PROCEEDINGS OF THE YEREVAN STATE UNIVERSITY

Physical and Mathematical Sciences 2011, № 1, p. 28–35

Informatics

UNSOLVABILITY OF TYPE CORRECTNESS PROBLEM FOR FUNCTIONAL PROGRAMS

A. H. ARAKELYAN*

Chair of Programming and Information Technologies, YSU

In the present paper the type correctness problem is considered for functional programs without the type information. The aim of this research is to prove that there is no algorithm to reject all programs, during execution of which the type error would occur and accept all programs, during execution of which the type error would not occur.

***Keywords*:** term, redex, reduction strategy, type error.

1. Introduction. The compile time type checking is important in any programming language for purposes such as the early detection of programming errors, for doing optimizations etc. In the present paper we consider the compile time type checking problem for functional programs, where no explicit type information is provided by the programmer. Ideally, it would be excellent to have such a type checking algorithm to determine whether during the execution of any given arbitrary program by the interpreter the type error occurs or not. It will be shown that no such an algorithm exist even for the terms that can be also treated as functional programs with one non-recursive equation. For proving the main theorems of this paper, we shall widely use important properties of λI system introduced in [1].

2. Definitions Used and Preliminary Results. Let the $TermVariable$ be a countable set of term variables and the $Constant$ be a countable set of term constants.

Definition 2.1. The set of terms $Term$ is defined as follows:

1) $\perp \in Term$ and is for representing type errors;
2) if $x \in TermVariable$, then $x \in Term$;
3) if $c \in Constant$, then $c \in Term$;
4) if $x \in TermVariable$ and $M \in Term$, then $(\lambda x M) \in Term$, and we say that the term $(\lambda x M)$ is obtained from term variable x and term M by means of abstraction operation;
5) if $M_1, M_2 \in Term$, then $(M_1 M_2) \in Term$, and we say that the term $(M_1 M_2)$ is obtained from terms M_1 and M_2 by means of application operation.

* E-mail: ara_arakelyan@yahoo.com

The notions of a free and bound occurrence of a variable in a term and the notion of a free variable of a term are introduced in a conventional way. The set of all free variables of a term M is denoted by $FV(M)$. The notion of congruency $(\equiv)$ of two terms is also introduced in a conventional way. We will use the following abridged notations:

1. The term $(\ldots(M_1M_2)\ldots M_k)$ is denoted by $M_1M_2\ldots M_k$, where $M_i \in Term$, $i=1,\ldots,k$ and $k\geq 2$;

2. The term $(\lambda x_1(\lambda x_2(\ldots(\lambda x_k M)\ldots)))$ is denoted by $\lambda x_1x_2\ldots x_k.M$, where $x_i \in TermVariable$, $i=1,\ldots,k$, $k\geq 1$, $M\in Term$;

3. By $M[x_1,\ldots,x_k]$ we denote the term M mentioning also mutually different term variables $x_1,\ldots,x_k$ that interest us, where $k\geq 1$;

4. By $M[x_1:=M_1,\ldots,x_k:=M_k]$ we denote the term obtained by the simultaneous substitution of the terms $M_1,\ldots,M_k$ for all free occurrences of the variables $x_1,\ldots,x_k$ respectively into the term M, where $k\geq 1$.

The notion of β-reduction, one-step β-reduction $(\to_\beta)$, β-reduction $(\twoheadrightarrow_\beta)$, β-equality $(=_\beta)$, β-redex, β-reduct, β-normal form and strongly β-normalizable terms are defined in a standard way. The set of all β-normal forms is denoted as $\beta-NF$.

Definition 2.2. Any relation $\delta\subset Term^2$ is called a notion of δ-eduction, if the following conditions are satisified:

1) if $(M,M')\in\delta$, then $M\equiv cM_1\ldots M_k$, where $c\in Constant$, $M_i\in Term$, $i=1,\ldots,k$, $k\geq 1$;

2) if $(M,M')\in\delta$ and $(M,M'')\in\delta$, then $M'\equiv M''$;

3) there exists an algorithm such that for any input term $cM_1\ldots M_k$, where $c\in Constant$, $M_i\in Term$, $i=1,\ldots,k$, $k\geq 1$, it returns term M' such that $(cM_1\ldots M_k,M')\in\delta$ or returns *no*, if there does not exist any term M' such that $(cM_1\ldots M_k,M')\in\delta$.

