Almost Every Simply Typed λ-Term Has a Long β-Reduction Sequence

Ryoma Sin'ya(✉), Kazuyuki Asada, Naoki Kobayashi, and Takeshi Tsukada

The University of Tokyo, Tokyo, Japan
ryoma@kb.is.s.u-tokyo.ac.jp

Abstract. It is well known that the length of a β-reduction sequence of a simply typed λ-term of order k can be huge; it is as large as k-fold exponential in the size of the λ-term in the *worst* case. We consider the following relevant question about *quantitative* properties, instead of the worst case: *how many* simply typed λ-terms have very long reduction sequences? We provide a partial answer to this question, by showing that asymptotically almost every simply typed λ-term of order k has a reduction sequence as long as $(k-2)$-fold exponential in the term size, under the assumption that the arity of functions and the number of variables that may occur in every subterm are bounded above by a constant. The work has been motivated by quantitative analysis of the complexity of higher-order model checking.

1 Introduction

It is well known that the length of a β-reduction sequence of a simply typed λ-term can be extremely long. Beckmann [1] showed that, for any $k \geq 0$,

$$\max\{\beta(t) \mid t \text{ is a simply typed } \lambda\text{-term of order } k \text{ and size } n\} = \mathbf{exp}_k(\Theta(n))$$

where $\beta(t)$ is the maximum length of the β-reduction sequences of the term t, and $\mathbf{exp}_k(x)$ is defined by: $\mathbf{exp}_0(x) \triangleq x$ and $\mathbf{exp}_{k+1}(x) \triangleq 2^{\mathbf{exp}_k(x)}$. Indeed, the following order-k term [1]:

$$(Twice_k)^n\, Twice_{k-1} \cdots Twice_2(\lambda x.a\, x\, x)c,$$

where $Twice_j$ is the twice function $\lambda f^{\sigma_{j-1}}.\lambda x^{\sigma_{j-2}}.f(f\, x)$ (with σ_j being the order-j type defined by: $\sigma_0 = \mathsf{o}$ and $\sigma_j = \sigma_{j-1} \to \sigma_{j-1}$), has a β-reduction sequence of length $\mathbf{exp}_k(\Omega(n))$.

Although the worst-case length of the longest β-reduction sequence is well known as above, much is not known about the *average-case* length of the longest β-reduction sequence: *how often* does one encounter a term having a very long β-reduction sequence? In other words, suppose we pick a simply-typed λ-term t of order k and size n *randomly*; then what is the probability that t has a β-reduction sequence longer than a certain bound, like $\mathbf{exp}_k(cn)$ (where c is some constant)? One may expect that, although there exists a term (like the

J. Esparza and A.S. Murawski (Eds.): FOSSACS 2017, LNCS 10203, pp. 53–68, 2017.
DOI: 10.1007/978-3-662-54458-7_4

one above) whose reduction sequence is as long as $\mathbf{exp}_k(\Omega(n))$, such a term is rarely encountered.

In the present paper, we provide a partial answer to the above question, by showing that almost every simply typed λ-term of order k has a reduction sequence as long as $(k-2)$-fold exponential in the term size, under a certain assumption. More precisely, we shall show:

$$\lim_{n\to\infty} \frac{\#\left(\{[t]_\alpha \in \Lambda_n^\alpha(k,\iota,\xi) \mid \beta(t) \geq \mathbf{exp}_{k-2}(n)\}\right)}{\#(\Lambda_n^\alpha(k,\iota,\xi))} = 1$$

where $\Lambda_n^\alpha(k,\iota,\xi)$ is the set of (α-equivalence classes $[-]_\alpha$ of) simply-typed λ-terms such that the term size is n, the order is up to k, the (internal) arity is up to $\iota \geq k$ and the number of variable names is up to ξ (see the next section for the precise definition).

To obtain the above result, we use techniques inspired by the quantitative analysis of *untyped* λ-terms [2–4]. For example, David et al. [2] have shown that almost all untyped λ-terms are strongly normalizing, whereas the result is opposite in the corresponding combinatory logic. A more sophisticated analysis is, however, required in our case, for considering only well-typed terms, and also for reasoning about the *length* of a reduction sequence instead of a qualitative property like strong normalization.

This work is a part of our long-term project on the quantitative analysis of the complexity of higher-order model checking [5,6]. The higher-order model checking asks whether the (possibly infinite) tree generated by a ground-type term of the λY-calculus (or, a higher-order recursion scheme) satisfies a given regular property, and it is known that the problem is k-EXPTIME complete for order-k terms [6]. Despite the huge worst-case complexity, practical model checkers [7–9] have been built, which run fast for many typical inputs, and have successfully been applied to automated verification of functional programs [10–13]. The project aims to provide a theoretical justification for it, by studying *how many* inputs actually suffer from the worst-case complexity. Since the problem appears to be hard due to recursion, as an intermediate step towards the goal, we aimed to analyze the variant of the problem considered by Terui [14]: given a term of the simply-typed λ-calculus (without recursion) of type Bool, decide whether it evaluates to true or false (where Booleans are Church-encoded; see [14] for the precise definition). Terui has shown that even for the problem, the complexity is k-EXPTIME complete for order-$(2k+2)$ terms. If, contrary to the result of the present paper, the upper-bound of the lengths of β-reduction sequences *were* small for almost every term, then we could have concluded that the decision problem above is easily solvable for most of the inputs. The result in the present paper does not necessarily provide a negative answer to the question above, because one need not necessarily apply β-reductions to solve Terui's decision problem.

The present work may also shed some light on other problems on typed λ-calculi with exponential or higher worst-case complexity. For example, despite DEXPTIME-completeness of ML typability [15,16], it is often said that the

exponential behavior is rarely seen *in practice*. That is, however, based on only empirical studies. Our technique may be used to provide a theoretical justification (or possibly *un*justification).

The rest of this paper is organized as follows. Section 2 states our main result formally. Section 3 analyzes the asymptotic behavior of the number of typed λ-terms of a given size. Section 4 proves the main result. Section 5 discusses related work, and Sect. 6 concludes the paper. For the space restriction, we omit formal proofs and give only sketches instead; see the full version [17] for details.

2 Main Result

In this section we give the precise statement of our main theorem. We denote the cardinality of a set S by $\#(S)$, and the domain and image of a function f by $\mathrm{Dom}(f)$ and $\mathrm{Im}(f)$, respectively.

The set of (*simple*) *types*, ranged over by τ and σ, is given by: $\tau ::= \mathsf{o} \mid \sigma \to \tau$. Let V be a countably infinite set, which is ranged over by x, x_1, x_2, etc. The set of λ-*terms* (or *terms*), ranged over by t, is defined by:

$$t ::= x \mid \lambda \overline{x}^{\tau}.t \mid t\,t \qquad\qquad \overline{x} ::= x \mid *$$

We call elements of $V \cup \{*\}$ *variables*; $V \cup \{*\}$ is ranged over by $\overline{x}, \overline{x}_1, \overline{x}_2$, etc. We call the special variable $*$ an *unused variable*. We sometimes omit type annotations and just write $\lambda\overline{x}.t$ for $\lambda\overline{x}^{\tau}.t$.

Terms of our syntax can be translated to usual λ-terms by regarding elements in $V \cup \{*\}$ as usual variables. We define the notions of free variables, closed terms, and α-equivalence $\sim_\alpha$ through this identification. The α-equivalence class of a term t is written as $[t]_\alpha$. In this paper, we do not consider a term as an α-equivalence class, and we always use $[-]_\alpha$ explicitly. For a term t, we write $\mathbf{FV}(t)$ for the set of all the free variables of t.

For a term t, we define the set $\mathbf{V}(t)$ of variables (except $*$) in t by:

$$\mathbf{V}(x) \triangleq \{x\} \quad \mathbf{V}(\lambda x^{\tau}.t) \triangleq \{x\} \cup \mathbf{V}(t) \quad \mathbf{V}(\lambda *^{\tau}.t) \triangleq \mathbf{V}(t) \quad \mathbf{V}(t_1 t_2) \triangleq \mathbf{V}(t_1) \cup \mathbf{V}(t_2).$$

Note that neither $\mathbf{V}(t)$ nor even $\#(\mathbf{V}(t))$ is preserved by α-equivalence. For example, $t = \lambda x_1.(\lambda x_2.x_2)(\lambda x_3.x_1)$ and $t' = \lambda x_1.(\lambda x_1.x_1)(\lambda *.x_1)$ are α-equivalent, but $\#(\mathbf{V}(t)) = 3$ and $\#(\mathbf{V}(t')) = 1$.

