that intuitive considerations must be objectified adequately and supplemented with precise researches and developments as far as possible. The matter is in the main intuition carrier of PA — programming as process of software creation. Questions connected with revelation of programming languages semantic play here vital role. That is why research of following problematic is **objective** of the paper. Paramount role here plays compositional paradigm [4–8], as methodological consideration basis of whole diversity of general as well as particular software creation methods. Namely that methods, to be more precise, their explications in the form of different classes of compositions build up the **object** of the research. The **subject** of the research is problem of computable functions characteristics classes creation. The functions are on different carriers in primitive programming algebras (PPAs) [9–11]. The main attention is paid to search of such algebra's generative sets and bases. In process of research, along with general results, description of computable functions class on records also received. It supplements analogical results for natural numbers tuples, graph transformers and multi-sets [10–15].

All undefined generic mathematical conceptions and designations are interpreted in sense [16], and concepts of numerations theory and theory of algorithms interpreted as in [17, 18].

GENERAL STATEMENTS

The carrier of PPA are n-ary functions and n-ary predicates (or simply functions and predicates) $(n = 1,2,\dots)$. The signature of PPA (denoted as Ω) consists of superposition, branch, loop operations, which are represent adequate specifications of the main methods of software or computational hardware design, which are peculiar to majority of high-level programming languages [1, 4–7, 9]. Let us make formal definitions of the operations. Termal, rather then operator notation of functions will be preferred for convenience and compactness [18]. Usage of specific notation form in every individual case conditioned by the fact that different notations present fundamentally different viewpoints on the entity they describe. In other words, operator notation used in cases when it is important to reveal genesis of entity described, termal notation is important for description of result genesis. Although, those forms are interchangeable like texts in certain senses, those keypoints arrangement is important because it presents completely natural dominant of genesis relatively to its result.

Let m functions $f_1,\dots,f_m$ of the same arity (for example, k) of type $A^k \to B$ be defined on certain set A with values from set B (it is no need to preserve $A \cap B = \varnothing$, moreover $A \cap B \neq \varnothing$ is acceptable case too). Also, let m-ary function f with values in certain set C be defined on set B. Consider k-ary function $g : A^k \to C$ with value $g(<a_1,\dots,a_k>) \cong f(f_1(<a_1,\dots,a_k>),\dots$ $\dots, f_m(<a_1,\dots,a_k>)))$ on argument $<a_1,\dots,a_k>$. In this case function g is the result of a $(m+1)$-ary *superposition* application, denoted as S^{m+1}, to m-tuple of functions $<f, f_1,\dots,f_m>$, i.e. $g = S^{m+1}(<f, f_1,\dots,f_m>)$. Hereinafter in this document the designation "$\cong$" means the generalized equality [19].

Now, additionally let function h of type $A^k \to B$ and m-valued function $\beta : B \to \{1,2,\dots,m\}$ be defined. k-ary function $g : A^k \to B$ is built from functions $h, f_1, \ldots, f_m$ by (m+1)-ary parametric *branch* operation $\Diamond_\beta^{m+1}$ if for any argument $< a_1,...,a_k > \in A^k$ the value of function $g(< a_1,...,a_k >)$ defined as:

$$g(< a_1,...,a_k >) \cong f_r(< a_1,...,a_k >), \text{ if } \beta(h(< a_1,...,a_k >)) = r, \ (1 \le r \le m).$$

Note, that described parametric branch operation represents adequate specification of well-known method of software design — $case\_of$. Ternary *branch* operation $\Diamond$, which puts in correspondence to two functions $f_i : A^k \to B,\ i = 1,2$ and one predicate $p : A^k \to \{T, F\}$ a k-ary function $g \equiv \Diamond(< p, f_1, f_2 >)$ with values defined on any argument $< a_1,...,a_k > \in A^k$ as:

$$g(< a_1,...,a_k >) \cong \begin{cases} f_1(< a_1,...,a_k >), p(< a_1,...,a_k >) \cong T \\ f_2(< a_1,...,a_k >), p(< a_1,...,a_k >) \cong F \end{cases}$$

may be useful partial case.

Finally, let us complete our list of definitions with k-ary predicate $p : A^k \to \{T, F\}$. Consider k-ary function $g : A^k \to B$ with value $g(< a_1,...,a_k >)$ on arbitrary argument $< a_1,...,a_k > \in A^k$ equal to the first component of the first tuple from sequence of tuples $[< a_1^i,...,a_k^i >]_{i=0,1,2,...}$, where $a_j^0 = a_j,\ j = 1,2,...,k$ and $a_j^{i+1} = f_j(< a_1^i,...,a_k^i >),\ j = 1,2,...,k$, for which (denote it as $< a_1^s,...,a_k^s >$, for example) $p(< a_1^s,...,a_k^s >) = F$ in case if for all $r < s$, if such argument exists, value of $p(< a_1^r,...,a_k^r >) \cong T$. Function g built by application of $(m+1)$-loop operation to functions of $(m+1)$-tuple $< p, f_1, \ldots, f_m >$, i.e. $g = *^{m+1}(< p, f_1, \ldots, f_m >)$. Thus, according to statements made before, $g(< a_1,...,a_k >) = a_1^s$.

Note, previously, in order to denote introduced operations, we used exclusively operator notation. By using termal notation of operations from PPA signature, we will evidently denote only variables with considerably used values. For instance, for loop operation notation like $p(x_1, \ldots, x_m) \underset{y_1 \cdots y_k}{*} < f_1(z_1^1, \ldots, z_{m_1}^1), \ldots$ $\ldots, f_m(z_1^k, \ldots, z_{m_k}^k) >$ will be used with those variables denoted, on which functions and predicate considerably depend. At the same time, function $f_j(z_1^j, \ldots, z_{m_j}^j)$ changes variable $y_j,\ j = 1,2,...,k$ and variable y_1 considered as an "output". Operator notation can be easily reconstructed from this notation.

