FIXPOINT TECHNIQUE FOR COUNTING TERMS IN TYPED λ CALCULUS [1]

Marek Zaionc [2]
email: zaionc@cs.buffalo.edu

Abstract: Typed λ calculus with denumerable set of ground types is considered. The aim of the paper is to show procedure for counting closed terms in long normal forms. It is shown that there is a surprising correspondence between the number of closed terms and the fixpoint solution of the polynomial equation in some complete lattice. It is proved that counting of terms in typed lambda calculus can be reduced to the problem of finding least fixpoint for the system of polynomial equations. The algorithm for finding the least fixpoint of the system of polynomials is considered. By the well known Curry Howard isomorphism the result can be interpreted as a method for counting proofs in the implicational fragment of the propositional intuitionistic logic. The problem of number of terms was studied but never published by Ben- Yelles (see [3]). Similarly in [4] it was proved that complexity of the question whether given type possess an infinite number of normal terms is polynomial space complete.

1. Typed λ calculus.

The set $TYPES$ is defined as follows: there is a denumerable set of ground types $\{A, B, ...\}$ and if τ and μ are types then $\tau \rightarrow \mu$ is a type. We will use the following notation: if $\mu, \tau_1, ..., \tau_n$ are types then by $\otimes_{j=1}^{n} \tau_j \rightarrow \mu$ we mean the type $\tau_1 \rightarrow (...(\tau_n \rightarrow \mu)...)$. Therefore, every type τ has the form $\otimes_{j=1}^{n} \tau_j \rightarrow \mu$ where μ is a ground type. If τ is a type of the form $\otimes_{j=1}^{n} \tau_j \rightarrow \mu$ where μ is a ground type, then τ_i for $i \leq n$ are called components of type τ and are denoted by $\tau[i]$. For any type τ we define $rank(\tau)$ and $arg(\tau)$ as follows: $arg(\tau) = rank(\tau) = 0$ for every ground type τ and $arg(\otimes_{j=1}^{n} \tau[j] \rightarrow \mu) = n$ and $rank(\otimes_{j=1}^{n} \tau[j] \rightarrow \mu) = \max_{i=1...n} rank(\tau[i]) + 1$. We define inductively types $\tau[i_1, ..., i_k]$ by $(\tau[i_1, ..., i_{k-1}])[i_k]$ for $1 \leq i_k \leq arg(\tau[i_1, ..., i_{k-1}])$. Ground types τ and μ are equivalent (denoted by $\tau \sim \mu$) if $\tau = \mu$. This relation is extended to all types by the following rules: $(\tau \rightarrow \nu) \sim \mu$ iff $\nu \sim \mu$ and $\tau \sim (\mu \rightarrow \nu)$ iff $\tau \sim \nu$.
For any type τ there is given a denumerable set of variables $V(\tau)$. Any variable of type τ is a term of type τ. If T is a term of type $\tau \rightarrow \mu$ and S is a term of type τ, then TS is a term of type μ. If T is a term of type μ and x is a variable of type τ, then $\lambda x.T$ is a term of type $\tau \rightarrow \mu$. The axioms of equality between terms have the form of $\beta\eta$ conversions and the convertible terms will be written as $T =_{\beta\eta} S$. Term T is in long normal form if $T = \lambda x_1...x_n.yT_1...T_k$ where y is an x_i for some $i \leq n$ or y is a free variable, T_j for $j \leq k$ are in the long normal form and $yT_1...T_k$ is a term of some ground type. For a closed term $T = \lambda x_1...x_n.x_iT_1...T_k$ variable x_i is called a *heading*. Long normal forms exist and are unique for $\beta\eta$ conversions (compare [6]).

[1] Supported by KBN research grant 0384/P4/93

[2] Permanent address of the author is: Computer Science Department, Jagiellonian University Nawojki 11, 30-072 Krakow, Poland E-mail `zaionc@ii.uj.edu.pl`. This paper was partially prepared while author was visiting Computer Science Department at State University of New York at Buffalo, USA

Let us introduce a complexity measure π for closed terms. If T is a closed term written in the long normal form and $T = \lambda x_1...x_n.x_i$ then $\pi(T) = 1$. If $T = \lambda x_1...x_n.x_i T_1...T_k$ then $\pi(T) = \max_{j=1...k}(\pi(\lambda x_1...x_n.T_j)) + 1$. By $|\tau|$ we mean the number of closed terms in normal forms of the type τ. For any $\alpha \in N$ by $|\tau|_\alpha$ we mean the number of closed terms T in long normal forms of the type τ with the complexity $\pi(T)$ not greater then α. For more detailed treatment of typed lambda calculus see [8].

Definition 1.1 Type τ *is embeddable in* μ if $\tau \sim \mu$ and there is a surjection $f : \{1, ..., arg(\tau)\} \to \{1, ..., arg(\mu)\}$ such that $\tau[i] = \mu[f(i)]$ for all $i \leq arg(\tau)$.

Definition 1.2 Type τ *is a direct prolongation of type* μ if $\tau = \otimes_{j=1}^{n} \mu[j] \to \mu[i, p]$ for some $i \leq arg(\mu)$ and $p \leq arg(\mu[i])$ where $n = arg(\mu)$ such that $\mu[i] \sim \mu$. Relation of *prolongation* is a transitive and reflexive closure of direct prolongation.

Definition 1.3 Type τ *is an extension of type* μ if there is a finite sequence of types $\tau_1, ..., \tau_n$ for $n \geq 1$ such that $\tau_1 = \mu$ and $\tau_n = \tau$ and the following conditions are satisfied:

1. τ_i is a direct prolongation of τ_{i-1} for all $2 \leq i \leq n$.
2. τ_i is not embeddable in any τ_k for $k < i$.

Definition 1.4 Types τ and μ are of the *same size* if both $|\tau|$ and $|\mu|$ are 0, or both $|\tau|$ and $|\mu|$ are different then 0 and finite, or both $|\tau|$ and $|\mu|$ are ω. Obviously the *same size* is an equivalence relation with three equivalence classes.