Here we mention only those properties of notion of δ-reduction that we need in this paper. The real notion of δ-reduction must also have other properties that are necessary, when proving some important propositions, e.g. uniqueness of $\beta\delta$-normal form etc. One-step δ-reduction $(\to_\delta)$, δ-reduction $(\twoheadrightarrow_\delta)$, δ-equality $(=_\delta)$, δ-redex, δ-reduct and δ-normal form are defined in a standard way.

Definition 2.3. Let δ be some notion of δ-reduction. The relation $\beta\cup\delta$ is called a notion of $\beta\delta$-reduction. One-step $\beta\delta$-reduction $(\to_{\beta\delta})$, $\beta\delta$-reduction $(\twoheadrightarrow_{\beta\delta})$, $\beta\delta$-equality $(=_{\beta\delta})$, $\beta\delta$-edex, $\beta\delta$-reduct and $\beta\delta$-normal form are defined in a standard way.

Now let us define the notion of $\beta\delta$-reduction strategy.

Defition 2.4. The map $R: Term\to Term$ is called a $\beta\delta$-reduction strategy, if $M\twoheadrightarrow_{\beta\delta} R(M)$ for any term M and if M is not $\beta\delta$-normal form, then there exists $M_1 \in Term$ such that $M \to_{\beta\delta} M_1 \twoheadrightarrow_{\beta\delta} R(M)$. The strategy R is called one-step strategy, if for any term M, which is not $\beta\delta$-normal form, $M \to_{\beta\delta} R(M)$. The strategy R is called an effective strategy, if there exists an algorithm such that for any input term M it returns $R(M)$.

Let us define the subset of terms $TermI$, which plays an important role in this paper.

Definition 2.5. The subset of terms $TermI \subset Term$ is defined as follows:

1) if $x \in TermVariable$, then $x \in Term$;
2) if $x \in TermVariable$, $M \in TermI$ and $x \in FV(M)$, then $(\lambda xM) \in TermI$;
3) if $M_1, M_2 \in TermI$, then $(M_1 M_2) \in TermI$.

Now we will present Church–Rosser Theorem [1] about the terms of $TermI$.

Theorem 2.1. Let $M \in TermI$. If M has a β-normal form, then M is a strongly β-normalizable term, i.e. any sequence of β-reductions in term M reduces it to its β-normal form.

Let us present some abridged notations and coding of natural numbers using terms of $TermI$.

1. $I \equiv \lambda x.x$, $T \equiv \lambda xy.yIIx$, $F \equiv \lambda x.xIII$, $Zero \equiv \lambda x.x(TF)TTF$;
2. $M^0 M' \equiv M'$, $M^{n+1} M' \equiv MM^n M'$, where $M, M' \in Term$ and $n \geq 0$;
3. $C_0 \equiv \lambda xy.xIIy$, $C_n \equiv \lambda fx.f^n x$, where $n \geq 1$.

Next two lemmas from [1] present some simple β-equalities, which are used later in this paper.

Lemma 2.1. The following β-equalities hold:

1. $ZeroC_0 =_\beta T$ and $ZeroC_n =_\beta F$, where $n \geq 1$;
2. $III =_\beta I$, $TII =_\beta I$, $FII =_\beta I$, $C_n II =_\beta I$, where $n \geq 0$.

Lemma 2.2. Let $P, Q \in TermI$, $PII =_\beta I$ and $QII =_\beta I$. Then the following β-equalities hold: $TPQ =_\beta P$ and $FPQ =_\beta Q$.

It is obvious, that the set of terms $TermI$ is a countable set. Hence we can enumerate terms of $TermI$ using natural numbers. Let us fix one such effective enumeration and denote number of term $M \in TermI$ in this enumeration by $\frown{M}$.

Definition 2.6. Let $\varphi : A \to N$ be an arithmetic function, where $A \subset N^k$ and $k \geq 1$. Then φ is said to be defined by the term $M \in TermI$, if the following conditions are satisfied:

1) if $(n_1, \ldots, n_k) \in A$ and $\varphi(n_1, \ldots, n_k) = m$, then $MC_{n_1} \ldots C_{n_k} =_\beta C_m$;
2) if $(n_1, \ldots, n_k) \notin A$, then $MC_{n_1} \ldots C_{n_k}$ has no β-normal form.

Theorem 2.2 (Kleene [1]). The arithmetic function $\varphi : A \to N$, where $A \subset N^k$ and $k \geq 1$, can be defined by the term of $TermI$, iff φ is a partial recursive function.