A *type environment* Γ is a finite set of type bindings of the form $x : \tau$ such that if $(x : \tau), (x : \tau') \in \Gamma$ then $\tau = \tau'$; sometimes we regard an environment also as a function. Note that $(* : \tau)$ cannot belong to a type environment; we do not need any type assumption for $*$ since it does not occur in terms. We give the typing rules as follows:

$$\frac{}{x : \tau \vdash x : \tau} \qquad \frac{\Gamma_1 \vdash t_1 : \sigma \to \tau \qquad \Gamma_2 \vdash t_2 : \sigma}{\Gamma_1 \cup \Gamma_2 \vdash t_1 t_2 : \tau}$$

$$\frac{\Gamma' \vdash t : \tau \qquad \Gamma' = \Gamma \text{ or } \Gamma' = \Gamma \cup \{\overline{x} : \sigma\} \qquad \overline{x} \notin \mathrm{Dom}(\Gamma)}{\Gamma \vdash \lambda \overline{x}^{\sigma}.t : \sigma \to \tau}$$

The above typing rules are equivalent to the usual ones for closed terms, and if $\Gamma \vdash t : \tau$ is derivable, then the derivation is unique. Moreover, if $\Gamma \vdash t : \tau$ then $\mathrm{Dom}(\Gamma) = \mathbf{FV}(t)$. Below we consider only well-typed λ-terms. A pair $\langle \Gamma; \tau \rangle$ of Γ and τ is called a *typing*. We use θ as a metavariable for typings. When $\Gamma \vdash t : \tau$ is derived, we call $\langle \Gamma; \tau \rangle$ a *typing of* a term t, and call t an *inhabitant of* $\langle \Gamma; \tau \rangle$ or a $\langle \Gamma; \tau \rangle$*-term*.

Definition 1 (size, order and internal arity of a term). The *size* of a term t, written $|t|$, is defined by:

$$|\overline{x}| \triangleq 1 \qquad |\lambda \overline{x}^{\tau}.t| \triangleq |t| + 1 \qquad |t_1 t_2| \triangleq |t_1| + |t_2| + 1.$$

The *order* and *internal arity* of a type τ, written $\mathtt{ord}(\tau)$ and $\mathtt{iar}(\tau)$, are defined respectively by:

$$\begin{aligned} \mathtt{ord}(\mathtt{o}) &\triangleq 0 \qquad \mathtt{iar}(\mathtt{o}) \triangleq 0 \\ \mathtt{ord}(\tau_1 \to \cdots \to \tau_n \to \mathtt{o}) &\triangleq \max\{\mathtt{ord}(\tau_i) + 1 \mid 1 \le i \le n\} \qquad (n \ge 1) \\ \mathtt{iar}(\tau_1 \to \cdots \to \tau_n \to \mathtt{o}) &\triangleq \max(\{n\} \cup \{\mathtt{iar}(\tau_i) \mid 1 \le i \le n\}) \quad (n \ge 1). \end{aligned}$$

For a $\langle \Gamma; \tau \rangle$-term t, we define the order and internal arity of $\Gamma \vdash t : \tau$ written $\mathtt{ord}(\Gamma \vdash t : \tau)$ and $\mathtt{iar}(\Gamma \vdash t : \tau)$ by:

$$\begin{aligned} \mathtt{ord}(\Gamma \vdash t : \tau) &\triangleq \max\{\mathtt{ord}(\tau') \mid (\Gamma' \vdash t' : \tau') \text{ occurs in } \Delta\} \\ \mathtt{iar}(\Gamma \vdash t : \tau) &\triangleq \max\{\mathtt{iar}(\tau') \mid (\Gamma' \vdash t' : \tau') \text{ occurs in } \Delta\} \end{aligned}$$

where Δ is the (unique) derivation tree for $\Gamma \vdash t : \tau$.

Note that the notions of size, order, internal arity, and $\beta(t)$ (defined in the introduction) are well-defined with respect to α-equivalence.

Definition 2 (terms with bounds on types and variables). Let $\delta, \iota, \xi \ge 0$ and $n \ge 1$ be integers. We denote by $\mathrm{Types}(\delta, \iota)$ the set of types $\{\tau \mid \mathtt{ord}(\tau) \le \delta, \mathtt{iar}(\tau) \le \iota\}$. For Γ and τ we define:

$$\begin{aligned} \Lambda^{\alpha}_n(\langle \Gamma; \tau \rangle, \delta, \iota, \xi) \triangleq \{[t]_\alpha \mid\ & \Gamma \vdash t : \tau,\ |t| = n,\ \min_{t' \in [t]_\alpha} \#(\mathbf{V}(t')) \le \xi, \\ & \mathtt{ord}(\Gamma \vdash t : \tau) \le \delta,\ \mathtt{iar}(\Gamma \vdash t : \tau) \le \iota\} \\ \Lambda^{\alpha}(\langle \Gamma; \tau \rangle, \delta, \iota, \xi) \triangleq & \bigcup_{n \ge 1} \Lambda^{\alpha}_n(\langle \Gamma; \tau \rangle, \delta, \iota, \xi). \end{aligned}$$

Also we define:

$$\Lambda^{\alpha}_n(\delta, \iota, \xi) \triangleq \bigcup_{\tau \in \mathrm{Types}(\delta, \iota)} \Lambda^{\alpha}_n(\langle \emptyset; \tau \rangle, \delta, \iota, \xi) \qquad \Lambda^{\alpha}(\delta, \iota, \xi) \triangleq \bigcup_{n \ge 1} \Lambda^{\alpha}_n(\delta, \iota, \xi).$$

Our main result is the following theorem, which will be proved in Sect. 4.

Theorem 1. *Let $\delta, \iota, \xi \ge 2$ be integers and let $k = \min\{\delta, \iota\}$. Then,*

$$\lim_{n \to \infty} \frac{\#\big(\{[t]_\alpha \in \Lambda^{\alpha}_n(\delta, \iota, \xi) \mid \beta(t) \ge \mathbf{exp}_{k-2}(n)\}\big)}{\#(\Lambda^{\alpha}_n(\delta, \iota, \xi))} = 1.$$

Remark 1. Note that in the above theorem, the order δ, the internal arity ι and the number ξ of variables are bounded above by a constant, independently of the term size n. It is debatable whether the assumption is reasonable, and a slight change of the assumption may change the result, as is the case for strong normalization of untyped λ-term [2,4]. When λ-terms are viewed as models of functional programs, our rationale behind the assumption is as follows. The assumption that the size of types (hence also the order and the internal arity) is fixed is sometimes assumed in the context of type-based program analysis [18]. The assumption on the number of variables comes from the observation that a large program usually consists of a large number of *small* functions, and that the number of variables is bounded by the size of each function.

3 Analysis of $\Lambda_n^\alpha(\delta, \iota, \xi)$

To prove our main theorem, we first analyze some formal language theoretic structure and properties of $\Lambda^\alpha(\delta, \iota, \xi)$: in Sect. 3.1, we construct a regular tree grammar such that there is a size preserving bijection between its tree language and $\Lambda^\alpha(\delta, \iota, \xi)$; in Sect. 3.2, we show that the grammar has two important properties: irreducibility and aperiodicity. Thanks to those properties, we can obtain a simple asymptotic formula for $\#(\Lambda_n^\alpha(\delta, \iota, \xi))$ using analytic combinatorics [19]. The irreducibility and aperiodicity properties will also be used in Sect. 4 for adjusting the size and typing of a λ-term.

3.1 $\Lambda^\alpha(\delta, \iota, \xi)$ as a Regular Tree Language

We first recall some basic definitions for regular tree grammars. A *ranked alphabet* Σ is a mapping from a finite set of symbols to the set of natural numbers. For a symbol $a \in \mathrm{Dom}(\Sigma)$, we call $\Sigma(a)$ the *rank* of a. A *Σ-tree* is a tree composed from symbols in Σ according to their ranks: (i) a is a Σ-tree if $\Sigma(a) = 0$, (ii) $a(T_1, \cdots, T_{\Sigma(a)})$ is a Σ-tree if $\Sigma(a) \geq 1$ and T_i is a Σ-tree for each $i \in \{1, \ldots, \Sigma(a)\}$. We use the meta-variable T for trees. The *size* of T, written as $|T|$, is the number of nodes and leaves of T. We denote the set of all Σ-trees by $\mathcal{T}_\Sigma$.

