PROCEEDINGS OF THE YEREVAN STATE UNIVERSITY

Physical and Mathematical Sciences 2017, **51**(2), p. 177–186

Informatics

ON TRANSLATION OF TYPED FUNCTIONAL PROGRAMS INTO UNTYPED FUNCTIONAL PROGRAMS

S. A. NIGIYAN [*], T. V. KHONDKARYAN [**]

Chair of Programming and Information Technologies YSU, Armenia

In this paper typed and untyped functional programs are considered. Typed functional programs use variables of any order and constants of order ≤ 1, where constants of order 1 are strong computable, λ-definable functions with indeterminate values of arguments. The basic semantics of a typed functional program is a function with indeterminate values of arguments, which is the main component of its least solution. The basic semantics of an untyped functional program is an untyped λ-term, which is defined by means of a fixed point combinator. An algorithm that translates typed functional program P into untyped functional program P' is suggested. It is proved that the basic semantics of the program P' λ-defines the basic semantics of the program P.

MSC2010: 68N18.

***Keywords*:** typed functional program, untyped functional program, basic semantics, translation, λ-definability.

Introduction. The paper is devoted to typed and untyped functional programs. A typed functional program is a system of equations (with separating variables) in the monotonic models of typed λ-calculus. Typed functional programs use variables of any order and constants of order ≤ 1, where constants of order 1 are strong computable, λ-definable functions with indeterminate values of arguments. The basic semantics of the typed functional program is a function with indeterminate values of arguments, which is the main component of its least solution (see [1–3]). An untyped functional program is a system of equations (with separating variables) in the untyped λ-calculus. The basic semantics of an untyped functional program is an untyped λ-term, which is defined by means of a fixed point combinator (see [4–6]). An algorithm that translates typed functional program P into untyped functional program P' is suggested. It is proved that the basic semantics of the program P' λ-defines the basic semantics of the program P.

[*] E-mail: nigiyan@ysu.am [**] E-mail: tigrankhondkaryan@gmail.com

Typed Functional Programs. The definitions of this section can be found in [1–3, 7, 8]. Let M be a partially ordered set, which has a least element $\perp$, which corresponds to an indeterminate value. For every $m \in M$ we have: $\perp \sqsubseteq m$ and $m \sqsubseteq m$, where $\sqsubseteq$ is a partial ordering on the set M.

Let us define the set of types (denoted by $Types$).

1. $M \in Types$;

2. If $\beta, \alpha_1, \ldots, \alpha_k \in Types$ $(k > 0)$, then the set of all monotonic mappings from $\alpha_1 \times \ldots \times \alpha_k$ into β (denoted by $[\alpha_1 \times \ldots \times \alpha_k \to \beta]$) belongs to $Types$.

Let $\alpha \in Types$, then the order of type α (denoted by $ord(\alpha)$) will be a natural number, which is defined in the following way: if $\alpha = M$, then $ord(\alpha) = 0$, if $\alpha = [\alpha_1 \times ... \times \alpha_k \to \beta]$, where $\beta, \alpha_1, ...\alpha_k \in Types,\ k > 0$, then $ord(\alpha) = 1 + max(ord(\alpha_1), \ldots,\ ord(\alpha_k), ord(\beta))$. If x is a variable of type α and a constant $c \in \alpha$, then $ord(x) = ord(c) = ord(\alpha)$.

Let $\alpha \in Types$ and V^T_α be a countable set of variables of type α, then $V^T = \bigcup\limits_{\alpha \in Types} V^T_\alpha$ is the set of all variables. The set of all terms denoted by $\Lambda^T = \bigcup\limits_{\alpha \in Types} \Lambda^T_\alpha$, where Λ^T_α is the set of terms of type α, is defined the following way:

1. If $c \in \alpha$, $\alpha \in Types$, then $c \in \Lambda^T_\alpha$;

2. If $x \in V^T_\alpha, \alpha \in Types$, then $x \in \Lambda^T_\alpha$;

3. If $\tau \in \Lambda^T_{[\alpha_1 \times \ldots \times \alpha_k \to \beta]}$, $t_i \in \Lambda^T_{\alpha_i}$, where $\beta, \alpha_i \in Types, i = 1, \ldots, k,\ k \geq 1$, then $\tau(t_1, \ldots, t_k) \in \Lambda^T_\beta$ (the operation of application);

4. If $\tau \in \Lambda^T_\beta$, $x_i \in V^T_{\alpha_i}$, where $\beta, \alpha_i \in Types,\ i \neq j \Rightarrow x_i \neq x_j,\ i, j = 1, \ldots, k$, $k \geq 1$, then $\lambda x_1 \ldots x_k[\tau] \in \Lambda^T_{[\alpha_1 \times \ldots \times \alpha_k \to \beta]}$ (the operation of abstraction).

The notions of free and bound occurrences of variables in terms as well as the notion of a free variable are introduced in the conventional way. The set of all free variables of a term t is denoted by $FV(t)$. A term which doesn't contain free variables is called a closed term. Terms t_1 and t_2 are said to be congruent (which is denoted by $t_1 \equiv t_2$), if one term can be obtained from the other by renaming bound variables. In what follows congruent terms are considered identical.

