

ALMOST EVERY SIMPLY TYPED λ -TERM HAS A LONG β -REDUCTION SEQUENCE

KAZUYUKI ASADA, NAOKI KOBAYASHI, RYOMA SIN'YA, AND TAKESHI TSUKADA

The University of Tokyo, Japan
e-mail address: asada@kb.is.s.u-tokyo.ac.jp

The University of Tokyo, Japan
e-mail address: koba@is.s.u-tokyo.ac.jp

Akita University, Japan
e-mail address: ryoma@math.akita-u.ac.jp

The University of Tokyo, Japan
e-mail address: tsukada@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 $(k - 1)$ -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. To prove it, we have extended the infinite monkey theorem for words to a parameterized one for regular tree languages, which may be of independent interest. The work has been motivated by quantitative analysis of the complexity of higher-order model checking.

1. INTRODUCTION

It is well known that 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]:

$$(\bar{2}_k)^n \bar{2}_{k-1} \cdots \bar{2}_2 (\lambda x^\circ . a x x) ((\lambda x^\circ . x) c),$$

where $\bar{2}_j$ is the twice function $\lambda f^{\tau^{(j-1)}} . \lambda x^{\tau^{(j-2)}} . f(f x)$ (with $\tau^{(j)}$ being the order- j type defined by: $\tau^{(0)} = \circ$ and $\tau^{(j)} = \tau^{(j-1)} \rightarrow \tau^{(j-1)}$), 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 (such as the 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 β -reduction sequence as long as $(k-1)$ -fold exponential in the term size, under a certain assumption. More precisely, we shall show:

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

for some constant $p > 0$, where $\Lambda_n(k, \iota, \xi)$ 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).

Related problems have been studied in the context of the quantitative analysis of *untyped* λ -terms [2, 3, 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.

To prove our main result above, we have extended the infinite monkey theorem (a.k.a. ‘‘Borges’s theorem’’ [5, p.61, Note I.35]) to a parameterized version for regular tree languages. The infinite monkey theorem states that for any word w , a sufficiently long word almost surely contains w as a subword (see Section 2 for a more precise statement). Our extended theorem, roughly speaking, states that, for any regular tree grammar \mathcal{G} that satisfies a certain condition and any *family* $(U_m)_{m \in \mathbb{N}}$ of *trees* (or tree contexts) generated by \mathcal{G} such that $|U_m| = O(m)$, a sufficiently large tree T generated by \mathcal{G} almost surely contains $U_{\lceil p \log |T| \rceil}$ as a subtree (where p is some positive constant). Our main result is then obtained by preparing a regular tree grammar for simply-typed λ -terms, and using as U_m a term having a very long β -reduction sequence, like $(\bar{2}_k)^m \bar{2}_{k-1} \cdots \bar{2}_2 (\lambda x^\circ. a x x) ((\lambda x^\circ. x) c)$ given above. The extended infinite monkey theorem mentioned above may be of independent interest and applicable to other problems.

Our work is a part of our long-term project on the quantitative analysis of the complexity of higher-order model checking [6, 7]. 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 [7]. Despite the huge worst-case complexity, practical model checkers [8, 9, 10] have been built, which run fast for many typical inputs, and have successfully been applied to automated verification of functional programs [11, 12, 13, 14]. 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 [15]: 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 [15] 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 [16, 17], it is often said that the exponential behavior is rarely seen *in practice*. That is, however, based on only empirical studies. A variation of our technique may be used to provide a theoretical justification (or possibly *unjustification*).

A preliminary version of this article appeared in Proceedings of FoSSaCS 2017 [18]. Compared with the conference version, we have strengthened the main result (from “almost every λ -term of size n has β -reduction sequence as long as $\exp_{k-2}(n)$ ” to “... as long as $\exp_{k-1}(n^p)$ ”), and added proofs. We have also generalized the argument in the conference version to formalize the parameterized infinite monkey theorem for regular tree languages.

The rest of this paper is organized as follows. Section 2 states our main result formally. Section 3 proves the extended infinite monkey theorem, and Section 4 proves the main result. Section 5 discusses related work, and Section 6 concludes this article.

2. MAIN RESULTS

This section states the main results of this article: the result on the quantitative analysis of the length of β -reduction sequences of simply-typed λ -terms (Section 2.1) and the parameterized infinite monkey theorem for regular tree grammars (Section 2.2). The latter is used in the proof of the former in Section 4. In the course of giving the main results, we also introduce various notations used in later sections; the notations are summarized in Appendix A. We assume some familiarity with the simply-typed λ -calculus and regular tree grammars and omit to define some standard concepts (such as β -reduction); readers who are not familiar with them may wish to consult [19] about the simply-typed λ -calculus and [20, 21] about regular tree grammars.

For a set A , we denote by $\#(A)$ the cardinality of A ; by abuse of notation, we write $\#(A) = \infty$ to mean that A is infinite. For a sequence s , we also denote by $\#(s)$ the length of s . For a sequence $s = a_1 a_2 \cdots a_{\#(s)}$ and $i \leq \#(s)$, we write $s.i$ for the i -th element a_i . We write $s_1 \cdot s_2$ or just $s_1 s_2$ for the concatenation of two sequences s_1 and s_2 . We write ϵ for the empty sequence. We use \uplus to denote the union of disjoint sets. For a set I and a family of sets $(A_i)_{i \in I}$, we define $\coprod_{i \in I} A_i \triangleq \uplus_{i \in I} (\{i\} \times A_i) = \{(i, a) \mid i \in I, a \in A_i\}$. For a map f , we denote the domain and image of f by $\text{Dom}(f)$ and $\text{Im}(f)$, respectively. We denote by $\log x$ the binary logarithm $\log_2 x$.

2.1. The Main Result on Quantitative Analysis of the Length of β -Reductions.

The set of (*simple*) *types*, ranged over by $\tau, \tau_1, \tau_2, \dots$, is defined by the grammar:

$$\tau ::= \mathbf{o} \mid \tau_1 \rightarrow \tau_2.$$

Note that the set of types can also be generated by the following grammar:

$$\tau ::= \tau_1 \rightarrow \cdots \rightarrow \tau_k \rightarrow \mathbf{o}$$

where $k \geq 0$; we sometimes use the latter grammar for inductive definitions of functions on types. We also use σ as a metavariable for types.

Remark 2.1. We have only a single base type \mathbf{o} above. The main result (Theorem 2.5) would not change even if there are multiple base types.

Let V be a countably infinite set, which is ranged over by x, y, z (and x', x_1 , etc.). The set of λ -terms (or *terms*), ranged over by t, t_1, t_2, \dots , is defined by:

$$t ::= x \mid \lambda \bar{x}^\tau. t \mid t_1 t_2 \quad \bar{x} ::= x \mid *.$$

We call elements of $V \cup \{*\}$ *variables*, and use meta-variables $\bar{x}, \bar{y}, \bar{z}$ (and \bar{x}', \bar{x}_1 , etc.) for them. We sometimes omit type annotations and just write $\lambda \bar{x}. t$ for $\lambda \bar{x}^\tau. t$. We call the special variable $*$ an *unused variable*, which may be bound by λ , but must not occur in the body. In our quantitative analysis below, we will count the number of variable names occurring in a term, except $*$. For example, the term $\lambda x. \lambda y. \lambda z. x$ in the standard syntax can be represented by $\lambda x. \lambda *. \lambda *. x$, and the number of used variables in the latter is counted as 1.

Terms of our syntax can be translated to usual λ -terms by regarding elements in $V \cup \{*\}$ as usual variables. Through this identification we define the notions of free variables, closed terms, and α -equivalence \sim_α . The α -equivalence class of a term t is written as $[t]_\alpha$. In this article, we distinguish between a term and its 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:

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

Note that neither $\mathbf{V}(t)$ nor even $\#(\mathbf{V}(t))$ is preserved by α -equivalence. For example, $t = \lambda x. (\lambda y. y)(\lambda z. x)$ and $t' = \lambda x. (\lambda x. x)(\lambda *. x)$ are α -equivalent, but $\#(\mathbf{V}(t)) = 3$ and $\#(\mathbf{V}(t')) = 1$. We write $\#\mathbf{vars}(t)$ for $\min_{t' \in [t]_\alpha} \#(\mathbf{V}(t'))$, i.e., the minimum number of variables required to represent an α -equivalent term. For example, for $t = \lambda x. (\lambda y. y)(\lambda z. x)$ above, $\#\mathbf{vars}(t) = 1$, because $t \sim_\alpha t'$ 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. When we write $\Gamma_1 \cup \Gamma_2$, we implicitly require that $(x : \tau) \in \Gamma_1$ and $(x : \tau') \in \Gamma_2$ imply $\tau = \tau'$, so that $\Gamma_1 \cup \Gamma_2$ is well formed. Note that $(* : \tau)$ cannot belong to a type environment; we do not need any type assumption for $*$ since it does not occur in terms.