A *regular tree grammar* is a triple $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ where (i) Σ is a ranked alphabet; (ii) $\mathcal{N}$ is a finite set of *non-terminals*; (iii) $\mathcal{R}$ is a finite set of *rewriting rules* of the form $N \longrightarrow a(N_1, \cdots, N_{\Sigma(a)})$ where $a \in \mathrm{Dom}(\Sigma)$, $N \in \mathcal{N}$ and $N_i \in \mathcal{N}$ for every $i \in \{1, \ldots, \Sigma(a)\}$. A *$(\Sigma \cup \mathcal{N})$-tree* T is a tree composed from symbols in $\Sigma \cup \mathcal{N}$ according to their ranks where the rank of every symbol in $\mathcal{N}$ is zero (thus non-terminals appear only in leaves of T). For a tree grammar $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ and a non-terminal $N \in \mathcal{N}$, *the language $\mathcal{L}(\mathcal{G}, N)$ of N* is defined by $\mathcal{L}(\mathcal{G}, N) \triangleq \{T \in \mathcal{T}_\Sigma \mid N \longrightarrow_{\mathcal{G}}^* T\}$ where $\longrightarrow_{\mathcal{G}}^*$ denotes the reflexive and transitive closure of the rewriting relation $\longrightarrow_{\mathcal{G}}$. We also define $\mathcal{L}_n(\mathcal{G}, N) \triangleq \{T \in \mathcal{T}_\Sigma \mid N \longrightarrow^* T, |T| = n\}$. We often omit $\mathcal{G}$ and write $N \longrightarrow^* N'$, $\mathcal{L}(N)$, and $\mathcal{L}_n(N)$ for $N \longrightarrow_{\mathcal{G}}^* N'$, $\mathcal{L}(\mathcal{G}, N)$, and $\mathcal{L}_n(\mathcal{G}, N)$ respectively, if $\mathcal{G}$ is clear from the context. We say that N' is *reachable* from N if there exists a $(\Sigma \cup \mathcal{N})$-tree T

such that $N \longrightarrow^* T$ and T contains N' as a leaf. A grammar $\mathcal{G}$ is *unambiguous* if, for every pair of a non-terminal N and a tree T, there exists at most one leftmost reduction sequence from N to T.

Definition 3 (grammar of $\Lambda^\alpha(\delta,\iota,\xi)$). Let $\delta,\iota,\xi \geq 0$ be integers and $X_\xi = \{x_1,\cdots,x_\xi\}$ be a subset of V. The regular tree grammar $\mathcal{G}(\delta,\iota,\xi)$ is defined as $(\Sigma(\delta,\iota,\xi),\mathcal{N}(\delta,\iota,\xi),\mathcal{R}(\delta,\iota,\xi))$ where:

$$\begin{aligned}
\Sigma(\delta,\iota,\xi) &\triangleq \{x \mapsto 0 \mid x \in X_\xi\} \cup \{@ \mapsto 2\} \\
&\quad \cup \{\lambda\overline{x}^\tau \mapsto 1 \mid \overline{x} \in \{*\} \cup X_\xi,\ \tau \in \mathrm{Types}(\delta-1,\iota)\} \\
\mathcal{N}(\delta,\iota,\xi) &\triangleq \{N_{\langle\Gamma;\tau\rangle} \mid \tau \in \mathrm{Types}(\delta,\iota), \mathrm{Dom}(\Gamma) \subseteq X_\xi,\ \mathrm{Im}\,(\Gamma) \subseteq \mathrm{Types}(\delta-1,\iota), \\
&\qquad\qquad \Gamma \vdash t : \tau \text{ for some } t\ \} \\
\mathcal{R}(\delta,\iota,\xi) &\triangleq \{N_{\langle\{x_i:\tau\};\tau\rangle} \longrightarrow x_i\} \cup \{N_{\langle\Gamma;\sigma\to\tau\rangle} \longrightarrow \lambda *^\sigma (N_{\langle\Gamma;\tau\rangle})\} \\
&\quad \cup \{N_{\langle\Gamma;\sigma\to\tau\rangle} \longrightarrow \lambda x_i^\sigma (N_{\langle\Gamma\cup\{x_i:\sigma\};\tau\rangle}) \mid i = \min\{j \geq 1 \mid x_j \notin \mathrm{Dom}(\Gamma)\}, \\
&\qquad \#(\Gamma) < \xi\} \cup \{N_{\langle\Gamma;\tau\rangle} \longrightarrow @(N_{\langle\Gamma_1;\sigma\to\tau\rangle}, N_{\langle\Gamma_2;\sigma\rangle}) \mid \Gamma = \Gamma_1 \cup \Gamma_2\}
\end{aligned}$$

Here, the special symbol $@ \in \mathrm{Dom}(\Sigma(\delta,\iota,\xi))$ corresponds to application. For a technical convenience, the above definition excludes from $\mathcal{N}(\delta,\iota,\xi)$ typings which have no inhabitant. Note that $\Sigma(\delta,\iota,\xi)$, $\mathcal{N}(\delta,\iota,\xi)$ and $\mathcal{R}(\delta,\iota,\xi)$ are finite. To see the finiteness of $\mathcal{N}(\delta,\iota,\xi)$, notice that X_ξ and $\mathrm{Types}(\delta-1,\iota)$ are finite, hence so is $\{\Gamma \mid \mathrm{Dom}(\Gamma) \subseteq X_\xi,\ \mathrm{Im}\,(\Gamma) \subseteq \mathrm{Types}(\delta-1,\iota)\}$. The finiteness of $\mathcal{R}(\delta,\iota,\xi)$ follows immediately from that of $\mathcal{N}(\delta,\iota,\xi)$.

Example 1. Let us consider the case where $\delta = \iota = \xi = 1$. The grammar $\mathcal{G}(1,1,1)$ consists of the following.

$$\Sigma(1,1,1) = \{x_1, @, \lambda x_1^\circ, \lambda *^\circ\} \qquad \mathcal{N}(1,1,1) = \{N_{\langle\emptyset;\mathtt{o}\to\mathtt{o}\rangle}, N_{\langle\{x_1:\mathtt{o}\};\mathtt{o}\rangle} N_{\langle\{x_1:\mathtt{o}\};\mathtt{o}\to\mathtt{o}\rangle}\}$$

$$\mathcal{R}(1,1,1) = \begin{cases} N_{\langle\emptyset;\mathtt{o}\to\mathtt{o}\rangle} \longrightarrow \lambda x_1^\circ (N_{\langle\{x_1:\mathtt{o}\};\mathtt{o}\rangle}) \\ N_{\langle\{x_1:\mathtt{o}\};\mathtt{o}\rangle} \longrightarrow x_1 | @(N_{\langle\{x_1:\mathtt{o}\};\mathtt{o}\to\mathtt{o}\rangle}, N_{\langle\{x_1:\mathtt{o}\};\mathtt{o}\rangle}) | @(N_{\langle\emptyset;\mathtt{o}\to\mathtt{o}\rangle}, N_{\langle\{x_1:\mathtt{o}\};\mathtt{o}\rangle}) \\ N_{\langle\{x_1:\mathtt{o}\};\mathtt{o}\to\mathtt{o}\rangle} \longrightarrow \lambda *^\circ (N_{\langle\{x_1:\mathtt{o}\};\mathtt{o}\rangle}). \end{cases}$$

There is the obvious embedding $e^{(\delta,\iota,\xi)}$ (e for short) from trees in $\mathcal{T}_{\Sigma(\delta,\iota,\xi)}$ into λ-terms. For $N_{\langle\Gamma;\tau\rangle} \in \mathcal{N}(\delta,\iota,\xi)$ we define

$$\pi^{(\delta,\iota,\xi)}_{\langle\Gamma;\tau\rangle} \triangleq [-]_\alpha \circ e : \mathcal{L}\left(N_{\langle\Gamma;\tau\rangle}\right) \to \Lambda^\alpha(\langle\Gamma;\tau\rangle,\delta,\iota,\xi).$$

We sometimes omit the superscript and/or the subscript.

Proposition 1. *For $\delta,\iota,\xi \geq 0$, $\pi_{\langle\Gamma;\tau\rangle}$ is a size-preserving bijection, and $\mathcal{G}(\delta,\iota,\xi)$ is unambiguous.*

The former part of Proposition 1 says that $\mathcal{G}(\delta,\iota,\xi)$ gives a complete representation system of the α-equivalence classes. For $[t]_\alpha \in \Lambda^\alpha(\langle\Gamma;\tau\rangle,\delta,\iota,\xi)$, we define $\nu^{(\delta,\iota,\xi)}_{\langle\Gamma;\tau\rangle}([t]_\alpha)$ (or $\nu([t]_\alpha)$ for short) as $e^{(\delta,\iota,\xi)} \circ \left(\pi^{(\delta,\iota,\xi)}_{\langle\Gamma;\tau\rangle}\right)^{-1}([t]_\alpha)$. The function ν *normalizes* variable names. For example, $t = \lambda x.x(\lambda y.\lambda z.z)$ is normalized to $\nu([t]_\alpha) = \lambda x_1.x_1(\lambda *.\lambda x_1.x_1)$.