Let $t \in \Lambda^T_\alpha,\ \alpha \in Types$ and $FV(t) \subset \{y_1, \ldots, y_n\}$, $\bar{y_0} = \langle y^0_1, \ldots, y^0_n \rangle$, where $y_i \in V^T_{\beta_i}$, $y^0_i \in \beta_i$, $\beta_i \in Types,\ i = 1, \ldots, n,\ n \geq 0$. The value of the term t for the values of the variables $y_1, \ldots, y_n$ equal to $\bar{y_0} = \langle y^0_1, \ldots, y^0_n \rangle$ is denoted by $Val_{\bar{y_0}}(t)$ and defined as follows:

1. If $t \equiv c$ and $c \in \alpha$, then $Val_{\bar{y_0}}(c) = c$;

2. If $t \equiv x, x \in V^T_\alpha$, then $Val_{\bar{y_0}}(x) = y^0_i$, where $FV(x) = \{x\} \subset \{y_1, \ldots, y_n\}$ and $x \equiv y_i,\ i = 1, \ldots, n,\ n \geq 1$;

3. If $t \equiv \tau(t_1, ..., t_k) \in \Lambda^T_\alpha$, where $\tau \in \Lambda^T_{[\alpha_1 \times \ldots \times \alpha_k \to \alpha]}, t_i \in \Lambda^T_{\alpha_i},\ \alpha_i \in Types$; $i = 1, \ldots, k,\ k \geq 1$, then $Val_{\bar{y_0}}(\tau(t_1, \ldots, t_k)) = Val_{\bar{y_0}}(\tau)(Val_{\bar{y_0}}(t_1), \ldots, Val_{\bar{y_0}}(t_k))$;

4. If $t \equiv \lambda x_1 \ldots x_k[\tau] \in \Lambda^T_\alpha$, where $\alpha = [\alpha_1 \times \ldots \times \alpha_k \to \beta], \tau \in \Lambda^T_\beta, x_i \in V^T_{\alpha_i}$, $\beta, \alpha_i \in Types,\ i = 1, \ldots, k,\ k \geq 1$, then $Val_{\bar{y_0}}(\lambda x_1 \ldots x_k[\tau]) \in [\alpha_1 \times \ldots \times \alpha_k \to \beta]$ and is defined as follows: let $\{y_1, \ldots, y_n\} \setminus \{x_1, \ldots, x_k\} = \{y_{j_1}, \ldots, y_{j_s}\},\ s \geq 0$, and

$\bar{z_0} = \langle y^0_{j_1}, \dots, y^0_{j_s} \rangle$, then for any $\bar{x_0} = \langle x^0_1, \dots, x^0_k \rangle$, where $x^0_i \in \alpha_i$, $i = 1, \dots, k$, $Val_{\bar{y_0}}(\lambda x_1 \dots x_k[\tau])(x^0_1, \dots, x^0_k) = Val_{\bar{x_0}, \bar{z_0}}(\tau)$, where $\bar{x_0}, \bar{z_0} = \langle x^0_1, \dots, x^0_k, y^0_{j_1}, \dots, y^0_{j_s} \rangle$.

It follows from [1], that for any $\bar{y_0} = \langle y^0_1, \dots, y^0_n \rangle$ and $\bar{y_1} = \langle y^1_1, \dots, y^1_n \rangle$ such that $\bar{y_0} \sqsubseteq \bar{y_1}$, where $y^0_i, y^1_i \in \beta_i$ $(1 \le i \le n)$, we have the following:

1. $Val_{\bar{y_0}}(t) \in \alpha$;
2. $Val_{\bar{y_0}}(t) \sqsubseteq Val_{\bar{y_1}}(t)$.

Let terms $t_1, t_2 \in \Lambda^T_\alpha$, $\alpha \in Types$, $FV(t_1) \cup FV(t_2) = \{y_1, \dots, y_n\}$, $y_i \in V^T_{\beta_i}$, $\beta_i \in Types$, $i = 1, \dots, n$, $n \ge 0$, then terms t_1 and t_2 are called equivalent (denoted by $t_1 \sim t_2$), if for any $\bar{y_0} = \langle y^0_1, \dots, y^0_n \rangle$, where $y^0_i \in \beta_i$, $i = 1, \dots, n$, we have the following: $Val_{\bar{y_0}}(t_1) = Val_{\bar{y_0}}(t_2)$. A term $t \in \Lambda^T_\alpha, \alpha \in Types$, is called a constant term with $a \in \alpha$ value, if $t \sim a$.

To show mutually different variables of interest $x_1, \dots, x_k$, $k \ge 1$, of a term t, the notation $t[x_1, \dots, x_k]$ is used. The notation $t[t_1, \dots, t_k]$ denotes the term obtained by the simultaneous substitution of the terms $t_1, \dots, t_k$ for all free occurrences of variables $x_1, \dots, x_k$ respectively, where $x_i \in V^T_{\alpha_i}, i \ne j \Rightarrow x_i \not\equiv x_j, t_i \in \Lambda^T_{\alpha_i}, \alpha_i \in Types$, $i, j = 1, \dots, k$, $k \ge 1$. A substitution is said to be admissible, if all free variables of the term being substituted remain free after the substitution. We will consider only admissible substitutions.

A term $t \in \Lambda^T$ with a fixed occurrence of a subterm $\tau_1 \in \Lambda^T_\alpha$, where $\alpha \in Types$, is denoted by t_{τ_1} and a term with this occurrence of τ_1 replaced by τ_2, where $\tau_2 \in \Lambda^T_\alpha$, is denoted by t_{τ_2}.

Further, we assume that M is a recursive set and considered terms use variables of any order and constants of order ≤ 1, where constants of order 1 are strong computable, monotonic functions with indeterminate values of arguments. A function $f : M^k \to M, k \ge 1$, with indeterminate values of arguments is said to be strong computable, if there exists an algorithm, which stops with value $f(m_1, \dots, m_k)$ for all $m_1, \dots, m_k \in M$ (see [3]). We suppose that each strong computable function with indeterminate values of arguments is given by its algorithm. Hereafter all such terms will be denoted by Λ^T and all such terms of type α will be denoted by Λ^T_α.

A term of the form $\lambda x_1 \dots x_k[\tau[x_1, \dots, x_k]](t_1, \dots, t_k)$, where $x_i \in V^T_{\alpha_i}, i \ne j \Rightarrow x_i \not\equiv x_j$, $\tau \in \Lambda^T$, $t_i \in \Lambda^T_{\alpha_i}, \alpha_i \in Types$, $i, j = 1, \dots, k$, $k \ge 1$, is called a β-redex, its convolution is the term $\tau[t_1, \dots, t_k]$.

