Counting a type's principal inhabitants

Sabine Broda, Luís Damas

Technical Report Series: DCC-98-12

Departamento de Ciência de Computadores – Faculdade de Ciências

&

Laboraório de Inteligência Artificial e Ciência de Computadores

Universidade do Porto

Rua do Campo Alegre, 823 4150 Porto, Portugal

Tel: +351+2+6001672 – Fax: +351+2+6003654

http://www.ncc.up.pt/fcup/DCC/Pubs/treports.html

Counting a type's principal inhabitants

Sabine Broda, Luís Damas
DCC & LIACC, Universidade do Porto
R. do Campo Alegre 823, 4150-180 Porto, Portugal
Phone: 351 2 6078830, Fax: 351 2 6003654

e-mail: {sbb,luis}@ncc.up.pt

Abstract

We present a Counting Algorithm that computes the number of λ-terms in β-normal form that have a given type τ as a principal type and produces a list of these terms. The design of the algorithm follows the lines of Ben-Yelles' algorithm for counting normal (not necessarily principal) inhabitants of a type τ.

1 Introduction

In [2], Ben-Yelles presented a Counting Algorithm, also described in [3], which given a type τ computes the number of λ-terms in β-normal form that can receive type τ in $\mathbf{TA}_\lambda$. For each type τ the algorithm decides in a finite number of steps whether the number of closed β-normal forms with type τ is finite or infinite, computes this number in the finite case, and lists all relevant terms in both cases. Related to this is the problem of counting the number of β-normal forms that have a given type τ as a principal type. As pointed out in ([3], p. 127), this problem is still open and in this paper we present a Counting Algorithm which solves this case. Analogous to Ben-Yelles' algorithm, our algorithm for counting (and listing) principal normal inhabitants of a type τ is based on the following facts. First, it is sufficient to look for a special kind of principal normal inhabitants of τ, called long terms. Second, there are integers $0 < \mathbf{d_p}(\tau) < \mathbf{D_p}(\tau)$ such that the cardinality of the set of principal normal inhabitants of τ depends directly on the number of long principal normal inhabitants of τ with depth in $[0; \mathbf{d_p}(\tau)[$ and on the number of those with depth in $[\mathbf{d_p}(\tau); \mathbf{D_p}(\tau)[$, where depth is a measure on the structure of a λ-term in β-normal form. Finally, Ben-Yelles defined in [2] a Search Algorithm (others are in [6], [7] and [5]), that given a type τ and any integer $d \geq 0$ can be used to compute all (a finite number) long normal inhabitants of a type τ with depth $\leq d$. Thus, using any principal-type checking algorithm (for example in [3]) it is possible to compute the long principal normal inhabitants of τ with depth $\leq d$. Thus the problem of counting principal normal inhabitants of a type τ is essentially solved by computing the long principal normal inhabitants of τ with depth $< \mathbf{D_p}(\tau)$.

In section 2 we describe the Counting Algorithm for normal principal inhabitants based on the existence of $\mathbf{d_p}(\tau)$ and $\mathbf{D_p}(\tau)$ and on Ben-Yelles' Search Algorithm. In section 3 we obtain a characterization of long normal principal inhabitants, that will give us a better insight on principal deductions for long terms in β-normal form and will thereby enable us in section 4 to establish and prove the correctness of the limits $\mathbf{d_p}(\tau)$ and $\mathbf{D_p}(\tau)$.

2 The Counting Algorithm

We use standard notation from [1] and [3]. Type-variables (atoms) are denoted by "a,b,c,..." and arbitrary types are denoted by lower-case Greek letters. It has been pointed out in [3] that it is equivalent to count typed or untyped inhabitants of a type τ. In sake of simplicity, we restrict this paper to the untyped case. A term M has a bound-variable clash iff M contains an abstractor $\underline{\lambda x}$ and a (free, bound or binding) occurrence of x that is not in its scope. Note that for any λ-term

M exists a λ-term N without bound-variable clashes and such that $M =_\alpha N$. In this paper we will only consider λ-terms without bound-variable clashes.

Definition 2.1 *A type-assignment is an expression of the form $M : \tau$, where M is a λ-term and τ is a type. The type τ is the predicate and M is the subject of the type-assignment. A type-context or basis Γ is any finite, perhaps empty, set of type-assignments with distinct variables as subjects. If $\Gamma = \{x_1, \rho_1, \ldots, x_m : \rho_m\}$ define $Subjects(\Gamma) = \{x_1, \ldots, x_m\}$. A* **TA***-formula is any expression of the form $\Gamma \vdash M : \tau$, where M is a term, Γ a type-context and τ a type.*

In the following we describe a system to assign types to λ-terms in β-nf.

Definition 2.2 *Given a λ-term M in β-nf, a type τ and a context Γ, we say that $M : \tau$ is derivable from Γ, and write $\Gamma \vdash M : \tau$ if the formula $\Gamma \vdash M : \tau$ can be produced by the following rules.*

$$(axiom)\ \frac{}{\Gamma \vdash x : \alpha} \qquad (if\ x : \alpha \in \Gamma)$$

$$(app)\ \frac{\Gamma \vdash M_1 : \alpha_1 \quad \ldots \quad \Gamma \vdash M_n : \alpha_n}{\Gamma \vdash x M_1 \ldots M_n : \beta} \qquad (if\ x : \alpha_1 \to \ldots \to \alpha_n \to \beta \in \Gamma,\ n \geq 1)$$

$$(abs)\ \frac{\Gamma, x : \alpha \vdash M : \beta}{\Gamma \vdash \lambda x.M : \alpha \to \beta}$$

A **TA***-deduction Δ of $\Gamma \vdash M : \tau$ (where M denotes a β-nf) is a tree of* **TA***-formulae, those at the tops of branches being axioms and those below being deduced from those immediately above them by a rule (app) or (abs)) and with bottom formula $\Gamma \vdash M : \tau$.*

Proposition 2.3 *Given a β-nf M, basis Γ and type τ such that $\Gamma \vdash M : \tau$, there is exactly one deduction Δ of $\Gamma \vdash M : \tau$.*

Proof Straightforward. •

Definition 2.4 *Let M be a β-nf and (Γ, τ) a pair such that $\Gamma \vdash M : \tau$. We say that M is long with respect to (Γ, τ) iff for every formula of the form $\Gamma' \vdash x M_1 \ldots M_n : \alpha$, $n \geq 0$, in the unique deduction of $\Gamma \vdash M : \tau$, α is an atom. We call M a normal inhabitant of a type τ iff M is in β-nf and $\vdash M : \tau$ and denote the set of normal inhabitants of a type τ by $\mathsf{Nhabs}(\tau)$. The set of principal normal inhabitants of a type τ is called $\mathsf{Nprinc}(\tau)$. The set of normal inhabitants of τ which are long with respect to $(\emptyset, \tau)$ is called $\mathsf{Long}(\tau)$. The set of long principal normal inhabitants is called $\mathsf{Lprinc}(\tau)$.*

Thus,

$$\mathsf{Lprinc}(\tau) = \mathsf{Long}(\tau) \cap \mathsf{Nprinc}(\tau) \subseteq \mathsf{Long}(\tau) \subseteq \mathsf{Nhabs}(\tau).$$

The (finite) set of all terms obtained by η-reducing a λ-term M is called the η-family of M and denoted by $\{M\}_\eta$. It has been shown (cf. [3]) that the η-families of the long normal inhabitants of τ partition $\mathsf{Nhabs}(\tau)$ into non-overlapping finite subsets, each η-family containing just one long member. From this and from the two following results (in [3]) we conclude, that in order to count $\mathsf{Nprinc}(\tau)$ it is sufficient to count $\mathsf{Lprinc}(\tau)$.