Due to technical reasons, we restrict the grammar $\mathcal{G}(\delta,\iota,\xi)$ to $\mathcal{G}^{\emptyset}(\delta,\iota,\xi)$, which contains only non-terminals reachable from $N_{\langle\emptyset;\sigma\rangle}$ for some σ (see the full version [17] for details).

$$\mathcal{N}^{\emptyset}(\delta,\iota,\xi) \triangleq \{N_\theta \in \mathcal{N}(\delta,\iota,\xi) \mid N_\theta \text{ is reachable from some } N_{\langle\emptyset;\sigma\rangle} \in \mathcal{N}(\delta,\iota,\xi)\}$$
$$\mathcal{R}^{\emptyset}(\delta,\iota,\xi) \triangleq \{N_\theta \longrightarrow T \in \mathcal{R}(\delta,\iota,\xi) \mid N_\theta \in \mathcal{N}^{\emptyset}(\delta,\iota,\xi)\}$$
$$\mathcal{G}^{\emptyset}(\delta,\iota,\xi) \triangleq (\Sigma(\delta,\iota,\xi), \mathcal{N}^{\emptyset}(\delta,\iota,\xi), \mathcal{R}^{\emptyset}(\delta,\iota,\xi)).$$

For $N_\theta \in \mathcal{N}^{\emptyset}(\delta,\iota,\xi)$, clearly $\mathcal{L}\left(\mathcal{G}^{\emptyset}(\delta,\iota,\xi), N_\theta\right) = \mathcal{L}\left(\mathcal{G}(\delta,\iota,\xi), N_\theta\right)$. Through the bijection π, we can show that, for any $N_{\langle\Gamma;\tau\rangle} \in \mathcal{N}(\delta,\iota,\xi)$, $N_{\langle\Gamma;\tau\rangle}$ also belongs to $\mathcal{N}^{\emptyset}(\delta,\iota,\xi)$ if and only if there exists a term in $\Lambda^{\alpha}(\delta,\iota,\xi)$ whose derivation contains a type judgment of the form $\Gamma \vdash t : \tau$.

3.2 Irreducibility and Aperiodicity

We discuss two important properties of the grammar $\mathcal{G}^{\emptyset}(\delta,\iota,\xi)$ where $\delta,\iota,\xi \geq 2$: irreducibility and aperiodicity [19].[1]

Definition 4 (irreducibility and aperiodicity). Let $\mathcal{G} = (\Sigma,\mathcal{N},\mathcal{R})$ be a regular tree grammar. We say that $\mathcal{G}$ is:

- *non-linear* if $\mathcal{R}$ contains at least one rule of the form $N \longrightarrow a(N_1,\cdots,N_{\Sigma(a)})$ with $\Sigma(a) \geq 2$,
- *strongly connected* if for any pair of non-terminals $N_1, N_2 \in \mathcal{N}$, N_1 is reachable from N_2,
- *irreducible* if $\mathcal{G}$ is both non-linear and strongly connected,
- *aperiodic* if for any non-terminal $N \in \mathcal{N}$ there exists an integer $m > 0$ such that $\#(\mathcal{L}_n(N)) > 0$ for any $n > m$.

Proposition 2. *$\mathcal{G}^{\emptyset}(\delta,\iota,\xi)$ is irreducible and aperiodic for any $\delta,\iota,\xi \geq 2$.*

The following theorem is a minor modification of Theorem VII.5 in [19], which states the asymptotic behavior of an irreducible and aperiodic *context-free specification* (see the full version [17] for details). Below, $\sim$ means the *asymptotic equality*, i.e., $f(n) \sim g(n) \iff \lim_{n\to\infty} f(n)/g(n) = 1$.

Theorem 2 ([19]). *Let $\mathcal{G} = (\Sigma,\mathcal{N},\mathcal{R})$ be an unambiguous, irreducible and aperiodic regular tree grammar. Then there exists a constant $\gamma(\mathcal{G}) > 1$ such that, for any non-terminal $N \in \mathcal{N}$, there exists a constant $C_N(\mathcal{G}) > 0$ such that*

$$\#(\mathcal{L}_n(N)) \sim C_N(\mathcal{G})\gamma(\mathcal{G})^n n^{-3/2}.$$

As a corollary of Proposition 2 and Theorem 2 above, we obtain:

$$\#(\Lambda_n^{\alpha}(\delta,\iota,\xi)) \sim C\gamma^n n^{-3/2} \tag{1}$$

where $C > 0$ and $\gamma > 1$ are some real constants determined by δ, ι, $\xi \geq 2$. For proving our main theorem, we use a variation of the formula (1) above, stated as Lemma 1 later.

[1] In [19], irreducibility and aperiodicity are defined for *context-free specifications*. Our definition is a straightforward adaptation of the definition for regular tree grammars.

4 Proof of the Main Theorem

We give a proof of Theorem 1 in this section. In the rest of the paper, we denote by $\log^{(2)}(n)$ the 2-fold logarithm: $\log^{(2)}(n) \triangleq \log\log n$. All logarithms are base 2. The outline of the proof is as follows. We prepare a family $(t_n)_{n\in\mathbb{N}}$ of λ-terms such that t_n is of size $\Omega(\log^{(2)}(n))$ and has a β-reduction sequence of length $\mathbf{exp}_k(\Omega(|t_n|))$, i.e., $\mathbf{exp}_{k-2}(\Omega(n))$. Then we show that almost every λ-term of size n contains t_n as a subterm. The latter is shown by adapting (a parameterized version of) the *infinite monkey theorem*[2] for words to simply-typed λ-terms.

To clarify the idea, let us first recall the infinite monkey theorem for words. Let A be an alphabet, i.e., a finite non-empty set of symbols. For a word $w = a_1 \cdots a_n$, we write $|w| = n$ for the length of w. As usual, we denote by A^n the set of all words of length n over A, and by A^* the set of all finite words over A: $A^* = \bigcup_{n\geq 0} A^n$. For two words $w, w' \in A^*$, we say w' is a *subword of* w and write $w' \sqsubseteq w$ if $w = w_1 w' w_2$ for some words $w_1, w_2 \in A^*$. The infinite monkey theorem states that, for any word $w \in A^*$, the probability that a randomly chosen word of size n contains w as a subword tends to one if n tends to infinity.

To prove our main theorem, we need to extend the above infinite monkey theorem to the following parameterized version[3], and then further extend it for simply-typed λ-terms instead of words. We give a proof of the following proposition, because it will clarify the overall structure of the proof of the main theorem.

Proposition 3. (parameterized infinite monkey theorem). *Let A be an alphabet and $(w_n)_n$ be a family of words over A such that $|w_n| = \lceil \log^{(2)}(n) \rceil$. Then, we have:*

$$\lim_{n\to\infty} \frac{\#(\{w \in A^n \mid w_n \sqsubseteq w\})}{\#(A^n)} = 1.$$

Proof. Let $p(n)$ be $1 - \#(\{w \in A^n \mid w_n \sqsubseteq w\})/\#(A^n)$, i.e., the probability that a word of size n does *not* contain w_n. We write $s(n)$ for $\lceil \log^{(2)}(n) \rceil$ and $c(n)$ for $\lfloor n/s(n) \rfloor$. Given a word $w = a_1 \cdots a_n \in A^n$, let us partition it to subwords of length $s(n)$ as follows.

$$w = \underbrace{a_1 \cdots a_{s(n)}}_{\text{1-st subword}} \cdots \underbrace{a_{(c(n)-1)s(n)+1} \cdots a_{c(n)s(n)}}_{c(n)\text{-th subword}} a_{c(n)s(n)+1} \cdots a_n$$

Then,

$$\begin{aligned} p(n) &\leq \text{“the probability that none of the } i\text{-th subword is } w_n'' \\ &= \left(\frac{\#(A^{s(n)} \setminus \{w_n\})}{\#(A^{s(n)})}\right)^{c(n)} = \left(\frac{\#(A^{s(n)}) - 1}{\#(A^{s(n)})}\right)^{c(n)} = \left(1 - \frac{1}{\#(A)^{s(n)}}\right)^{c(n)}. \end{aligned}$$

[2] It is also called as “Borges's theorem” (*cf.* [19, p. 61, Note I.35]).

[3] Although it is a simple extension, we are not aware of literature that explicitly states this parameterized version.