A term t_1 is said to be obtained from a term t_0 by one-step β-reduction (denoted by $t_0 \to_\beta t_1$), if $t_0 \equiv t_{\tau_0}$, $t_1 \equiv t_{\tau_1}, \tau_0$ is a β-redex and τ_1 is its convolution. A term t is said to be obtained from a term t_0 by β-reduction (denoted by $t_0 \to\to_\beta t$), if there exists a finite sequence of terms $t_1, \dots, t_n$ $(n \ge 1)$ such that $t_1 \equiv t_0$, $t_n \equiv t$ and $t_i \to_\beta t_{i+1}$, where $i = 1, \dots, n-1$.

δ-redex has a form $f(t_1, \dots, t_k)$, where $f \in [M^k \to M]$, $t_i \in \Lambda^T_M$, $i = 1, \dots, k$, $k \ge 1$, its convolution is either $m \in M$ and in this case $f(t_1, \dots, t_k) \sim m$ or a subterm t_i and in this case $f(t_1, \dots, t_k) \sim t_i$, $i = 1, \dots, k$. A term t_1 is said to be obtained from a term t_0 by one-step δ-reduction (denoted by $t_0 \to_\delta t_1$), if $t_0 \equiv t_{\tau_0}$, $t_1 \equiv t_{\tau_1}, \tau_0$ is a δ-redex and τ_1 is its convolution. A term t is said to be obtained from a term t_0 by δ-reduction (denoted by $t_0 \to\to_\delta t$), if there exists a finite sequence of terms $t_1, \dots, t_n$

$(n \geq 1)$ such that $t_1 \equiv t_0, t_n \equiv t$ and $t_i \to_\delta t_{i+1}$, where $i = 1, \ldots, n-1$.

A term t_1 is said to be obtained from a term t_0 by one-step $\beta\delta$-reduction (denoted by $t_0 \to_{\beta\delta} t_1$), if either $t_0 \to_\beta t_1$ or $t_0 \to_\delta t_1$. A term t is said to be obtained from a term t_0 by $\beta\delta$-reduction (denoted by $t_0 \to\to_{\beta\delta} t$), if there exists a finite sequence of terms $t_1, \ldots, t_n$ $(n \geq 1)$ such that $t_1 \equiv t_0$, $t_n \equiv t$ and $t_i \to_{\beta\delta} t_{i+1}$, where $i = 1, \ldots, n-1$. It follows from [7], that if $t_1 \to\to_{\beta\delta} t_2$, then $t_1 \sim t_2$, where $t_1, t_2 \in \Lambda^T_\alpha, \alpha \in Types$.

A term containing no $\beta\delta$-redexes is called a normal form. The set of all normal forms is denoted by NF^T. For every term $t \in \Lambda^T$ there exists a term $\tau \in NF^T$ such that $t \to\to_{\beta\delta} \tau$ (see [7]).

A notion of δ-reduction is called a single-valued notion of δ-reduction, if δ is a single-valued relation, i.e. if $\langle \tau_0, \tau_1 \rangle \in \delta$ and $\langle \tau_0, \tau_2 \rangle \in \delta$, then $\tau_1 \equiv \tau_2$, where $\tau_0, \tau_1, \tau_2 \in \Lambda^T_M$.

A notion of δ-reduction is called effective notion of δ-reduction, if there exists an algorithm, which for any term $f(t_1, \ldots, t_k)$, where $f \in [M^k \to M]$, $t_i \in \Lambda^T_M$, $i = 1, \ldots, k$, $k \geq 1$, gives its convolution, if $f(t_1, \ldots, t_k)$ is a δ-redex and stops with a negative answer otherwise.

Definition 1. An effective, single-valued notion of δ-reduction is called a canonical notion of δ-reduction if:

1. $t \in \Lambda^T_M,\ t \sim m,\ m \in M \setminus \{\perp\} \Rightarrow t \to\to_{\beta\delta} m$;
2. $t \in \Lambda^T_M,\ FV(t) = \emptyset,\ t \sim \perp \Rightarrow t \to\to_{\beta\delta} \perp$.

In [8] it was proved that for every recursive set of strong computable, monotonic functions with indeterminate values of arguments there exists a canonical notion of δ-reduction. Further we will only use the canonical notion of δ-reduction.

Typed functional program P is the following system of equations:

$$\begin{cases} F_1 = t_1[F_1, \ldots, F_n], \\ \qquad \cdots \\ F_n = t_n[F_1, \ldots, F_n], \end{cases} \tag{1}$$

where $F_i \in V_{\alpha_i}, i \neq j \Rightarrow F_i \not\equiv F_j, t_i[F_1, ..., F_n] \in \Lambda_{\alpha_i}, FV(t_i[F_1, \ldots, .F_n]) \subset \{F_1, \ldots, F_n\}$, $\alpha_i \in Types,\ i, j = 1, \ldots, n,\ n \geq 1, \alpha_1 = [M^k \to M],\ k \geq 1$.