The type judgment relation $\Gamma \vdash t : \tau$ is inductively defined by the following typing rules.

$$\frac{}{x : \tau \vdash x : \tau} \quad \frac{\Gamma_1 \vdash t_1 : \sigma \rightarrow \tau \quad \Gamma_2 \vdash t_2 : \sigma}{\Gamma_1 \cup \Gamma_2 \vdash t_1 t_2 : \tau}$$

$$\frac{\Gamma' \vdash t : \tau \quad \Gamma' = \Gamma \text{ or } \Gamma' = \Gamma \cup \{\bar{x} : \sigma\} \quad \bar{x} \notin \text{Dom}(\Gamma)}{\Gamma \vdash \lambda \bar{x}^\sigma. t : \sigma \rightarrow \tau}$$

The type judgment relation is equivalent to the usual one for the simply-typed λ -calculus, except that a type environment for a term may contain only variables that occur free in the term (i.e., if $\Gamma \vdash t : \tau$ then $\text{Dom}(\Gamma) = \mathbf{FV}(t)$). Note that if $\Gamma \vdash t : \tau$ is derivable, then

the derivation is unique. Below we consider only well-typed λ -terms, i.e., those that are inhabitants of some typings.

Definition 2.2 (size, order and internal arity of a term). The *size* of a term t , written $|t|$, is defined by:

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

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

$$\begin{aligned} \text{ord}(\tau_1 \rightarrow \cdots \rightarrow \tau_n \rightarrow \mathbf{o}) &\triangleq \max(\{0\} \cup \{\text{ord}(\tau_i) + 1 \mid 1 \leq i \leq n\}) \\ \text{iar}(\tau_1 \rightarrow \cdots \rightarrow \tau_n \rightarrow \mathbf{o}) &\triangleq \max(\{n\} \cup \{\text{iar}(\tau_i) \mid 1 \leq i \leq n\}) \end{aligned}$$

where $n \geq 0$. We denote by $\text{Types}(\delta, \iota)$ the set of types $\{\tau \mid \text{ord}(\tau) \leq \delta, \text{iar}(\tau) \leq \iota\}$. For a judgment $\Gamma \vdash t : \tau$, we define the *order* and *internal arity* of $\Gamma \vdash t : \tau$, written $\text{ord}(\Gamma \vdash t : \tau)$ and $\text{iar}(\Gamma \vdash t : \tau)$ respectively, by:

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

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

Note that the notions of size, order, internal arity, and $\beta(t)$ (the maximum length of β -reduction sequences of t , as defined in Section 1) are well-defined with respect to α -equivalence.

Example 2.3. Recall the term $\bar{2}_j \triangleq \lambda f^{\tau^{(j-1)}}. \lambda x^{\tau^{(j-2)}}. f(fx)$ (with $\tau^{(j)}$ being the order- j type defined by: $\tau^{(0)} = \mathbf{o}$ and $\tau^{(j)} = \tau^{(j-1)} \rightarrow \tau^{(j-1)}$) in Section 1. For $t = \bar{2}_3 \bar{2}_2 (\lambda x^\mathbf{o}. x) y$, we have $\text{ord}(y : \mathbf{o} \vdash t : \mathbf{o}) = \text{iar}(y : \mathbf{o} \vdash t : \mathbf{o}) = 3$. Note that the derivation for $y : \mathbf{o} \vdash t : \mathbf{o}$ contains the type judgment $\emptyset \vdash \bar{2}_3 : \tau^{(3)}$, where $\tau^{(3)} = ((\mathbf{o} \rightarrow \mathbf{o}) \rightarrow (\mathbf{o} \rightarrow \mathbf{o})) \rightarrow (\mathbf{o} \rightarrow \mathbf{o}) \rightarrow \mathbf{o} \rightarrow \mathbf{o}$, and $\text{ord}(\tau^{(3)}) = \text{iar}(\tau^{(3)}) = 3$.

We now define the sets of (α -equivalence classes of) terms with bounds on the order, the internal arity, and the number of variables.

Definition 2.4 (terms with bounds on types and variables). Let $\delta, \iota, \xi \geq 0$ and $n \geq 1$ be integers. For each Γ and τ , $\Lambda(\langle \Gamma; \tau \rangle, \delta, \iota, \xi)$ and $\Lambda_n(\langle \Gamma; \tau \rangle, \delta, \iota, \xi)$ are defined by:

$$\begin{aligned} \Lambda(\langle \Gamma; \tau \rangle, \delta, \iota, \xi) &\triangleq \{[t]_\alpha \mid \Gamma \vdash t : \tau, \text{ord}(\Gamma \vdash t : \tau) \leq \delta, \text{iar}(\Gamma \vdash t : \tau) \leq \iota, \# \text{vars}(t) \leq \xi\} \\ \Lambda_n(\langle \Gamma; \tau \rangle, \delta, \iota, \xi) &\triangleq \{[t]_\alpha \in \Lambda(\langle \Gamma; \tau \rangle, \delta, \iota, \xi) \mid |t| = n\}. \end{aligned}$$

We also define:

$$\Lambda(\delta, \iota, \xi) \triangleq \bigsqcup_{\tau \in \text{Types}(\delta, \iota)} \Lambda(\langle \emptyset; \tau \rangle, \delta, \iota, \xi) \quad \Lambda_n(\delta, \iota, \xi) \triangleq \{[t]_\alpha \in \Lambda(\delta, \iota, \xi) \mid |t| = n\}.$$

Intuitively, $\Lambda(\langle \Gamma; \tau \rangle, \delta, \iota, \xi)$ is the set of (the equivalence classes of) terms of type τ under Γ , within given bounds δ , ι , and ξ on the order, (internal) arity, and the number of variables respectively; and $\Lambda_n(\langle \Gamma; \tau \rangle, \delta, \iota, \xi)$ is the subset of $\Lambda(\langle \Gamma; \tau \rangle, \delta, \iota, \xi)$ consisting of terms of size n . The set $\Lambda(\delta, \iota, \xi)$ consists of (the equivalence classes of) all the closed (well-typed) terms, and $\Lambda_n(\delta, \iota, \xi)$ consists of those of size n . For example, $t = [\lambda x^\mathbf{o}. \lambda y^\mathbf{o}. \lambda z^\mathbf{o}. x]_\alpha$ belongs to $\Lambda_4(1, 3, 1)$; note that $|t| = 4$ and $\# \text{vars}(t) = \# \text{vars}(\lambda x^\mathbf{o}. \lambda y^\mathbf{o}. \lambda z^\mathbf{o}. x) = 1$.

We are now ready to state our main result, which is proved in Section 4.

Theorem 2.5. *For $\delta, \iota, \xi \geq 2$ and $k = \min\{\delta, \iota\}$, there exists a real number $p > 0$ such that*

$$\lim_{n \rightarrow \infty} \frac{\#\{[t]_\alpha \in \Lambda_n(\delta, \iota, \xi) \mid \beta(t) \geq \mathbf{exp}_{k-1}(n^p)\}}{\#(\Lambda_n(\delta, \iota, \xi))} = 1.$$

As we will see later (in Section 4), the denominator $\#(\Lambda_n(\delta, \iota, \xi))$ is nonzero if n is sufficiently large. The theorem above says that if the order, internal arity, and the number of used variables are bounded independently of term size, most of the simply-typed λ -terms of size n have a very long β -reduction sequence, which is as long as $(k - 1)$ -fold exponential in n .

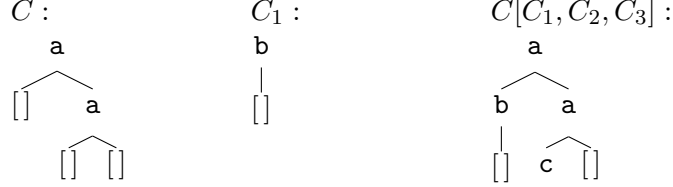
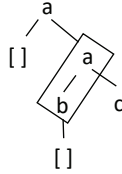
Remark 2.6. Recall that, when $k = \delta$, the worst-case length of β -reduction sequence is k -fold exponential [1]. We do not know whether $k - 1$ can be replaced with k in the theorem above.

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* . Our proof of the theorem (given in Section 4) makes use of this assumption to model the set of simply-typed λ -terms as a regular tree language. It is debatable whether our assumption is reasonable. A slight change of the assumption may change the result, as is the case for strong normalization of untyped λ -terms [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 [22]. 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.

2.2. Regular Tree Grammars and Parameterized Infinite Monkey Theorem. To prove Theorem 2.5 above, we extend the well-known *infinite monkey theorem* (a.k.a. “Borges’s theorem” [5, p.61, Note I.35]) to a parameterized version for regular tree grammars, and apply it to the regular tree grammar that generates the set of (α -equivalence classes of) simply-typed λ -terms. Since the extended infinite monkey theorem may be of independent interest, we state it (as Theorem 2.13) in this section as one of the main results. The theorem is proved in Section 3. We first recall some basic definitions for regular tree grammars in Sections 2.2.1 and 2.2.2, and then state the parameterized infinite monkey theorem in Section 2.2.3.