Let us declare certain countable set D and for any natural number $k > 0$ consider classes Φ^k of partial k-ary functions and predicates of types: $D^k \to D$ and $D^k \to \{T, F\}$ accordingly, and $\Phi \equiv \bigcup_k \Phi^k,\ k = 1,2,3,...$ of partial multiplace functions and predicates on set D. Further, functions and predicates on D will be denoted as D-functions (D-predicates) and will belong to set Φ. Computability on D is defined as numeric computability [17, 20]. PPA with carrier formed from partial recoursive functions (computable functions, pr-functions) and partial recoursive predicates (computable predicates, pr-predicates on D will be denoted as A_D^{pr}. Generative set of A_D^{pr} will be defined as its *comlete system* (CS). Complete system of PPA will be its I_m^n*-basis*, if any system produced by exclusion of any elements from the CS, except selective function, will not be complete.

Some terms, designations, properties and associated results, showed below, may be useful during study of PPA complete systems [10].

Property 1. n-ary function f preserves set $L \subseteq D$, $L \neq \varnothing$, if $f(\underbrace{L \times \ldots \times L}_{n}) \subseteq L$. Here $f(\underbrace{L \times \ldots \times L}_{n}) \equiv \{f(< a_1, \ldots, a_n >) \mid < a_1, \ldots, a_n > \in$ $\in \underbrace{L \times \ldots \times L}_{n}\}$.

Now, let D be set of composite objects (composed from certain components). Assume that universal set of such components is countable. Denote it as B. Lets declare surjective map $\beta : D \to 2^B$, where 2^B is set of all finite subsets of set B. Hence, for any $d \in D$ $\beta(d) \in 2^B$ is set of elements from B, which d composed of. Those elements are called denotates of d.

Property 2. n-ary function f β-preserves denotates, if finite set $B_f \subset B$ exists and for any $d \equiv < d_1, \ldots, d_n > \in \mathrm{dom}\, f$ expression $\beta(f(< d_1, \ldots, d_n >)) \subseteq$ $\subseteq \bigcup_{i=1}^{n} \beta(d_i) \cup B_f$ is correct.

Note that described properties of D-functions are preserved in signature Ω [10]. This allows to form several simple and essential conditions of completeness of A_D^{pr} CS [10].

Statement 1. Any complete system of A_D^{pr} contains at least one D-function that does not preserve set L for any non-empty set L $(L \subset D, L \neq \varnothing)$.

Statement 2. Any complete system of A_D^{pr} contains at least one D-function, which does not β-preserve denotates.

THE CONCEPT OF COMPLETENESS IN CLASSES OF PR-FUNCTIONS AND PR-PREDICATES

Consider general method for PPA of D-functions and D-predicates complete systems finding. It will be represented by series of interconnected results, introduced as proved lemmas and theorems. First, let us define some notions, useful conventions and denotations.

Let two countable sets D_1 and D_2 be defined. Assume that for every of those sets exists effective numeration $\alpha_1 : N \to D_1$ and $\alpha_2 : N \to D_2$. Also, PPAs

$A_{D_1}^{\text{pr}}$ and $A_{D_2}^{\text{pr}}$ are defined. Elements of sets D_1 and D_2 designated with lowercase letters: $a^1, b^1,...$ and $a^2, b^2,...$, may be with subscript. Let complete system σ_{D_1} of PPA $A_{D_1}^{\text{pr}}$ is defined and injective constructive mappings $\varphi: D_2 \to D_1$ and $\Phi: D_1 \to D_2$ are given. Sets $\varphi(D_2) \equiv \{\varphi(d) \mid d \in D_2\}$ and $\Phi(D_1) \equiv$ $\equiv \{\Phi(d) \mid d \in D_1\}$ are recursive [18]. Consider approach to solution of completeness problem for algebraic structure $A_{D_2}^{\text{pr}}$.

To designate D_1- and D_2-functions, lowercase ($f, g,...$) and uppercase ($F, G,...$) letters accordingly will be used. Letters $p, r,...$ and $P, R,...$ are used for designation of D_1- and D_2- predicates accordingly. When using termal notation, variables for D_1-functions and D_1-predicates are designated with lowercase roman letters $x, y, z,...$, and D_2-function and D_2-predicates — with lowercase Greek letters τ, ξ, $\pi,...$, subscripts and superscripts may be used in both cases.

Definition 1. $\varphi(D_2)$-function $f(x_1,...,x_n)$ is D_1-image of D_2-function $F(\tau_1,...,\tau_n)$, if for any $a_1^2,...,a_n^2 \in D_2$ expression $f(\varphi(a_1^2), ..., \varphi(a_n^2)) \cong$ $\cong \varphi(F(a_1^2,...,a_n^2))$ is true.

Definition 2. $\varphi(D_2)$-predicate $p(x_1,...,x_n)$ is D_1-image of D_2-predicate $P(\tau_1,...,\tau_n)$, if for any $a_1^2,...,a_n^2 \in D_2$ expression $p(\varphi(a_1^2),...,\varphi(a_n^2)) \cong$ $\cong P(a_1^2,...,a_n^2)$ is true.

Lets show that relations «to be an image of function» and «to be an image of predicate», declared with definitions listed, preserve property of partial recursiveness. In other words, listed theorem below is true.

Theorem 1. D_1-image of D_2-pr-function (D_2-pr-predicate) is D_1-pr-function (D_1-pr-predicate).