Theorem 1.5 Any type admits a finite number of extensions.
Proof: We start the proof with defining the sets $DC(\tau)$ of direct components of τ . For ground type $DC(A) = \emptyset$. For type $\tau = \otimes_{j=1}^{n} \tau[j] \to A$ $DC(\tau) = \{\tau[1], ..., \tau[n]\}$. The set $C(\tau)$ of all components of τ is defined by allowing: $\tau \in C(\tau)$ and if $\mu \in C(\tau)$ and $arg(\mu) = n$ then $\mu[1], ..., \mu[n] \in C(\tau)$. Obviously, for type $\tau = \otimes_{j=1}^{n} \tau[j] \to A$ we have $C(\tau) = \bigcup_1^n C(\tau[j]) \cup \{\tau\}$, and therefore for any type τ $C(\tau)$ is finite. Evidently, if τ is a direct prolongation of μ then $DC(\mu) \subset DC(\tau)$. Of course we have $DC(\mu) \subset DC(\tau)$ and $DC(\tau) \subset C(\mu)$ if τ is a prolongation of μ .
Now, let us assume that there is a type τ with infinite number of extensions. Since each type admits at most finite number of direct prolongations then by the König lemma argument there must be an infinite sequence of types $\tau_1, \tau_2, ...$ such that τ_{i+1} is a direct prolongation of τ_i for all $i \geq 1$ and for every i τ_i is not embeddable in any of τ_k for $k < i$. Since there is only finite number of ground types in the type τ then we can find an infinite subsequence $\tau'_1, \tau'_2, ...$ such τ'_{i+1} is a prolongation of τ'_i for all $i \geq 1$ and τ'_i is not embeddable in any of τ'_k for $k < i$, and additionally for all $i, j, \tau'_i \sim \tau'_j$. By an earlier observation we have

$$DC(\tau'_1) \subset DC(\tau'_2) \subset ... \subset C(\tau)$$

Since $C(\tau)$ is finite then there must be an i such that $DC(\tau'_i) = DC(\tau'_{i+1})$. Therefore τ'_{i+1} is embeddable in τ'_i.

We can display the set of extensions of type τ in the form of a tree. The tree consists of type τ as the root and each extension of type τ is connected with all its prolongations.

Definition 1.6 The *extension tree* for type τ is a finite branching tree such that any node is labeled by type. The extension tree is constructed in the following way:

1. root of the tree is τ ,
2. if a current node μ is an extension of τ and μ has direct prolongations $\mu_1, ..., \mu_n$, then draw an arc from μ to each μ_i for $1 \leq i \leq n$,
3. if the current node μ is an extension of τ and has no direct prolongations, then μ has no successors in the tree. Such μ is a terminal node in the tree,
4. if the current node μ is not an extension of τ (μ is embeddable in one of its predecessors), then μ has no successors in the tree. Such μ is a terminal node in the tree.

Theorem 1.7 Extension tree for type τ is finite.
Proof: This is a simple consequence of theorem 1.5

2. Polynomials.

We consider the set $\bar{N} = N \cup \{\omega\}$. The natural order is extended by $i \leq \omega$ for every $i \in N$. Addition and multiplication are extended by $a + \omega = \omega$, $0 \cdot \omega = 0$, $a \cdot \omega = \omega$ for every $a \neq 0$. The set $(\bar{N}^k, \leq)$ is naturally ordered by $(a_1, ..., a_k) \leq (b_1, ..., b_k)$ if $\forall i \leq k \; a_i \leq b_i$

Lemma 2.1 $\bar{N}^k$ is a complete lattice.
Proof. For each nonempty set $X \subseteq \bar{N}^k$ there exists the least upper bound which is denoted by $\bigcup X$ and the greatest lower bound denoted by $\bigcap X$.A bottom element of $\bar{N}^k$ is $(0, ..., 0)$ and a top element is $(\omega, ..., \omega)$.

Function $f : \bar{N}^n \rightarrow \bar{N}^k$ is called a *system of polynomials* (or simply a polynomial) if $f = (f_1, ..., f_k)$ and each $f_i : \bar{N}^n \rightarrow \bar{N}$ is a composition of addition, multiplication and constant functions. Since addition, multiplication and constant functions are monotonic on $\bar{N}^k$, it is easy to notice that polynomials are monotonic.

Definition 2.2 For any polynomial $f : \bar{N}^k \rightarrow \bar{N}^k$ and $i \in N$ we define polynomial $f^i : \bar{N}^k \rightarrow \bar{N}^k$ by $f^{i+1} = f \circ f^i$. and $f^1 = f$.

Definition 2.3 We say that $x_0 \in \bar{N}^k$ is the *least fixpoint* of $f : \bar{N}^k \rightarrow \bar{N}^k$ if x_0 is a fixpoint, that is $f(x_0) = x_0$ and for all fixpoints x of f, we have $x_0 \sqsubseteq x$

Next theorem follows from the so called Knaster-Tarski lemma:

Theorem 2.4 Let $f : \bar{N}^k \to \bar{N}^k$ be a polynomial. Then $\bigcup_{i=0}^{\infty} f^i(\underbrace{0, ..., 0}_{k})$ is the least fixed point of f.

Lemma 2.5 Let $f : \bar{N} \times \bar{N}^k \to \bar{N}$ be a polynomial. For each $\vec{y} \in \bar{N}^k$ we define polynomial $g^{\vec{y}} : \bar{N} \to \bar{N}$ by $g^{\vec{y}}(x) = f(x, \vec{y})$. According to Knaster Tarski lemma polynomial $g^{\vec{y}}$ has the least fixed point. Let $g : \bar{N}^k \to \bar{N}$ be a function which maps $\vec{y} \in \bar{N}^k$ to the least fixpoint of $g^{\vec{y}}$. Then g is a polynomial.
Proof: Polynomial f can be effectively turned in to the form $f(x, \vec{y}) = \sum_{i=1}^{p} \alpha_i(\vec{y})x^i + \alpha_0(\vec{y})$ where α_i for $0 \leq i \leq p$ are polynomials $\bar{N}^k \to \bar{N}$. Let $\alpha(\vec{y})$ be $\sum_{i=1}^{p} \alpha(\vec{y})$. Let us define $g(\vec{y})$ by $\alpha_0(\vec{y}) + \alpha_0(\vec{y})\alpha(\vec{y})\omega$. We can check that $g(\vec{y})$ is a fixpoint for $g^{\vec{y}}$. If $\alpha_0(\vec{y}) = 0$ then the least fixpoint of f is 0. If $\alpha_0(\vec{y}) \neq 0$ and $\alpha(\vec{y}) = 0$, then for all $i \leq p$ $\alpha_i(\vec{y}) = 0$. So function $g^{\vec{y}}$ has a form $g^{\vec{y}}(x) = \alpha_0(\vec{y})$. Therefore the least fixpoint is $\alpha_0(\vec{y})$. In the case of $\alpha_0(\vec{y}) \neq 0$ and $\alpha(\vec{y}) \neq 0$, the least fixpoint is ω. Hence function g always returns the fixpoint of $g^{\vec{y}}$. It can be easily checked that g is also minimal fixpoint. This lemma is valid even in the case when $k = 0$. In such case all α_i are just elements of $\bar{N}$.