2.2.1. Trees and Tree Contexts. A *ranked alphabet* Σ is a map from a finite set of *terminal* symbols to the set of natural numbers. We use the metavariable a for a terminal, and often write $\mathbf{a}, \mathbf{b}, \mathbf{c}, \dots$ for concrete terminal symbols. For a terminal $a \in \text{Dom}(\Sigma)$, we call $\Sigma(a)$ the *rank* of a . A Σ -*tree* is a tree constructed from terminals in Σ according to their ranks: $a(T_1, \dots, T_{\Sigma(a)})$ is a Σ -tree if T_i is a Σ -tree for each $i \in \{1, \dots, \Sigma(a)\}$. Note that $\Sigma(a)$ may be 0: $a()$ is a Σ -tree if $\Sigma(a) = 0$. We often write just a for $a()$. For example, if $\Sigma = \{\mathbf{a} \mapsto 2, \mathbf{b} \mapsto 1, \mathbf{c} \mapsto 0\}$, then $\mathbf{a}(\mathbf{b}(\mathbf{c}), \mathbf{c})$ is a tree; see (the lefthand side of) Figure 1 for a graphical illustration of the tree. We use the meta-variable T for trees. The *size* of T , written $|T|$, is the number of occurrences of terminals in T . We denote the set of all Σ -trees by $\mathcal{T}(\Sigma)$, and the set of all Σ -trees of size n by $\mathcal{T}_n(\Sigma)$.

Before introducing grammars, we define tree contexts, which play an important role in our formalization and proof of the (parameterized) infinite monkey theorem for regular tree

FIGURE 1. A tree $a(b(c), c)$ (left) and a context $a(a([], c), b([]))$ (right).FIGURE 2. Context filling (where $C_2 = c$ and $C_3 = []$).FIGURE 3. A graphical illustration of a subcontext. The part surrounded by the rectangle is the subcontext $a(b([], []))$.

languages. The set of *contexts* over a ranked alphabet Σ , ranged over by C , is defined by:

$$C ::= [] \mid a(C_1, \dots, C_{\Sigma(a)}).$$

In other words, a context is a tree where the alphabet is extended with the special nullary symbol $[]$. For example, $a(a([], c), b([]))$ is a context over $\{a \mapsto 2, b \mapsto 1, c \mapsto 0\}$; see (the righthand side of) Figure 1 for a graphical illustration. We write $\mathcal{C}(\Sigma)$ for the set of contexts over Σ . For a context C , we write $\mathbf{hn}(C)$ for the number of occurrences of $[]$ in C . We call C a k -context if $\mathbf{hn}(C) = k$, and an *affine context* if $\mathbf{hn}(C) \leq 1$. A 1-context is also called a *linear context*. We use the metavariable S for linear contexts and U for affine contexts. We write $[]_i$ for the i -th hole occurrence (in the left-to-right order) of C .

For contexts $C, C_1, \dots, C_{\mathbf{hn}(C)}$, we write $C[C_1, \dots, C_{\mathbf{hn}(C)}]$ for the context obtained by replacing each $[]_i$ in C with C_i . For example, if $C = a([], a([], []))$, $C_1 = b([])$, $C_2 = c$, and $C_3 = []$, then $C[C_1, C_2, C_3] = a(b([], a(c, []))$; see Figure 2 for a graphical illustration. Also, for contexts C, C' and $i \in \{1, \dots, \mathbf{hn}(C)\}$, we write $C[C']_i$ for the context obtained by replacing $[]_i$ in C with C' . For example, for C and C_1 above, $C[C_1]_2 = a([], a(b([], []))$.

For contexts C and C' , we say that C is a *subcontext* of C' and write $C \preceq C'$ if there exist $C_0, C_1, \dots, C_{\mathbf{hn}(C)}$, and $1 \leq i \leq \mathbf{hn}(C_0)$ such that $C' = C_0[C[C_1, \dots, C_{\mathbf{hn}(C)}]]_i$. For example, $C = a(b([], []))$ is a subcontext of $C' = a([], a(b([], c)))$, because $C' = C_0[C[], c]_2$ for $C_0 = a([], [])$; see also Figure 3 for a graphical illustration. The subcontext relation \preceq may be regarded as a generalization of the subword relation. In fact, a word $w = a_1 \cdots a_k$ can be viewed as a linear context $w^\# = a_1(\cdots(a_k[]) \cdots)$; and if w is a subword of w' , i.e., if $w' = w_1 w w_2$, then $w^\# = w_1^\# [w^\# [w_2^\#]]$. The *size* of a context U , written $|U|$, is defined by:

$||| \triangleq 0$ and $|a(C_1, \dots, C_{\Sigma(a)})| \triangleq 1 + |C_1| + \dots + |C_{\Sigma(a)}|$. Note that $|||$ and 0-ary terminal a have different sizes: $||| = 0$ but $|a| = 1$. For a 0-context C , $|C|$ coincides with the size of C as a tree.

2.2.2. Tree Grammars. A *regular tree grammar* [20, 21] (*grammar* for short) 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 symbols called *nonterminals*; (iii) \mathcal{R} is a finite set of *rewriting rules* of the form $N \longrightarrow C[N_1, \dots, N_{\text{hn}(C)}]$ where $C \in \mathcal{C}(\Sigma)$ and $N, N_1, \dots, N_{\text{hn}(C)} \in \mathcal{N}$. We use the metavariable N for nonterminals. We write $\Sigma \cup \mathcal{N}$ for the ranked alphabet $\Sigma \cup \{N \mapsto 0 \mid N \in \mathcal{N}\}$ and often regard the right-hand-side of a rule as a $(\Sigma \cup \mathcal{N})$ -tree.

The rewriting relation $\longrightarrow_{\mathcal{G}}$ on $\mathcal{T}(\Sigma \cup \mathcal{N})$ is inductively defined by the following rules:

$$\frac{(N \longrightarrow C[N_1, \dots, N_{\text{hn}(C)}]) \in \mathcal{R}}{N \longrightarrow_{\mathcal{G}} C[N_1, \dots, N_{\text{hn}(C)}]} \quad \frac{T_i \longrightarrow_{\mathcal{G}} T'_i}{a(T_1, \dots, T_k) \longrightarrow_{\mathcal{G}} a(T_1, \dots, T_{i-1}, T'_i, T_{i+1}, \dots, T_k)}$$

We write $\longrightarrow_{\mathcal{G}}^*$ for the reflexive and transitive closure of $\longrightarrow_{\mathcal{G}}$. For a tree grammar $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ and a nonterminal $N \in \mathcal{N}$, the *language* $\mathcal{L}(\mathcal{G}, N)$ of N is defined by $\mathcal{L}(\mathcal{G}, N) \triangleq \{T \in \mathcal{T}(\Sigma) \mid N \longrightarrow_{\mathcal{G}}^* T\}$. We also define $\mathcal{L}_n(\mathcal{G}, N) \triangleq \{T \in \mathcal{L}(\mathcal{G}, N) \mid |T| = n\}$. If \mathcal{G} is clear from the context, we often omit \mathcal{G} and just write $T \longrightarrow^* T'$, $\mathcal{L}(N)$, and $\mathcal{L}_n(N)$ for $T \longrightarrow_{\mathcal{G}}^* T'$, $\mathcal{L}(\mathcal{G}, N)$, and $\mathcal{L}_n(\mathcal{G}, N)$, respectively.