Lemma 2.5 (Completeness of Long(τ); Ben-Yelles 1979) *Every normal inhabitant of τ can be η-expanded to a long normal inhabitant of τ. And this long inhabitant is unique (modulo $=_\alpha$); i.e.*

$$\{M, N \in Long(\tau)\ and\ M =_\eta N\} \implies M =_\alpha N.$$

Lemma 2.6 (in Hindley'97) *Let M^+ be the unique member of Long(τ) to which M η-expands. Then,*

$$M \in \mathsf{Nprinc}(\tau) \quad \Longrightarrow \quad M^+ \in \mathsf{Nprinc}(\tau).$$

On the other hand, η-reduction preserves typing, hence $M \in \mathsf{Nprinc}(\tau)$ iff $M^+ \in \mathsf{Lprinc}(\tau)$. Thus, $\mathsf{Nprinc}(\tau) = \emptyset$ iff $\mathsf{Lprinc}(\tau) = \emptyset$ and $\mathsf{Nprinc}(\tau)$ is infinite iff $\mathsf{Lprinc}(\tau)$ is. Furthermore,

$$\mathsf{Nprinc}(\tau) = \bigcup_{M \in \mathsf{Lprinc}(\tau)} \{M\}_\eta.$$

Hence, our algorithm will focus on long normal principal inhabitants and, following Ben-Yelles' algorithm, the searching will be done in order of increasing depth of terms.

Definition 2.7 *The depth of a λ-term M in β-nf is defined as follows and denoted by $Depth(M)$.*

i. *$Depth(y) = Depth(\lambda x_1 \ldots x_m.y) = 0$;*

ii. *$Depth(\lambda x_1 \ldots x_m.yM_1 \ldots M_n) = 1 + max_{1 \leq j \leq n} Depth(M_j)$, $\quad$ if $n > 0$.*

In [2] Ben-Yelles defined an algorithm, called Search Algorithm, that given a composite type τ, i.e. non-atomic (note that atomic types have no inhabitants at all), produces a sequence $\mathcal{A}(\tau, 0), \mathcal{A}(\tau, 1), \mathcal{A}(\tau, 2), \ldots$ of finite sets of expressions, called nf-schemes, such that each member of $\mathcal{A}(\tau, d+1)$, which is a λ-term, is a closed β-nf with depth d. More precisely, one has the following, where $\mathsf{Long}(\tau, d)$ denotes the set of long normal inhabitants of τ with depth $\leq d$.

Theorem 2.8 (Search Theorem for $\mathsf{Long}(\tau)$, Ben-Yelles'79) *The Search Algorithm accepts as input any composite type τ and outputs a finite or infinite sequence of sets $\mathcal{A}(\tau, d)$ ($d = 0, 1, 2, \ldots$) such that for all $d \geq 0$,*

i. *each member of $\mathcal{A}(\tau, d)$ is a closed nf-scheme with type τ and long with respect to $(\emptyset, \tau)$, and is either*
   (a) *a proper nf-scheme with depth d, or*
   (b) *a λ-term with depth $d - 1$;*

ii. *$\mathcal{A}(\tau, d)$ is finite;*

iii. *$\mathsf{Long}(\tau, d) \subseteq \mathcal{A}(\tau, 0) \cup \ldots \cup \mathcal{A}(\tau, d + 1)$;*

iv. *if we call the set of all λ-terms in $\mathcal{A}(\tau, d)$ "$\mathcal{A}_{terms}(\tau, d)$", then*

$$\mathsf{Long}(\tau) = \bigcup_{d \geq 0} \mathcal{A}_{terms}(\tau, d).$$

Now, and analogous to the Counting Algorithm for $\mathsf{Long}(\tau)$, the Counting Algorithm for $\mathsf{Nprinc}(\tau)$ is based on the fact that $\mathsf{Lprinc}(\tau)$ is infinite iff it has some member whose depth lies between two integers $\mathbf{d_p}(\tau)$ and $\mathbf{D_p}(\tau)$, that can be computed from τ. Furthermore, if $\mathsf{Lprinc}(\tau)$ has no member with depth in $[\mathbf{d_p}(\tau); \mathbf{D_p}(\tau)[$, then $\mathsf{Lprinc}(\tau)$ is finite or empty according as the number of long principal inhabitants of τ with depth $< \mathbf{d_p}$ is finite or zero.

Definition 2.9 *The total number of occurrences of type-variables in a type τ will be denoted by $|\tau|$ and is defined as follows*

$$|a| = 1, \qquad |\rho \to \sigma| = |\rho| + |\sigma|.$$

The number of distinct type-variables occurring in τ will be denoted by $||\tau||$. Furthermore, if τ is a type let

$$\mathbf{d_p}(\tau) = |\tau| \qquad \text{and} \qquad \mathbf{D_p}(\tau) = |\tau|^4.$$

In section 4 we will prove the following.

Theorem 2.10 *For any type τ, there is* *i.* $\mathsf{Lprinc}(\tau) = \emptyset$ *iff* $\mathsf{Lprinc}(\tau)$ *has no member with depth* $< \mathbf{D_p}(\tau)$*;*

ii. $\mathsf{Lprinc}(\tau)$ *is infinite iff it has a member* M *with* $\mathbf{d_p}(\tau) \leq \mathit{depth}(M) < \mathbf{D_p}(\tau)$*;*

iii. $\mathsf{Lprinc}(\tau)$ *is finite iff all its members with depth* $< \mathbf{D_p}(\tau)$ *have depth* $< \mathbf{d_p}(\tau)$.

Thus, we have the following algorithm to count $\mathsf{Nprinc}(\tau)$.

Counting Algorithm for $\mathsf{Nprinc}(\tau)$ **2.11** *If* τ *is an atom,* $\mathsf{Nprinc}(\tau)$ *is empty. If* τ *is composite, apply the Search Algorithm to* τ *and compute* $\mathcal{A}_{terms}(\tau, d)$ *for* $d = 0, \ldots, \mathbf{D_p}(\tau)$*. Determine the set* $\mathcal{A}_p$ *of all* λ*-terms in* $\mathcal{A}_{terms}(\tau, 0) \cup \ldots \cup \mathcal{A}_{terms}(\tau, \mathbf{D_p}(\tau))$ *that are principal inhabitants of* τ *(using any algorithm for checking principal types).*

Case I *If* $\mathcal{A}_p = \emptyset$*, then* $\mathsf{Nprinc}(\tau) = \emptyset$*.*

Case II *If* $\mathcal{A}_p$ *has a member with depth* $\geq \mathbf{d_p}(\tau)$*, then* $\mathsf{Nprinc}(\tau)$ *is infinite. Apply the Search Algorithm to enumerate* $\mathcal{A}_{terms}(\tau, d)$ *for* $d = 0, 1, 2, \ldots$*, outputting for each of these sets its members which are principal inhabitants of* τ *as well as their* η*-families.*

Case III *If all members in* $\mathcal{A}_p$ *have depth* $< \mathbf{d_p}(\tau)$*, then* $\mathsf{Nprinc}(\tau)$ *is finite. Output all members of* $\mathcal{A}_p$ *as well as their* η*-families.*

3 Principal type inference for long normal λ-terms

In this section we introduce the typing system $\mathbf{TA_{pln}}$ which will give us a better insight on deductions of principal types for long normal inhabitants, and will thus enable us to prove the Shrinking and Stretching Lemmas for $\mathsf{Lprinc}(\tau)$, which have Theorem 2.10 as a consequence.