3. Main Results. Before presenting main theorems of this paper, let us introduce several definitions and see what does the phrase "term contains type error" mean. We will use the following abridged notation: $R^0(M) \equiv M$ and $R^{n+1}(M) \equiv R(R^n(M))$, where $M \in Term$, R is a $\beta\delta$-reduction strategy and $n \geq 0$.

Definition 3.1. Let $M \in Term$ and R is a $\beta\delta$-reduction strategy. We say that M contains a type error in R, if there exists $n \geq 0$ such that $\perp$ is a subterm of term $R^n(M)$.

Definition 3.2. Let $M \in Term$ and R is a $\beta\delta$-reduction strategy. We say that R terminates on M, if there exists $n \geq 0$ such that $R^{n+1}(M) \equiv R^n(M)$. Otherwise, we say that R does not terminate on M.

Definition 3.3. The set $A \subset Term$ is called recursive, if the set $\{\overset{\frown}{M} \mid M \in A\}$ of natural numbers is recursive.

Definition 3.4. Let δ be some notion of δ-reduction. We say that $c \in Constant$ is a constant of 0-order for δ, if for any $M_1, \ldots, M_n \in Term$, $(cM_1 \ldots M_n, \perp) \in \delta$, where $n \geq 1$.

For the reminder of this paper assume that there exists at least one common constant of 0-order for all notions of δ-reduction to be considered by us. The following lemmas are needed for proving the main theorems of this paper.

Lemma 3.1. Let $M \in TermI$, $xM_1 \ldots M_n$ is a subterm of term M, where $x \in TermVariable$, $M_i \in TermI$, $i = 1, \ldots, n$, $n \geq 0$, and for at least one occurrence of $xM_1 \ldots M_n$ in M, occurrence of variable x at the beginning of that subterm is free in M. If $M \to_\beta M'$, then there exists $xM_1' \ldots M_n'$ subterm of term M' such that $M_i' \in TermI$, $i = 1, \ldots, n$, and for at least one occurrence of $xM_1' \ldots M_n'$ in M', the occurrence of variable x at the beginning of that subterm is free in M'.

Proof. To avoid mentioning the occurrence of subterm $xM_1 \ldots M_n$ every time during the proof, assume that we deal only with such an occurrence of subterm $xM_1 \ldots M_n$, for which the occurrence of variable x at the beginning of that subterm is free in M. Assume that $(\lambda y.M_0[y])M_0'$ is a β-redex corresponding to the one step β-reduction $M \to_\beta M'$. Let us consider 4 possible cases:

1. The occurences of $xM_1 \ldots M_n$ and $(\lambda y.M_0[y])M_0'$ in M have no common parts. It is evident, that in this case the same occurrence of $xM_1 \ldots M_n$ also exists in M'.

2. The occurrence of $xM_1 \ldots M_n$ is in M_0'. Since $M \in TermI$, then $y \in FV(M_0[y])$. Therefore, there exists at least one occurrence of $xM_1 \ldots M_n$ in $M_0[y := M_0']$ and hence in M', for which occurrence of variable x at the beginning of it is free in M'.

3. The occurrence of $xM_1 \ldots M_n$ is in $M_0[y]$. Because of our agreement $x \neq y$. Therefore, it is evident that there exists an occurrence of $xM_1[y := M_0'] \ldots M_n[y := M']$ in $M_0[y := M_0']$ and hence in M', for which the occurrence of variable x at the beginning of it is free in M'.

4. There exists $k \in \{1,\ldots,n\}$ such that the occurrence of $(\lambda y.M_0[y])M_0'$ is in M_k. It is evident, that in this case there exists an occurrence of $xM_1\ldots M_k'\ldots M_n$ in term M', for which the occurrence of variable x at the beginning of it is free in M', where M_k' is obtained from M_k by replacing $(\lambda y.M_0[y])M_0'$ with its β-reduct.

Lemma 3.1 is proved.

Corollary 3.1. Let $M \in TermI$, $xM_1\ldots M_n$ is a subterm of term M, where $x \in TermVariable$, $M_i \in TermI$, $i = 1,\ldots,n$, $n \geq 0$, and for at least one occurrence of $xM_1\ldots M_n$ in M, the occurrence of variable x at the beginning of that subterm is free in M. If $M \twoheadrightarrow_\beta M'$, then there exists $xM_1'\ldots M_n'$ subterm of term M' such that $M_i' \in TermI$, $i = 1,\ldots,n$, and for at least one occurrence of $xM_1'\ldots M_n'$ in M', the occurrence of variable x at the beginning of that subterm is free in M'.