We are interested in, not only trees, but also contexts generated by a grammar. Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be a grammar. The set of *context types* of \mathcal{G} , ranged over by κ , is defined by:

$$\kappa ::= N_1 \cdots N_k \Rightarrow N$$

where $k \in \mathbb{N}$ and $N, N_1, \dots, N_k \in \mathcal{N}$. Intuitively, $N_1 \cdots N_k \Rightarrow N$ denotes the type of k -contexts C such that $N \longrightarrow^* C[N_1, \dots, N_k]$. Based on this intuition, we define the sets $\mathcal{L}(\mathcal{G}, \kappa)$ and $\mathcal{L}_n(\mathcal{G}, \kappa)$ of contexts by:

$$\begin{aligned} \mathcal{L}(\mathcal{G}, N_1 \cdots N_k \Rightarrow N) &\triangleq \{C \in \mathcal{C}(\Sigma) \mid \text{hn}(C) = k, N \longrightarrow_{\mathcal{G}}^* C[N_1, \dots, N_k]\} \\ \mathcal{L}_n(\mathcal{G}, \kappa) &\triangleq \{C \in \mathcal{L}(\mathcal{G}, \kappa) \mid |C| = n\}. \end{aligned}$$

We also write $C : \kappa$ when $C \in \mathcal{L}(\mathcal{G}, \kappa)$ (assuming that \mathcal{G} is clear from the context). Note that $\mathcal{L}(\mathcal{G}, \Rightarrow N) = \mathcal{L}(\mathcal{G}, N)$, and we identify a context type $\Rightarrow N$ with the nonterminal N . We call an element in $\mathcal{L}(\mathcal{G}, \kappa)$ a κ -*context*, and also a κ -*tree* if it is a tree. It is clear that, if $C \in \mathcal{L}(\mathcal{G}, N_1 \cdots N_k \Rightarrow N)$ and $C_i \in \mathcal{L}(\mathcal{G}, N_1^i \cdots N_{\ell_i}^i \Rightarrow N_i)$ ($i = 1, \dots, k$), then $C[C_1, \dots, C_k] \in \mathcal{L}(\mathcal{G}, N_1^1 \cdots N_{\ell_1}^1 \cdots N_1^k \cdots N_{\ell_k}^k \Rightarrow N)$. We also define:

$$\begin{aligned} \mathcal{L}(\mathcal{G}) &\triangleq \bigcup_{N \in \mathcal{N}} \mathcal{L}(\mathcal{G}, N) & \mathcal{L}_n(\mathcal{G}) &\triangleq \{T \in \mathcal{L}(\mathcal{G}) \mid |T| = n\} \\ \mathcal{S}(\mathcal{G}) &\triangleq \bigcup_{N, N' \in \mathcal{N}} \mathcal{L}(\mathcal{G}, N \Rightarrow N') & \mathcal{S}_n(\mathcal{G}) &\triangleq \{S \in \mathcal{S}(\mathcal{G}) \mid |S| = n\} \\ \mathcal{U}(\mathcal{G}) &\triangleq \mathcal{L}(\mathcal{G}) \cup \mathcal{S}(\mathcal{G}) & \mathcal{U}_n(\mathcal{G}) &\triangleq \{U \in \mathcal{U}(\mathcal{G}) \mid |U| = n\}. \end{aligned}$$

Example 2.7. Consider the grammar $\mathcal{G}_0 = (\{\mathbf{a} \mapsto 2, \mathbf{b} \mapsto 1, \mathbf{c} \mapsto 0\}, \{A_0, B_0\}, \mathcal{R}_0)$, where \mathcal{R}_0 consists of:

$$A_0 \longrightarrow \mathbf{a}(B_0, B_0) \quad A_0 \longrightarrow \mathbf{c} \quad B_0 \longrightarrow \mathbf{b}(A_0).$$

Then, $\mathbf{a}([], []) \in \mathcal{L}_1(\mathcal{G}_0, B_0 \Rightarrow A_0)$ and $\mathbf{a}(\mathbf{b}([]), \mathbf{b}(c)) \in \mathcal{L}_4(\mathcal{G}_0, A_0 \Rightarrow A_0)$.

This article focuses on grammars that additionally satisfy two properties called *strong connectivity* and *unambiguity*. We first define strong connectivity.

Definition 2.8 (Strong Connectivity [5]¹). Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be a regular tree grammar. We say that N' is *reachable* from N if $\mathcal{L}(\mathcal{G}, N' \Rightarrow N)$ is non-empty, i.e., if there exists a linear context $S \in \mathcal{C}(\Sigma)$ such that $N \longrightarrow^* S[N']$. We say that \mathcal{G} is *strongly connected* if for any pair $N, N' \in \mathcal{N}$ of nonterminals, N' is reachable from N .

Remark 2.9. There is another reasonable, but slightly weaker condition of reachability: N' is *reachable* from N if there exists a $(\Sigma \cup \mathcal{N})$ -tree T such that $N \longrightarrow^* T$ and N' occurs in T . The two notions coincide if $\mathcal{L}(\mathcal{G}, N) \neq \emptyset$ for every $N \in \mathcal{N}$. Furthermore this condition can be easily fulfilled by simply removing all the nonterminals N with $\mathcal{L}(\mathcal{G}, N) = \emptyset$ and the rules containing those nonterminals.

Example 2.10. The grammar \mathcal{G}_0 in Example 2.7 is strongly connected, since $\mathbf{b}([]) \in \mathcal{L}(\mathcal{G}_0, A_0 \Rightarrow B_0)$ and $\mathbf{a}([], \mathbf{b}(c)) \in \mathcal{L}(\mathcal{G}_0, B_0 \Rightarrow A_0)$.

Next, consider the grammar $\mathcal{G}_1 = (\{\mathbf{a} \mapsto 2, \mathbf{b} \mapsto 1, \mathbf{c} \mapsto 0\}, \{A_1, B_1\}, \mathcal{R}_1)$, where \mathcal{R}_1 consists of:

$$A_1 \longrightarrow \mathbf{a}(c, A_1) \quad A_1 \longrightarrow \mathbf{b}(B_1) \quad B_1 \longrightarrow \mathbf{b}(B_1) \quad B_1 \longrightarrow c.$$

\mathcal{G}_1 is not strongly connected, since $\mathcal{L}(\mathcal{G}_1, A_1 \Rightarrow B_1) = \emptyset$.

To define the unambiguity of a grammar, we define (the standard notion of) leftmost rewriting. The *leftmost rewriting relation* $\longrightarrow_{\mathcal{G}, \ell}$ is the restriction of the rewriting relation that allows only the leftmost occurrence of nonterminals to be rewritten. Given $T \in \mathcal{T}(\Sigma)$ and $N \in \mathcal{N}$, a *leftmost rewriting sequence of T from N* is a sequence T_1, \dots, T_n of $(\Sigma \cup \mathcal{N})$ -trees such that

$$N = T_1 \longrightarrow_{\mathcal{G}, \ell} T_2 \longrightarrow_{\mathcal{G}, \ell} \dots \longrightarrow_{\mathcal{G}, \ell} T_n = T.$$

It is easy to see that $T \in \mathcal{L}(\mathcal{G}, N)$ if and only if there exists a leftmost rewriting sequence from N to T .

Definition 2.11 (Unambiguity). A grammar $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ is *unambiguous* if, for each $N \in \mathcal{N}$ and $T \in \mathcal{T}(\Sigma)$, there exists at most one leftmost rewriting sequence from N to T .

2.2.3. Parameterized Infinite Monkey Theorem for Tree Grammars. We call a partial function f from \mathbb{N} to \mathbb{R} with infinite definitional domain (i.e., $\text{Dom}(f) = \{n \in \mathbb{N} \mid f(n) \text{ is defined}\}$ is infinite) a *partial sequence*, and write f_n for $f(n)$. For a partial sequence $f = (f_n)_{n \in \mathbb{N}}$, we define $\text{def}_{(-)}^f$ as the bijective monotonic function from \mathbb{N} to $\text{Dom}(f)$, and then we define:

$$\lim_{n \rightarrow \infty} \text{def } f_n \triangleq \lim_{n \rightarrow \infty} f_{\text{def}_n^f}.$$

All the uses of $\lim_{n \rightarrow \infty} \text{def}$ in this article are of the form: $\lim_{n \rightarrow \infty} \text{def } \frac{b_n}{a_n}$ where $(a_n)_{n \in \mathbb{N}}$ and $(b_n)_{n \in \mathbb{N}}$ are (non-partial) sequences (hence, $\frac{b_n}{a_n}$ is undefined only when $a_n = 0$). The following property on $\lim_{n \rightarrow \infty} \text{def}$ is straightforward, which we leave for the reader to check.

¹ In [5], strong connectivity is defined for *context-free specifications*. Our definition is a straightforward adaptation of the definition for regular tree grammars.

Lemma 2.12. *If $\limdef_{n \rightarrow \infty} \frac{b_n^{(i)}}{a_n^{(i)}} = 1$ for $i = 1, \dots, k$, and if $0 \leq b_n^{(i)} \leq a_n^{(i)}$ for $i = 1, \dots, k$ and for $n \in \mathbb{N}$, then $\limdef_{n \rightarrow \infty} \frac{\sum_i b_n^{(i)}}{\sum_i a_n^{(i)}} = 1$.*

Now we are ready to state the second main result of this article.

Theorem 2.13 (Parameterized Infinite Monkey Theorem for Regular Tree Languages). *Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be an unambiguous and strongly-connected regular tree grammar such that $\#(\mathcal{L}(\mathcal{G})) = \infty$, and $(S_n)_{n \in \mathbb{N}}$ be a family of linear contexts in $\mathcal{S}(\mathcal{G})$ such that $|S_n| = O(n)$. Then there exists a real constant $p > 0$ such that for any $N \in \mathcal{N}$ the following equation holds:*

$$\limdef_{n \rightarrow \infty} \frac{\#(\{T \in \mathcal{L}_n(\mathcal{G}, N) \mid S_{\lfloor p \log n \rfloor} \preceq T\})}{\#(\mathcal{L}_n(\mathcal{G}, N))} = 1.$$

Intuitively, the equation above says that if n is large enough, almost all the trees of size n generated from N contain $S_{\lfloor p \log n \rfloor}$ as a subcontext.