Definition 3.1 *The system* $\mathbf{TA_{pln}}$ *has an infinite set of axioms and three deduction rules as follows.*

$$\textit{(axiom)} \quad \frac{}{\Gamma \vdash^{pln} x : a \;\|\; \emptyset} \quad (\textit{if } x : a \in \Gamma)$$

$$\textit{(app)} \quad \frac{\Gamma \vdash^{pln} M_1 : \alpha_1 \;\|\; \phi_1 \quad \ldots \quad \Gamma \vdash^{pln} M_n : \alpha_n \;\|\; \phi_n}{\Gamma \vdash^{pln} x M_1 \ldots M_n : b \;\|\; \emptyset} \quad (\textit{if } x : \alpha_1 \to \ldots \to \alpha_n \to b \in \Gamma \textit{ and } n \geq 1)$$

$$\textit{(abs)} \quad \frac{\Gamma, x : \alpha \vdash^{pln} M : \beta \;\|\; \phi}{\Gamma \vdash^{pln} \lambda x.M : \alpha \to \beta \;\|\; \emptyset}$$

$$\textit{(U)} \quad \frac{\Gamma \vdash^{pln} M : a \;\|\; \emptyset}{\Gamma \vdash^{pln} M : b \;\|\; (a, b)}$$

A $\mathbf{TA_{pln}}$*-deduction* Δ *is a tree of* $\mathbf{TA_{pln}}$*-formulae, those at the tops of branches being axioms and those below being deduced from those immediately above them by a rule. The bottom formula in* Δ *is called its conclusion; if it is* $\Gamma \vdash^{pln} M : \tau \;\|\; \phi$*, we call* Δ *a deduction of* $\Gamma \vdash^{pln} M : \tau$*.*

In the following we are going to define a transformation-algorithm that given the unique **TA**-deduction Δ of a formula $\Gamma \vdash M : \tau$, where M is a β-nf which is long with respect to (Γ, τ), and a pair (Γ', τ'), obtained from (Γ, τ) by renaming occurrences of variables (note that different occurrences of variables may have been substituted by different variables), constructs a $\mathbf{TA_{pln}}$-deduction Δ' of $\Gamma' \vdash^{pln} M : \tau'$, such that several arrows occurring in Δ' are possibly marked with a $\star$. In the first step an unmarked version of Δ' will be constructed bottom-up from Δ as follows.

- if $\Gamma \vdash M : \tau$ is an axiom, i.e. $x : a \in \Gamma$, $M = x$ and $\tau = a$, then $x : a_k \in \Gamma'$ and $\tau' = a_l$. Take $\Gamma' \vdash^{pln} x : a_l \;\|\; (a_k, a_l)$ as the bottom formula of Δ' (U-rule) and precede it by $\Gamma' \vdash^{pln} x : a_k \;\|\; \emptyset$ (axiom).

- if $\Gamma \vdash M : \tau$ was obtained by the (app)-rule from $\Gamma \vdash M_1 : \alpha_1, \dots, \Gamma \vdash M_n : \alpha_n$, thus $M = xM_1 \dots M_n$ and $\tau = b$, then there is a type-assignment $x : \alpha'_1 \to \dots \to \alpha'_n \to b_k$ in Γ' and $\tau' = b_l$. Take $\Gamma' \vdash^{pln} xM_1 \dots M_n : b_l \parallel (b_k, b_l)$ as the bottom formula of Δ' (U-rule), precede it by $\Gamma' \vdash^{pln} xM_1 \dots M_n : b_k \parallel \emptyset$ (app) and precede this formula by the deductions $\Delta'_1, \dots, \Delta'_n$ constructed from the **TA**-deductions $\Delta_1, \dots, \Delta_n$ of $\Gamma \vdash M_1 : \alpha_1, \dots, \Gamma \vdash M_n : \alpha_n$ and pairs $(\Gamma', \alpha'_1), \dots, (\Gamma', \alpha'_n)$.
- if $\Gamma \vdash M : \tau$ was obtained by the (abs)-rule, i.e. $M = \lambda x.N$ and $\tau = \alpha \to \beta$, from $\Gamma, x : \alpha \vdash N : \beta$, then take $\Gamma' \vdash^{pln} \lambda x.N : \alpha' \to \beta' \parallel \emptyset$ as the bottom-rule in Δ' (abs) and precede it by the deduction Δ'_1 constructed from the deduction Δ_1 of $\Gamma, x : \alpha \vdash N : \beta$ and pair $(\Gamma' \cup \{x : \alpha'\}, \beta')$.

In the second step we mark arrows, starting top-down from axioms as follows.

- No arrows are marked for axioms.
- If $\Gamma \vdash^{pln} xM_1 \dots M_n : b \parallel \emptyset$ results from $\Gamma \vdash M_1 : \alpha_1 \parallel \phi_1, \dots, \Gamma \vdash M_n : \alpha_n \parallel \phi_n$ by the (app)-rule, then we mark $x : \alpha_1 \to \dots \to \alpha_n \to b$ in Γ as follows: $x : \alpha_1 \to^{\star} \dots \to^{\star} \alpha_n \to^{\star} b$ (no arrows are marked in $\alpha_1, \dots, \alpha_n$).
- If $\Gamma \vdash^{pln} \lambda x.M : \alpha \to \beta \parallel \emptyset$ was obtained from $\Gamma, x : \alpha \vdash^{pln} M : \beta \parallel \phi$ by the (abs)-rule, then we mark the following arrow: $\Gamma \vdash^{pln} \lambda x.M : \alpha \to^{\star} \beta \parallel \emptyset$.
- Finally, no arrows are marked in formulae obtained by the U-rule.

Definition 3.2 *The indexed counterpart* $(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})$ *of a pair* (Γ, τ) *is obtained by successively indexing all occurrences of type variables and arrows in* (Γ, τ).

Note 3.3 *If* $(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})$ *is the indexed counterpart of* (Γ, τ)*, then* $||\tau|| \le |\tau| = |\tau_{\mathtt{i}}| = ||\tau_{\mathtt{i}}||$.