Since $\left(1-\frac{1}{\#(A)^{s(n)}}\right)^{c(n)} = \left(1-\frac{1}{\#(A)^{\lceil \log^{(2)}(n)\rceil}}\right)^{\lfloor n/\lceil \log^{(2)}(n)\rceil \rfloor}$ tends to zero (see the full version [17]) if n tends to infinity, we have the required result. □

To prove an analogous result for simply-typed λ-terms, we consider below subcontexts of a given term instead of subwords of a given word. To consider "contexts up to α-equivalence", in Sect. 4.1 we introduce the set $\mathcal{U}^{\nu}_{n}(\delta,\iota,\xi)$ of "normalized" contexts (of size n and with the restriction by δ, ι and ξ), where $\mathcal{U}^{\nu}_{s(n)}(\delta,\iota,\xi)$ corresponds to $A^{s(n)}$ above, and give an upper bound of $\#(\mathcal{U}^{\nu}_{n}(\delta,\iota,\xi))$. A key property used in the above proof was that any word of length n can be partitioned to sufficiently many subwords of length $\log^{(2)}(n)$. Section 4.2 below shows an analogous result that any term of size n can be decomposed into sufficiently many subcontexts of a given size. Section 4.3 constructs a family of contexts $Expl^{k}_{n}$ (called "explosive contexts") that have very long reduction sequences; $(Expl^{k}_{n})_{n}$ corresponds to $(w_n)_n$ above. Finally, Sect. 4.4 proves the main theorem using an argument similar to (but more involved than) the one used in the proof above.

4.1 Normalized Contexts

We first introduce some basic definitions of contexts, and then we define the notion of a normalized context, which is a context normalized by the function ν given in Sect. 3.1.

The set of *contexts*, ranged over by C, is defined by

$$C ::= [\,] \mid x \mid \lambda\overline{x}^{\tau}.C \mid CC$$

The *size* of C, written $|C|$, is defined by:

$$|[\,]| \triangleq 0 \qquad |x| \triangleq 1 \qquad |\lambda\overline{x}^{\tau}.C| \triangleq |C|+1 \qquad |C_1C_2| \triangleq |C_1|+|C_2|+1.$$

We call a context C an *n-context* (and define $\mathtt{hn}(C) \triangleq n$) if C contains n occurrences of $[\,]$. We use the metavariable S for 1-contexts. A *0/1-context* is a term t or a 1-context S and we use the metavariable u to denote 0/1-contexts. The holes in C occur as leaves and we write $[\,]_i$ for the i-th hole, which is counted in the left-to-right order.

For C, C_1, …, $C_{\mathtt{hn}(C)}$, we write $C[C_1]\dots[C_{\mathtt{hn}(C)}]$ for the context obtained by replacing $[\,]_i$ in C with C_i for each $i \le \mathtt{hn}(C)$. For C and C', we write $C[C']_i$ for the context obtained by replacing the i-th hole $[\,]_i$ in C with C'. As usual, these substitutions may capture variables; e.g., $(\lambda x.[\,])[x]$ is $\lambda x.x$. We say that C is a *subcontext of* C' and write $C \preceq C'$ if there exist C'', $1 \le i \le \mathtt{hn}(C'')$ and $C_1, \cdots, C_{\mathtt{hn}(C)}$ such that $C' = C''[C[C_1]\cdots[C_{\mathtt{hn}(C)}]]_i$.

The set of *context typings*, ranged over by κ, is defined by: $\kappa ::= \theta_1 \cdots \theta_k \Rightarrow \theta$ where $k \in \mathbb{N}$ and θ_i is a typing of the form $\langle \Gamma_i; \tau_i\rangle$ for each $1 \le i \le k$ (recall that we use θ as a metavariable for typings). A $\langle \Gamma_1;\tau_1\rangle \cdots \langle \Gamma_k;\tau_k\rangle \Rightarrow \langle \Gamma;\tau\rangle$-context is a k-context C such that $\Gamma \vdash C : \tau$ is derivable from $\Gamma_i \vdash [\,]_i : \tau_i$. We identify a context typing $\Rightarrow\theta$ with the typing θ, and call a θ-context also a *θ-term*.

From now, we begin to define normalized contexts. First we consider contexts in terms of the grammar $\mathcal{G}^{\emptyset}(\delta,\iota,\xi)$ given in Sect. 3.1. Let $\delta,\iota,\xi \geq 0$. For $\kappa = \theta_1 \cdots \theta_n \Rightarrow \theta$ such that $N_{\theta_1},\ldots,N_{\theta_n},N_\theta \in \mathcal{N}(\delta,\iota,\xi)$, a ($\kappa$-)*context-tree* is a tree $\widehat{T}$ in $\mathcal{T}_{\Sigma(\delta,\iota,\xi)\cup\mathcal{N}(\delta,\iota,\xi)}$ such that there exists a reduction $N_\theta \longrightarrow^* \widehat{T}$ and the occurrences of non-terminals in $\widehat{T}$ (in the left-to-right order) are exactly $N_{\theta_1},\ldots,N_{\theta_n}$. We use $\widehat{T}$ as a metavariable for context-trees. We write $\mathcal{L}(\kappa,\delta,\iota,\xi)$ for the set of all κ-context-trees. For $\theta_1\cdots\theta_n\Rightarrow\theta$-context-tree $\widehat{T}$ and $\theta^i_1\cdots\theta^i_{k_i}\Rightarrow\theta_i$-context-trees $\widehat{T}_i$ $(i=1,\ldots,n)$, we define the *substitution* $\widehat{T}[\widehat{T}_1]\cdots[\widehat{T}_n]$ as the $\theta^1_1\cdots\theta^1_{k_1}\cdots\theta^n_1\cdots\theta^n_{k_n}\Rightarrow\theta$- context-tree obtained by replacing N_{θ_i} in $\widehat{T}$ with $\widehat{T}_i$.

The set $\mathcal{C}^\nu(\kappa,\delta,\iota,\xi)$ of *normalized κ-contexts* is defined by:

$$\mathcal{C}^\nu(\kappa,\delta,\iota,\xi) \triangleq e_\kappa^{(\delta,\iota,\xi)}(\mathcal{L}(\kappa,\delta,\iota,\xi))$$

where $e_\kappa^{(\delta,\iota,\xi)}$ is the obvious embedding from κ-context-trees to κ-contexts that preserves the substitution (i.e., $e_\kappa^{(\delta,\iota,\xi)}(T[T']) = e_\kappa^{(\delta,\iota,\xi)}(T)[e_\kappa^{(\delta,\iota,\xi)}(T')]$). Further, the sets $\mathcal{U}^\nu(\delta,\iota,\xi)$ and $\mathcal{U}^\nu_n(\delta,\iota,\xi)$ of *normalized 0/1-contexts* are defined by:

$$\mathcal{U}^\nu(\delta,\iota,\xi) \triangleq \Big(\bigcup_{N_\theta\in\mathcal{N}^\emptyset(\delta,\iota,\xi)} \mathcal{C}^\nu(\theta,\delta,\iota,\xi)\Big) \bigcup \Big(\bigcup_{N_\theta,N_{\theta'}\in\mathcal{N}^\emptyset(\delta,\iota,\xi)} \mathcal{C}^\nu(\theta\Rightarrow\theta',\delta,\iota,\xi)\Big)$$

$$\mathcal{U}^\nu_n(\delta,\iota,\xi) \triangleq \{u \in \mathcal{U}^\nu(\delta,\iota,\xi) \mid |u| = n\}.$$

In our proof of the main theorem, the set $\mathcal{U}^\nu_{s(n)}(\delta,\iota,\xi)$ plays a role corresponding to $A^{s(n)}$ in the word case explained above. Note that in the word case we calculated the limit of some upper bound of $p(n)$; similarly, in our proof, we only need an upper bound of $\#(\mathcal{U}^\nu_n(\delta,\iota,\xi))$, which is given as follows.