Remark 2.14. The standard infinite monkey theorem says that for any finite word $w = a_1 \cdots a_k \in A^*$, the probability that a randomly chosen word of length n contains w as a subword converges to 1, i.e.,

$$\lim_{n \rightarrow \infty} \frac{\#(\{w' \in A^n \mid \exists w_0, w_1 \in A^*. w' = w_0 w w_1\})}{\#(A^n)} = 1.$$

This may be viewed as a special case of our parameterized infinite monkey theorem above, where (i) the components of the grammar are given by $\Sigma = \{a \mapsto 1 \mid a \in A\} \cup \{e \mapsto 0\}$, $\mathcal{N} = \{N\}$, and $\mathcal{R} = \{N \rightarrow a(N) \mid a \in A\} \cup \{N \rightarrow e\}$, and (ii) $S_n = a_1(\cdots a_k([\])\cdots)$ independently of n .

Note that in the theorem above, we have used \limdef rather than \lim . To avoid the use of \limdef , we need an additional condition on \mathcal{G} , called aperiodicity.

Definition 2.15 (Aperiodicity [5]). Let \mathcal{G} be a grammar. For a nonterminal N , \mathcal{G} is called *N -aperiodic* if there exists n_0 such that $\#(\mathcal{L}_n(N)) > 0$ for any $n \geq n_0$. Further, \mathcal{G} is called *aperiodic* if \mathcal{G} is N -aperiodic for any nonterminal N .

In Theorem 2.13, if we further assume that \mathcal{G} is N -aperiodic, then $\limdef_{n \rightarrow \infty}$ in the statement can be replaced with $\lim_{n \rightarrow \infty}$.

In the rest of this section, we reformulate the theorem above in the form more convenient for proving Theorem 2.5. In Definition 2.4, $\Lambda_n(\delta, \iota, \xi)$ was defined as the disjoint union of $(\Lambda_n(\langle \emptyset; \tau \rangle, \delta, \iota, \xi))_{\tau \in \text{Types}(\delta, \iota)}$. To prove Theorem 2.5, we will construct a grammar \mathcal{G} so that, for each τ , $\Lambda_n(\langle \emptyset; \tau \rangle, \delta, \iota, \xi) \cong \mathcal{L}(\mathcal{G}, N)$ holds for some nonterminal N . Thus, the following style of statement is more convenient, which is obtained as an immediate corollary of Theorem 2.13 and Lemma 2.12.

Corollary 2.16. *Let \mathcal{G} be an unambiguous and strongly-connected regular tree grammar such that $\#(\mathcal{L}(\mathcal{G})) = \infty$, and $(S_n)_{n \in \mathbb{N}}$ be a family of linear contexts in $\mathcal{S}(\mathcal{G})$ such that $|S_n| = O(n)$. Then there exists a real constant $p > 0$ such that for any non-empty set \mathcal{I} of nonterminals of \mathcal{G} the following equation holds:*

$$\limdef_{n \rightarrow \infty} \frac{\#(\{(N, T) \in \coprod_{N \in \mathcal{I}} \mathcal{L}_n(\mathcal{G}, N) \mid S_{\lfloor p \log n \rfloor} \preceq T\})}{\#(\coprod_{N \in \mathcal{I}} \mathcal{L}_n(\mathcal{G}, N))} = 1.$$

We can also replace a family of linear contexts $(S_n)_n$ above with that of trees $(T_n)_n$. We need only this corollary in the proof of Theorem 2.5.

Corollary 2.17. *Let \mathcal{G} be an unambiguous and strongly-connected regular tree grammar such that $\#(\mathcal{L}(\mathcal{G})) = \infty$ and there exists $C \in \cup_{\kappa} \mathcal{L}(\mathcal{G}, \kappa)$ with $\mathbf{hn}(C) \geq 2$. Also, let $(T_n)_{n \in \mathbb{N}}$ be a family of trees in $\mathcal{L}(\mathcal{G})$ such that $|T_n| = O(n)$. Then there exists a real constant $p > 0$ such that for any non-empty set \mathcal{I} of nonterminals of \mathcal{G} the following equation holds:*

$$\lim_{n \rightarrow \infty} \frac{\#(\{(N, T) \in \coprod_{N \in \mathcal{I}} \mathcal{L}_n(\mathcal{G}, N) \mid T_{\lceil p \log n \rceil} \preceq T\})}{\#(\coprod_{N \in \mathcal{I}} \mathcal{L}_n(\mathcal{G}, N))} = 1.$$

Proof. Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$. Let $C \in \mathcal{L}(\mathcal{G}, \kappa)$ be the context in the assumption and let κ be of the form $N'_1 \cdots N'_k \Rightarrow N'$, i.e., $N' \rightarrow^* C[N'_1, \dots, N'_k]$, where $k = \mathbf{hn}(C) \geq 2$. By Corollary 2.16, it suffices to construct a family of linear contexts $(S_n)_n$ such that (i) $S_n \in \mathcal{S}(\mathcal{G})$, (ii) $T_n \preceq S_n$, and (iii) $|S_n| = O(n)$. For each n , there exists N_n such that $N_n \rightarrow^* T_n$. Since $\#(\mathcal{L}(\mathcal{G})) = \infty$ and by the strong connectivity, for each $i \leq k$ there exists T'_i such that $N'_i \rightarrow^* T'_i$. For each n , by the strong connectivity, there exists S'_n such that $N'_1 \rightarrow^* S'_n[N_n]$, and since \mathcal{N} is finite, we can choose S'_n so that the size is bounded above by a constant (that is independent of n). Let S_n be $C[S'_n[T_n], [], T'_3, \dots, T'_k]$. Then (i) $S_n \in \mathcal{L}(\mathcal{G}, N'_2 \Rightarrow N') \subseteq \mathcal{S}(\mathcal{G})$ because $N' \rightarrow^* C[N'_1, \dots, N'_k] \rightarrow^* C[S'_n[T_n], N'_2, T'_3, \dots, T'_k]$, (ii) $T_n \preceq S_n$, and (iii) $|S_n| = O(n)$, as required. \square

Remark 2.18. In each of Corollaries 2.16 and 2.17, if we further assume that \mathcal{G} is N -aperiodic for any $N \in \mathcal{I}$, then $\lim_{n \rightarrow \infty}$ in the statement can be replaced with $\lim_{n \rightarrow \infty}$.

Remark 2.19. On Theorem 2.5 (and similarly on Corollaries 2.16 and 2.17), one might be interested in the following form of probability:

$$\lim_{n \rightarrow \infty} \frac{\#(\uplus_{m \leq n} \{T \in \mathcal{L}_m(\mathcal{G}, N) \mid S_{\lceil p \log m \rceil} \preceq T\})}{\#(\uplus_{m \leq n} \mathcal{L}_m(\mathcal{G}, N))} = 1,$$

which discusses trees of size *at most* n rather than *exactly* n . Under the assumption of aperiodicity, the above equation follows from Theorem 2.5, by the following *Stolz-Cesàro theorem* [23, Theorem 1.22]: Let $(a_n)_{n \in \mathbb{N}}$ and $(b_n)_{n \in \mathbb{N}}$ be a sequence of real numbers and assume $b_n > 0$ and $\sum_{n=0}^{\infty} b_n = \infty$. Then for any $c \in [-\infty, +\infty]$,

$$\lim_{n \rightarrow \infty} \frac{a_n}{b_n} = c \quad \text{implies} \quad \lim_{n \rightarrow \infty} \frac{\sum_{m \leq n} a_m}{\sum_{m \leq n} b_m} = c.$$

(We also remark that the inverse implication also holds if there exists $\gamma \in \mathbb{R} \setminus \{1\}$ such that $\lim_{n \rightarrow \infty} \frac{\sum_{m \leq n} b_m}{\sum_{m \leq n+1} b_m} = \gamma$ [23, Theorem 1.23].)

3. PROOF OF THE PARAMETERIZED INFINITE MONKEY THEOREM FOR REGULAR TREE LANGUAGES (THEOREM 2.13)

Here we prove Theorem 2.13. In Section 3.1, we first prove a parameterized version² of infinite monkey theorem for *words*, and explain how the proof for the word case can be

²Although the parameterization is a simple extension, we are not aware of literature that explicitly states this parameterized version.

extended to deal with regular tree languages. The structure of the rest of the section is explained at the end of Section 3.1.