Lemma 2.6 (vectorial fixed points) Let (f, F) be a system of polynomials $(f, F) : \bar{N}^{k+1} \to \bar{N}^{k+1}$ such that $f : \bar{N}^{k+1} \to \bar{N}$ is a single polynomial and $F : \bar{N}^{k+1} \to \bar{N}^k$ is a system of k polynomials. According to Knaster Tarski lemma (theorem 2.4) (f, F) admits least fixpoint $(x_0, \vec{y}_0)$. For each $\vec{y} \in \bar{N}^k$ let us define polynomial $g^{\vec{y}} : \bar{N} \to \bar{N}$ by $g^{\vec{y}}(x) = f(x, \vec{y})$. Let $g : \bar{N}^k \to \bar{N}$ be the function (see lemma 2.5) such that $g(\vec{y})$ is the least fixpoint of $g^{\vec{y}}$. From lemma 2.5 it follows that $g(\vec{y})$ is a polynomial. Let us define $F' : \bar{N}^k \to \bar{N}^k$ by $F'(\vec{y}) = F(g(\vec{y}), \vec{y})$. As a composition of polynomials F' must be a polynomial. Let $\vec{y'}_0$ be a fixpoint of F'. Define $x'_0 = g(\vec{y'}_0)$.
Thesis: $(x_0, \vec{y}_0) = (x'_0, \vec{y'}_0)$.
Proof: This lemma is a well known fact in the fixpoint theory. Consult for example [1] (page 165, chapter 2.2 Vectorial Fixed Points)
($\sqsubseteq$) Since $g(\vec{y})$ is a fixpoint of $g^{\vec{y}}$ it means that $g(\vec{y}) = f(g(\vec{y}), \vec{y})$. Since $\vec{y'}_0$ is a fixpoint of F' it means that $F(g(\vec{y'}), \vec{y'}_0) = \vec{y'}_0$. First we show that $(x'_0, \vec{y'}_0)$ is a fixpoint of (f, F).

$$f(x'_0, \vec{y'}_0) = f(g(\vec{y'}_0), \vec{y'}_0) = g(\vec{y'}_0) = x'_0$$

$$W(x'_0, \vec{y'}_0) = W(g(\vec{y'}_0), \vec{y'}_0) = \vec{y'}_0$$

Since $(x'_0, \vec{y'}_0)$ is a fixpoint of (f, F) then $(x_0, \vec{y}_0) \sqsubseteq (x'_0, \vec{y'}_0)$.
($\sqsupseteq$) Since x_0 is a fixpoint of the mapping $x \to f(x, \vec{y})$ then $g(\vec{y}) \sqsubseteq x_0$. By monotonicity of F, $F'(\vec{y}_0) = F(g(\vec{y}_0), \vec{y}_0) \sqsubseteq F(x_0, \vec{y}_0) = \vec{y}_0$. But the least fixed point of F' may be characterized by $\bigcap\{x : F'(x) \sqsubseteq x\}$ (see [5] proposition 5.1 page 26). Hence $\vec{y'}_0 \sqsubseteq y_0$. By monotonicity of g, $x'_0 = g(\vec{y'}_0) \sqsubseteq g(\vec{y}_0) \sqsubseteq x_0$.

This result is a base for the algorithm which for a given polynomial $f : \bar{N}^k \to \bar{N}^k$ returns $x_0 \in \bar{N}^k$ which is the least fixpoint of f.

Least Fixpoint Algorithm

Input: polynomial $f : \bar{N}^k \to \bar{N}^k$.
Output: $x_0 \in \bar{N}^k$.
Specification: x_0 is the least fixpoint of f.

By k we mean the dimension of the problem f. The algorithm is recursive on k.

First we solve a one dimensional problem.
Any polynomial $f : \bar{N} \to \bar{N}$ can be effectively transformed to the form $f(x) = \sum_{i=1}^{p} \alpha_i x^i + \alpha_0$ for some array of $\alpha_i \in \bar{N}$. The least fixpoint of f is $\alpha_0 + \alpha_0\alpha\omega$ where $\alpha = \sum_{i=1}^{p} \alpha_i$ (see lemma 2.5 for $k = 0$).

Solving $(k+1)$-dimensional problem.
Let $(f, F) : \bar{N} \times \bar{N}^k \to \bar{N}^{k+1}$ be a $(k+1)$-dimensional polynomial. Therefore $f : \bar{N} \times \bar{N}^k \to \bar{N}$ is a polynomial and $F : \bar{N} \times \bar{N}^k \to \bar{N}^k$ is a system of polynomials. The polynomial f can be effectively turned into the form $f(x, \vec{y}) = \sum_{i=1}^{p} \alpha_i(\vec{y})x^i + \alpha_0(\vec{y})$. Let us define $\alpha(\vec{y}) = \sum_{i=1}^{p} \alpha(\vec{y})$. Let us construct polynomial $g : \bar{N}^k \to \bar{N}$ by $g(\vec{y}) = \alpha_0(\vec{y}) + \alpha_0(\vec{y})\alpha(\vec{y})\omega$. We define $F' : \bar{N}^k \to \bar{N}^k$ by the formula $F'(\vec{y}) = F(g(\vec{y}), \vec{y})$. Since we assume that there is a procedure for solving any k dimensional problem, let $\vec{y_0}$ be the least fixpoint of F'. Let us define x_0 as $g(\vec{y_0})$. The pair $(x_0, \vec{y_0})$ is the least fixpoint for the main problem. The correctness of this procedure is guaranteed by lemmas 2.6, 2.5.