Proof. The proof follows directly from Lemma 3.1.

Corollary 3.1 is proved.

Lemma 3.2. Let $M \in TermI$ has a β-normal form, R is a $\beta\delta$-reduction strategy and c is a constant of 0-order for δ. If Mc contains type error in R, then $M'c$ also will contain type error in R, where M' is a β-normal form of M.

Proof. First of all let us note, that during $\beta\delta$-reductions in term Mc β-redexes can be in the form $cM_1\ldots M_n$ only, where $M_i \in TermI$, $i = 1,\ldots,n$ and $n \geq 1$. Hence, the type error will occur, when replacing an arbitrary δ-redex with its δ-reduct, because c is a constant of 0-order. We will prove Lemma 3.2 by contradiction. Let us suppose that Mc contains a type error in R, but $M'c$ does not contain any type error in R. Then it is evident that $M'c$ does not contain any δ-redex. Since M' is a β-normal form, $M'c$ can contain β-redex only when $M' \equiv \lambda y.M_0'[y]$. In that case at follows that $M'c \to_\beta M_0'[y := c]$ and $M_0'[y := c]$ does not β-redex anymore. The term $M_0'[y := c]$ also does not contain δ-redex, otherwise, $M'c$ would have contained a type error in an arbitrary $\beta\delta$-reduction strategy and hence in R too. Here we can conclude, that if $M'c$ is not $\beta\delta$-normal form, then it reduces to $\beta\delta$-normal form after one β-reduction. Let us denote $\beta\delta$-normal form of term $M'c$ by M''. Since Mc contains a type error in R, after doing finite β-reductions in Mc, the strategy R finally does one δ-reduction, which brings to type error. Let us denote the term directly prior to that first δ-reduction mentioned above by M_0. So M_0 has a subterm of the form $cM_1\ldots M_n$. Let us forget that c is a constant and assume just for a moment that c is a variable. In that case it is evident that $Mc, M_0, M'' \in TermI$, $Mc \twoheadrightarrow_\beta M_0$ and M'' is a β-normal form of Mc. Hence $M_0 \twoheadrightarrow_\beta M''$. Bassed on Corollary 3.1, we can conclude, that M'' has a subterm of the form $cM_1\ldots M_n$, which contradicts the fact that M'' is a $\beta\delta$-normal form.

Lemma 3.2 is proved.

Lemma 3.3. Let $M \in TermI$ has a β-normal form, R is a $\beta\delta$-reduction strategy and c is a constant of 0-order for δ. If Mc does not contain a type error in

R, then $M'c$ also will not contain any type error in R, where M' is a β-normal form of M.

Proof. First of all let us note, that during $\beta\delta$-reductions in term Mc δ-redexes can be in the form $cM_1 \dots M_n$ only, where $M_i \in TermI$, $i = 1, \dots, n$ and $n \geq 1$. Hence, the type error will occur, when replacing arbitrary δ-redex with its δ-reduct, because c is a constant of 0-order. Since Mc does not contain the type error in R, the strategy R does only β-reductions in term Mc. We will prove Lemma 3.3 by contradiction. Let us suppose that Mc does not contain a type error in R, but $M'c$ contains a type error in R. It is evident that $M'c$ does not contain δ-redex. Since $M'c$ must contain a type error in R, the term M', which is a β-normal form, must have the following form: $M' \equiv \lambda y.M'_0[y]$. Hence

$$M'c \equiv (\lambda y.M'_0[y])c \to_\beta M'_0[y := c]. \tag{1}$$

It is obvious that $M'_0[y := c]$ does not contain β-redex anymore, but it must have a subterm of the form $cM_1 \dots M_n$, otherwise, $M'c$ will not contain a type error in R. Let us forget that c is a constant and think that c is a term variable just for a moment. In that case it is evident that $Mc, M'c, M'_0[y := c] \in TermI$ and $Mc \twoheadrightarrow_\beta M'c$. Hence, according to (1), $Mc \twoheadrightarrow_\beta M'_0[y := c] \in \beta - NF$. Since the strategy R does only β-reductions in term Mc, by Theorem 2.1 the strategy R will reduce Mc to the $M'_0[y := c]$ after finite β-reductions. On the other hand, $M'_0[y := c]$ must have a subterm of the form $cM_1 \dots M_n$ as is seen above. Thus, Mc contains a type error in R in contravention of the Lemma's condition.