3.1. Proof for Word Case and Outline of this Section. 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 $|w|$ for n and call it the *size* 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 (recall Remark 2.14). The following theorem is a parameterized version, where w may depend on n . It may also be viewed as a special case of Theorem 2.13, where $\Sigma = \{a \mapsto 1 \mid a \in A\} \cup \{e\}$, $\mathcal{N} = \{N\}$, $\mathcal{R} = \{N \longrightarrow a(N) \mid a \in A\} \cup \{N \longrightarrow e\}$, and $S_n = a_1(\cdots a_k([\])\cdots)$ for each $w_n = a_1 \cdots a_k$.

Proposition 3.1 (Parameterized Infinite Monkey Theorem for Words). *Let A be an alphabet and $(w_n)_n$ be a family of words over A such that $|w_n| = O(n)$. Then, there exists a real constant $p > 0$ such that we have:*

$$\lim_{n \rightarrow \infty} \frac{\#(\{w \in A^n \mid w_{[p \log n]} \sqsubseteq w\})}{\#(A^n)} = 1.$$

Proof. Let Z_n be $1 - \#(\{w \in A^n \mid w_{[p \log n]} \sqsubseteq w\}) / \#(A^n)$, i.e., the probability that a word of size n does *not* contain $w_{[p \log n]}$. By the assumption $|w_n| = O(n)$, there exists $b > 0$ such that $|w_n| \leq bn$ for sufficiently large n . Let $0 < q < 1$ be an arbitrary real number, and we define $p \triangleq \frac{q}{2b \log \#(A)}$. We write $s(n)$ for $|w_{[p \log n]}|$ and $c(n)$ for $\lfloor n/s(n) \rfloor$. Let $n \in \mathbb{N}$. Given a word $w = a_1 \cdots a_n \in A^n$, let us decompose it to subwords of length $s(n)$ as follows.

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

Then,

$$\begin{aligned} 0 \leq Z_n &\leq \text{"the probability that none of the } i\text{-th subword is } w_{[p \log n]} \text{"} \\ &= \frac{\#((A^{s(n)} \setminus \{w_{[p \log n]}\})^{c(n)} A^{n-c(n)s(n)})}{\#(A^n)} \\ &= \left(\frac{\#(A^{s(n)} \setminus \{w_{[p \log n]}\})}{\#(A^{s(n)})} \right)^{c(n)} = \left(\frac{\#(A^{s(n)}) - 1}{\#(A^{s(n)})} \right)^{c(n)} = \left(1 - \frac{1}{\#(A)^{s(n)}} \right)^{c(n)} \end{aligned}$$

and we can show $\lim_{n \rightarrow \infty} \left(1 - \frac{1}{\#(A)^{s(n)}} \right)^{c(n)} = 0$ as follows. For sufficiently large n , we have $s(n) \leq b[p \log n] \leq 2bp \log n = q \frac{\log n}{\log \#(A)} = q \log_{\#(A)} n$, and hence

$$\left(1 - \frac{1}{\#(A)^{s(n)}} \right)^{c(n)} \leq \left(1 - \frac{1}{n^q} \right)^{c(n)} \leq \left(1 - \frac{1}{n^q} \right)^{\frac{n}{s(n)} - 1} \leq \left(1 - \frac{1}{n^q} \right)^{\frac{\log \#(A)}{q} \frac{n}{\log n} - 1}.$$

Also we have $\lim_{n \rightarrow \infty} \left(1 - \frac{1}{n^q} \right)^{\frac{n}{\log n}} = 0$ (which we leave for the reader to check); therefore $\lim_{n \rightarrow \infty} Z_n = 0$. \square

The key observations in the above proof were:

(W1) Each word $w = a_1 \cdots a_n$ can be decomposed to

$$(w'_{\text{rem}}, w'_1, \dots, w'_{c(n)}) \in A^{n-c(n)s(n)} \times \left(\prod_{i=1}^{c(n)} A^{s(n)} \right)$$

where $w'_{\text{rem}} = a_{c(n)s(n)+1} \cdots a_n$ and $w'_i = a_{(i-1)s(n)+1} \cdots a_{is(n)}$.

(W2) The word decomposition above induces the following decomposition of the set A^n of words of length n :

$$\begin{aligned} A^n &\cong A^{n-c(n)s(n)} \times \left(\prod_{i=1}^{c(n)} A^{s(n)} \right) \\ &\cong \coprod_{w \in A^{n-c(n)s(n)}} \prod_{i=1}^{c(n)} A^{s(n)} \end{aligned}$$

Here, \cong denotes the existence of a bijection.

(W3) Choose $s(n)$ and $c(n)$ so that (i) $A^{s(n)}$ contains at least one element that contains w_n as a subword and (ii) $c(n)$ is sufficiently large. By condition (i), the probability that an element of A^n does not contain w as a subword can be bounded above the probability that none of w'_i ($i \in \{1, \dots, c(n)\}$) contains w_n as a subword, i.e., $(1 - \frac{1}{\#(A^{s(n)})})^{c(n)}$, which converges to 0 by condition (ii).

The proof of Theorem 2.13 in the rest of this section is based on similar observations:

(T1) Each tree T of size n can be decomposed to

$$(E, U_1, \dots, U_{c_E}),$$

where U_1, \dots, U_{c_E} are (affine) subcontexts, E , called a *second-order context* (which will be formally defined later), is the “remainder” of T obtained by extracting U_1, \dots, U_{c_E} , and c_E is a number that depends on E . For example, the tree on the lefthand side of Figure 4 can be decomposed to the second-order context and affine contexts shown on the righthand side. By substituting each affine context for $\llbracket \rrbracket$ in the preorder traversal order, we recover the tree on the lefthand side. This decomposition of a tree may be regarded as a generalization of the word decomposition above, (by viewing a word as an affine context), where the part E corresponds to the remainder $a_{c(n)s(n)+1} \cdots a_n$ of the word decomposition.

(T2) The tree decomposition above induces the decomposition of the set $\mathcal{L}_n(\mathcal{G}, N)$ in the following form:

$$\mathcal{L}_n(\mathcal{G}, N) \cong \prod_{E \in \mathcal{E}} \prod_{j=1}^{c_E} \mathcal{U}_{E,j} \quad (3.1)$$

where \mathcal{E} is a set of second-order contexts and $\mathcal{U}_{E,j}$ is a set of affine contexts. (At this point, the reader need not be concerned about the exact definitions of \mathcal{E} and $\mathcal{U}_{E,j}$, which will be given later.)

(T3) Design the above decomposition so that (i) each $\mathcal{U}_{E,j}$ contains at least one element that contains $S_{\lceil p \log n \rceil}$ as a subcontext, and (ii) c_E is sufficiently large. By condition (i), the probability that an element of $\mathcal{L}_n(\mathcal{G}, N)$ does not contain $S_{\lceil p \log n \rceil}$ as a subcontext is bounded above by the probability that none of U_i ($i \in \{1, \dots, c_E\}$) contain $S_{\lceil p \log n \rceil}$ as

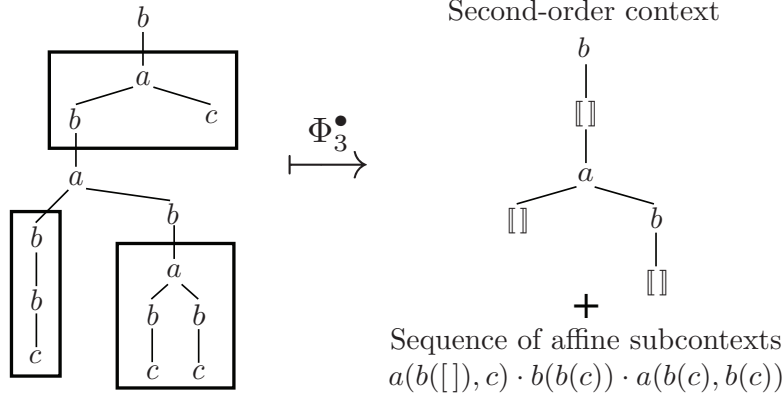


FIGURE 4. An example of tree decomposition. The parts surrounded by rectangles on the lefthand side show the extracted affine contexts, and the remaining part of the tree is the second-order tree context.

a subcontext, which is further bounded above by $\max_{E \in \mathcal{E}} \left(\prod_{j=1}^{c_E} \left(1 - \frac{1}{\#(\mathcal{U}_{E,j})} \right) \right)$. The bound can be proved to converge to 0 by using condition (ii).

The rest of the section is organized as follows. Before we jump to the decomposition of $\mathcal{L}_n(\mathcal{G}, N)$, we first present a decomposition of $\mathcal{T}_n(\Sigma)$, i.e., the set of Σ -trees of size n in Section 3.2. The decomposition of $\mathcal{T}_n(\Sigma)$ may be a special case of that of $\mathcal{L}_n(\mathcal{G}, N)$, where \mathcal{G} generates all the trees in $\mathcal{T}_n(\Sigma)$. For a technical convenience in extending the decomposition of $\mathcal{T}_n(\Sigma)$ to that of $\mathcal{L}_n(\mathcal{G}, N)$, we normalize grammars to *canonical form* in Section 3.3. We then give the decomposition of $\mathcal{L}_n(\mathcal{G}, N)$ given in Equation (3.1) above, and prove that it satisfies the required properties in Sections 3.4 and 3.5. Finally, we prove Theorem 2.13 in Section 3.6.