We consider the mapping $\Psi_P : \alpha_1 \times \ldots \times \alpha_n \to \alpha_1 \times \ldots \times \alpha_n$, which is defined as follows: if $\bar{g} = \langle g_1, \ldots, g_n \rangle$, where $g_i \in \alpha_i$, $i = 1, \ldots, n$, then $\Psi_P(\bar{g}) = \langle Val_{\bar{g}}(t_1[F_1, \ldots, F_n]), \ldots, Val_{\bar{g}}(t_n[F_1, \ldots, F_n]) \rangle$. $\bar{g}$ is said to be the solution of the program P, if $\Psi_P(\bar{g}) = \bar{g}$, i.e. $\langle Val_{\bar{g}}(t_1[F_1, \ldots, F_n]), \ldots, Val_{\bar{g}}(t_n[F_1, \ldots, F_n]) \rangle = \langle g_1, \ldots, g_n \rangle$. Every typed functional program P has a least solution (see [1]). Let $\langle f_1, \ldots, f_n \rangle \in \alpha_1 \times \ldots \times \alpha_n$ be the least solution of P, then the first component $f_1 \in [M^k \to M]$ of the least solution is said to be the basic semantics of the program P and is denoted by f_P.

Theorem 1. (On basic semantics of typed functional programs). Let $f_P \in [M^k \to M]$, $k \geq 1$, be the basic semantics of a typed functional program P of the form (1), then for all $m_1, \ldots, m_k \in M$ we have:

$$f_P(m_1, \ldots m_k) = \sup\{\Psi^s_P(\bar{\Omega})_1(m_1, \ldots, m_k) \mid s < \omega\},$$

where $\bar{\Omega}$ is the least element of the set $\alpha_1 \times \ldots \times \alpha_n$, $\Psi_P^0(\bar{\Omega}) = \bar{\Omega}$, $\Psi_P^{s+1}(\bar{\Omega}) = \Psi_P(\Psi_P^s(\bar{\Omega}))$ and $\Psi_P^s(\bar{\Omega})_1$ is the first component of the $\Psi_P^s(\bar{\Omega})$, $s < \omega$, ω is the ordinal corresponding to the set of the natural numbers.

Proof. Follows from the results of [2].

Theorem 2. (On substitutions for typed functional programs). Let $f_P \in [M^k \to M]$, $k \geq 1$, be the basic semantics of a typed functional program P of the form (1), then for all $m_1, \ldots, m_k \in M$ we have:

$$f_P(m_1, \ldots, m_k) = m \neq \bot \Leftrightarrow \exists s \geq 1, t_1^s[F_1, \ldots, F_n](m_1, \ldots, m_k) \twoheadrightarrow_{\beta\delta} m \neq \bot,$$

where $t_i^0[F_1, \ldots, F_n] \equiv F_i$, $t_i^r[F_1, \ldots, F_n] \equiv t_i[t_1^{r-1}[F_1, \ldots, F_n], \ldots, t_n^{r-1}[F_1, \ldots, F_n]]$, $r \geq 1$, $i = 1, \ldots, n$, $n \geq 1$.

Proof. If $f_P(m_1, \ldots, m_k) = m \neq \bot$, then, according to the Theorem 1, $\exists s \geq 1, \Psi_P^s(\bar{\Omega})_1(m_1, \ldots, m_k) = m$, where $\bar{\Omega}$ is the least element of the set $\alpha_1 \times \ldots \times \alpha_n$. Therefore, $Val_{\bar{\Omega}}(t_1^s[F_1, \ldots, F_n](m_1, \ldots, m_k)) = m$. Since for every $\bar{g} \in \alpha_1 \times \ldots \times \alpha_n$, $\bar{\Omega} \sqsubseteq \bar{g}$, then $Val_{\bar{g}}(t_1^s[F_1, \ldots, F_n](m_1, \ldots, m_k)) = m$ and $t_1^s[F_1, \ldots, F_n](m_1, \ldots, m_k) \sim m$. Therefore, according to the Definition 1 (point 1), $t_1^s[F_1, \ldots, F_n](m_1, \ldots, m_k) \twoheadrightarrow_{\beta\delta} m$.

If $\exists s \geq 1, t_1^s[F_1, \ldots, F_n](m_1, \ldots, m_k) \twoheadrightarrow_{\beta\delta} m \neq \bot$, then, according to [7], $t_1^s[F_1, \ldots, F_n](m_1, \ldots, m_k) \sim m$ and $Val_{\bar{\Omega}}(t_1^s[F_1, \ldots, F_n](m_1, \ldots, m_k)) = m$. Therefore, $\Psi_P^s(\bar{\Omega})_1(m_1, \ldots, m_k) = m$ and, according to the Theorem 1, $f_P(m_1, \ldots, m_k) = m$. $\square$

Untyped Functional Programs. The definitions of this section can be found in [4–6]. Let us fix a countable set of variables V. The set Λ of terms is defined as follows:

1. If $x \in V$, then $x \in \Lambda$;
2. If $t_1, t_2 \in \Lambda$, then $(t_1 t_2) \in \Lambda$ (the operation of application);
3. If $x \in V$ and $t \in \Lambda$, then $(\lambda x t) \in \Lambda$ (the operation of abstraction).

The following shorthand notations are introduced: a term $(...(t_1 t_2)...t_k)$, where $t_i \in \Lambda, i = 1, \ldots, k$, $k > 1$, is denoted by $t_1 t_2 \ldots t_k$ and a term $(\lambda x_1(\lambda x_2(\ldots(\lambda x_n t)\ldots)))$, where $x_j \in V, j = 1, \ldots, n$, $n > 0$, $t \in \Lambda$, is denoted by $\lambda x_1 x_2 \ldots x_n.t$.

The notions of free and bound occurrences of variables in terms as well as the notion of free variable are introduced in the conventional way. The set of all free variables of a term t is denoted by $FV(t)$. A term, which does not contain free variables, is called a closed term. Terms t_1 and t_2 are said to be congruent (which is denoted by $t_1 \equiv t_2$), if one term can be obtained from the other by renaming bound variables. In what follows congruent terms are considered identical.