Example 3.4 *The pair*

$$(\{z : ((a_1 \to_1 b_1) \to_2 c_1 \to_3 c_2) \to_4 d_1\}, d_2)$$

is the indexed counterpart of $(\{z : ((a \to b) \to c \to c) \to d\}, d)$*. The* **TA***-deduction of*

$$z : ((a \to b) \to c \to c) \to d \vdash z(\lambda x.y) : d$$

is

$z : ((a \to b) \to c \to c) \to d, x : a \to b, y : c$	$\vdash$	$y : c$
$z : ((a \to b) \to c \to c) \to d, x : a \to b$	$\vdash$	$\lambda y.y : c \to c$
$z : ((a \to b) \to c \to c) \to d$	$\vdash$	$\lambda xy.y : (a \to b) \to c \to c$
$z : ((a \to b) \to c \to c) \to d$	$\vdash$	$z(\lambda xy.y) : d$

and the corresponding $\mathbf{TA_{pln}}$*-deduction of*

$$z : ((a_1 \to_1 b_1) \to_2 c_1 \to_3 c_2) \to_4 d_1 \vdash z(\lambda x.y) : d_2$$

is

$z : ((a_1 \to_1 b_1) \to_2 c_1 \to_3 c_2) \to_4 d_1, x : a_1 \to_1 b_1, y : c_1$	$\vdash$	$y : c_1$	$\Vert \emptyset$
$z : ((a_1 \to_1 b_1) \to_2 c_1 \to_3 c_2) \to_4 d_1, x : a_1 \to_1 b_1, y : c_1$	$\vdash$	$y : c_2$	$\Vert (c_1, c_2)$
$z : ((a_1 \to_1 b_1) \to_2 c_1 \to_3 c_2) \to_4 d_1, x : a_1 \to_1 b_1$	$\vdash$	$\lambda y.y : c_1 \to_3^{\star} c_2$	$\Vert \emptyset$
$z : ((a_1 \to_1 b_1) \to_2 c_1 \to_3 c_2) \to_4 d_1$	$\vdash$	$\lambda xy.y : (a_1 \to_1 b_1) \to_2^{\star} c_1 \to_3 c_2$	$\Vert \emptyset$
$z : ((a_1 \to_1 b_1) \to_2 c_1 \to_3 c_2) \to_4^{\star} d_1$	$\vdash$	$z(\lambda xy.y) : d_1$	$\Vert \emptyset$
$z : ((a_1 \to_1 b_1) \to_2 c_1 \to_3 c_2) \to_4 d_1$	$\vdash$	$z(\lambda xy.y) : d_2$	$\Vert (d_1, d_2)$

Definition 3.5 *Given a type* τ *and a binary relation* Φ *defined over the set of type-variables in* τ*, let* $\mathcal{C}_{\Phi}$ *be the set of equivalence classes of the reflexive, symmetric and transitive closure of* Φ.

Lemma 3.6 *Consider any finite non-empty set A and n binary relations over A*

$$\Phi_1 \subseteq \ldots \subseteq \Phi_n$$

such that $\mathcal{C}_{\Phi_i} \neq \mathcal{C}_{\Phi_j}$ for $1 \leq i \neq j \leq n$. Then $n \leq \#A$.

Proof If $\Phi_i \subseteq \Phi_j$, then for every $X \in \mathcal{C}_{\Phi_i}$ there is $Y \in \mathcal{C}_{\Phi_j}$ such that $X \subseteq Y$. On the other hand, $\mathcal{C}_{\Phi_i} \neq \mathcal{C}_{\Phi_j}$ implies that there are $X \in \mathcal{C}_{\Phi_i}$ and $Y \in \mathcal{C}_{\Phi_j}$ with $X \subseteq Y$ but $X \neq Y$. Thus, there is at least another $X' \in \mathcal{C}_{\Phi_i}$, $X' \cup X = \emptyset$, such that $X' \subseteq Y$ and consequently $\mathcal{C}_{\Phi_j}$ has less elements than $\mathcal{C}_{\Phi_i}$. But the maximum of equivalence classes in $\mathcal{C}_{\Phi_1}$ is $\#A$ and the minimum of classes in $\mathcal{C}_{\Phi_n}$ is n. •

The following result is a consequence of observing that the algorithm for computing the principal pair of a λ-term only introduces arrows required by the typing rules and only unifies two variables if this is absolutely required by the term structure. Note that the binary relation Φ corresponds directly to the connection relation for TA-figures in [4].

Proposition 3.7 *Let M be a λ-term in normal form, Γ a type-context and τ a type such that, $Subjects(\Gamma) = FV(M)$, $\Gamma \vdash M : \tau$ and M is long with respect to (Γ, τ). Consider the indexed counterpart $(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})$ of (Γ, τ). Let $\Delta_{(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})}$ be the* **TA**$_{\mathbf{pln}}$*-deduction of $\Gamma_{\mathtt{i}} \vdash^{pln} M : \tau_{\mathtt{i}}$ constructed from the unique* **TA***-deduction of $\Gamma \vdash M : \tau$ and let Φ be the set of all binary pairs in $\Delta_{(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})}$. Then,*

(Γ, τ) is a principal pair for M
iff
for each type-variable a in (Γ, τ) there is one equivalence class
in $\mathcal{C}_\Phi$ containing exactly all indexed occurrences of a in $(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})$
and all indexed arrows have a marked occurrence in $\Delta_{(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})}$.

Proof It is straightforward to show that whenever $(a, b) \in \Phi$, then a and b are indexed versions of the same type-variable. Now, suppose that (Γ, τ) is no principal pair for M, i.e. there exists a (more general) pair (Γ_0, τ_0) such that $\Gamma_0 \vdash M : \tau_0$, $\Gamma_0^\star = \Gamma$ and $\tau_0^\star = \tau$ for some substitution $\star$, but (Γ, τ) is no variant of (Γ_0, τ_0). If (Γ_0, τ_0) has the same structure as (Γ, τ), but at least two type-variables in (Γ_0, τ_0) are given the same name, say a, in (Γ, τ), then, considering the observation made at the beginning of the proof, it is easy to see that there are at least two equivalence classes in $\mathcal{C}_\Phi$ containing indexed occurrences of a in $(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})$. If, on the other hand, one variable in (Γ_0, τ_0) has a composite type $\alpha \to \beta$ in (Γ, τ), then the corresponding occurrences of the arrow will never be marked in $\Delta_{(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})}$. This follows from the fact that the **TA**-deduction of $\Gamma \vdash M : \tau$ is a copy of the **TA**-deduction of $\Gamma_0 \vdash M : \tau_0$ where all type-variables a are substituted by $\star(a)$. Analysing the transformation algorithm, one sees that the same holds for the corresponding **TA**$_{\mathbf{pln}}$-deductions.

Conversely, suppose that M is a β-nf, Γ a type-context and τ a type such that $\Gamma \vdash M : \tau$, M is long with respect to (Γ, τ) and $FV(M) \subseteq Subjects(\Gamma)$. Furthermore, consider $(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})$, $\Delta_{(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})}$ and Φ as before. Let $\Phi' \supseteq \Phi$ be any binary relation over the variables in $(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})$ such that there is some type-variable, say d, such that there are at least two different equivalence classes $\{d_{i_1}, \ldots, d_{i_m}\}$ and $\{d_{j_1}, \ldots, d_{j_n}\}$ in $\mathcal{C}_{\Phi'}$ containing indexed occurrences of d. We prove by induction on the structure of M that, if we substitute in $\Delta_{(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})}$ the occurrences $d_{j_1}, \ldots, d_{j_n}$ by $c_{j_1}, \ldots, c_{j_n}$, for some new type-variable c, then we obtain a new **TA**$_{\mathbf{pln}}$-deduction Δ'_c that corresponds to the **TA**-deduction Δ_c of $\Gamma_c \vdash M : \tau_c$, where (Γ_c, τ_c) is obtained from (Γ, τ) by substitution of the occurrences of d corresponding to $d_{j_1}, \ldots, d_{j_n}$ by c. Then (Γ_c, τ_c) is a more general pair such that $\Gamma_c \vdash M : \tau_c$ and consequently (Γ, τ) is no principal pair for M.