Lemma 2.7 The pair (F, f) is a system of polynomials where $f : \bar{N}^k \times \bar{N} \to \bar{N}$ and $F : \bar{N}^k \times \bar{N} \to \bar{N}^k$. Let $(\vec{x}_0, y_0)$ be a least fixed point of (F, f). Let $\vec{x} \in \bar{N}^k, y \in \bar{N}, z \in \bar{N}$ be such that $y \leq z$ and both are grater then 0 and finite and $(\vec{x}, y) \sqsubseteq (\vec{x}_0, y_0)$ then $(f, F)(\vec{x}, z) \sqsubseteq (\vec{x}_0, y_0)$.
Proof: For $y_0 = \omega$ it is trivial. Let us assume that y_0 if finite. Then it must be greater than 0. Let the polynomial f have the form $f(\vec{x}, y) = \sum_{i=1}^{p} \alpha_i(\vec{x})y^i + \alpha_0(\vec{x})$. Since $(\vec{x}_0, y_0)$ is a fixpoint of (F, f) then we have $y_0 = \sum_{i=1}^{p} \alpha_i(\vec{x}_0)y_0^i + \alpha_0(\vec{x}_0)$. We will prove that for all $1 \leq i \leq p$, $\alpha_i(\vec{x}_0) = 0$. If not, $\alpha_0(\vec{x}_0) = 0$. So for all $x \sqsubseteq x_0$, $\alpha_0(\vec{x}) = 0$. Let $(\vec{t}^{(\alpha)}, s^{\alpha})$ for $\alpha \in N$ be a chain of iterations of (F, f). Hence for all α, $\alpha_0(\vec{t}^{(\alpha)}) = 0$. Therefore $s^{(\alpha+1)} = f(\vec{t}^{(\alpha)}, s^{(\alpha)}) = \sum_{i=1}^{p} \alpha_i(\vec{t}^{(\alpha)})(s^{(\alpha)})^i = 0$. So $s^{(\alpha)} = 0$ for all α which contradicts with the assumption that fixpoint y_0 is greater then 0. So we proved that for all $i \leq p$, $\alpha_i(\vec{x}) = 0$. Hence for all t, $f(\vec{x}_0, t) = \alpha_0(\vec{x}_0)$. In particular $f(\vec{x}_0, z) = \alpha_0(\vec{x}_0)$. Consequently for $\vec{x} \sqsubseteq \vec{x}_0$ we have $f(\vec{x}, z) \leq \alpha_0(\vec{x}_0) = y_0$.

Theorem 2.8 Let $F : \bar{N}^k \to \bar{N}^k$ be a system of polynomials. Let $y^{(0)} \sqsubseteq y^{(1)} \sqsubseteq ...$ and $z^{(0)} \sqsubseteq z^{(1)} \sqsubseteq ...$ are two sequences of k-tuples which satisfy following conditions:
(1) $y^{(\alpha)} \sqsubseteq z^{(\alpha)}$ for all $\alpha \in N$
(2) if $y_i^{(\alpha)} = 0$ then $z_i^{(\alpha)} = 0$ for all $\alpha \in N$ and $i \leq k$
(3) if $\bigcup y_i^{(\alpha)} < \omega$ then $\bigcup z_i^{(\alpha)} < \omega$ for all $i \leq k$
(4) $y^{(\alpha+1)} = F(z^{(\alpha)})$ for all $\alpha \in N$
(5) $y^{(0)} = z^{(0)} = (0, ..., 0)$
Then least fixpoint of F is $\bigcup y^{(\alpha)}$.

Proof: Let $\vec{x}^{\,fix}$ be the least fixpoint of F. Let $\vec{t}^{(0)}, \vec{t}^{(1)}, ...$ be the chain of iterations of F. It means that $\vec{t}^{(0)} = (0...0)$ and $\vec{t}^{(\alpha+1)} = F(\vec{t}^{(\alpha)})$. We will see that $\vec{t}^{(\alpha)} \sqsubseteq \vec{y}^{\,(\alpha)}$. For $\alpha = 0$ it is obvious. For $\alpha > 0$ by induction using (1) and (4)

$$\vec{t}^{(\alpha+1)} = F(\vec{t}^{(\alpha)}) \sqsubseteq F(\vec{y}^{\,(\alpha)}) \sqsubseteq F(\vec{z}^{\,(\alpha)}) = \vec{y}^{\,(\alpha+1)}$$

Therefore $\vec{x}^{\,fix} = \bigcup t^{(\alpha)} \sqsubseteq \bigcup \vec{y}^{\,(\alpha)}$. We will prove also by induction that $\vec{y}^{\,(\alpha)} \sqsubseteq \vec{x}^{\,fix}$. For $\alpha = 0$ it holds. If $\vec{z}^{\,(\alpha)} = \vec{y}^{\,(\alpha)}$ then

$$\vec{y}^{\,(\alpha+1)} = F(\vec{z}^{\,(\alpha)}) = F(\vec{y}^{\,(\alpha)}) \sqsubseteq F(\vec{x}^{\,fix}) = \vec{x}^{\,fix}$$

Let us assume that $\vec{z}^{\,(\alpha)} \sqsupset \vec{y}^{\,(\alpha)}$. It means that there are coordinates $c_1, .., c_p$ such that $\vec{z}^{\,(\alpha)}_{c_i} > \vec{y}^{\,(\alpha)}_{c_i}$. Because of assumptions (2) and (3) $\vec{z}^{\,(\alpha)}_{c_i}$ and $\vec{y}^{\,(\alpha)}_{c_i}$ are finite and different then 0. Let we define $p+1$ k-tuples $\vec{x}_0, ..., \vec{x}_p$ such that $\vec{y}^{\,(\alpha)} = \vec{x}_0 \sqsubset \vec{x}_1 \sqsubset, ..., \sqsubset \vec{x}_p = \vec{z}^{\,(\alpha)}$ where consequent vectors $\vec{x}_i$ and $\vec{x}_{i+1}$ differs on exactly one c_i coordinate. By lemma 2.7 $F(\vec{x}_i) \sqsubseteq \vec{x}^{\,fix}$. Hence

$$\vec{y}^{\,(\alpha+1)} = F(\vec{z}^{\,(\alpha)}) \sqsubseteq \vec{x}^{\,fix}$$

3. The number of closed terms

Lemma 3.1 Let T be a closed term of type $\tau = \otimes_{j=1}^{n}\tau_j \rightarrow A$. There exists $i \leq arg(\tau)$ such that $\tau[i] \sim \tau$ and if $k = arg(\tau[i]) > 0$ then there are closed terms $G_1, ..., G_k$ of types $\otimes_{j=1}^{n}\tau_j \rightarrow \tau[i,1]$,..., $\otimes_{j=1}^{n}\tau_j \rightarrow \tau[i,k]$, respectively, such that

$$T = \lambda x_1...x_n.x_i(G_1x_1...x_n)...(G_kx_1...x_n)$$

Proof. If the long normal form of T is a projection $\lambda x_1...x_n.x_i$ then $\tau[i] = A$. In this case $arg(\tau[i]) = 0$ and obviously $\tau[i] \sim \tau$. If T has a long normal form $T = \lambda x_1...x_n.x_iK_1...K_k$ then let us define $G_1, ..., G_k$ by $G_j = \lambda x_1...x_n.K_j$ for $j \leq k$.

Definition 3.2 For any type τ we define $S_\tau = \{i \leq arg(\tau) : \tau \sim \tau[i] \;\; and \;\; rank(\tau[i]) \geq 1\}$ and $T_\tau = \{i \leq arg(\tau) : \tau \sim \tau[i] \;\; and \;\; rank(\tau[i]) = 0\}$.