To show mutually different variables of interest $x_1, \ldots, x_k$, $k \geq 1$, of a term t the notation $t[x_1, \ldots, x_k]$ is used. The notation $t[t_1, \ldots, t_k]$ denotes the term obtained by the simultaneous substitution of the terms $t_1, \ldots, t_k$ for all free occurrences of variables $x_1, \ldots, x_k$ respectively, $i \neq j \Rightarrow x_i \not\equiv x_j$, $i, j = 1, \ldots, k$, $k \geq 1$. A substitution is said to be admissible, if all free variables of the term being substituted remain free after the substitution. We will consider only admissible substitutions.

A term t with a fixed occurrence of a subterm τ_1 is denoted by t_{τ_1} and a term with this occurrence of τ_1 replaced by a term τ_2 is denoted by t_{τ_2}.

A term of the form $(\lambda x.t[x])\tau$ is called a β-redex and the term $t[\tau]$ is called its convolution. A term t_1 is said to be obtained from a term t_0 by one-step β-reduction (denoted by $t_0 \to_\beta t_1$), if $t_0 \equiv t_{\tau_0}, t_1 \equiv t_{\tau_1}, \tau_0$ is a β-redex and τ_1 is its convolution. A term t is said to be obtained from a term t_0 by β-reduction (denoted by $t_0 \twoheadrightarrow_\beta t$), if there exists a finite sequence of terms $t_1, \dots, t_n$ $(n \geq 1)$ such that $t_1 \equiv t_0,\ t_n \equiv t$ and $t_i \to_\beta t_{i+1}$, where $i = 1, \dots, n-1$. A term containing no β-redexes is called a normal form. The set of all normal forms is denoted by NF and the set of all closed normal forms is denoted by NF^0. A term t is said to have a normal form, if there exists a term τ such that $\tau \in NF$ and $t \twoheadrightarrow_\beta \tau$. Of the Church–Rosser theorem [4] it follows, that if $t \twoheadrightarrow_\beta \tau_1, t \twoheadrightarrow_\beta \tau_2, \tau_1, \tau_2 \in NF$, then $\tau_1 \equiv \tau_2$.

If a term has a form $\lambda x_1 \dots x_k.xt_1 \dots t_n$, where $x_1, \dots, x_k, x \in V,\ t_1, \dots, t_n \in \Lambda$, $k, n \geq 0$, it is called a head normal form and x is called its head variable. The set of all head normal forms is denoted by HNF. A term t is said to have a head normal form, if there exists a term τ such that $\tau \in HNF$ and $t \twoheadrightarrow_\beta \tau$. It is known that $NF \subset HNF$, but $HNF \not\subset NF$ (see [4]).

Let $\lambda x_1, \dots, x_k.((\lambda x.t)\tau)t_1 \dots t_n$ be a term, where $x_1, \dots, x_k, x \in V,\ t_1, \dots, t_n, t,$ $\tau \in \Lambda,\ k, n \geq 0$, then the β-redex $(\lambda x.t)\tau$ is called a head β-redex. It is obvious that every head β-redex of the term is its left β-redex, but not every left β-redex of the term is its head β-redex.

Recall, that if a term has a head normal form, then the reducing chain, where always the head β-redex is chosen, leads to a head normal form, and if the term has a normal form, then the reducing chain, where always the left β-redex is chosen, leads to the normal form (see [4]).

Let us give the notion of β-equality (denoted by $=_\beta$):

1. $t_1 \twoheadrightarrow_\beta t_2 \Rightarrow t_1 =_\beta t_2$;
2. $t_1 =_\beta t_2 \Rightarrow t_2 =_\beta t_1$;
3. $t_1 =_\beta t_2, t_2 =_\beta t_3 \Rightarrow t_1 =_\beta t_3$, where $t_1, t_2, t_3 \in \Lambda$.

A term $Z \in \Lambda$ is called a fixed point combinator, if for all terms $t \in \Lambda$ we have:

$$Zt =_\beta t(Zt).$$

L e m m a 1. Let Z be a fixed point combinator and a term $t \in \Lambda$, then there exists a sequence of terms $Z_0^t, Z_1^t, Z_2^t, \dots$ such that $Z_0^t \equiv Zt$ and $Z_s^t \twoheadrightarrow_\beta tZ_{s+1}^t$, $s = 0, 1, 2, \dots$

P r o o f . Let $x \in V$ and $Z_0^x \equiv Zx =_\beta x(Zx)$, according to the Church–Rosser's theorem, there exists such term Z_1^x that $x(Zx) \twoheadrightarrow_\beta xZ_1^x$ and $Z_0^x \twoheadrightarrow_\beta xZ_1^x$, therefore, $Z_1^x =_\beta Zx =_\beta x(Zx)$ and there exists such term Z_2^x that $x(Zx) \twoheadrightarrow_\beta xZ_2^x$ and $Z_1^x \twoheadrightarrow_\beta$ xZ_2^x and so on. Thus, for every $s \geq 0$ we get a term Z_s^x, such that $Z_s^x =_\beta Zx =_\beta$ $x(Zx)$ and there exists a term Z_{s+1}^x such that $x(Zx) \twoheadrightarrow_\beta xZ_{s+1}^x$ and $Z_s^x \twoheadrightarrow xZ_{s+1}^x$. Let $Z_s^x \equiv Z_s^x[x], Z_{s+1}^x \equiv Z_{s+1}^x[x]$ and $Z_s^t \equiv Z_s^x[t], Z_{s+1}^t \equiv Z_{s+1}^x[t]$, then $Z_s^t \twoheadrightarrow_\beta tZ_{s+1}^t$, $s = 0, 1, 2, \dots$ □

We introduce notations for some terms:
$\langle t_1, \dots, t_n \rangle \equiv \lambda x.xt_1 \dots t_n$, where $x \in V,\ t_i \in \Lambda,\ x \notin FV(t_i),\ i = 1, \dots, n,\ n \geq 1$;
$U_i^n \equiv \lambda x_1 \dots x_n.x_i$, where $x_j \in V,\ k \neq j \Rightarrow x_k \not\equiv x_j,\ k, j = 1, \dots, n,\ 1 \leq i \leq n,\ n \geq 1$;
$P_i^n \equiv \lambda x.xU_i^n$, where $x \in V,\ 1 \leq i \leq n,\ n \geq 1$; $\Omega \equiv (\lambda x.xx)(\lambda x.xx)$, where $x \in V$.