Lemma 3.3 is proved.

Corollary 3.2. Let $M \in TermI$ have a β-normal form, R be a $\beta\delta$-reduction strategy and c be a constant of 0-order for δ. Then Mc contains a type error in R, iff $M'c$ contains a type error in R, where M' is a β-normal form of M.

Proof. The proof follows directly from Lemma 3.2 and Lemma 3.3.

Corollary 3.2 is proved.

Now it is the time to present the main theorems of this paper.

Theorem 3.1. Let R be some $\beta\delta$-reduction strategy. There is no algorithm that for input term $M \in Term$ returns *yes*, if M does not contain a type error in R and returns *no*, if M contains a type error in R.

Proof. We will prove the Theorem by contradiction. Let us suppose that such an algorithm does exist. Assume that c is a constant of 0-order for δ. It is evident that the set $A = \{M \in TermI \mid Mc$ does not contain a type error in $R\}$ is recursive, because for determining whether the term $M \in TermI$ belongs to A or not, it is sufficient to run the above existing algorithm for the input term Mc. From recursiveness of A follows recursiveness of the set $A' = \{M \in TermI \mid MC_{\overline{M}} \in A\}$. Hence there exists a total recursive function $\varphi : N \to N$ such that $\varphi(\widehat{M}) = 1$, when $M \in A'$ and $\varphi(\widehat{M}) = 0$, when $M \notin A'$. According to Theorem 2.2, there exists $M_0 \in TermI$ such that φ is defined by term M_0, i.e.

$$M \in A' \Rightarrow M_0 C_{\overline{M}} =_\beta C_1, \tag{2}$$

$$M \notin A' \Rightarrow M_0 C_{\overline{M}} =_\beta C_0. \tag{3}$$

Now we will construct a new term using the term M_0: $P \equiv \lambda x.Zero(M_0 x)M'M'' \in TermI$, where $M' \equiv \lambda x.x \in A$ and $M'' \equiv \lambda x.xx \notin A$. Let us show the following:

$$M \in A' \Rightarrow PC_{\overline{M}} \notin A, \tag{4}$$

$$M \notin A' \Rightarrow PC_{\overline{M}} \in A. \tag{5}$$

1. First let us prove (4). We have that $M \in A'$. Hence by (2), Lemma 2.1 and Lemma 2.2, $PC_{\overline{M}} \equiv (\lambda x.Zero(M_0 x)M'M'')C_{\overline{M}} \rightarrow_\beta Zero(M_0 C_{\overline{M}})M'M'' \twoheadrightarrow$ $\twoheadrightarrow_\beta ZeroC_1 M'M'' \twoheadrightarrow_\beta FM'M'' \twoheadrightarrow_\beta M'' \in \beta - NF$. So M'' is a β-normal form of $PC_{\overline{M}}$. Since $M''c \equiv (\lambda x.xx)c$ contains a type error in an arbitrary $\beta\delta$-reduction strategy and hence in R too, by Corollary 3.2, $PC_{\overline{M}}c$ will also contain a type error in R, which means that $PC_{\overline{M}} \notin A$.

2. Now let us prove (5). We have that $M \notin A'$. Hence by (3), Lemma 2.1 and Lemma 2.2, $PC_{\overline{M}} \equiv (\lambda x.Zero(M_0 x)M'M'')C_{\overline{M}} \rightarrow_\beta Zero(M_0 C_{\overline{M}})M'M'' \twoheadrightarrow$ $\twoheadrightarrow_\beta ZeroC_0 M'M'' \twoheadrightarrow_\beta TM'M'' \twoheadrightarrow_\beta M' \in \beta - NF$. So M' is a β-normal form of $PC_{\overline{M}}$. Since $M'c \equiv (\lambda x.x)c$ does not contain a type error in every $\beta\delta$-reduction strategy and hence in R too, according to Corollary 3.2, $PC_{\overline{M}}c$ also will not contain any type error in R, which means that $PC_{\overline{M}} \in A$.

The term $P \in TermI$ constructed by us either belongs to or does not belong to A'. If $P \in A'$, then, by (4), $PC_{\overline{P}} \notin A$, i.e. $P \notin A'$, which is not possible. If $P \notin A'$, then, by (5), $PC_{\overline{P}} \in A$, i.e. $P \in A'$, which is not possible either. We came to contradiction. Hence, such an algorithm does not exist.