Lemma 3.3 If $S_\tau \cup T_\tau = \emptyset$ then $|\tau| = 0$
Proof is a trivial consequence of lemma 3.1.

Theorem 3.4 If $S_\tau \cup T_\tau \neq \emptyset$ then

(1) $\quad |\tau| = |T_\tau| + \sum_{i \in S_\tau} \prod_{p=1}^{arg(\tau[i])} |\otimes_{j=1}^{n}\tau[j] \rightarrow \tau[i,p]|$

(2) $\quad |\tau|_\alpha = |T_\tau| + \sum_{i \in S_\tau} \prod_{p=1}^{arg(\tau[i])} |\otimes_{j=1}^{n}\tau(j) \rightarrow \tau(i,p)|_{\alpha-1} \quad \alpha > 0$

Proof: (equation 1) For each $i \in T_\tau$ there is exactly one projection $\lambda x_1...x_n.x_i$ of type τ . If $S_\tau = \emptyset$ then $\sum_{i \in S_\tau} = 0$. Assuming $S_\tau \neq \emptyset$ for every $i \in S_\tau$ we count the number of closed terms which are on the form $T = \lambda x_1...x_n.x_iK_1...K_k$, that is the closed terms with the heading x_i where $k = arg(\tau[i])$. The equivalent form of the term of type τ with heading x_i is

$$T = \lambda x_1 ... x_n . x_i (G_1 x_1 ... x_n) ... (G_k x_1 ... x_n)$$

where $G_1, ..., G_k$ are closed terms defined by $G_j = \lambda x_1 ... x_n . K_p$ for $p \leq k$. Each term G_p is of type $\otimes_{j=1}^{n} \tau[j] \to \tau[i,p]$ for $p \leq k$. Since any combination of closed terms $G_1, ..., G_k$ uniquely determines a closed term given by the formula

$$\lambda x_1 ... x_n . x_i (G_1 x_1 ... x_n) ... (G_k x_1 ... x_n)$$

of type τ with heading x_i, therefore the global number of terms of type τ with headings x_i is given as the product $\prod_{p=1}^{k} |\otimes_{j=1}^{n} \tau[j] \to \tau[i,p]|$. Note that the formula is correct even when $|\otimes_{j=1}^{n} \tau[j] \to \tau[i,p]| = 0$ for some p and $|\otimes_{j=1}^{n} \tau[j] \to \tau[i,p']| = \omega$ for some p' since by the definition of multiplication $0 * \omega = 0$.

Proof: (equation 2) It follows from previous equation and the definition of complexity π of closed terms (see page 2).

Lemma 3.5 Let τ be a type and $\tau_1, ..., \tau_n$ are all direct prolongations of τ . Let $x = |\tau|$ and $x_1 = |\tau_1|, ..., x_n = |\tau_n|$. There is a polynomial $f : N^n \to N$ such that $x = f(x_1, ..., x_n)$.
Proof: From lemma 3.4 we have that

$$|\tau| = |T_\tau| + \sum_{i \in S_\tau} \prod_{p=1}^{arg(\tau[i])} |\otimes_{j=1}^{n} \tau[j] \to \tau[i,p]|$$

Since on right hand side of the formula there are only multiplications and additions of $x_1, ..., x_n$ and possibly some constants, therefore there is a polynomial f such that $x = f(x_1, ..., x_n)$.

Example 3.6 Let $x, y, z \in \bar{N}$ be numbers of closed terms of types respectively:
$(B, C \to C), C, C, (B, C \to B), (C, C \to A), (A, B \to A) \to A$
$(B, C \to C), C, C, (B, C \to B), (C, C \to A), (A, B \to A) \to B$
$(B, C \to C), C, C, (B, C \to B), (C, C \to A), (A, B \to A) \to C$
By Theorem 3.4 we have following equations between x, y and z

$$\begin{aligned} x &= \; xy + z^2 \\ y &= \; yz \\ z &= \; yz + 2 \end{aligned}$$

Lemma 3.7 Let $i \geq 1$. Types $\tau \to \mu$ and $\tau^i \to \mu$ are of the same size.
Proof: Let N be a number of closed terms for the type $\tau \to \mu$ and P be a number of closed terms for the type $\tau^i \to \mu$. We are going to prove that:

(1)	$N = 0$	if and only if	$P = 0$
(2)	$0 < N < \omega$	if and only if	$0 < P < \omega$
(3)	$N = \omega$	if and only if	$P = \omega$

For $i = 1$ there is nothing to prove. We start the proof with $i = 2$. Certainly in any case $N \leq P$. (1) If T is a closed term of type $\tau \to \mu$ then $\lambda x_\tau y_\tau . Tx$ is a term of type $\tau^2 \to \mu$. If S is a term of type $\tau^2 \to \mu$ then $\lambda x_\tau . Sxx$ is a closed term of type $\tau \to \mu$.

(2) Obviously if $P < \omega$ then $N < \omega$. Suppose $0 < N < \omega$. Certainly $0 < P$. Let $\lambda x_\tau.T_1, ..., \lambda x_\tau.T_N$ be all closed terms of type $\tau \to \mu$. We are going to prove that P is finite. Let k_j be a number of occurrences of the variable x_τ in T_j for $j \leq N$. We will write $T_j[x_\tau, ..., x_\tau]$ instead of T_j. There is 2^{k_j} of different strings of x_τ and y_τ which can be replaced in the term T_j. Accordingly for any term $\lambda x_\tau.T_j[x_\tau, ..., x_\tau]$ there is the 2^{k_j} terms $\lambda x_\tau y_\tau.T_j$ with x_τ and y_τ inserted. Therefore the number P of closed terms of type $\tau^2 \to \mu$ is $2^{k_1} + ... + 2^{k_N}$. The minimum of the expression $2^{k_1} + ... + 2^{k_N}$ is N when $k_1 = ... = k_N = 0$. So we have that P is finite and $N \leq P$.
(3) this case is proved by (1) and (2).
Let $\cong$ stand for a *same size* relation between types (see definition 1.4). For $i \geq 3$ proof by induction.

$$\tau^i \to \mu = \tau^2 \to (\tau^{i-2} \to \mu) \overset{case\, i=2}{\cong} \tau \to (\tau^{i-2} \to \mu) = \tau^{i-1} \to \mu \overset{induction}{\cong} \tau \to \mu$$

Lemma 3.8 Let $\tau = \otimes_{j=1}^{n} \tau[j] \to A$ and let $f : \{1, ...n\} \to \{1, ..., n\}$ be a permutation. If type μ is $\otimes_{j=1}^{n} \tau[f(j)] \to A$ then $|\tau| = |\mu|$.
Proof: If T is a closed term of type $\otimes_{j=1}^{n} \tau[j] \to A$ then $\lambda x_1 ... x_n.T x_{f^{-1}(1)} ... x_{f^{-1}(n)}$ is an equivalent term of type μ.