Lemma 1. (upper bound of $\#(\mathcal{U}^\nu_n(\delta,\iota,\xi))$). *For any $\delta,\iota,\xi \geq 2$, there exists some constant $\overline{\gamma}(\delta,\iota,\xi) > 1$ such that $\#(\mathcal{U}^\nu_n(\delta,\iota,\xi)) = \mathrm{O}(\overline{\gamma}(\delta,\iota,\xi)^n)$.*

Proof Sketch. Given an unambiguous, irreducible and aperiodic regular tree grammar, adding a new terminal of the form a_N and a new rule of the form $N \longrightarrow a_N$ for each non-terminal N does not change the unambiguity, irreducibility and aperiodicity. Let $\overline{\mathcal{G}}^{\emptyset}(\delta,\iota,\xi)$ be the grammar obtained by applying this transformation to $\mathcal{G}^{\emptyset}(\delta,\iota,\xi)$. We can regard a tree of $\overline{\mathcal{G}}^{\emptyset}(\delta,\iota,\xi)$ as a normalized context, with a_{N_θ} considered a hole with typing θ. Then, clearly we have

$$\#(\mathcal{U}^\nu_n(\delta,\iota,\xi)) \leq \#\Big(\cup_{N\in\mathcal{N}^\emptyset(\delta,\iota,\xi)}\mathcal{L}_n\left(\overline{\mathcal{G}}^{\emptyset}(\delta,\iota,\xi),N\right)\Big).$$

Thus the required result follows from Theorem 2. □

4.2 Decomposition

As explained in the beginning of this section, to prove the parameterized infinite monkey theorem for terms, we need to decompose a λ-term into sufficiently many subcontexts of the term. Thus, in this subsection, we will

define a decomposition function $\widehat{\Phi}_m$ (where m is a parameter) that decomposes a term t into (i) a (sufficiently long) sequence P of 0/1-subcontexts of t such that every component u of P satisfies $|u| \geq m$, and (ii) a "second-order" context E (defined later), which is a remainder of extracting P from t. Figure 1 illustrates how a term is decomposed by $\widehat{\Phi}_3$. Here, the symbols $[\![\,]\!]$ in the second-order context on the right-hand side represents the original position of each subcontext $(\lambda y.[\,])x$, $\lambda z.\lambda *.z$, and $(\lambda *.y)\lambda z.z$.

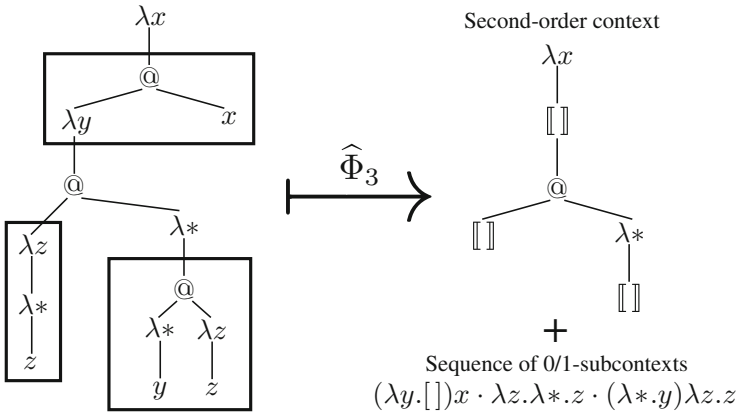

Fig. 1. Example of a decomposition

In order to define $\widehat{\Phi}_m$, let us give a precise definition of second-order contexts. The set of *second-order contexts*, ranged over by E, is defined by:

$$E ::= [\![\,]\!]_n^{\theta_1 \cdots \theta_k \Rightarrow \theta}[E_1] \cdots [E_k] \ (n \in \mathbb{N}) \mid x \mid \lambda \overline{x}^{\tau}.E \mid E_1 E_2.$$

Intuitively, the second-order context is an expression having holes of the form $[\![\,]\!]_n^{\kappa}$ (called *second-order holes*). In the second-order context $[\![\,]\!]_n^{\theta_1 \cdots \theta_k \Rightarrow \theta}[E_1] \cdots [E_k]$, $[\![\,]\!]_n^{\theta_1 \cdots \theta_k \Rightarrow \theta}$ should be filled with a $\theta_1 \cdots \theta_k \Rightarrow \theta$-context of size n, yielding a term whose typing is θ. We use the metavariable P for sequences of contexts. For a sequence of contexts $P = C_1 \cdot C_2 \cdots C_\ell$ and $i \leq \ell$, we write $\#(P)$ for the length ℓ, and $P.i$ for the i-th component C_i.

We define $|[\![\,]\!]_n^{\kappa}| \triangleq n$. We write $\mathtt{shn}(E)$ for the number of the second-order holes in E. For $i \leq \mathtt{shn}(E)$, we write $E.i$ for the i-th second-order hole (counted in the depth-first left-to-right pre-order). For a context C and a second-order hole $[\![\,]\!]_n^{\kappa}$, we write $C : [\![\,]\!]_n^{\kappa}$ if C is a κ-context of size n. For E and $P = C_1 \cdot C_2 \cdot \cdots \cdot C_{\mathtt{shn}(E)}$, we write $P : E$ if $C_i : E.i$ for each $i \leq \mathtt{shn}(E)$. We distinguish between second-order contexts with different annotations; for example, $[\![\,]\!]_0^{\langle\{x:\mathtt{o}\};\mathtt{o}\rangle \Rightarrow \langle\{x:\mathtt{o}\};\mathtt{o}\rangle}[x]$, $[\![\,]\!]_2^{\langle\{x:\mathtt{o}\};\mathtt{o}\rangle \Rightarrow \langle\{x:\mathtt{o}\};\mathtt{o}\rangle}[x]$ and $[\![\,]\!]_2^{\langle\{x:\mathtt{o}\to\mathtt{o}\};\mathtt{o}\to\mathtt{o}\rangle \Rightarrow \langle\{x:\mathtt{o}\to\mathtt{o}\};\mathtt{o}\to\mathtt{o}\rangle}[x]$ are different from each other. Note that every term can be regarded as a second-order context E such that $\mathtt{shn}(E) = 0$.

The bracket $[-]$ in a second-order context is just a syntactical representation rather than the substitution operation of contexts. Given E and C such that $\mathtt{shn}(E) \geq 1$ and $C : E\bullet 1$, we write $E[\![C]\!]$ for the second-order context obtained by replacing the leftmost second-order hole of E (i.e., $E\bullet 1$) with C (and by interpreting the syntactical bracket $[-]$ as the substitution operation). For example, we have: $((\lambda x.[\,][x][x])[\,])[\![\lambda y.y[\,][\,]]\!] = (\lambda x.(\lambda y.y[\,][\,])[x][x])[\,]\!] = (\lambda x.\lambda y.yxx)[\,]\!]$. Below we actually consider only second-order contexts whose second-order holes are of the form $[\,]^{\theta}_{n}$ or $[\,]^{\theta' \Rightarrow \theta}_{n}$.

We are now ready to define the decomposition function $\widehat{\Phi}_m$. We first prepare an auxiliary function $\Phi_m(t) = (E, u, P)$ such that (i) u is an auxiliary 0/1-subcontext, (ii) $E[\![u \cdot P]\!] = t$, and (iii) the size of each context in P is between m and $2m-1$. It is defined by induction on the size of t as follows:
If $|t| < m$, then $\Phi_m(t) \triangleq ([\,], t, \epsilon)$.
If $|t| \geq m$, then:

$$\Phi_m(\lambda\overline{x}^{\tau}.t_1) \triangleq \begin{cases} (E_1, \lambda\overline{x}^{\tau}.u_1, P_1) & \text{if } |\lambda\overline{x}^{\tau}.u_1| < m \\ ([\,][E_1], [\,], (\lambda\overline{x}^{\tau}.u_1) \cdot P_1) & \text{if } |\lambda\overline{x}^{\tau}.u_1| = m \end{cases}$$
$$\text{where } (E_1, u_1, P_1) = \Phi_m(t_1).$$

$$\Phi_m(t_1 t_2) \triangleq \begin{cases} ([\,][(E_1[\![u_1]\!])(E_2[\![u_2]\!])], [\,], P_1 \cdot P_2) & \text{if } |t_i| \geq m \ (i = 1, 2) \\ (E_1, u_1 t_2, P_1) & \text{if } |t_1| \geq m,\ |t_2| < m,\ |u_1 t_2| < m \\ ([\,][E_1], [\,], (u_1 t_2) \cdot P_1) & \text{if } |t_1| \geq m,\ |t_2| < m,\ |u_1 t_2| \geq m \\ (E_2, t_1 u_2, P_2) & \text{if } |t_1| < m,\ |t_1 u_2| < m \\ ([\,][E_2], [\,], (t_1 u_2) \cdot P_2) & \text{if } |t_1| < m,\ |t_1 u_2| \geq m \end{cases}$$
$$\text{where } (E_i, u_i, P_i) = \Phi_m(t_i) \quad (i = 1, 2).$$

Above, we have omitted the context-typing/size annotations of second-order holes for simplicity (see the full version [17] for details). The decomposition function $\widehat{\Phi}_m$ is then defined by $\widehat{\Phi}_m(t) \triangleq (E[\![u]\!], P)$ where $(E, u, P) = \Phi_m(t)$.

In the rest of this subsection, we show key properties of $\widehat{\Phi}_m$. We say that a 0/1-context u is *good for* m if u is either (i) a λ-abstraction where $|u| = m$; or (ii) an application $u_1 u_2$ where $|u_j| < m$ for each $j = 1, 2$. By the definition of $\widehat{\Phi}_m(t) = (E, P)$, every component u of P is good for m.