Untyped functional program P is the following system of equations:

$$\begin{cases} F_1 = t_1[F_1,\dots,F_n], \\ \quad \dots \\ F_n = t_n[F_1,\dots,F_n], \end{cases} \qquad (2)$$

where $F_i \in V, i \neq j \Rightarrow F_i \not\equiv F_j,\ \ t_i[F_1,...,F_n] \in \Lambda,\ \ FV(t_i[F_1,\dots,F_n]) \subset \{F_1,\dots,F_n\}$, $i,j = 1,\dots,n,\ \ n \geq 1$.

A sequence of terms $(\tau_1,\dots,\tau_n)$ will be a solution of the program P, if for all $i = 1,\dots,n$ we have:

$$\tau_i =_\beta t_i[\tau_1,\dots,\tau_n].$$

We consider the solution $(\tau_1,\dots,\tau_n)$ of the program P, where $\tau_i \equiv P_i^n(Z(\lambda x.\langle t_1[P_1^n x,\dots,P_n^n x],\dots,t_n[P_1^n x,\dots,P_n^n x]\rangle)),\ \ x \in V,\ \ Z \in \Lambda$ and is a fixed point combinator, $i = 1,\dots,n$. The term τ_1 is the basic semantics of the program P denoted by τ_P.

$Fix(P,Z) = \{(\upsilon_1,\dots,\upsilon_k,t_0) | \tau_P \upsilon_1 \dots \upsilon_k \to\to_\beta t_0,\ \ \ t_0 \in NF^0,\ \upsilon_j \in NF^0$ or $\upsilon_j \equiv \Omega, j = 1,\dots,k,\ \ k \geq 0\}$.

Theorem 3. (On the invariance of the baic semantics of untyped functional programs). For any untyped functional program P and for any fixed point combinators Z_1, Z_2 we have:

$$Fix(P,Z_1) = Fix(P,Z_2).$$

Proof. Follows from the results of [6].

Theorem 4. (On substitutions for untyped functional programs). Let τ_P be the basic semantics of an untyped functional program P of the form (2), $\upsilon_1,\upsilon_2,\dots,\upsilon_k$ be terms, where $\upsilon_j \in NF^0$ or $\upsilon_j \equiv \Omega,\ \ \ j = 1,\dots,k,\ \ \ k \geq 0$ and $t_0 \in NF^0$, then:

$$\tau_P \upsilon_1 \dots \upsilon_k \to\to_\beta t_0 \Leftrightarrow \exists s \geq 1,\ \ t_1^s[F_1,\dots,F_n]\upsilon_1\upsilon_2\dots\upsilon_k \to\to_\beta t_0,$$

where $t_i^0[F_1,\dots,F_n] \equiv F_i,\ \ \ t_i^r[F_1,\dots,F_n] \equiv t_i[t_1^{r-1}[F_1,\dots,F_n],\dots,t_n^{r-1}[F_1,\dots,F_n]]$, $r \geq 1,\ \ i = 1,\dots,n,\ \ n \geq 1$.

Proof. Follows from the results of [5,6].

Lemma 2. Let P be an untyped functional program of the form (2) and Z be a fixed point combinator. Then, there exists a sequence of terms $T_0, T_1,\dots$ such that $T_0 \equiv Z(\lambda x.\langle t_1[P_1^n x,\dots,P_n^n x],\dots,t_n[P_1^n x,\dots,P_n^n x]\rangle)$ and for any $s \geq 0$ we have: $P_i^n T_s \to\to_\beta t_i[P_1^n T_{s+1},\dots,P_n^n T_{s+1}],\ \ 1 \leq i \leq n,\ \ n \geq 1$.

Proof. Let $t \equiv \lambda x.\langle t_1[P_1^n x,\dots,P_n^n x],\dots,t_n[P_1^n x,\dots,P_n^n x]\rangle$, from the Lemma 1 it follows that there exists a sequence of the terms $Z_0^t, Z_1^t, Z_2^t,\dots$ such that $Z_s^t \to\to_\beta tZ_{s+1}^t$ for any $s \geq 0$. Let $T_s \equiv Z_s^t$ for any $s \geq 0$, then:

$P_i^n T_s \to\to_\beta P_i^n((\lambda x.\langle t_1[P_1^n x,...,P_n^n x],...,t_n[P_1^n x,...,P_n^n x]\rangle)T_{s+1}) \to\to_\beta$
$P_i^n(\langle t_1[P_1^n T_{s+1},...,P_n^n T_{s+1}],...,t_n[P_1^n T_{s+1},...,P_n^n T_{s+1}]\rangle) \to\to_\beta t_i[P_1^n T_{s+1},...,P_n^n T_{s+1}]$. □

3. Translation. Let M be a recursive, partially ordered set, which has a least element $\perp$ and every element of M is comparable with itself and with $\perp$. Every $m \in M$ is mapped to an untyped term m' in the following way:

if $m \in M \setminus \{\perp\}$, then $m' \in NF^0$ and for any $m_1, m_2 \in M \setminus \{\perp\}, m_1 \neq m_2 \Rightarrow m_1' \not\equiv m_2'$;
if $m \equiv \perp$, then $m' \equiv \Omega \equiv (\lambda x.xx)(\lambda x.xx)$.