Lemma 3.9 If type τ can be embedded in μ then τ and μ are of the same size.
Proof: Let μ has the form $\otimes_{j=1}^{n} \mu[j] \to A$. Since τ can be embedded in μ then any component of τ is a component of μ. By appropriate permutation of components of τ we can obtain type $\tau' = (\mu[1])^{i_1}, ..., (\mu[n])^{i_n} \to A$ for some $i_1, ..., i_n \geq 1$. By lemma 3.8 $|\tau| = |\tau'|$. Now, let we define the sequence of types $\nu_0, ..., \nu_n$ by

$$\begin{aligned}
\nu_0 = &\quad (\mu[1])^{i_1}, ..., (\mu[n])^{i_n} \to A \\
&\vdots \\
\nu_j = &\quad \mu[1], ..., \mu[j], (\mu[j+1])^{i_{j+1}}, ..., (\mu[n])^{i_n} \to A \\
&\vdots \\
\nu_n = &\quad (\mu[1]), ..., (\mu[n]) \to A
\end{aligned}$$

We are going to prove by induction that all types $\nu_0, ..., \nu_n$ are of the same size as $\tau' = (\mu[1])^{i_1}, ..., (\mu[n])^{i_n} \to A$. Suppose it is true for $\nu_0, ..., \nu_j$. By appropriate permutation we turn the type ν_j into form of type $\nu'_j = (\mu[j+1])^{i_{j+1}}, \mu[1], ..., \mu[j], (\mu[j+2])^{i_{j+2}}, ..., (\mu[n])^{i_n} \to A$. Utilizing lemma 3.7 we can turn ν'_j into a *same size* type $\nu''_j = (\mu[j+1]), \mu[1], ..., \mu[j], (\mu[j+2])^{i_{j+2}}, ..., (\mu[n])^{i_n} \to A$. The type ν''_j can be permutated in to ν_{j+1}.

Lemma 3.10 If type τ can be embedded in μ then for every $\alpha \in N$ $|\mu|_\alpha \leq |\tau|_\alpha$ and moreover if $|\mu|_\alpha = 0$ then $|\tau|_\alpha = 0$.
Proof: It follows from the proof of lemma 3.9.

Polynomial Construction Algorithm

Input: type τ
Output: number $n \in N$ and list of types $(\tau_1, ..., \tau_n)$ and polynomial $F : \bar{N}^n \to \bar{N}^n$.

Specification: $\tau_1 = \tau$ and if $(x_1, ..., x_n)$ is the least fixed point of F, then $x_i = |\tau_i|$ for all $i \leq n$.
Let us consider the extension tree for type τ. According to theorem 1.7 the tree is finite. Let $\tau_1, ..., \tau_n$ be all types in the tree. Let us put $\tau_1 = \tau$. For each type τ_i we introduce a new variable x_i which will denote $|\tau_i|$. Let $k \leq n$ be the number of extensions of τ in the tree. Note that some terminal nodes in the extension tree may not be extensions of τ (see definition 1.6). Let $\tau_1, ..., \tau_k$ be all extensions of τ in the tree. We are going to construct k polynomials $f_1, ..., f_k$ one for each extension of type τ. If a current node $\mu = \tau_i$ in the tree is an extension of τ and μ has direct prolongations $\mu_1, ..., \mu_p$ then according to lemma 3.5 there is a polynomial say f_i such that $f_i : \bar{N}^p \to \bar{N}$. f_i can be viewed as a function of all $x_1, ..., x_n$ arguments. Therefore

$$(1)\ x_i = f_i(x_1, ..., x_n)$$

Let τ_i be a terminal node in the tree such that τ_i is an extension of τ and has no direct prolongations. Let $s = |T_{\tau_i}|$ (see definition 3.2). Therefore

$$(2)\ x_i = s$$

Let τ_i be a terminal node which is not an extension of τ (for $i > k$). Therefore there is a predecessor of τ_i, say τ_s such that τ_i is embeddable in τ_s. Type τ_s is an extension of τ. Therefore

$$(3)\ x_i = x_s$$

Hence we have exactly n polynomial equations between variables.

$$\begin{array}{lll}
x_1 & = f_1(x_1, ..., x_n) & \\
\vdots & & \\
x_k & = f_k(x_1, ..., x_n) & \\
x_{k+1} & = x_{s_1} & \text{where } \tau_{k+1} \text{ is embeddable in } \tau_{s_1} \in \{\tau_1, ..., \tau_k\} \\
\vdots & & \vdots \\
x_n & = x_{s_{n-k}} & \text{where } \tau_n \text{ is embeddable in } \tau_{s_{n-k}} \in \{\tau_1, ..., \tau_k\}
\end{array}$$

Using last $n - k$ equations we can replace variables $x_{k+1}, ..., x_n$ denoting types which are not extensions of τ, by other variables denoting some extensions of τ. Also we may assume that for every i variable x_i have an appearance at the polynomial f_i. If not, we may reduce the number of equations by substituting polynomial $f_i(x_1, ..., x_k)$ for x_i in all remaining polynomials. For instance we may reduce this way all equations which are on the form $x_i = s$.

Lemma 3.11 (correctness) Let $(\tau_1, ..., \tau_k, \tau_{k+1}, ..., \tau_n)$ and $F = (f_1, ..., f_k, f_{k+1}, ...f_n)$ be all types and all polynomials introduced by the polynomial construction algorithm. We prove that, if $(x_1, ..., x_n)$ is the least fixpoint of $F = (f_1, ..., f_n)$ then $x_i = |\tau_i|$ for all $i \leq n$.
Proof. We divide the proof of correctness into two separate cases. The first one is the following: In the extension tree for type τ each terminal node is an extension of τ. It means that $n = k$. In this case the construction algorithm introduces polynomials only by construction (1) and (2) but not (3).
For any number $\alpha \in N$ we define $|\tau|_\alpha$ as a number of closed terms T of type τ with complexity $\pi(T)$ not grater then α.