3.2. Grammar-Independent Tree Decomposition. In this subsection, we will define a decomposition function Φ_m^\bullet (where m is a parameter) that decomposes a tree T into (i) a (sufficiently long) sequence $P = U_1 \cdots U_k$ consisting of affine subcontexts of size no less than m , and (ii) a “second-order” context E (defined shortly in Section 3.2.1), which is the remainder of extracting P from T . Recall Figure 4, which illustrates how a tree is decomposed by Φ_3^\bullet . Here, the symbols $[]$ in the second-order context on the right-hand side represents the original position of each subcontext. By filling the i -th occurrence (counted in the depth-first, left-to-right pre-order) of $[]$ with the i -th affine context, we can recover the original tree on the left hand side.

As formally stated later (in Corollary 3.8), the decomposition function Φ_m^\bullet provides a witness for a bijection of the form

$$\mathcal{T}_n(\Sigma) \cong \prod_{E \in \mathcal{E}} \prod_{j=1}^{c_E} \mathcal{U}_{E,j},$$

a special case of Equation (3.1) mentioned in Section 3.1.

3.2.1. *Second-Order Contexts.* We first define the notion of second-order contexts and operations on them.

The set of *second-order contexts* over Σ , ranged over by E , is defined by:

$$E ::= \llbracket \rrbracket_k^n[E_1, \dots, E_k] \mid a(E_1, \dots, E_{\Sigma(a)}) \quad (a \in \text{Dom}(\Sigma)).$$

Intuitively, the second-order context is an expression having holes of the form $\llbracket \rrbracket_k^n$ (called *second-order holes*), which should be filled with a k -context of size n . By filling all the second-order holes, we obtain a Σ -tree. Note that k may be 0. In the technical development below, we only consider second-order holes $\llbracket \rrbracket_k^n$ such that k is 0 or 1. We write $\text{shn}(E)$ for the number of the second-order holes in E . Note that Σ -trees can be regarded as second-order contexts E such that $\text{shn}(E) = 0$, and vice versa. For $i \leq \text{shn}(E)$, we write $E.i$ for the i -th second-order hole (counted in the depth-first, left-to-right pre-order). We define the *size* $|E|$ by: $|\llbracket \rrbracket_k^n[E_1, \dots, E_k]| \triangleq n + |E_1| + \dots + |E_k|$ and $|a(E_1, \dots, E_{\Sigma(a)})| \triangleq |E_1| + \dots + |E_k| + 1$. Note that $|E|$ includes the size of contexts to fill the second-order holes in E .

Example 3.2. The second-order context on the right hand side of Figure 4 is expressed as $E = \mathbf{b}(\llbracket \rrbracket_1^3[\mathbf{a}(\llbracket \rrbracket_0^3, \mathbf{b}(\llbracket \rrbracket_0^5))])$, where $\text{shn}(E) = 3$, $|E| = 14$, $E.1 = \llbracket \rrbracket_1^3$, $E.2 = \llbracket \rrbracket_0^3$, and $E.3 = \llbracket \rrbracket_0^5$.

Next we define the substitution operation on second-order contexts. For a context C and a second-order hole $\llbracket \rrbracket_k^n$, we write $C : \llbracket \rrbracket_k^n$ if C is a k -context of size n . Given E and C such that $\text{shn}(E) \geq 1$ and $C : E.1$, we write $E[C]$ for the second-order context obtained by replacing the leftmost second-order hole of E (i.e., $E.1$) with C (and by interpreting the syntactical bracket $[-]$ as the substitution operation). Formally, it is defined by induction on E as follows:

$$\begin{aligned} (\llbracket \rrbracket_k^n[E_1, \dots, E_k]) [C] &\triangleq C[E_1, \dots, E_k] \\ (a(E_1, \dots, E_{\Sigma(a)})) [C] &\triangleq a(E_1, \dots, E_i[C], \dots, E_{\Sigma(a)}) \\ &\quad \text{where } i = \min\{j \mid \text{shn}(E_j) \geq 1, 1 \leq j \leq \Sigma(a)\}. \end{aligned}$$

In the first clause, $C[E_1, \dots, E_k]$ is the second-order context obtained by replacing $\llbracket \rrbracket_i$ in C with E_i for each $i \leq k$. Note that we have $|E[C]| = |E|$ whenever $E[C]$ is well-defined, i.e., if $C : E.1$ (cf. Lemma 3.4 below).

We extend the substitution operation for a sequence of contexts. We use metavariable P for sequences of contexts. For E and $P = C_1 C_2 \dots C_{\text{shn}(E)}$, we write $P : E$ if $C_i : E.i$ for each $i \leq \text{shn}(E)$. Given E and a sequence of contexts $P = C_1 C_2 \dots C_\ell$ such that $\ell \leq \text{shn}(E)$ and $C_i : E.i$ for each $i \leq \ell$, we define $E[P]$ by induction on P :

$$E[\epsilon] \triangleq E \qquad E[C \cdot P] \triangleq (E[C])[P]$$

Note that $\text{shn}(E[C]) = \text{shn}(E) - 1$, so if $P : E$ then $E[P]$ is a tree.

Example 3.3. Recall the second-order context $E = \mathbf{b}(\llbracket \rrbracket_1^3[\mathbf{a}(\llbracket \rrbracket_0^3, \mathbf{b}(\llbracket \rrbracket_0^5))])$ in Figure 4 and Example 3.2. Let P be the sequence of affine contexts given in Figure 4:

$$\mathbf{a}(\mathbf{b}(\llbracket \rrbracket), c) \cdot \mathbf{b}(\mathbf{b}(c)) \cdot \mathbf{a}(\mathbf{b}(c), \mathbf{b}(c)).$$

Then

$$E[P.1] = \mathbf{b}(P.1[\mathbf{a}(\llbracket \rrbracket_0^3, \mathbf{b}(\llbracket \rrbracket_0^5))]) = \mathbf{b}(\mathbf{a}(\mathbf{b}(\mathbf{a}(\llbracket \rrbracket_0^3, \mathbf{b}(\llbracket \rrbracket_0^5))), c)),$$

and

$$\begin{aligned} E[P] &= ((E[P.1])[P.2])[P.3] \\ &= \mathbf{b}(\mathbf{a}(\mathbf{b}(\mathbf{a}(P.2, \mathbf{b}(P.3))), c)) = \mathbf{b}(\mathbf{a}(\mathbf{b}(\mathbf{a}(\mathbf{b}(\mathbf{b}(c))), \mathbf{b}(\mathbf{a}(\mathbf{b}(c), \mathbf{b}(c)))))), c)), \end{aligned}$$

which is the tree shown on the left hand side of Figure 4.

Thanks to the size annotation, the substitution operation preserves the size of a second-order context.

Lemma 3.4. *Let E be a second-order context and P be a sequence of contexts such that $P : E$. Then $|E| = |E[P]|$.*

Proof. The proof is given by a straightforward induction on E . In the case $E = \llbracket \rrbracket_k^n[E_1, \dots, E_k]$, we use the fact $|C[E_1, \dots, E_k]| = |C| + |E_1| + \dots + |E_k|$ (which can be shown by induction on C). \square

3.2.2. Grammar-Independent Decomposition Function. Now we define the decomposition function Φ_m^\bullet . Let Σ be a ranked alphabet and $m \geq 1$. We define $r_\Sigma \triangleq \max(\text{Im}(\Sigma))$, which is also written as r for short. We always assume that $r \geq 1$, since it holds whenever $\#(\mathcal{L}(\mathcal{G})) = \infty$, which is an assumption of Theorem 2.13. We shall define the decomposition function $\Phi_m^{\bullet, \Sigma}$ (we omit Σ and just write Φ_m^\bullet for $\Phi_m^{\bullet, \Sigma}$ below) so that $\Phi_m^\bullet(T) = (E, P)$ where (i) E is the second-order context, and (ii) P is a sequence of affine contexts, (iii) $E[P] = T$, and (iv) $m \leq |P \cdot i| \leq r(m-1) + 1$ for each $i \in \{1, \dots, \#(P)\}$.

The function Φ_m^\bullet is defined as follows, using an auxiliary decomposition function Φ_m given below.

$$\Phi_m^\bullet(T) \triangleq (U[E], P) \text{ where } (U, E, P) = \Phi_m(T).$$

The auxiliary decomposition function (just called “decomposition function” below) Φ_m traverses a given tree T in a bottom-up manner, extracts a sequence P of subcontexts, and returns it along with a linear context U and a second-order context E ; U and E together represent the “remainder” of extracting P from T . During the bottom-up traversal, the U -component for each subtree T' represents a context containing the root of T' ; whether it is extracted as (a part of) a subcontext or becomes a part of the remainder will be decided later based on the surrounding context. The E -component will stay as a part of the remainder during the decomposition of the whole tree (unless the current subtree T' is too small, i.e., $|T'| < m$).