Indeed, it is easy to check that φ, as mapping of numerated set $\langle D_2, \alpha_2 \rangle$ to numerated set $\langle \varphi(D_2), \alpha_2 \cdot \varphi \rangle, \alpha_2 \cdot \varphi : N \to \varphi(D_2) : \forall k = 1,2,... \alpha_2 \cdot \varphi(k) \cong \varphi(\alpha_2(k))$ is pr-equivalence ([21], p. 150–160), because of constructiveness of mapping φ and effectiveness of numerations α_1 and α_2. Hereinafter in this document the designation $\alpha_2 \cdot \varphi$ means standard multiplication of functions α_2 and φ: $\text{dom}(\alpha_2 \cdot \varphi) \subseteq \text{dom}(\alpha_2)$, $\text{ran}(\alpha_2 \cdot \varphi) \subseteq \text{ran}(\varphi)$ and for any $d \in \text{dom}(\alpha_2 \cdot \varphi)$ value of this function $\alpha_2 \cdot \varphi(d) \equiv \varphi(\alpha_2(d))$.

After application of theorem 2.1.5 [17], lemma 1 will be true.

Lemma 1. D_1-image of D_2-pr-function (D_2-pr-predicate) is $\varphi(D_2)$-pr-function ($\varphi(D_2)$ - pr-predicate).

Hence, recursiveness of set $\varphi(D_2)$ results.

Lemma 2. Any $\varphi(D_2)$-pr-function is D_1-pr-function. The same for $\varphi(D_2)$-pr-predicates.

This lemma results truth of theorem 1.

Definition 3. D_2-function $F(\tau_1,...,\tau_n)$ is D_2-model of D_1-function $f(x_1,...,x_n)$, if expression $F(\Phi(a_1^1),...,\Phi(a_n^1)) \cong \Phi(f(a_1^1,...,a_n^1))$ holds for any $a_1^1,...,a_n^1 \in D_1$. D_2-model of D_1-predicate defined in the same way.

Let $\psi = \varphi \cdot \Phi$ ($f \cdot g$ is standard function multiplication, i.e. such function for which $\mathrm{dom}(f \cdot g) \subseteq \mathrm{dom}(f)$, $\mathrm{ran}(f \cdot g) \subseteq \mathrm{ran}(g)$ and which any $d \in \mathrm{dom}(f \cdot g)$ maps to value $f \cdot g(d) \equiv g(f(d))$, if $f(d) \in \mathrm{dom}(g)$). Obviously, that $\psi : D_2 \to \Phi(\varphi(D_2))$, $\Phi(\varphi(D_2)) \subseteq D_2$ — bijection. Thus, it is possible to assume that mapping, which inverse mapping to ψ, exists. Some mapping extension $\psi^{-1} : \Phi(\varphi(D_2)) \to D_2$ will be designated as $\chi : D_2 \to D_2$. In other words, D_2-functions ψ and χ are playing roles of coding and decoding functions accordingly. Let σ_{D_2} is designator for set of D_2-functions and D_2-predicates for which, firstly, D_2-model of D_1-function (D_1-predicate) from CS σ_{D_1} may be built from D_2-functions and D_2-predicates of set σ_{D_2} by finite number of application of operations from signature Ω and, secondly, D_2-functions ψ and χ may be built from D_2-functions and D_2-predicates of σ_{D_2} in analogical manner.

Definition 4. Sextuple $\sum \equiv < D_1, D_2, \sigma_{D_1}, \sigma_{D_2}, \psi, \chi >$ is called allowable system (AS) and tuple $< D_1, \sigma_{D_1} >$ is its basis.

Obviously, that in context of coding and decoding function lemma 3 true.

Lemma 3. Let $F(\tau_1,...,\tau_n)$ is D_2-pr-function, and $H(\pi_1,...,\pi_n)$ is D_2-model of D_1-image of function $F(\tau_1,...,\tau_n)$, then $F(a_1^2,...,a_n^2) \cong$ $\cong \chi(H(\psi(a_1^2),...,\psi(a_n^2)))$ is true for any $a_1^2,...,a_n^2 \in D_2$.

Lemma 4. Let $P(\tau_1,...,\tau_n)$ be D_2-pr-predicate, and $R(\pi_1,...,\pi_n)$ be D_2-model of D_1-image of predicate $P(\tau_1,...,\tau_n)$, then $P(a_1^2,...,a_n^2) \cong$ $\cong R(\psi(a_1^2),...,\psi(a_n^2))$ is true for any $a_1^2,...,a_n^2 \in D_2$.

Hence, theorem 2 is true.

Theorem 2. σ_{D_2} is CS of PPA $A_{D_2}^{\mathrm{pr}}$.

Considered, that there are few as general as possible requirements to sets D_1, D_2 and to its elements the nature of our constructions are maximally general. This allows to formulate simple, but effective condition of completeness of functions system in PPA.

So, if $D_1, D_2, \sigma_{D_1}, \sigma_{D_2}, \psi, \chi$ are objects, mentioned above, then theorem 3 is true.

Theorem 3. If $D_1, D_2, \sigma_{D_1}, \sigma_{D_2}, \psi, \chi$ is AS, then σ_{D_2} is CS of PPA $A_{D_2}^{\mathrm{pr}}$.

Results gained are giving complete enough idea about building method of complete systems for PPA of partially recursive functions and predicates on countable sets. This method will be applied below in order to solve problem of PPA completeness in class of pr-functions and pr-predicates on pragmatically significant in programming data type — set of records.