If $M = x$, then $(\Gamma, \tau) = (\{x : a\}, a)$ and the result holds vacuously, since $\mathcal{C}_\Phi = \{\{a_1, a_2\}\}$.

If $M = xM_1 \ldots M_n$, then $\tau = b$, $x : \alpha_1 \to \ldots \alpha_n \to b \in \Gamma$ and $\Gamma \vdash xM_1 \ldots M_n : b$ results from $\Gamma \vdash M_1 : \alpha_1, \ldots, \Gamma \vdash M_n : \alpha_n$ by the $(\to E)$-rule. From the induction hypothesis we conclude that $\Gamma_c \vdash M_1 : \alpha_1^c, \ldots, \Gamma_c \vdash M_n : \alpha_n^c$. If $\Delta'_{c,1}, \ldots, \Delta'_{c,n}$ are the corresponding **TA**$_{\mathbf{pln}}$-deductions, then Δ'_c has the form

$$\cfrac{\cfrac{\Delta'_{c,1} \ldots \Delta'_{c,n}}{\Gamma_c \vdash^{pln} xM_1 \ldots M_n : b_k^c \parallel \emptyset}}{\Gamma_c \vdash^{pln} xM_1 \ldots M_n : b_l^c \parallel (b_k^c, b_l^c).}$$

Thus $(b_k, b_l) \in \Phi \subseteq \Phi'$ belong to the same class in $\mathcal{C}_\Phi$ and the corresponding occurrences in (Γ^c, τ^c) are occurrences of the same type-variable. Hence, $\Gamma^c \vdash xM_1 \dots M_n : b^c$ can be inferred from $\Gamma_c \vdash M_1 : \alpha_1^c, \dots, \Gamma_c \vdash M_n : \alpha_n^c$ by the (app)-rule and Δ_c' corresponds to the **TA**-deduction Δ_c of $\Gamma_c \vdash xM_1 \dots M_n : b^c$.

The result is straightforward for $M = \lambda x.N$.

Thus we showed, that whenever (Γ, τ) is a principal pair for M then for each type-variable a in (Γ, τ) there is one equivalence class in $\mathcal{C}_\Phi$ containing exactly all indexed occurrences of a in $(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})$. It remains to show that all (indexed) arrows have a marked occurrence in $\Delta_{(\Gamma_{\mathtt{i}}, \tau_{\mathtt{i}})}$. We proceed by induction on the structure of M.

If $M = x$, then the result holds vacuously.

For $M = xM_1 \dots M_n$ let (Γ, b) be a principal pair of $xM_1 \dots M_n$ obtained from $\Gamma \vdash M_1 : \alpha_1, \dots, \Gamma \vdash M_n : \alpha_n$ by the (app)-rule and $x : \alpha_1 \to \dots \to \alpha_n \to b \in \Gamma$. Then there are $(\Gamma_1, \alpha_1^0), \dots, (\Gamma_n, \alpha_n^0)$ respectively principal pairs of $M_1, \dots, M_n$, such that $\Gamma = \Gamma_1^\star \cup \dots \cup \Gamma_n^\star \cup \{x : v_1 \to \dots \to v_n \to v\}^\star$, where $\star$ is the substitution resulting from the unification of the types assigned to variables in $\Gamma_1, \dots, \Gamma_n$ as well as in $\{x : v_1 \to \dots \to v_n \to v\}$ ($v_1, \dots, v_n, v$ are new type-variables) and such that $\alpha_1 = \star(v_1) = \star(\alpha_1^0), \dots, \alpha_n = \star(v_n) = \star(\alpha_n^0)$ and $b = \star(v)$. Now, by the induction hypothesis, all occurrences of arrows in $\Gamma_1, \dots, \Gamma_n, \alpha_1^0, \dots, \alpha_n^0$ are marked in the $\mathbf{TA_{pln}}$-deductions corresponding to the **TA**-deductions of $\Gamma_1 \vdash M_1 : \alpha_1, \dots, \Gamma_n \vdash M_n : \alpha_n$. Thus, the arrows resulting from the unification and corresponding to arrows in $\Gamma_1, \dots, \Gamma_n, \alpha_1^0, \dots, \alpha_n^0$ will also be marked in the $\mathbf{TA_{pln}}$-deduction that corresponds to the **TA**-deduction of $\Gamma \vdash xM_1 \dots M_n : b$. Finally, note that the n main arrows in $\alpha_1 \to \dots \to \alpha_n \to b$ will be marked too.

The case $M = \lambda x.N$ is trivial, since if $\Gamma \vdash \lambda x.N : \alpha \to \beta$ results from $\Gamma, x : \alpha \vdash N : \beta$ by the (abs)-rule and $(\Gamma, \alpha \to \beta)$ is a principal pair for $\lambda x.N$, then $(\Gamma \cup \{x : \alpha\}, \beta)$ is a principal pair for N and the result follows almost directly from the induction hypothesis. •

Example 3.8 *Although* $\mathcal{C}_\Phi = \{\{a_1\}, \{b_1\}, \{c_1, c_2, \}, \{d_1, d_2\}\}$ *we conclude that* $(\{z : ((a \to b) \to c \to c) \to d\}, d)$ *is no principal pair for* $z(\lambda x.y)$*, since* $\to_1$ *has no marked occurrence in the* $\mathbf{TA_{pln}}$*-deduction of* $z : ((a_1 \to_1 b_1) \to_2 c_1 \to_3 c_2) \to_4 d_1 \vdash z(\lambda x.y) : d_2$.

4 Correctness

We begin this section with several definitions and results (mostly from [3]) on the structure of types and terms, that we will need later on in order to prove Theorem 2.10. Note that every type τ can be written uniquely in the form

$$\tau_1 \to \dots \to \tau_m \to e,$$

where e is an atom and $m \geq 0$. Iff $m \geq 1$ we call τ a composite type.

Definition 4.1 *The significant subtypes or s-subtypes of a type* $\tau = \tau_1 \to \dots \to \tau_m \to e$*, where* e *is an atom and* $m \geq 0$*, are defined recursively as follows.*

- τ *is an s-subtype of* τ*;*
- *every s-subtype of one of* $\tau_1, \dots, \tau_m, e$ *is an s-subtype of* τ*.*

A proper s-subtype of τ *is an s-subtype* $\neq \tau$*. Particular occurrences of s-subtypes of* τ *are also called s-components of* τ *and are distinguished by underlining their names. An s-component of a type* τ *is defined to be positive or negative as follows.*

- $\underline{\tau}$ *is a positive s-component of* τ*;*
- *if* $\tau = \tau_1 \to \dots \to \tau_m \to e$*, then* $\underline{\tau}_1, \dots, \underline{\tau}_m$ *are negative s-components of* τ *and* $\underline{e}$ *is a positive s-component of* τ*;*
- *if* $\tau = \tau_1 \to \dots \to \tau_m \to e$ *and if* $\underline{\rho}$ *is an s-component of one of* $\tau_1, \dots, \tau_m$*, then* $\underline{\rho}$ *is a positive or negative s-component of* τ *according as it is a negative or positive s-component of* $\tau_1, \tau_2, \dots$ *or* τ_m*.*

Definition 4.2 *If $\underline{\rho}$ is a composite s-component of a type τ and $\rho = \rho_1 \to \ldots \to \rho_n \to a$ $(n \geq 1)$, the s-components $\underline{\rho}_1, \ldots, \underline{\rho}_n$ are called the premises of $\underline{\rho}$ and $\underline{a}$ is called the conclusion or tail-component of $\underline{\rho}$.*

An s-component of τ is called a subpremise or subtail of τ according as it is a premise or tail of another s-component of τ.