By iteration of F we define the sequence of k-tuples $(x_1^{(\alpha)}, ..., x_k^{(\alpha)})$ for $\alpha \in N$ by $(x_1^{(0)}, ..., x_k^{(0)}) = (0, ..., 0)$ and $(x_1^{(\alpha+1)}, ..., x_k^{(\alpha+1)}) = F(x_1^{(\alpha)}, ..., x_k^{(\alpha)})$. We shall show by induction on α that $|\tau_s|_\alpha = x_s^\alpha$. For $\alpha = 0$ there is nothing to prove. For $\alpha = 1$ it is obvious since the number of terms with complexity 1 (projections) is the same as the number of elements of the set T_τ. Suppose it is true for all $s \leq k$ and for all $\beta < \alpha$. According to lemma 3.4

$$|\tau_s|_\alpha = |T_{\tau_s}| + \sum_{i \in S_\tau} \prod_{p=1}^{arg(\tau[i])} |\otimes_{j=1}^n \tau_s(j) \rightarrow \tau_s(i,p)|_{\alpha-1}$$

which can be translated into form

$$|\tau_s|_\alpha = f_s(|\tau_1|_{\alpha-1}, ..., |\tau_k|_{\alpha-1})$$

By induction it can be turned in to

$$|\tau_s|_\alpha = f_s(x_1^{(\alpha-1)}, ..., x_k^{(\alpha-1)})$$

So finally we have

$$|\tau_s|_\alpha = x_s^{(\alpha)}$$

Now we will prove the second case when $k < n$. Types $\tau_1, ..., \tau_k$ are extensions of τ and $\tau_{k+1}, ..., \tau_n$ are not. Accordingly τ_{k+i} can be embedded in $\tau_{s_i} \in \{\tau_1, ..., \tau_k\}$ for $1 \leq i \leq n-k$. Let us consider two chains of n-tuples of numbers. The first one $y^{(0)} \sqsubseteq y^{(1)} \sqsubseteq ...$ is defined by $y^{(\alpha)} = (|\tau_1|_\alpha, ..., |\tau_k|_\alpha, |\tau_{s_1}|_\alpha, ..., |\tau_{s_{n-k}}|_\alpha)$. The second chain $z^{(0)} \sqsubseteq z^{(1)} \sqsubseteq ...$ is defined by $z^{(\alpha)} = (|\tau_1|_\alpha, ..., |\tau_k|_\alpha, |\tau_{k+1}|_\alpha, ..., |\tau_n|_\alpha)$. We are going to clarify whether chains $y^{(0)}, y^{(1)}, ...$ and $z^{(0)}, z^{(1)}, ...$ satisfy conditions (1), (2), (3) (4) and (5) from the theorem 2.8.

(1) $y^{(\alpha)} \sqsubseteq z^{(\alpha)}$. First k- coordinates of $y^{(\alpha)}$ and $z^{(\alpha)}$ are identical. For coordinates $k+1, ..., n$ it follows from lemma 3.10 and from the fact that type τ_{k+j} is embeddable in type τ_{s_j}.

(2) if $y_i^{(\alpha)} = 0$ then $z_i^{(\alpha)} = 0$ follows from the second part of lemma 3.10. since types either are identical or embeddable.

(3) if $\bigcup y_i^{(\alpha)} < \omega$ then $\bigcup z_i^{(\alpha)} < \omega$ for all $i \leq n$ follows from the fact that types in chains are identical or the first is embeddable in to the second one. By lemma 3.9 it follows that types like this are of the same size.

(4) $y^{(\alpha+1)} = F(z^{(\alpha)})$ follows from the construction of the algorithm and theorem 3.4 equation 2.

(5) $y^{(0)} = z^{(0)} = (0, ..., 0)$ follows from the definition of chains.

Since conditions (1), (2), (3) ,(4) and (5) are true then it follows from theorem 2.8 that $\bigcup y^{(\alpha)}$ is the same as the least fixpoint of F. So the replacement of types by other types of the same size can be done without effecting the solution of the system of polynomial equations.

Example 3.12 This is a continuation of example 3.6. The problem is to find the number of normal forms for type $(B \rightarrow C \rightarrow C) \rightarrow C \rightarrow C \rightarrow (B \rightarrow C \rightarrow B) \rightarrow (C \rightarrow C \rightarrow A) \rightarrow (A \rightarrow B \rightarrow A) \rightarrow A$. Performing the polynomial construction algorithm we have to introduce few types and variables denoting the number of normal forms of those types. Let

$$\begin{array}{l}
x = |(B \to C \to C) \to C \to C \to (B \to C \to B) \to (C \to C \to A) \to (A \to B \to A) \to A| \\
y = |(B \to C \to C) \to C \to C \to (B \to C \to B) \to (C \to C \to A) \to (A \to B \to A) \to B| \\
z = |(B \to C \to C) \to C \to C \to (B \to C \to B) \to (C \to C \to A) \to (A \to B \to A) \to C|
\end{array}$$

By the polynomial construction algorithm we have the following equations between x, y and z.

$$\begin{array}{rl}
x = & xy + z^2 \\
y = & yz \\
z = & yz + 2
\end{array}$$

Using the least fixpoint algorithm we obtain the least fixed point of the equations which is:

$$\begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 4 \\ 0 \\ 2 \end{pmatrix}$$

Using Knaster Tarski theorem we can reach the fixpoint after 2 iterations of the polynomial:

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \xrightarrow{f} \begin{pmatrix} 0 \\ 0 \\ 2 \end{pmatrix} \xrightarrow{f} \begin{pmatrix} 4 \\ 0 \\ 2 \end{pmatrix} \xrightarrow{f} \begin{pmatrix} 4 \\ 0 \\ 2 \end{pmatrix} \xrightarrow{f}$$

Therefore there are exactly 4 closed terms of type $(B \to C \to C) \to C \to C \to (B \to C \to B) \to (C \to C \to A) \to (A \to B \to A) \to A$

Example 3.13 In this example we will construct the system of polynomials and accordingly number of proofs for the classical formula which is intuitionistically not provable, namely for Pierce law $((A \supset B) \supset A) \supset A$. Connective $\supset$ stands for the classical implication and is definable by means of the intuitionistic implication $\to$ and the falsity symbol $\perp$ as follows: $\sim A$ is $A \to\perp$ and $A \supset B$ is $A \to (\sim B \to\perp)$. Therefore after translation the Pierce law has the form:

$$(((A \to (B \to\perp) \to\perp) \to (A \to\perp) \to\perp) \to (A \to\perp) \to\perp$$