PPA OF PR-FUNCTIONS AND PR-PREDICATES ON SET OF RECORDS

Number of different intuitive interpretations of term «record» exists in information technologies and programming. Despite the fact that some interpretations of «record» significantly differ one from another, all of them tend to use adequately a concept of *named structures* to describe complex aggregated entities. Often, those interpretations burdened with minor partial details, blurring significance of naming mechanisms. However, as experience shows, named structures form «common denominator», through which all other aspects of problems solved should be considered. Namely, this tendency is basis for all following constructions.

Let V and W are non-empty countable sets of elements, interpreted as sets of names and values (denotates) accordingly. In general case it is allowed, that some names may play role of values and vice versa, i.e. it is possible that $V \cap W \neq \varnothing$.

We need to define some denotations, introduce main and auxiliary definitions in order to go further. Some of definitions will be given now, others — later, as may be necessary. All undefined terms and designations are given in [7].

One of the main concepts of this section is record. Set of all records on sets of names V and values W designated as $Z^{(V,W)}$. Now, introduce definition of record.

Definition 5. Record on sets of names V and values W (or simply record, it is clear from context what do V and W mean) is finite functional binary relation between set of names V and set of values W.

To designate record uppercase letters $I, J, K,...$ will be used. Lowercase letters $u, v, w,...$ are used to designate names of record elements, letters $a, b, c, d,...$ are their values and letters $\lambda, \mu, \eta,...$ are elements of records. In all cases subscripts may be used. Left subscripts and (or) superscripts may be used to designate names and (or) values of elements of record may be used. For example, let $\lambda = (v, a)$. Then such designations jf this element of record as ${}^{v}\lambda$, ${}_{a}\lambda$ and ${}^{v}_{a}\lambda$ may be used,

Hereinafter in the article so called «schemes», which represent name templates of correspondent records, may be used along with records.

Definition 6. The scheme of record K is finite set of names $\{v_1,...,v_n\}$, which represent projection of the record by the first component, i.e. $\{v_1,...,v_n\} = \mathrm{pr}_1(K)$, where pr_i is function of projection by i-th component of m-ary relation ($1 \le i \le m$) [7].

Scheme of the record I is designated as $sh(I) = \{v_1,...,v_n\}$, and record itself named for compactness as $sh(I)$-record or record of $sh(I)$ type. In case when type of record I must be defined explicitly, designation $I^{sh(I)}$ will be used. Set of all records of $\{v_1,...,v_n\}$ type designated as $Z[\{v_1,...,v_n\}]$. Couple particular cases of those notations take place: $I^{\varnothing} = \varnothing$ and $Z[\varnothing] = \{I^{\varnothing}\}$. As follows from above, it is obvious that $Z^{(V,W)} = \bigcup\limits_{V' \in 2^V} Z[V']$.

In unfolded notation record is designates as $I^{\{v_1,...,v_n\}} \equiv \{(v_1,d_1),...,(v_n,d_n)\}$.

For correct usage of numeric computability on set of records, it is required to prove existence of effective numeration of set $Z^{(V,W)}$. Given the countability of sets $V, W,$ as well as the fact that in this case is not so much important form of presentation names and values, how important their fundamentally different role, without limitation of subsequent constructions generality, we can assume that $V = W = N.$ It can be deduced from a context which of the roles meant in every single case. Therefore, any further formal constructions will be carried out on set $Z^{(N,N)}$, which is special case of $Z^{(V,W)}$.

Few steps need to be done to construct the numeration. Firstly, we need to take in account that for any non-empty record $I = \{(v_1,d_1),...,(v_m,d_m)\}$ its number concur with number of finite set $M_I = \{n_1,...,n_m\}$, where n_i is number of named element of record (v_i,d_i) (effective numeration of set N^2 defined in [12]). Number (unique identifier) for set M_I itself defined, for example, as $\alpha'(M_I) \equiv 2^{n_{j_1}+1} \cdot 3^{n_{j_2}+1} \cdot ... \cdot p_{m-1}^{n_{j_m}+1}$, where $n_{j_1} < n_{j_2} < ... < n_{j_m}$, $n_{j_s} \in M$, $s = 1,...,m$, and p_i, $i = 0, 1, 2, ...$ is i-th prime number ($p_0 = 2$, $p_1 = 3$, $p_2 = 5$, …). Than numeration $\alpha_{Z^{(N,N)}}$ of set $Z^{(N,N)}$ defined through piecewise scheme

$$\alpha'_{Z^{(N,N)}}(K) = \begin{cases} 1, & \text{if } I = \varnothing, \\ \alpha'(M_K), & \text{else}, \end{cases}$$

where K is certain record. Namely, $\alpha_{Z^{(N,N)}} \equiv (\alpha'_{Z^{(N,N)}})^{-1}$.

Now, consider to find of complete system of PPA $A^{\text{pr}}_{Z^{(N,N)}}$ itself. From the results gained above, conclusion followed that the solution of this problem reduces to the corresponding AS construction. Refer to concept of multi-set, mentioned in [14, 15]. Let U be some finite, may be empty, set.

Definition 7. Multi-set α with U basis is finitely defined function of $\alpha : U \to N^+$ type, where $N^+ \equiv N \setminus \{0\} = \{1, 2, 3, ...\}$. If designation of α basis is necessary, notation α^U will be used.

It would not be a great loss of generality, if we would assume that $U \subseteq N.$ Collection of all multi-sets with basis U designated as M_U. Then, obvious, that $M \equiv \bigcup_{U \in 2^N} M_U$ is set of all multi-sets (on N).