We say that an untyped term Φ λ-defines (see [3]) the function $f : M^k \to M$ $(k \geq 0)$ with indeterminate values of arguments, if for any $m_1, \ldots, m_k \in M$ we have:

$f(m_1, \ldots, m_k) = m \neq \perp \Rightarrow \Phi m_1{}' \ldots m_k{}' \to\to_\beta m'$;

$f(m_1, \ldots, m_k) = \perp \Rightarrow \Phi m_1{}' \ldots m_k{}'$ does not have a head normal form.

We consider typed terms using a recursive set of functions C, every $f \in C$ is a strong computable function with intermediate values of arguments, which has an untyped λ-term that λ-defines it. From [3] it follows, that every $f \in C$ is a strong computable, monotonic function with indeterminate values of arguments. Therefore, according to [8], there exists a canonical notion of δ-reduction for the set C. Let us consider the algorithm of translation of any typed term t to an untyped term t' studied in [8]:

if $t \equiv m \in M$, then $t' \equiv m'$;

if $t \in C$, then $FV(t') = \emptyset$ and t' λ-defines t;

if $t \equiv x \in V^T$, then $x' \in V$ and for any $x_1, x_2 \in V^T$, $x_1 \not\equiv x_2 \Rightarrow x_1{}' \not\equiv x_2{}'$;

if $t \equiv \tau(t_1, \ldots, t_k)$, $k \geq 1$, then $t' \equiv \tau' t_1{}' \ldots t_k{}'$;

if $t \equiv \lambda x_1 \ldots x_n[\tau]$, $n \geq 1$, then $t' \equiv \lambda x_1{}' \ldots x_n{}'.\tau'$.

The main result of [8] is the Lemma 3 (Theorem 3 in [8]).

L e m m a 3 .

1. $t \in \Lambda_M^T, t \to\to_{\beta\delta} m,\ m \in M \setminus \{\perp\} \Rightarrow t' \to\to_\beta m'$;
2. $t \in \Lambda_M^T,\ FV(t) = \emptyset,\ t \to\to_{\beta\delta} \perp \Rightarrow t'$ does not have a head normal form.

Let P be a typed functional program of the form (1) and P' be the untyped functional program (the result of the translation) obtained by replacing typed terms for corresponding untyped terms in program P. Program P':

$$\begin{cases} F_1' = t_1'[F_1', \ldots, F_n'], \\ \quad\ldots \\ F_n' = t_n'[F_1', \ldots, F_n']. \end{cases} \tag{3}$$

T h e o r e m 5. (On translation). Let $f_P \in [M^k \to M], k \geq 1$, be the basic semantics of a typed functional program P of the form (1) and let $\tau_{P'}$ be the basic semantics of the untyped functional program P' of the form (3), then for all $m_1, \ldots, m_k \in M$ we have:

$f_P(m_1, \ldots, m_k) = m \neq \perp \Rightarrow \tau_{P'} m_1' \ldots m_k' \to\to_\beta m'$;

$f_P(m_1, \ldots, m_k) = \perp \Rightarrow \tau_{P'} m_1' \ldots m_k'$ does not have a head normal form.

P r o o f . Let $f_P(m_1, \ldots, m_k) = m \neq \perp$, then, according to the Theorem 2, $\exists s \geq 1, t_1^s[F_1, \ldots, F_n](m_1, \ldots, m_k) \to\to_{\beta\delta} m$, therefore, according to the point 1 of the Lemma 3, $t_1'^s[F_1', \ldots, F_n']m_1' \ldots m_k' \to\to_\beta m'$ and, according to the Theorem 4, $\tau_{P'} m_1' \ldots m_k' \to\to_\beta m'$.

Let $f_P(m_1, \ldots, m_k) = \perp$. There are 2 possible cases:

a) there exists $s \geq 1$ and term $b \in \Lambda$ such that
$t_1'^s[F_1', \ldots, F_n']m_1' \ldots m_k' \to\to_\beta b$ and $FV(b) = \emptyset$;

b) for any $s \geq 1$ and for any term $b \in \Lambda$ we have:
$t_1'^s[F_1', \ldots, F_n']m_1' \ldots m_k' \to\to_\beta b \Rightarrow FV(b) \neq \emptyset$.

Case (a). Let Ω_i be a term corresponding to the least element of the type of variable F_i obtained by the operation of abstraction and using $\perp$, $i = 1, \dots, n$. According to the Theorem 1, $t_1^s[\Omega_1, \dots, \Omega_n](m_1, \dots, m_k) \sim \perp$, therefore, according to the Definition 1 (point 2), $t_1^s[\Omega_1, \dots, \Omega_n](m_1, \dots, m_k) \to\to_{\beta\delta} \perp$ and, according to the point 2 of the Lemma 3, the term $t_1'^s[\Omega_1', \dots, \Omega_n']m_1' \dots m_k'$ does not have a head normal form. Since $t_1'^s[F_1', \dots, F_n']m_1' \dots m_k' \to\to_\beta b$ and $FV(b) = \emptyset$, then $t_1'^s[\Omega_1', \dots, \Omega_n']m_1' \dots m_k' \to\to_\beta b$ and the term b does not have a head normal form, therefore, the term $t_1'^s[F_1', \dots, F_n']m_1' \dots m_k'$ does not have a head normal form. It follows that the term $t_1'^s[P_1^n T_s, \dots, P_n^n T_s]m_1' \dots m_k'$ does not have a head normal form too (see [4]). According to the Lemma 2, $\tau_{P'} m_1' \dots m_k' \to\to_\beta t_1'^s[P_1^n T_s, \dots, P_n^n T_s]m_1' \dots m_k'$ and, therefore, the term $\tau_{P'} m_1' \dots m_k'$ does not have a head normal form.