For $m \geq 2$, E and $1 \leq i \leq \mathtt{shn}(E)$, we define $\widehat{U}^m_{E\bullet i}(\delta, \iota, \xi)$, $\Lambda^m_E(\delta, \iota, \xi)$, and $\mathcal{B}^m_n(\delta, \iota, \xi)$ by:

$$\widehat{U}^m_{E\bullet i}(\delta, \iota, \xi) \triangleq \{u \in \mathcal{U}^{\nu}(\delta, \iota, \xi) \mid u : E\bullet i \text{ and } u \text{ is good for } m\}$$
$$\Lambda^m_E(\delta, \iota, \xi) \triangleq \{[E[\![u_1 \cdots u_{\mathtt{shn}(E)}]\!]]_{\alpha} \mid u_i \in \widehat{U}^m_{E\bullet i}(\delta, \iota, \xi) \text{ for } 1 \leq i \leq \mathtt{shn}(E)\}$$
$$\mathcal{B}^m_n(\delta, \iota, \xi) \triangleq \{E \mid (E, P) = \widehat{\Phi}_m(\nu([t]_{\alpha})) \text{ for some } [t]_{\alpha} \in \Lambda^{\alpha}_n(\delta, \iota, \xi)\}.$$

Intuitively, $\widehat{U}^m_{E\bullet i}(\delta, \iota, \xi)$ is the set of good contexts that can fill $E.i$, $\Lambda^m_E(\delta, \iota, \xi)$ is the set of terms obtained by filling the second-order holes of E with good contexts, and $\mathcal{B}^m_n(\delta, \iota, \xi)$ is the set of second-order contexts that can be obtained

by decomposing a term of size n. The following lemma states the key properties of $\widehat{\Phi}_m$.

Lemma 2. (decomposition). *Let $\delta, \iota, \xi \geq 0$ and $2 \leq m \leq n$.*

1. *$\Lambda_n^\alpha(\delta,\iota,\xi)$ is the disjoint union of $\Lambda_E^m(\delta,\iota,\xi)$'s, i.e., $\Lambda_n^\alpha(\delta,\iota,\xi) = \biguplus_{E\in\mathcal{B}_n^m(\delta,\iota,\xi)} \Lambda_E^m(\delta,\iota,\xi)$. Moreover, $\widehat{\Phi}_m(E[\![P]\!]) = (E,P)$ holds for any $P \in \prod_{1\leq i\leq \mathtt{shn}(E)} \widehat{U}_{E.i}^m(\delta,\iota,\xi)$.*
2. *$m \leq |E.i| < 2m$ $(1 \leq i \leq \mathtt{shn}(E))$ for every $E \in \mathcal{B}_n^m(\delta,\iota,\xi)$.*
3. *$\mathtt{shn}(E) \geq n/4m$ for every $E \in \mathcal{B}_n^m(\delta,\iota,\xi)$.*

The second and third properties say that $\widehat{\Phi}_m$ decomposes each term into sufficiently many contexts of appropriate size.

4.3 Explosive Context

Here, we show that each $\widehat{U}_{E.i}^m(\delta,\iota,\xi)$ contains at least one context that has a very long reduction sequence. To this end, we first prepare a special context $Expl_k^m$ that has a long reduction sequence, and shows that at least one element of $\widehat{U}_{E.i}^m(\delta,\iota,\xi)$ contains $Expl_k^m$ as a subcontext.

We define a "duplicating term" $Dup \triangleq \lambda x^{\mathtt{o}}.(\lambda x^{\mathtt{o}}.\lambda *^{\mathtt{o}}.x)xx$, and $Id \triangleq \lambda x^{\mathtt{o}}.x$. For two terms t, t' and integer $n \geq 1$, we define the "n-fold application" operation $\uparrow^n$ as $t \uparrow^0 t' \triangleq t'$ and $t \uparrow^n t' \triangleq t(t \uparrow^{n-1} t')$. For an integer $k \geq 2$, we define an order-k term

$$\overline{2}_k \triangleq \lambda f^{\tau(k)\to\tau(k)}.\lambda x^{\tau(k)}.f(fx)$$

where $\tau(i)$ is defined by $\tau(2) \triangleq \mathtt{o}$ and $\tau(i+1) \triangleq \tau(i)\to\tau(i)$.

Definition 5. (explosive context). Let $m \geq 1$ and $k \geq 2$ be integers and let

$$t \triangleq \nu\left(\lambda x^{\mathtt{o}}.\left((\overline{2}_k \uparrow^m \overline{2}_{k-1})\overline{2}_{k-2}\cdots\overline{2}_2\, Dup(Id\, x^\dagger)\right)\right)$$

where $x^\dagger$ is just variable x but we put $\dagger$ to refer to the occurrence. We define the *explosive context* $Expl_m^k$ (of m-fold and order k) as the 1-context obtained by replacing the "normalized" variable ${x_1}^\dagger$ in t with $[\,]$.

We state key properties of $Expl_m^k$ below. The proof of Item 3 is the same as that in [1]. The other items follow by straightforward calculation.

Lemma 3 (explosive).

1. *$\emptyset \vdash Expl_m^k[x_1] : \mathtt{o}\to\mathtt{o}$ is derivable.*
2. *$|Expl_m^k| = 8m + 8k - 3$.*
3. *$\mathtt{ord}(Expl_m^k[x_1]) = k$, $\mathtt{iar}(Expl_m^k[x_1]) = k$ and $\#(\mathbf{V}(Expl_m^k)) = 2$.*
4. *$Expl_m^k \in \mathcal{U}^\nu(\delta,\iota,\xi)$ if δ, $\iota \geq k$ and $\xi \geq 2$.*
5. *If a term t satisfies $Expl_m^k \preceq t$, then $\beta(t) \geq \mathbf{exp}_k(m)$ holds.*

We show that at least one element of $\widehat{U}_{E.i}^m(\delta,\iota,\xi)$ contains $Expl_{m'}^k$ as a subcontext.

Lemma 4 *Let $\delta, \iota, \xi \geq 2$ be integers and $k = \min\{\delta, \iota\}$. There exist integers $b, c \geq 2$ such that, for any $n \geq 1$, $m' \geq b$, $E \in \mathcal{B}_n^{cm'}(\delta, \iota, \xi)$ and $i \in \{1, \ldots, \mathtt{shn}(E)\}$, $\widehat{U}_{E.i}^{cm'}(\delta, \iota, \xi)$ contains u' such that $Expl_{m'}^k \preceq u'$.*

Proof Sketch. We pick $u'' \in \widehat{U}_{E.i}^{cm'}(\delta, \iota, \xi)$ and construct u' by replacing some subcontext u_0 of u'' with a 0/1-context of the form $S^\circ[Expl_{m'}^k[u^\circ]]$. Here S° and u° adjust the context typing and size of $Expl_{m'}^k$ and these can be obtained by using Proposition 2. The subcontext u_0 is chosen so that the goodness of u'' is preserved by this replacement. □

4.4 Proof Sketch of Theorem 1

We are now ready to prove the main theorem; see the full version [17] for details. For readability, we omit the parameters (δ, ι, ξ), and write $\Lambda_n^\alpha, \mathcal{U}_n^\nu, \Lambda_E^m, \widehat{U}_{E.i}^m$ and $\mathcal{B}_n^m$ for $\Lambda_n^\alpha(\delta, \iota, \xi), \mathcal{U}_n^\nu(\delta, \iota, \xi), \Lambda_E^m(\delta, \iota, \xi), \widehat{U}_{E.i}^m(\delta, \iota, \xi)$ and $\mathcal{B}_n^m(\delta, \iota, \xi)$ respectively.

Let $\overline{p}(n)$ be the probability that a randomly chosen normalized term t in Λ_n^α does *not* contain $Expl_{\lceil \log^{(2)}(n) \rceil}^k$ as a subcontext. By Item 3 of Lemma 3, it suffices to show $\lim_{n\to\infty} \overline{p}(n) = 0$. Let b and c be the constants in Lemma 4 and let $n \geq 2^{2^b}$, $m' = \lceil \log^{(2)}(n) \rceil$ and $m = cm'$. Then $m' \geq \log^{(2)}(n) \geq b$.