Elements of set M are designated with lowercase Greek letters $\alpha, \beta, \delta, ...,$ may be with subscripts and superscripts. Element of multi-set will be designated as tuple $< a, d >$, every component of which may be with subscript and superscript. Here a as the first component of tuple, its argument, the second is d — value (denotate, multiplicity). Two terms are related to multi-sets for convenient: characteristics χ_α and full image $f[\alpha]$. The first one is parametric function $\chi_\alpha : D \to N$ with values defined with piecewise scheme:

$$\chi_\alpha(a)=\begin{cases}\alpha(a), & if\ a\in \operatorname{dom}\alpha,\\ 0, & \text{else},\end{cases}\quad \text{for all } a\in N.$$

The second one is creates multi-set $f[\alpha]^{f(U)}$ from multi-set α^U and given function $f:N\to N$, where $f(U)$ is full image of set U relatively to function f, and characteristics of arbitrary argument a of this multi-set defined as: $\chi_{f[\alpha]^{f(U)}}(a)=\sum_{a'\in f^{-1}(a)}\chi_\alpha(a')$. Here $f^{-1}(a)$ is full image of element a relatively to function f. In case of empty set of summarands, sum assumed to be 0.

Now consider PPA A_M^{pr} of M-pr-functions and M-pr-predicates. Following collection σ_M of M-pr-functions and M-pr-predicates is of interest for us. It includes *predicate of equality* $\alpha=\beta$: $\alpha=\beta\Leftrightarrow\forall a(a\in N\Rightarrow\chi_\alpha(a)=\chi_\beta(a))$; *function of unification* $\bigcup_{All}$, which from two multi-sets α and β produces such multi-set $\alpha\bigcup_{All}\beta$, that for any argument a its characteristics equals to $\max(\chi_\alpha(a),\chi_\beta(a))$, i.e. $\chi_{\alpha\bigcup_{All}\beta}(a)=\max(\chi_\alpha(a),\chi_\beta(a))$; *function of direct junction* $\otimes$, which from two arbitrary multi-sets α^{U_α} and β^{U_β} produces new multi-set $(\alpha^{U_\alpha}\otimes\beta^{U_\beta})^{U_\alpha\times U_\beta}$, characteristics of arguments $<a_1,a_2>$ defined as: $\chi(<a_1,a_2>)=\chi_{\alpha^{U_\alpha}}(a_1)\cdot\chi_{\beta^{U_\beta}}(a_2),\forall a_1,a_2\in N$; *functions of addition* $\oplus$ *and subtraction* $\div$, defined with expressions $\alpha\oplus\beta=+[\alpha\otimes\beta]$ and $\alpha\div\beta=$ $=\dot{-}[\alpha\otimes\beta]$ accordingly (here «$\dot{-}$» — truncated distinction [18, 19]); *constant functions* $\{1^1\}(\alpha)$ and $\varnothing_m(\alpha)$, which produce multi-sets $\{<1,1>\}$ and $\varnothing_m\equiv\alpha^\varnothing$ accordingly; *function of multiplicity* φ which produces from two multi-sets $\{<n,1>\}$ and $\{<r,1>\}$ multi-set $\{<n,r>\}$; and selection functions I_m^n. The significance of collection σ_M described above is in the truth of.

Theorem (about multi-set PPA completeness). Collection $\sigma_M\equiv$ $\equiv\{=,\bigcup_{All},\oplus,\div,\{1^1\},\varnothing_m,\varphi,I_m^n\}_{m=1,\dots,n}^{n=1,2,3,\dots}$ is complete system of PPA $A_M^{чр}$ [14, 15].

The choice of multi-sets caused by relative simplicity of injective mappings $\varphi:Z^{(N,N)}\to M$ and $\Phi:M\to Z^{(N,N)}$ nature and obvious recursiveness of sets $\varphi(Z^{(N,N)})$ and $\Phi(M)$. These facts are creating reliable basis for solution completeness problem of PPA $A_{Z^{(N,N)}}^{чр}$. Injective mapping of set of records to multi-sets $\varphi:Z^{(V,W)}\to M$ is defined as

$$\varphi(K)=\begin{cases}\{<v_1,d_1+1>,<v_2,d_2+1>,\dots<v_m,d_m+1>\},\\ \qquad\qquad \text{if } K=\{(v_1,d_1),\dots,(v_m,d_m)\}\in Z^{(N,N)},\\ \alpha^\varnothing, \qquad \text{if } K=\varnothing.\end{cases}$$

Inverse mapping $\Phi:M\to Z^{(N,N)}$ is defined analogically

$$\Phi(\delta)=\begin{cases}\{(a_1,d_1-1),(a_2,d_2-1),...(a_m,d_m-1)\}, \\ \qquad\qquad \text{if } \delta=\{<a_1,d_1>,...,<a_m,d_m>\}\in M, \\ \varnothing, \qquad \text{if } \delta=\varnothing.\end{cases}$$

Lemma 5. M-image of $Z^{(N,N)}$-pr-function ($Z^{(N,N)}$-pr-predicate) is $\varphi(Z^{(N,N)})$-pr-function ($\varphi(Z^{(N,N)})$-pr-predicate).

From the lemma 5 it can be concluded that any $\varphi(Z^{(N,N)})$-pr-function is M-pr-function. Analogical conclusion may be done for $\varphi(Z^{(N,N)})$-pr-predicates. Thus consequence 1 is true.

Consequence 1. M-image of $Z^{(N,N)}$-pr-function ($Z^{(N,N)}$-pr-predicate) is M-function (M-predicate).