Case (b). If there exists such $s \geq 1$ that $t_1'^s[F_1', \dots F_n']m_1' \dots m_k'$ does not have a head normal form, then $t_1'^s[P_1^n T_s, \dots, P_n^n T_s]m_1' \dots m_k'$ does not have a head normal form [4]. Since, according to the Lemma 2, $\tau_{P'} m_1' \dots m_k' \to\to_\beta t_1'^s[P_1^n T_s, \dots, P_n^n T_s]m_1' \dots m_k'$, then the term $\tau_{P'} m_1' \dots m_k'$ does not have a head normal form.

The only remaining case is when for any $s \geq 1$, $t_1'^s[F_1', \dots, F_n']m_1' \dots m_k'$ has a head normal form.We prove that these head normal forms have one of $F_1', \dots, F_n'$ as a head variable. We assume the opposite: there exists such $s \geq 1$ that $t_1'^s[F_1', ..., F_n']m_1' ... m_k' \to\to_\beta \lambda x_1 ... x_h . y d_1 ... d_l$, where $x_1, ..., x_h, y \in V$, $y \notin \{F_1', ..., F_n'\}$, $d_1, \dots, d_l \in \Lambda$, $h, l \geq 0$. Let Ω_i be the term corresponding to the least element of the type of the variable F_i obtained by the operation of abstraction and using $\perp, i = 1, \dots, n$. According to the Theorem 1, $t_1^s[\Omega_1, \dots, \Omega_n](m_1, \dots, m_k) \sim \perp$, therefore, according to the point 2 of the Definition 1, $t_1^s[\Omega_1, \dots, \Omega_n](m_1, \dots, m_k) \to\to_{\beta\delta} \perp$ and, according to the point 2 of the Lemma 3, the term $t_1'^s[\Omega_1', \dots, \Omega_n']m_1' \dots m_k'$ does not have a head normal form. But $t_1'^s[\Omega_1', \dots, \Omega_n']m_1', \dots, m_k' \to\to_\beta$ $\lambda x_1 \dots x_h . y d_1[\Omega_1', \dots, \Omega_n'] \dots d_l[\Omega_1', \dots, \Omega_n']$, which is a head normal form.

Let for any $s \geq 1$ $t_1'^s[F_1', \dots, F_n']m_1' \dots m_k'$ have a head normal form and let its head variable be one of the $F_1', \dots, F_n'$.

Let $t_1'^1[F_1', \dots, F_n']m_1' \dots m_k' \to\to_\beta \theta_1[F_1', \dots, F_n'] \in HNF$, then $t_1'^2[F_1', \dots, F_n']m_1' \dots m_k' \to\to_\beta \theta_1[t_1'[F_1', \dots, F_n'], \dots, t_n'[F_1', \dots, F_n']] \to\to_\beta \theta_2[F_1', \dots, F_n'] \in HNF$ and so on. Therefore, for all $s \geq 1$ we have: $t_1'^{s+1}[F_1', \dots, F_n']m_1' \dots m_k' \to\to_\beta$ $\theta_s[t_1'[F_1', \dots, F_n'], \dots, t_n'[F_1', \dots, F_n']] \to\to_\beta \theta_{s+1}[F_1', \dots, F_n'] \in HNF$ and the head variable is one of the variables $F_1', \dots, F_n'$.

According to the Lemma 2, we have:

$$\tau_{P'} m_1' \dots m_k' \to\to_\beta t_1'[P_1^n T_1, \dots, P_n^n T_1]m_1' \dots m_k' \to\to_\beta$$
$$\theta_1[P_1^n T_1, \dots, P_n^n T_1] \to\to_\beta \theta_1[t_1'[P_1^n T_2, \dots, P_n^n T_2], \dots, t_n'[P_1^n T_2, \dots, P_n^n T_2]] \to\to_\beta$$
$$\theta_2[P_1^n T_2, \dots, P_n^n T_2] \to\to_\beta \theta_2[t_1'[P_1^n T_3, \dots, P_n^n T_3], \dots, t_n'[P_1^n T_3, \dots, P_n^n T_3]] \to\to_\beta \dots \to\to_\beta$$
$$\theta_s[P_1^n T_s, \dots, P_n^n T_s] \to\to_\beta \theta_s[t_1'[P_1^n T_{s+1}, \dots, P_n^n T_{s+1}], \dots, t_n'[P_1^n T_{s+1}, \dots, P_n^n T_{s+1}]] \to\to_\beta$$
$$\dots$$

In the reducing chain an infinite number of times the head β-redex is convoluted. Therefore, the term $\tau_{P'} m_1' \dots m_k'$ does not have a head normal form. $\square$

Received 18.10.2016

REFERENCES

1. **Nigiyan S.A.** Functional Programming Languages. // Programming and Computer Software, 1991, № 5, p. 77–86.
2. **Nigiyan S.A.** On Interpretation of Functional Programming Languages. // Programming and Computer Software, 1993, v. 19, № 2, p. 71–78.
3. **Nigiyan S.A.** On Non-classical Theory of Computability. // Proceedings of the YSU. Physical and Mathematical Sciences, 2015, № 1, p. 52–60.
4. **Barendregt H.** The Lambda Calculus. Its Syntax and Semantics. North-Holland Publishing Company, 1981.
5. **Nigiyan S.A., Avetisyan S.A.** Semantics of Untyped Functional Programs. // Programming and Computer Software, 2002, v. 28, № 3, p. 119–126.
6. **Hrachyan G.G.** On Basic Semantics of Untyped Functional Programs. // Programming and Computer Software, 2009, v. 35, № 3, p. 121–135.
7. **Budaghyan L.E.** Formalizing the Notion of δ-Reduction in Monotonic Models of Typed λ-Calculus. // Algebra, Geometry & Their Applications, 2002, v. 1, p. 48–57.
8. **Nigiyan S.A., Khondkaryan T.V.** On Canonical Notion of δ-reduction and on Translation of Typed λ-Terms into Untyped λ-Terms. // Proceedings of the YSU. Physical and Mathematical Science, 2017, v. 51, № 1, p. 46–52.