By Lemma 2, Λ_n^α can be represented as the disjoint union $\uplus_{E \in \mathcal{B}_n^m} \Lambda_E^m$. Let $\overline{\Lambda}_E^m$ be the subset of Λ_E^m that does not contain $Expl_{m'}^k$ as a subcontext. By Lemma 4, each of $\widehat{U}_{E.i}^m$ contains at least one element that has $Expl_{m'}^k$ as a subcontext. Furthermore, since $m \leq |E.i| < 2m$, we have $\#\left(\widehat{U}_{E.i}^m\right) \leq \#\left(\mathcal{U}_{2m+d}^\nu\right)$ for some constant d (see the full version [17]). Thus, we have

$$\frac{\#\left(\overline{\Lambda}_E^m\right)}{\#\left(\Lambda_E^m\right)} \leq \prod_{1 \leq i \leq \mathtt{shn}(E)} \left(1 - \frac{1}{\#\left(\widehat{U}_{E.i}^m\right)}\right) \leq \left(1 - \frac{1}{\#\left(\mathcal{U}_{2m+d}^\nu\right)}\right)^{\mathtt{shn}(E)}$$
$$\leq \left(1 - \frac{1}{\#\left(\mathcal{U}_{2m+d}^\nu\right)}\right)^{\frac{n}{4m}} \quad (\because \text{Item 3 of Lemma 2}).$$

Let $q(n)$ be the rightmost expression. Then we have

$$\overline{p}(n) = \frac{\sum_{E \in \mathcal{B}_n^m} \#\left(\overline{\Lambda}_E^m\right)}{\sum_{E \in \mathcal{B}_n^m} \#\left(\Lambda_E^m\right)} \leq \frac{\sum_{E \in \mathcal{B}_n^m} \left(q(n) \#\left(\Lambda_E^m\right)\right)}{\sum_{E \in \mathcal{B}_n^m} \#\left(\Lambda_E^m\right)}$$
$$= \frac{q(n) \sum_{E \in \mathcal{B}_n^m} \#\left(\Lambda_E^m\right)}{\sum_{E \in \mathcal{B}_n^m} \#\left(\Lambda_E^m\right)} = q(n) \leq \left(1 - \frac{1}{c' \overline{\gamma}(\delta, \iota, \xi)^{2m}}\right)^{\frac{n}{4m}} \quad (\because \text{Lemma 1})$$

for sufficiently large n. Finally, we can conclude that

$$\overline{p}(n) \leq \left(1 - \frac{1}{c' \overline{\gamma}(\delta, \iota, \xi)^{2c\lceil \log^{(2)}(n) \rceil}}\right)^{\frac{n}{4c\lceil \log^{(2)}(n) \rceil}} \longrightarrow 0 \quad (\text{as } n \longrightarrow \infty)$$

(see the full version [17] for the last convergence) as required. □

5 Related Work

As mentioned in Sect. 1, there are several pieces of work on probabilistic properties of untyped λ-terms [2–4]. David et al. [2] have shown that almost all untyped λ-terms are strongly normalizing, whereas the result is opposite for terms expressed in SK combinators.

Their former result implies that untyped λ-terms do *not* satisfy the infinite monkey theorem, i.e., for any term t, the probability that a randomly chosen term of size n contains t as a subterm tends to *zero*. Bendkowski et al. [4] proved that almost all terms in de Brujin representation are not strongly normalizing, by regarding the size of an index i is $i + 1$, instead of the constant 1. The discrepancies among those results suggest that this kind of probabilistic property is quite fragile and depends on the definition of the syntax and the size of terms. Thus, the setting of our paper, especially the assumption on the boundedness of internal arities and the number of variables is a matter of debate, and it would be interesting to study how the result changes for different assumptions.

We are not aware of similar studies on *typed* λ-terms. In fact, in their paper about combinatorial aspects of λ-terms, Grygiel and Lescanne [3] pointed out that the combinatorial study of typed λ-terms is difficult, due to the lack of (simple) recursive definition of typed terms. In the present paper, we have avoided the difficulty by making the assumption on the boundedness of internal arities and the number of variables (which is, as mentioned above, subject to a debate though).

In a larger context, our work may be viewed as an instance of the studies of average-case complexity ([20], Chap. 10), which discusses "typical-case feasibility". We are not aware of much work on the average-case complexity of problems with hyper-exponential complexity.

6 Conclusion

We have shown that almost every simply-typed λ-term of order k has a β-reduction sequence as long as $(k-2)$-fold exponential in the term size, under a certain assumption. To our knowledge, this is the first result of this kind for typed λ-terms. A lot of questions are left for future work, such as (i) whether our assumption (on the boundness of arities and the number of variables) is reasonable, and how the result changes for different assumptions, (ii) whether our result is optimal (e.g., whether almost every term has a k-fold exponentially long reduction sequence), and (iii) whether similar results hold for Terui's decision problems [14] and/or the higher-order model checking problem [6].

Acknowledgment. We would like to thank anonymous referees for useful comments. This work was supported by JSPS KAKENHI Grant Number JP15H05706.

References

1. Beckmann, A.: Exact bounds for lengths of reductions in typed lambda-calculus. J. Symb. Logic **66**(3), 1277–1285 (2001)
2. David, R., Grygiel, K., Kozic, J., Raffalli, C., Theyssier, G., Zaionc, M.: Asymptotically almost all λ-terms are strongly normalizing. Logical Method Comput. Sci. 9(2) (2013)
3. Grygiel, K., Lescanne, P.: Counting and generating lambda terms. J. Funct. Program. **23**(05), 594–628 (2013)
4. Bendkowski, M., Grygiel, K., Lescanne, P., Zaionc, M.: A natural counting of lambda terms. In: Freivalds, R.M., Engels, G., Catania, B. (eds.) SOFSEM 2016. LNCS, vol. 9587, pp. 183–194. Springer, Heidelberg (2016). doi:10.1007/978-3-662-49192-8_15
5. Knapik, T., Niwiński, D., Urzyczyn, P.: Higher-order pushdown trees are easy. In: Nielsen, M., Engberg, U. (eds.) FoSSaCS 2002. LNCS, vol. 2303, pp. 205–222. Springer, Heidelberg (2002). doi:10.1007/3-540-45931-6_15
6. Ong, C.H.L.: On model-checking trees generated by higher-order recursion schemes. In: LICS 2006, pp. 81–90. IEEE Computer Society Press (2006)
7. Kobayashi, N.: Model-checking higher-order functions. In: Proceedings of PPDP 2009, pp. 25–36. ACM Press (2009)
8. Broadbent, C.H., Kobayashi, N.: Saturation-based model checking of higher-order recursion schemes. In: Proceedings of CSL 2013, vol. 23, pp. 129–148. LIPIcs (2013)
9. Ramsay, S., Neatherway, R., Ong, C.H.L.: An abstraction refinement approach to higher-order model checking. In: Proceedings of POpPL 2014 (2014)
10. Kobayashi, N.: Types and higher-order recursion schemes for verification of higher-order programs. ACM SIGPLAN Not. **44**, 416–428 (2009). ACM Press
11. Kobayashi, N., Sato, R., Unno, H.: Predicate abstraction and CEGAR for higher-order model checking. ACM SIGPLAN Not. **46**, 222–233 (2011). ACM Press
12. Ong, C.H.L., Ramsay, S.: Verifying higher-order programs with pattern-matching algebraic data types. ACM SIGPLAN Not. **46**, 587–598 (2011). ACM Press
13. Sato, R., Unno, H., Kobayashi, N.: Towards a scalable software model checker for higher-order programs. In: Proceedings of PEpPM 2013, pp. 53–62. ACM Press (2013)
14. Terui, K.: Semantic evaluation, intersection types and complexity of simply typed lambda calculus. In: 23rd International Conference on Rewriting Techniques and Applications (RTA 2012), vol. 15, pp. 323–338. LIPIcs, Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2012)
15. Mairson, H.G.: Deciding ML typability is complete for deterministic exponential time. In: POPL, pp. 382–401. ACM Press (1990)
16. Kfoury, A.J., Tiuryn, J., Urzyczyn, P.: ML typability is dexptime-complete. In: Arnold, A. (ed.) CAAP 1990. LNCS, vol. 431, pp. 206–220. Springer, Heidelberg (1990). doi:10.1007/3-540-52590-4_50
17. Sin'ya, R., Asada, K., Kobayashi, N., Tsukada, T.: Almost every simply typed λ-term has a long β-reduction sequence (full version). http://www-kb.is.s.u-tokyo.ac.jp/ryoma/papers/fossacs17full.pdf
18. Heintze, N., McAllester, D.: Linear-time subtransitive control flow analysis. ACM SIGPLAN Not. **32**, 261–272 (1997)
19. Flajolet, P., Sedgewick, R.: Analytic Combinatorics, 1st edn. Cambridge University Press, New York (2009)
20. Goldreich, O.: Computational Complexity: A Conceptual Perspective. Cambridge University Press, Cambridge (2008)