Consider following $Z^{(N,N)}$-pr-functions and $Z^{(N,N)}$-pr-predicates with simple, but representative examples for some of them. Beforehand let us define auxiliary parametric operation of record projection $pr_{\{v_{i_1},...,v_{i_k}\}}$, which maps any record $I\in Z^{(N,N)}$ to new record $pr_{\{v_{i_1},...,v_{i_k}\}}(I)\equiv I\cap(\{v_{i_1},...,v_{i_k}\}\times N)$. So, *predicate of equality* $=_Z$ is analogical to predicate of equality for multi-sets; *deletion by example* $\div^Z$: $I\div^Z J\equiv \mathrm{pr}_{\mathrm{pr}_1(I)\backslash \mathrm{pr}_1(J)}(I)$, particularly if $\mathrm{pr}_1(I)\cap\mathrm{pr}_1(J)=$ $=\varnothing$, then $I\div^Z J=I$, for example: $\{(1,3),(2,10),(5,7)\}\div^Z\{(1,1),(2,5),(3,7)\}=$ $=\{(5,7)\}$; $\{(5,7)\}\div^Z\{(6,5),(3,7)\}=\{(5,7)\}$ and $\{(6,5),(3,7)\}\div^Z\varnothing=\{(6,5),$ $(3,7)\}$; *records overlapping* ∇: for any $I,J\in Z^{(N,N)}$ $I\nabla J\equiv$ $\equiv J\bigcup \mathrm{pr}_{\mathrm{pr}_1(I)\backslash \mathrm{pr}_1(J)}(I)$, particularly, $I\nabla I^{\varnothing}=I;$ $I^{\varnothing}\nabla J=I^{\varnothing}$, and in case if $\mathrm{pr}_1(I)\cap\mathrm{pr}_1(J)=\varnothing$ $I\nabla J=J\nabla I=J\bigcup I$, for example, $\{(1,3),(5,7)\}\nabla\{(1,1),$ $(2,5),(3,7)\}=\{(1,1),(2,5),(3,7),(5,7)\}$; *append to record*

$$\overset{+}{U}:\overset{+}{U}(I)\equiv\begin{cases}I\bigcup\{(\max(pr_1(I))+1,0)\}, & \text{if } I\neq I^{\varnothing}, \\ \{(0,0)\}, & \text{if } I=I^{\varnothing},\end{cases} \quad \text{for any } I\in Z^{(N,N)}.$$

For example, for $I=\{(1,3),(2,10)\}$ and $J=I^{\varnothing}$, function will result $\overset{+}{U}(I)=$ $=\{(1,3),(2,10),(3,0)\}$ and $\overset{+}{U}(J)=\{(0,0)\}$ accordingly; *selection by maximal name* max: $\max(I)\equiv pr_{\max(\mathrm{pr}_1(I))}(I)$, $\max(I^{\varnothing})=I^{\varnothing}$; *zeroing of values* $\{0\}:\{0\}(I)\equiv J$, where $\mathrm{pr}_1(J)=\mathrm{pr}_1(I)\,\&\,\mathrm{pr}_2(J)=\{0\}$. For example, $\{0\}(\{(1,3),(2,10)\})=\{(1,0),(2,0)\}$; *increment* $\uparrow$: maps any non-empty record $I\in Z^{(N,N)}$ to record $\uparrow(I)$, $\uparrow(I)\equiv\{(v,a+1)|\ \forall(v,a)\in I\}$. *decrement* $\downarrow$: maps any non-empty record $I\in Z^{(N,N)}$ to record $\downarrow(I)$, which $\downarrow(I)=\left\{(v,b)\middle|v:\exists(v,a)\in I\,\&\,b:b=\begin{cases}a-1, & a>0, \\ 0, & a=0\end{cases}\right\}$. In case if $I=I^{\varnothing}$, $\uparrow(I^{\varnothing})=\downarrow(I^{\varnothing})=I^{\varnothing}$.

Designate set of $Z^{(N,N)}$ -pr-functions and $Z^{(N,N)}$ -pr-predicates described as $\sigma_{Z(V,W)} = \left\{ =_Z, \nabla, \div^Z, \overset{+}{U}, \max, \{0\}, \uparrow, \downarrow, I_m^n \right\}_{\substack{n=1,2,... \\ m=1,...,n}}$.

Analogically to previous section, consider $Z^{(N,N)}$ -functions ψ and χ — coding and decoding functions, accordingly. $\psi \equiv \varphi \cdot \Phi$ and χ is certain extension of mapping ψ^{-1}.

Therefore lemma 6 takes place.

Lemma 6. $Z^{(N,N)}$ -model of M -function (M -predicate) from set σ_M may be created from functions of $\sigma_{Z(N,N)}$ set with PPA operations.

It is easy to build $Z^{(N,N)}$ -models for M -pr-predicate of equality and M -pr-functions $\bigcup_{All}, \div, \{1^1\}, \varnothing_m, \varphi$. Let us build model of function for multi-sets addition $\oplus$. Now, introduce few auxiliary $Z^{(N,N)}$ -functions and $Z^{(N,N)}$ -predicates. Namely *identically false* and *identically true* predicates Fal and Tru: $Fal \equiv S^3(=, I_1, S^2(\overset{+}{U}, I_1))$ and $Tru \equiv S^3(=, I_1, I_1)$; *predicate of inequality* Neq: $Neq \equiv \Diamond(S^3(=, I_1, I_2), Fal, Tru)$; *constant empty record* $\varnothing^Z$: $\varnothing^Z \equiv S(\div^Z, I_1, I_1)$; *selection by pattern* Sel: $Sel(I_1, I_2) \equiv S^3(\div^Z, I_1, S^3(\div^Z, I_1, I_2))$; for example, $Sel(\{(1,1),(2,2)\}, \{(2,3),(4,5)\}) = \{(2,2)\}$ — function «selects» from record I_1 those components, names of which are in record I_2, i.e. I_2 is a kind of pattern for selection from I_1; *maximal addition* $+^{\max}$ of pair of records with same schemes: $+^{\max} \equiv *^3 S^3(Neq(I_2^2, S^2(\{0\}, I_2^2)), S^2(\uparrow, I_1^2), S^2(\downarrow, I_2^2))$. For example, for records $I_1 = \{(1,1),(2,2)\}$ and $I_2 = \{(1,3),(2,5)\}$ we will get $+^{\max}(I_1, I_2) =$ $= \{(1,1+5),(2,2+5)\} = \{(1,6),(2,7)\}$. Note, that operation of maximal addition in general case is non-commutative, i.e. $+^{\max}(I_1, I_2) \neq +^{\max}(I_2, I_1)$. Commutative property preserved only for records of special type, for example, for same-scheme single-element records. For instance, if $I_1 = \{(2,2)\}$ and $I_2 = \{(2,5)\}$, then $+^{\max}(I_1, I_2) = +^{\max}(I_2, I_1) = \{(2,7)\}$.