Performing the polynomial construction algorithm we will introduce few types and variables denoting the number of normal forms of those types. Let

$$\begin{array}{rl}
x = & |(((A \to (B \to\perp) \to\perp) \to (A \to\perp) \to\perp) \to (A \to\perp) \to\perp| \\
y = & |(((A \to (B \to\perp) \to\perp) \to (A \to\perp) \to\perp) \to (A \to\perp) \to A \to (B \to\perp) \to\perp| \\
z = & |(((A \to (B \to\perp) \to\perp) \to (A \to\perp) \to\perp) \to (A \to\perp) \to A \to\perp| \\
u = & |(((A \to (B \to\perp) \to\perp) \to (A \to\perp) \to\perp) \to (A \to\perp) \to A| \\
s = & |(((A \to (B \to\perp) \to\perp) \to (A \to\perp) \to\perp) \to (A \to\perp) \to A \to A \to (B \to\perp) \to\perp| \\
t = & |(((A \to (B \to\perp) \to\perp) \to (A \to\perp) \to\perp) \to (A \to\perp) \to A \to A \to\perp| \\
w = & |(((A \to (B \to\perp) \to\perp) \to (A \to\perp) \to\perp) \to (A \to\perp) \to A \to A| \\
v = & |(((A \to (B \to\perp) \to\perp) \to (A \to\perp) \to\perp) \to (A \to\perp) \to A \to (B \to\perp) \to A \to (B \to\perp) \to\perp| \\
p = & |(((A \to (B \to\perp) \to\perp) \to (A \to\perp) \to\perp) \to (A \to\perp) \to A \to (B \to\perp) \to A \to\perp| \\
q = & |(((A \to (B \to\perp) \to\perp) \to (A \to\perp) \to\perp) \to (A \to\perp) \to A \to (B \to\perp) \to A| \\
r = & |(((A \to (B \to\perp) \to\perp) \to (A \to\perp) \to\perp) \to (A \to\perp) \to A \to (B \to\perp) \to B|
\end{array}$$

According to the polynomial construction algorithm we have the following equations:

$$\begin{aligned} x &= yz+u \\ u &= 0 \\ y &= vp+q+r \\ r &= 0 \\ q &= 1 \\ z &= st+w \\ w &= 1 \end{aligned}$$

$$\begin{aligned} v &= y && v \ is \ embeddable \ in \ y \\ p &= y && p \ is \ embeddable \ in \ y \\ s &= y && s \ is \ embeddable \ in \ y \\ t &= z && t \ is \ embeddable \ in \ z \end{aligned}$$

By substituting the system can be easily reduced to

$$\begin{aligned} x &= yz \\ y &= y^2+1 \\ z &= yz+1 \end{aligned}$$

So according to the least fixpoint algorithm we have the least solution of the system $y = \omega, z = \omega$ and consequently $x = \omega$. Therefore there exists an infinite number of normal forms (normal proofs) of type (formula) $((A \supset B) \supset A) \supset A$. Using Knaster Tarski theorem we have a divergent sequence of tuples.

$$\begin{pmatrix} 0 \\ 0 \\ 0 \end{pmatrix} \xrightarrow{f} \begin{pmatrix} 0 \\ 1 \\ 1 \end{pmatrix} \xrightarrow{f} \begin{pmatrix} 1 \\ 2 \\ 2 \end{pmatrix} \xrightarrow{f} \begin{pmatrix} 4 \\ 5 \\ 5 \end{pmatrix} \xrightarrow{f} \cdots$$

The opposite construction is also possible. From the given polynomial we can effectively construct types which will produce by the *polynomial construction Algorithm* the same polynomial we started with.

Lemma 3.14 . Let $F : \bar{N}^k \to \bar{N}^k$ be a system of polynomials. There are effectively rank 2 types $\tau_1, ..., \tau_k$ such that the least fixpoint $x_0 = (x_0^1, ..., x_0^k)$ of F is such that $x_0^i = |\tau_i|$ for all $i \leq k$.
Proof: By simple encoding polynomials into equivalent type system.

Lemma 3.15 For every type τ we can effectively construct rank 2 type τ' such that $|\tau| = |\tau'|$.
Proof: By polynomial construction algorithm and lemma 3.14.

4. Counting closed terms in λ calculus with a single ground type

The *number of normal forms* problem in the simplest case when the language of typed λ calculus is build over the type system based on single type can be reduced to the well known Statman hierarchy (see [6]). Suppose the only ground type is O. Type τ is reducible to μ written as $\tau \succ \mu$ if there is a closed term M of type $\tau \to \mu$ such that for all N_1, N_2 of type τ, $N_1 =_{\beta\eta} N_2$ if and only if $MN_1 =_{\beta\eta} MN_2$. Types $\tau \simeq \mu$ are equivalent if $\tau \succ \mu$ and $\mu \succ \tau$. The relation $\simeq$ is an equivalence relation. Equivalence classes are ordered by $[\tau] > [\mu]$ if $\tau \succ \mu$. Statman in [6] proved that equivalence classes are well ordered in ordinal $\omega + 3$ (see also [2]). It is easy to observe that equivalent types possess the same number of closed terms in normal forms. Therefore the following test is sufficient to decide the number of closed terms. First decide if the given type is an equivalent with respect to $\simeq$ with type O. This is the same as to ask whether or not the formula is unprovable. If the answer is YES then the type has no closed terms. Next, decide if the given type is equal to $O^n \to O$. If so then the type has exactly n closed terms. Otherwise type has infinite number of closed terms. This is based on the fact that all of types $\omega, \omega + 1, \omega + 2$ and $\omega + 3$ in the Statman hierarchy are having infinite number of closed terms.

Acknowledgments The author would like to thank Helmut Schwichtenberg for valuable comments and criticism on the early version of this paper and Roger Hindley for discussion and remarks. The author would like to thank as well Anna Nizegorodcew for her heroic efforts in careful proofreading.

References

[1] A. Arnold, D. Niwinski. Fixed point characterization of weak monadic logic definable sets of trees *Tree Automata and Languages* 1992, pp 159 -188.

[2] W. Dekkers. Reducibility of types in Typed Lambda Calculus *Information and Computation vol 77, No 2 (1988) pp 131– 137.*

[3] Private communication from Roger Hindlay. Will appear in forthcoming book in 1995.

[4] S. Hirokawa. Infiniteness of Proof (α) is polynomial space complete , private communication

[5] J.W. Lloyd. Foundations of Logic Programming, Springer- Verlag, 1984

[6] R. Statman. On the existence of closed terms in the typed λ-calculus. In: R. Hindley and J. Seldin, eds. Combinatory Logic, Lambda Calculus and Formalism (Academic Press, New York, 1980).

[7] R. Statman. On the Existence of closed terms in typed λ calculus II: Transformations of unification problems *Theoretical Computer Science 15 (1981) 329–338*

[8] D.A. Wolfram. The Clausual Theory of Types, Cambridge Tracts in Theoretical Computer Science 21, Cambridge University Press 1993.

[9] M. Zaionc. Mechanical Procedure for Proof Construction via Closed Terms in Typed λ calculus *Journal of Automated Reasoning 4 (1988) 173–190*