Definition 4.3 *If τ is composite, $\mathsf{NSS}(\tau)$ is the set of all finite sequences $< \sigma_1, \ldots, \sigma_n >$ $(n \geq 1)$ such that τ contains a positive composite s-component with form $\sigma_1 \to \ldots \to \sigma_n \to a$ for some atom a. Each member of $\mathsf{NSS}(\tau)$ is called a negative subpremise-sequence.*

Every non-atomic λ-term X can be expressed uniquely in the form

$$X = \lambda x_1 \ldots x_m.vY_1 \ldots Y_n, \qquad (m+n \geq 1).$$

The head and arguments of X are respectively $\underline{v}$ and $\underline{Y}_1, \ldots, \underline{Y}_n$.

Definition 4.4 *A subargument of a λ-term X is a component that is an argument of X or an argument of a proper component of X. If $\underline{Z}$ is a subargument of a λ-term X, the argument-branch from $\underline{X}$ to $\underline{Z}$ is the sequence $< \underline{Z}_0, \underline{Z}_1, \ldots, \underline{Z}_k >$, $(k \geq 1)$, such that $\underline{Z}_0 = \underline{X}$ and $\underline{Z}_i$ is an argument of $\underline{Z}_{i-1}$ for $i = 1, \ldots, k$, and $\underline{Z}_k = \underline{Z}$. It is called unextendable iff $\underline{Z}$ is an atom or abstracted atom. Its length is k (not $k+1$).*

Definition 4.5 *Let Δ be a $\mathbf{TA_{pln}}$-deduction of $\Gamma \vdash^{pln} M : \tau$, let $\underline{Z}$ be a subargument of M; say*

$$Z = \lambda x_1 \ldots x_m.yZ_1 \ldots Z_n \qquad (m, n \geq 0)$$

and let $\Gamma_Z \vdash^{pln} Z : \alpha_1 \to \ldots \alpha_m \to a \parallel \emptyset$ be the node in Δ which corresponds to $\underline{Z}$. The Initial Abstractors' Types sequence $\mathsf{IAT}(\underline{Z})$ is defined to be

$$\mathsf{IAT}(\underline{Z}) = < \alpha_1, \ldots, \alpha_m >$$

and has length m. The Initial Abstractors sequence $\mathsf{IA}(Z)$ is the (possibly empty) sequence

$$\mathsf{IA}(Z) = < x_1, \ldots, x_m > .$$

Lemma 4.6 *Let Δ be a deduction of $\vdash M : \tau$, where M is a long β-nf with respect to $(\emptyset, \tau)$ and let Δ' be the corresponding $\mathbf{TA_{pln}}$-deduction of $\vdash^{pln} M : \tau_{\mathtt{i}}$. Let $\underline{Z}$ be a subargument of M, and let $\Gamma_Z \vdash^{pln} Z : \sigma \parallel \emptyset$, with $\sigma = \sigma_1 \to \ldots \to \sigma_k \to s$, be the node in Δ' which corresponds to $\underline{Z}$. Then*

i. if σ is an atom, $\mathsf{IAT}(\underline{Z}) = \emptyset$;

ii. if σ is composite, $\mathsf{IAT}(\underline{Z}) \in \mathsf{NSS}(\tau_{\mathtt{i}})$.

Proof Part *i.* is trivial. For part *ii.*, we show by induction on the depth of $\underline{Z}$ in M, i.e. the length of the argument-branch from M to $\underline{Z}$, that whenever $x : \alpha \in \Gamma_Z$, then α occurs as a negative s-component in $\tau_{\mathtt{i}}$ and σ (composite) occurs as a positive s-component of τ. In fact, suppose that $\underline{Z}$ has depth 1 in M, i.e. $M = \lambda x_1 \ldots x_m.vY_1 \ldots Y_n$ with $m, n \geq 1$ and $\underline{Z} = Y_i$ for some $1 \leq i \leq n$. Then $\tau_{\mathtt{i}} = \alpha_1 \to \ldots \to \alpha_m \to a$ and the node in Δ' that corresponds to $\underline{Z}$ is $x_1 : \alpha_1, \ldots, x_m : \alpha_m \vdash^{pln} Z : \sigma \parallel \emptyset$ (note that σ is composite, so this node results from the (abs)-rule). By definition, $\alpha_1, \ldots, \alpha_m$ are negative s-components of $\tau_{\mathtt{i}}$. On the other hand, there is $v = x_j$ for some $1 \leq j \leq m$ and α_j is of the form $\beta_1 \to \ldots \to \beta_n \to b$ with $\beta_i = \sigma$. Thus, σ occurs as a negative s-component of a negative s-component of $\tau_{\mathtt{i}}$ and occurs consequently as a positive s-component of $\tau_{\mathtt{i}}$.

The induction step is mostly a repetition of the previous argument.

Thus, if σ is composite, then σ occurs as a positive s-component of $\tau_{\mathtt{i}}$ and by the definition of $\mathsf{NSS}(\tau_{\mathtt{i}})$ follows $\mathsf{IAT}(\underline{Z}) = < \sigma_1, \ldots, \sigma_k > \in \mathsf{NSS}(\tau_{\mathtt{i}})$. •

It has been shown in [3] that whenever τ is a composite type, then $\#(\mathsf{NSS}(\tau)) \leq |\tau| - 1$. Thus, if $(\emptyset, \tau_{\mathtt{i}})$ is the indexed counterpart of $(\emptyset, \tau)$, one has $\#(\mathsf{NSS}(\tau_{\mathtt{i}})) \leq |\tau_{\mathtt{i}}| - 1 = |\tau| - 1$.

Lemma 4.7 *If τ is composite and $(\emptyset, \tau_{\mathtt{i}})$ is the indexed counterpart of $(\emptyset, \tau)$, then*

$$\#(\mathsf{NSS}(\tau_{\mathtt{i}})) \leq |\tau| - 1. \quad \bullet$$

The proofs of the following two lemmas follow closely the schemes of Ben-Yelles' proofs of corresponding results for $\mathsf{Long}(\tau)$, from which they differ essentially in the justification of the encountered limits $\mathbf{d_p}(\tau)$ and $\mathbf{D_p}(\tau)$. As in the original case ($\mathsf{Long}(\tau)$), the construction of a term with smaller depth (or greater in the case of the Stretching Lemma) is done by substitution of a subterm by another subterm with smaller (greater) depth. But in the case of principal inhabitants one has to be more careful choosing these subterms, leading thus to greater limits, in order to guarantee the preservation of principality.