Theorem 3.1 is proved.

Theorem 3.2. There is no algorithm that for input term $M \in Term$ returns *yes*, if M does not contain type error in certain $\beta\delta$-reduction strategy and returns *no*, otherwise, i.e. when M contains a type error in arbitrary $\beta\delta$-reduction strategy.

Proof. The proof differs from that of previous theorem in choice of set A and in the proof of propositions (4) and (5). In this case $A = \{M \in TermI \mid Mc$ does not contain a type error in certain $\beta\delta$-reduction strategy$\}$. Now let us prove propositions (4) and (5).

1. First let us prove (4). In the same way we can conclude, that M'' is a β-normal form of $PC_{\overline{M}}$. Since $M''c \equiv (\lambda x.xx)c$ contains a type error in every $\beta\delta$-reduction strategy, by Corollary 3.2, $PC_{\overline{M}}c$ will also contain the type error in every $\beta\delta$-reduction strategy, which means that $PC_{\overline{M}} \notin A$.

2. Now let us prove (5). In the same way we can conclude that M' is a β-normal form of $PC_{\overline{M}}$. Since $M'c \equiv (\lambda x.x)c$ does not contain a type error in

every $\beta\delta$-reduction strategy, by Corollary 3.2, $PC_{\overline{M}}c$ also will not contain the type error in every $\beta\delta$-reduction strategy, which means that $PC_{\overline{M}} \in A$.

Theorem 3.2 is proved.

Theorem 3.3. There is no algorithm that for input term $M \in Term$ returns *yes*, if M does not contain a type error in arbitrary $\beta\delta$-reduction strategy and returns *no*, otherwise, i.e. when M contains the type error in certain $\beta\delta$-reduction strategy.

Proof. The proof differs from that of previous theorem only in the choice of set A. In this case $A = \{M \in TermI \mid Mc$ does not contain a type error in arbitrary $\beta\delta$-reduction strategy$\}$.

Theorem 3.3 is proved.

Theorem 3.4. Let R be some $\beta\delta$-reduction strategy. There is no algorithm that for input term $M \in Term$ returns *yes*, if M does not contain type error in R and R terminates on M, and returns *no*, otherwise, i.e. when M contains the type error in R or R does not terminate on M.

Proof. The proof differs from that of Theorem 3.1 in choice of set A and in proof of propositions (4) and (5). In this case $A = \{M \in TermI \mid Mc$ does not contain type error in R and R terminates on $M\}$. Now let us prove propositions (4) and (5).

1. First let us prove (4). In the same way we can conclude that M'' is a β-normal form of $PC_{\overline{M}}$. Since $M''c \equiv (\lambda x.xx)c$ contains a type error in every $\beta\delta$-reduction strategy, by Corollary 3.2, $PC_{\overline{M}}c$ will also contain the type error in every $\beta\delta$-reduction strategy and hence in R too, which means that $PC_{\overline{M}} \notin A$.

2. Now let us prove (5). In the same way we can conclude that M' is a β-normal form of $PC_{\overline{M}}$. Since $M'c \equiv (\lambda x.x)c$ does not contain type error in every $\beta\delta$-reduction strategy, by Corollary 3.2, $PC_{\overline{M}}c$ also will not contain type error in every $\beta\delta$-reduction strategy and hence in R too. Thus, after showing that R terminates on $PC_{\overline{M}}c$, we can conclude that $PC_{\overline{M}} \in A$. Since $PC_{\overline{M}}c$ does not contain type error in every $\beta\delta$-reduction strategy, only β-reductions will be done during work of arbitrary $\beta\delta$-reduction strategy on term $PC_{\overline{M}}c$. Let us forget that c is a constant and think that c is a variable of term just for a moment. In that case it is evident that $PC_{\overline{M}}c \in TermI$ and c is a β-normal form of $PC_{\overline{M}}c$, because $M'c \equiv (\lambda x.x)c \to_\beta c$. By Theorem 2.1, $PC_{\overline{M}}c$ is a strongly β-normalizable. Hence, any $\beta\delta$-reduction strategy terminates on $PC_{\overline{M}}c$.

Theorem 3.4 is proved.

Received 14.10.2010

REFERENCES

1. **Barendregt H.P.** The Lambda Calculus: Its Syntax and Semantics. Amsterdam, New York, Oxford: North-Holland Pub. Comp., 1981.