Now we can get down directly to building of $Z^{(N,N)}$ -pr-function $\oplus^Z$ — model of M -function $\oplus$. Assume that records $I_1 = \{(v_1^1, d_1^1),...,(v_{n_1}^1, d_{n_1}^1)\}$ and $I_2 = \{(v_1^2, d_1^2),...,(v_{n_2}^2, d_{n_2}^2)\}$ are given and $sh(I_1) \cap sh(I_2) = \{v_{r_1},...,v_{r_s}\}$. $\oplus^Z$ operation «breaks» records I_1 and I_2 to «segments», designated as $I_{1_1}, I_{1_2}, I_{2_1}$ and I_{2_2} with schemes $sh(I_{1_1}) = sh(I_1) \setminus \{v_{r_1},...,v_{r_s}\}$, $sh(I_{1_2}) = \{v_{r_1},...,v_{r_s}\}$, $sh(I_{2_1}) = \; = sh(I_2) \setminus \{v_{r_1},...,v_{r_s}\}$ and $sh(I_{2_2}) = \{v_{r_1},...,v_{r_s}\}$. Thus, resulting record may be represented as: $I_3 = I_{1_1} \cup I_{2_1} \cup I_{1_2} \oplus^Z I_{2_2}$. Note, $sh(I_{1_2}) = sh(I_{2_2})$. As for first two items I_{1_1} and I_{2_1}, they are easily created with earlier defined function $\div^Z$. Namely, $I_{1_1} = I_1 \div I_2$ and $I_{2_1} = I_2 \div I_1$. As for $I_{1_2} \oplus^Z I_{2_2}$, I_{1_2} and

I_{2_2} are easily defined with usage of function Sel: $I_{1_2} = Sel(I_1, I_2)$ and $I_{2_2} = Sel(I_2, I_1)$. Now model $Z^{(N,N)}$-function $\oplus^Z$ for same-scheme records I_{1_2} and I_{2_2}. It is easy to convince that $\oplus^Z$ may be represented as:

$$\oplus^Z = *^4(\underbrace{Neq(I_1^2, I_2^2))}_{p}, \underbrace{I_1^2 \nabla(\max(I_2^2) +^{\max} (Sel(I_1^2, \max(I_2^2)))}_{f_1}, \underbrace{I_2^2 \div^Z \max(I_2^2))}_{f_2}.$$

Obviously, in case when $sh(I_1) \cap sh(I_2) = \varnothing$, record $I_{1_2} \oplus^Z I_{2_2}$ is empty too, i.e. $I_{1_2} \oplus^Z I_{2_2} = \varnothing$ and, consequently, the result is $I_3 = I_1 \oplus^Z I_2 = I_{1_1} \cup I_{2_1}$. Taking in account that $sh(I_{1_1}) \cap sh(I_{2_1}) = \varnothing$, the result may be expressed as $I_3 = I_1 \oplus^Z I_2 = I_1 \nabla I_2 = I_{1_1} \nabla I_{2_1}$.

Lemma 7. Functions ψ and χ may be built from functions of set $\sigma_{Z(N,N)}$ by finite number of applications of PPA.

Correctness of the result is obvious, because of noted similarity of records and multi-sets, simplicity of coding and decoding mappings (φ and Φ), and adduced earlier statements. Hence, lemma 8 is correct.

Lemma 8. $\langle M, Z^{(N,N)}, \sigma_M, \sigma_{Z(N,N)}, \psi, \chi\rangle$ — AS with basis σ_M.

So, theorem 4 is true.

Theorem 4. $\sigma_{Z(N,N)} = \left\{ =_Z, \nabla, \div^Z, \overset{+}{U}, \max, \{0\}, \uparrow, \downarrow, I_m^n \right\}_{\substack{n=1,2,\ldots \\ m=1,\ldots,n}}$ — generative system of PPA $A^{чр}_{Z(N,N)}$.