Lemma 4.8 (Shrinking Lemma) *If $\mathsf{Lprinc}(\tau)$ has a member M with depth $\geq \mathbf{D_p}(\tau)$, then*

i. there exists $M^\star \in \mathsf{Lprinc}(\tau)$ with $Depth(M) - |\tau|^3 \leq Depth(M^\star) < Depth(M)$;

ii. there exists $N \in \mathsf{Lprinc}(\tau)$ with $\mathbf{D_p}(\tau) - |\tau|^3 \leq Depth(N) < \mathbf{D_p}(\tau)$.

Proof For part *i.* consider a λ-term $M \in \mathsf{Lprinc}(\tau)$ with depth $d \geq \mathbf{D_p}(\tau)$ and without bound-variable clashes. Let $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ be the $\mathbf{TA_{pln}}$-deduction of $\vdash^{pln} M : \tau_{\mathtt{i}}$ constructed from the unique $\mathbf{TA}$-deduction of $\vdash M : \tau$ and let Φ be the set of all binary pairs in $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$. It follows from Proposition 3.7 that for each type-variable a in τ there is one equivalence class in $\mathcal{C}_\Phi$ containing exactly all indexed occurrences of a in $\tau_{\mathtt{i}}$ and that all (indexed) arrows in τ_i are marked in $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$. In the following we are going to construct a term of depth $< Depth(M)$ which has $(\emptyset, \tau)$ as a principal pair. It has been shown in [3] that $d = Depth(M)$ is the maximum of the lengths of all the argument-branches in X. Thus M has at least one argument-branch with length d and in order to reduce the depth of M it is necessary to shrink all these branches. Let $< N_0, \ldots, N_d >$ be any such branch, with

$$N_i = \lambda x_{i,1} \ldots x_{i,m_i}.y_i P_{i,1} \ldots P_{i,n_i} \qquad (m_i, n_i \geq 0).$$

Let

$$\Gamma_i \vdash^{pln} N_i : \rho_{i,1} \to \ldots \to \rho_{i,m_i} \to a_i \parallel \emptyset$$

be the node in $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ that corresponds to $\underline{N}_i$, for $i = 0, \ldots, d$. Thus, $\mathsf{IAT}(\underline{N}_i) =< \rho_{i,1}, \ldots, \rho_{i,m_i} >$. For $i = 0, \ldots, d$ let $\underline{B}_i$ be the body of $\underline{N}_i$, i.e. $B_i = y_i P_{i,1} \ldots P_{i,n_i}$ and let $\Gamma_{B_i} \vdash^{pln} B_i : a_{b_i} \parallel \emptyset$ be the node in $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ that corresponds to $\underline{B}_i$. Furthermore, let Φ_i and $\mathcal{I}_i$ be respectively the set of binary pairs and of indexes of marked arrows in the subtree of $\Delta_{(\emptyset,\tau_{\mathtt{i}})}$ with bottom formula $\Gamma_{B_i} \vdash^{pln} B_i : a_{b_i} \parallel \emptyset$. Then, $\mathcal{I}_i \subseteq \mathcal{I}_{i-1}$ and $\Phi_i \subseteq \Phi_{i-1}$ for $i = 1, \ldots, d$.

As in [3] we define a sequence of integers $d_0, d_1, \ldots, d_n$ as follows: $d_0 = 0$ and d_{j+1} is the least $i > d_j$ such that $\mathsf{IAT}(\underline{N}_i)$ differs from all $\mathsf{IAT}(\underline{N}_{d_0}), \mathsf{IAT}(\underline{N}_{d_1}), \ldots, \mathsf{IAT}(\underline{N}_{d_j})$. Obviously, one has $n \leq d$ and $0 = d_0 < d_1 < \ldots < d_n \leq d$. Furthermore, for $0 \leq i \leq d$, $\mathsf{IAT}(\underline{N}_i)$ is identical to one of the $n + 1$

$$\mathsf{IAT}(\underline{N}_{d_0}), \ldots, \mathsf{IAT}(\underline{N}_{d_n}),$$

which are all distinct and by 4.6 are either empty or members of $\mathsf{NSS}(\tau_{\mathtt{i}})$. Hence, by lemma 4.7

$$n + 1 \leq 1 + \#\mathsf{NSS}(\tau_{\mathtt{i}}) \leq 1 + |\tau| - 1 = |\tau|.$$

For $i = 0, \ldots, n$ define the following non-empty sets, called IAT-intervals, as follows:

$$\begin{array}{lcll} \mathsf{I}_j & = & \{d_j, d_j + 1, \ldots, d_{j+1} - 1\} & \quad 0 \leq j \leq n - 1 \\ \mathsf{I}_n & = & \{d_n, d_n + 1, \ldots, d\}. & \end{array}$$

If I_j contains two numbers p, $p + r$ such that $r \geq 1$ and B_p and B_{p+r} have the same type (i.e. $a_{b_p} = a_{b_{p+r}}$), $\mathcal{C}_{\Phi_p} = \mathcal{C}_{\Phi_{p+r}}$ and $\mathcal{I}_p = \mathcal{I}_{p+r}$, we shall call $< p, p + r >$ a tail-repetition. It will be called minimal iff there is no other tail-repetition $< p', q' >$ with $p \leq p' < q' \leq p + r$. It follows that each IAT-interval I_j without a tail-repetition must have $\leq |\tau|^3$ members as well as $r \leq |\tau|^3$. In fact, there are $|\tau|$ distinct atoms in $\tau_{\mathtt{i}}$, there are at most $|\tau|$ distinct equivalence classes corresponding to $\Phi_0 \supseteq \Phi_1 \supseteq \ldots \supseteq \Phi_d$ (cf. Lemma 3.6), as well as at most $|\tau|$ distinct sets of indexes among $\mathcal{I}_0 \supseteq \ldots \supseteq \mathcal{I}_d$ (note that there are exactly $|\tau|$ arrows in $\tau_{\mathtt{i}}$). Thus, if none of the $n+1$ IAT-intervals contained a tail-repetition, then the branch would have $\leq |\tau|^4$ members. But the branch has $d+1$ members and

$$d+1 = Depth(M)+1 \geq \mathbf{D_p}(\tau)+1 > |\tau|^4.$$

Hence at least one IAT-interval contains a tail-repetition.

Now let I_j be the last interval containing a minimal tail-repetition, say $< p, p+r >$. Suppose that v is a variable that occurs free in B_{p+r} with type-assignment $v : \alpha \in \Gamma_{B_{p+r}}$. Since M is closed we conclude that $v \in \mathsf{IA}(\underline{N}_0) \cup \ldots \cup \mathsf{IA}(\underline{N}_{p+r})$. Furthermore, by the definition of IAT-intervals, $\alpha \in \mathsf{IAT}(\underline{N}_{d_q})$ for some $q \leq j \leq p < p+r$. Hence, there is some type-assignment $v' : \alpha \in \Gamma_{B_p}$.

Now let B'_{p+r} be a term obtained from B_{p+r} by substituting all free variables v in B_{p+r} by some variable v' as above (possibly itself if $v : \alpha \in \Gamma_{B_p}$). Finally let M' be obtained from M by replacing B_p by B'_{p+r}. Note that $\Gamma_{B_p} \subseteq \Gamma_{B_{p+r}}$, i.e. $\Gamma_{B_{p+r}} = \Gamma_{B_p} \cup \Gamma_r$ for some Γ_r with $\Gamma_{B_p} \cap \Gamma_r = \emptyset$ and such that no variable in $Subjects(\Gamma_r)$ occurs in B'_{p+r}. Consider the subtree Δ_{p+r} of $\Delta_{(\emptyset,\tau_\mathtt{i})}$ with bottom formula $\Gamma_{B_{p+r}} \vdash^{pln} B_{p+r} : a_{b_{p+r}} \parallel \emptyset$. Let Δ'_{p+r} be obtained from Δ_{p+r} by substituting every term variable v by the corresponding $v' \in Subjects(\Gamma_{B_p})$ and erasing in the bases of the formulae all type-assignments for variables in $Subjects(\Gamma_r)$. Then Δ'_{p+r} has bottom formula $\Gamma_{B_p} \vdash^{pln} B'_{p+r} : a_{b_{p+r}} \parallel \emptyset$. Now consider the tree obtained from $\Delta_{(\emptyset,\tau_\mathtt{i})}$ by substituting its subtree with bottom formula $\Gamma_{B_p} \vdash^{pln} B_p : a_{b_p} \parallel \emptyset$ by Δ'_{p+r} and by substituting B_p in all nodes below by B'_{p+r}. Then, it is straightforward to prove, that the resulting tree Δ' is a $\mathbf{TA_{pln}}$-deduction of $\vdash^{pln} M' : \tau_\mathtt{i}$ which corresponds to a $\mathbf{TA}$-deduction of $\vdash M' : \tau$ and such that for the set Φ' of binary pairs in Δ' one has $\mathcal{C}_{\Phi'} = \mathcal{C}_\Phi$ and all arrows marked in $\Delta_{(\emptyset,\tau_\mathtt{i})}$ are also marked in Δ'. Thus, we conclude that M' is a principal long inhabitant of τ. On the other hand, in a branch in M $r \leq |\tau|^3$ arguments have been removed. Thus $d - |\tau|^3 \leq Depth(M') \leq d$. If $Depth(M') < d$ let $M^\star = M'$. Otherwise, repeat shortening branches of length d until there are none left and define $M^\star$ to be the first term produced by this procedure whose depth is less than d. Then $d - |\tau|^3 \leq Depth(M^\star) \leq d$. For part *ii.* it is sufficient to repeat *i.* and take the first output with depth $< \mathbf{D_p}(\tau)$. •

Lemma 4.9 (Stretching Lemma) *If* $\mathsf{Lprinc}(\tau)$ *has a member* M *with depth* $\mathbf{d_p}(\tau)$*, then*

i. there exists $M^\star \in \mathsf{Lprinc}(\tau)$ *with* $Depth(M^\star) \geq Depth(M)+1$*;*

ii. $\mathsf{Lprinc}(\tau)$ *is infinite.*

Proof Part *ii.* follows from *i.* by repetition. The construction of $M^\star$ in *i.* is identical to the one in [3] for the Stretching Lemma for $\mathsf{Long}(\tau)$ (not $\mathsf{Lprinc}(\tau)$). Choose any argument branch $< N_0, \ldots, N_d >$ of length $d = Depth(M) \geq |\tau|$ and, as in the proof of the Shrinking Lemma, let $\underline{B}_i$ be the body of $\underline{N}_i$ for $i = 0, \ldots, d$. Let $\Gamma_{B_i} \vdash B_i : a_{b_i} \parallel \emptyset$ be the corresponding node in $\Delta_{(\emptyset,\tau_\mathtt{i})}$. Since $d+1 > |\tau| = ||\tau_\mathtt{i}||$, there are $0 \leq p < p+r \leq d$ such that $\underline{B}_p$ and $\underline{B}_{p+r}$ have the same type $a = a_{b_p} = a_{b_{p+r}}$. Define $M^\star$ to be the result of replacing $\underline{B}_{p+r}$ in M by a copy of $\underline{B}_p$, after (to avoid clashes) changing the names of the variables $x_1, \ldots, x_m$ in this copy to $x'_1, \ldots, x'_m$, where $\{x_1 : \alpha_1, \ldots, x_m : \alpha_m\} = Subjects(\Gamma_{B_p} \setminus \Gamma_{B_{p+r}})$. Then $M^\star$ has an argument branch with length $d+r > d$. In order to see that τ is a principal type of $M^\star$, consider the tree Δ'_{p+r} obtained from the subtree of $\Delta_{(\emptyset,\tau_\mathtt{i})}$ with bottom formula $\Gamma_{B_p} \vdash B_p : a \parallel \emptyset$ by first replacing in each node all occurrences of $x_1, \ldots, x_m$ respectively by $x'_1, \ldots, x'_m$ and then adding $\{x_1 : \alpha_1, \ldots, x_m : \alpha_m\}$ to the bases in all nodes. Note that Δ'_{p+r} has bottom formula $\Gamma_{B_{p+r}} \vdash B'_{p+r} : a \parallel \emptyset$. Finally, let Δ' be the tree obtained from $\Delta_{(\emptyset,\tau_\mathtt{i})}$ by replacing its subtree with bottom formula $\Gamma_{B_{p+r}} \vdash B_{p+r} : a \parallel \emptyset$ by Δ'_{p+r}. It is straightforward to see that Δ' is the $\mathbf{TA_{pln}}$-deduction corresponding to the $\mathbf{TA}$-deduction of $\vdash M^\star : \tau$ and that all binary pairs and marked arrows in $\Delta_{(\emptyset,\tau_\mathtt{i})}$ occur in Δ'. Thus we conclude that $M^\star \in \mathsf{Lprinc}(\tau)$. •

Whenever τ is a composite type, there is $|\tau| \geq 2$, thus

$$\mathbf{d_p}(\tau) = |\tau| < |\tau|^4 - |\tau|^3 = \mathbf{D_p}(\tau) - |\tau|^3 < \mathbf{D_p}(\tau).$$

Hence, Theorem 2.10 follows as a Corollary.

Acknowledgments The work presented in this paper has been partially supported by funds granted to *LIACC* through *Programa do Financiamento Plurianual, Fundação para a Ciência e Tecnologia* and *Programa PRAXIS*.

References

[1] H. Barendregt. Lambda calculi with types. In Abramsky, Gabbay, and Maibaum, editors, *Background: Computational Structures*, volume 2 of *Handbook of Logic in Computer Science*, pages 117–309. Oxford Science Publications, 1992.

[2] C.-B. Ben-Yelles. *Type-assignment in the lambda-calculus; syntax and semantics.* PhD thesis, Mathematics Dept., University of Wales Swansea, UK, 1979.

[3] J. R. Hindley. *Basic Simple Type Theory.* Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 1997.

[4] S. Hirokawa. Principal types of BCK-lambda-terms. *Theoretical Computer Science*, 107:253–276, 1993.

[5] M. Takahashi, Y. Akama, and S. Hirokawa. Normal proofs and their grammar. In M. Hagiya and J. Mitchell, editors, *Theoretical Aspects of Computer Software (TACS'94)*, volume 789 of *LNCS*, pages 465–493. Springer Verlag, 1994.

[6] M. Zaionc. The set of unifiers in typed λ-calculus as regular expression. In J.-P. Jouannaud, editor, *Rewriting Techniques and Applications*, volume 202 of *LNCS*, pages 430–440. Springer Verlag, 1985.

[7] M. Zaionc. Mechanical procedure for proof construction via closed terms in typed λ-calculus. *J. Automated Reasoning*, 4:173–190, 1988.