Considering given above statements 1, 2 certain conclusions respectively to possible reducability of $\sigma_{Z(N,N)}$ may be made. Equality predicate cannot be excluded from $\sigma_{Z(N,N)}$ because it is sole predicate in CS. $\div^Z$ ($\overset{+}{U}(I)$, ∇, $\max$, $\{0\}$) — the only function in CS that does not preserve set $Z^{(N,N)} \setminus \{I^{\varnothing}\}$ ($Z^{(N,N)} \setminus \{\{(0,0)\}\}$, $\bigcup_{i \in N} Z[\{i\}]$, $\bigcup_{i \in N} Z[\{i, i+1\}]$, $\bigcup_{i \in N} Z[\{i\}] \setminus \{\{(i,0)\}\}$). Moreover, $\overset{+}{U}(I)$ does not β-preserve denotations with given such estimation $\beta : Z^{(N,N)} \to 2^N$ that $\beta(I^{\varnothing}) = \varnothing$ and $\beta(\{(v_1, d_1), \ldots, (v_n, d_n)\}) == \{v_1, \ldots, v_n, d_1, \ldots, d_n\}$, $n = 1,2,3,\ldots$ As for increment $\uparrow$ and decrement $\downarrow$ functions, they are in $\sigma_{Z(N,N)}$ simultaneously for convenience and symmetry, however they are not independent. For example, decrement may be easily produced by PPA operations from rest of the functions and predicate of $\sigma_{Z(N,N)}$ ($S^4(*^4(S^3(Neq, I_2^3, I_3^3), S(\uparrow, I_1^3), (\uparrow, I_2^3), I_3^3), \{0\}, S(\uparrow, I_1^1), I_1^1))$. The fact that $\uparrow$ as well as $\downarrow$ does not preserve denotations with given estimation β: for which $\beta(I^{\varnothing}) = \varnothing$ and $\beta(\{(v_1, d_1), \ldots, (v_n, d_n)\}) = \{d_1, \ldots, d_n\}$, $n = 1,2,3,\ldots$ directly results truthfulness of theorem 5.

Theroem 5. $\sigma_{Z(N,N)}^{I_m^n} \equiv \left\{ =_Z, \nabla, \div^Z, \overset{+}{U}, \max, \{0\}, \uparrow, I_m^n \right\}_{\substack{n=1,2,\ldots \\ m=1,\ldots,n}}$ — I_m^n-basis of PPA $A_{Z(N,N)}^{pr}$.

CONCLUSIONS

Modern IT problematic needs direct consideration of not only programming problems solutions, but processes, which lead to them. That is why researches of such processes organization structures are of paramount importance today. A special place in those researches takes problematic, connected with building of algebraic characteristics of pragmatically conditioned function classes, particularly, with solutions of completeness problems in corresponding algebras. In the paper these questions discussed on basis of primitive program algebras. Method of generative sets finding in PPA presented here, and applied to research of class of partially-recursive functions on records, which is of theoretical and applied importance. Using concepts of complete and allowable systems, and results received, especially criteria of completeness, universality of proposed method in classes of computable functions on different carriers proved.

Results received form foundations for development of adaptive programming environments. Next steps in this direction will be related with investigation of general concept of composition and development of functions exploring reduction methods connected with it as means of pragmatically driven decomposition of programming problems.

REFERENCES

1. *Dijkstra E.* A Discipline of Programming. — Prentice Hall, Inc., 1978. — 275 p.
2. *Brooks F.P.* The Design of Design: Essays from a Computer Scientist. — Addison-Wesley, 2010. — 448 p.
3. Brooks F.P. The Mythical Man-Month: Essays on Software Engineering. — Addison-Wesley, 1995. — 304 c.
4. *Redko V.N.* Fundamentals of compositional programming // Cybernetics and System Analysis. — 1979. — № 3. — P. 3–13.
5. *Redko V.N.* Semantical structures of software // Cybernetics and System Analysis. — 1981. — № 1. — P. 3–19.
6. *Redko V.N.* Universal program logics and their application // Proc. of 4th soviet-wide symp. — Kishenev, 1983. — P. 310–326.
7. *Basarab I.A., Nikitchenko N.S., Redko V.N.* Compositional databases. — K.: Lybid, 1992. — 92 p.
8. *Redko I.V., Redko V.N.* Existential basis of compositional paradigm // Programming and Computer Softtware. — 2008. — № 2. — P. 3–12.
9. *Bui D.B., Redko V.N.* Primitive program algebras I, II // Cybernetics. – 1985. — № 1. — P. 28–33.
10. *Bui D.B., Redko I.V.* Primitive program algebras of computable functions // Cybernetics. — 1987. — № 3. — P. 68–74.
11. *Bui D.B., Redko I.V.* Primitive program algebras of functions, which preserve denotates // Report of Ukrainian AS. — 1988. — № 9. — P. 66–68.
12. *Yershov U.L.* Theory of numerations. — M.: Nauka, 1977. — 416 p.

13. *Redko I.V., Snigur N.M*. Primitive program algebra of computable functions on graph // Naukovi Visti NTUU "KPI". — 2011. — № 4. — P. 75–80.
14. *Bogatyryova Y.O.* Computability on finite sets and multi-sets // Taras Shevchenko Kiev Nation University bulletin. Ser.: phys.-math. scienses. — 2010. — № 4. — P. 88–96.
15. *Bogatyryova Y.O.* Concept of multi-set. Structure of multi-sets family // Academitian M. Kravtchuk 13th international scientific conference, proceedings of (Kyiv, may 13–15, 2010). — K.: NTUU "KPI", 2009. — 60 p.
16. *Maltsev A.I.* Algorythmic systems. — M.: Nauka. — 1970. — 392 p.
17. *Maltsev A.I.* Constructive algebras. 1 // Uspekhi matematicheskih nauk. — 1961. — **6**. — № 3. — P. 3–60.
18. *Maltsev A.I.* Algorithms and recursive functions // Groningen: Wolters-Noordhoff, 1970. — 391 p.
19. *Cutland N.* Computability. An introduction in recursive function. — M.: Mir, 1983. — 256 p.
20. *Yershov A.P.* Computability in arbitrary fields and bases // Semiotics and Informatics. — 1982. — № 19. — P. 3–58.
21. *Maltsev A.I.* Selected works // Mathematical logic and general theory of algebraic systems. — 1976. — **2**. — M.: Nauka, 1976. — 388 p.

Received 13.05.2015

From the Editorial Board: the article corresponds completely to submitted manuscript.