We define Φ_m by:

- If $|T| < m$, then $\Phi_m(T) \triangleq ([], T, \epsilon)$.
- If $|T| \geq m$, $T = a(T_1, \dots, T_{\Sigma(a)})$, and $\Phi_m(T_i) = (U_i, E_i, P_i)$ (for each $i \leq \Sigma(a)$), then:

$$\Phi_m(T) \triangleq \begin{cases} ([], a(U_1[E_1], \dots, U_{\Sigma(a)}[E_{\Sigma(a)}]), P_1 \cdots P_{\Sigma(a)}) & \text{if there exist } i, j \text{ such that } 1 \leq i < j \leq \Sigma(a) \text{ and } |T_i|, |T_j| \geq m \\ ([], \llbracket \rrbracket_1^n[E_i], a(T_1, \dots, U_i, \dots, T_{\Sigma(a)}) \cdot P_i) & \text{if } |T_j| < m \text{ for every } j \neq i, |T_i| \geq m, \text{ and} \\ & n \triangleq |a(T_1, \dots, U_i, \dots, T_{\Sigma(a)})| \geq m \\ (a(T_1, \dots, U_i, \dots, T_{\Sigma(a)}), E_i, P_i) & \text{if } |T_j| < m \text{ for every } j \neq i, |T_i| \geq m, \text{ and} \\ & |a(T_1, \dots, U_i, \dots, T_{\Sigma(a)})| < m \\ ([], T, \epsilon) & \text{if } |T_i| < m \text{ for every } i \leq \Sigma(a), \text{ and } n \triangleq |T| \end{cases} \quad (3.2)$$

Proof. (1)–(3) follow by straightforward simultaneous induction on $|T|$. (4) follows from (1) and Lemma 3.4. \square

The following lemma ensures that $\mathbf{shn}(E)$ is sufficiently large whenever $\Phi_m^\bullet(T) = (E, P)$. Recall condition (ii) of (T3) in Section 3.1; $\mathbf{shn}(E)$ corresponds to c_E .

Lemma 3.7 (Decomposition produces sufficiently many affine contexts). *For any tree T and m such that $1 \leq m \leq |T|$, if $\Phi_m(T) = (U, E, P)$, then*

$$\#(P) \geq \frac{|T|}{2rm}.$$

Proof. Recall that $r \triangleq \max(\text{Im}(\Sigma))$ and we assume $r \geq 1$. We show below that

$$|U| + 2rm\#(P) - rm \geq |T|$$

by induction on T . Then it follows that

$$2rm\#(P) \geq |T| + rm - |U| > |T| + rm - m \geq |T|$$

(where the second inequality uses Lemma 3.6(2)), which implies $\#(P) > \frac{|T|}{2rm}$ as required.

Since $|T| \geq m$, $\Phi_m(T)$ is computed by Equation (3.2), on which we perform a case analysis. Let $T = a(T_1, \dots, T_{\Sigma(a)})$ and $\Phi_m(T_i) = (U_i, E_i, P_i)$ ($i \leq \Sigma(a)$).

- The first case: In this case, we have

$$U = [] \quad P = P_{i_1} P_{i_2} \cdots P_{i_s}$$

where $\{i_1, \dots, i_s\} = \{i \leq \Sigma(a) \mid |T_i| \geq m\}$ and $s \geq 2$, since if $|T_i| < m$ then $P_i = \epsilon$. Note that we have $r \geq 2$ in this case. Then,

$$\begin{aligned} & |U| + 2rm\#(P) - rm \\ &= 2rm \left(\sum_{j \leq s} \#(P_{i_j}) \right) - rm \\ &\geq \left(\sum_{j \leq s} (|T_{i_j}| + rm - |U_{i_j}|) \right) - rm \quad (\because \text{by induction hypothesis for } T_{i_j}) \\ &\geq \left(\sum_{j \leq s} (|T_{i_j}| + rm - (m-1)) \right) - rm \quad (\because |U_{i_j}| \leq m-1 \text{ by Lemma 3.6(2)}) \\ &= \left(\sum_{j \leq s} |T_{i_j}| \right) + srm - sm + s - rm \\ &\geq \left(\sum_{j \leq s} |T_{i_j}| \right) + 2rm - sm + s - rm - r + 1 \quad (\because s \geq 2 \text{ and } r \geq 1) \\ &= \left(\sum_{j \leq s} |T_{i_j}| \right) + (r-s)(m-1) + 1 \\ &\geq \left(\sum_{j \leq s} |T_{i_j}| \right) + (\Sigma(a) - s)(m-1) + 1 \quad (\because r \geq \Sigma(a)) \\ &\geq \left(\sum_{j \leq s} |T_{i_j}| \right) + \left(\sum_{i \in \{1, \dots, \Sigma(a)\} \setminus \{i_1, \dots, i_s\}} |T_i| \right) + 1 \\ &\quad (\because m-1 \geq |T_i| \text{ for } i \in \{1, \dots, \Sigma(a)\} \setminus \{i_1, \dots, i_s\}) \\ &= |T| \end{aligned}$$

as required.

- The second case: In this case, we have

$$U = [] \quad P = a(T_1, \dots, U_i, \dots, T_{\Sigma(a)})P_i$$

and $|T_j| \geq m$ if and only if $j = i$. Also we have $r \geq 1$. Then,

$$\begin{aligned}
& |U| + 2rm\#(P) - rm \\
&= 2rm(1 + \#(P_i)) - rm \\
&= rm + 2rm\#(P_i) \\
&\geq rm + (|T_i| + rm - |U_i|) \quad (\because \text{by induction hypothesis for } T_i) \\
&\geq |T_i| + rm - (m - 1) \quad (\because rm \geq 0, |U_i| \leq m - 1 \text{ by Lemma 3.6(2)}) \\
&\geq |T_i| + rm - r - m + 2 \quad (\because r \geq 1) \\
&= |T_i| + (r - 1)(m - 1) + 1 \\
&\geq |T_i| + (\Sigma(a) - 1)(m - 1) + 1 \quad (\because r \geq \Sigma(a)) \\
&\geq |T_i| + (\sum_{j \in \{1, \dots, \Sigma(a)\} \setminus \{i\}} |T_j|) + 1 \\
&= |T|
\end{aligned}$$

as required.

- The third case of (3.2): In this case, we have

$$U = a(T_1, \dots, U_i, \dots, T_{\Sigma(a)}) \quad P = P_i$$

and $|T_j| \geq m$ if and only if $j = i$. Then,

$$\begin{aligned}
& |U| + 2rm\#(P) - rm \\
&= 1 + (\sum_{j \neq i} |T_j|) + |U_i| + 2rm\#(P_i) - rm \\
&\geq 1 + (\sum_{j \neq i} |T_j|) + |T_i| \quad (\because \text{by induction hypothesis for } T_i) \\
&= |T|
\end{aligned}$$

as required.

- The forth case of (3.2): In this case, we have

$$U = [] \quad P = T.$$

Then

$$\begin{aligned}
& |U| + 2rm\#(P) - rm \\
&= rm \\
&\geq r(m - 1) + 1 \quad (\because r \geq 1) \\
&\geq \Sigma(a)(m - 1) + 1 \quad (\because r \geq \Sigma(a)) \\
&\geq (\sum_{j \in \{1, \dots, \Sigma(a)\}} |T_j|) + 1 \\
&= |T|
\end{aligned}$$

as required. □

3.2.4. *Decomposition of $\mathcal{T}_n(\Sigma)$.* This subsection shows that the decomposition function Φ_m^\bullet above provides a witness for a bijection of the form

$$\mathcal{T}_n(\Sigma) \cong \coprod_{E \in \mathcal{E}} \prod_{j=1}^{c_E} \mathcal{U}_{E,j}.$$

We prepare some definitions to precisely state the bijection. We define the set \mathcal{E}_n^m of second-order contexts and the set $\mathcal{U}_{\llbracket k \rrbracket_n^m}^m$ of affine contexts by:

$$\begin{aligned}\mathcal{E}_n^m &\triangleq \{E \mid (E, P) = \Phi_m^\bullet(T) \text{ for some } T \in \mathcal{T}_n(\Sigma) \text{ and } P\} \\ \mathcal{U}_{\llbracket k \rrbracket_n^m}^m &\triangleq \{U \mid U : \llbracket k \rrbracket_n^m, U \text{ is good for } m\}.\end{aligned}$$

Intuitively, \mathcal{E}_n^m is the set of second-order contexts obtained by decomposing a tree of size n , and $\mathcal{U}_{\llbracket k \rrbracket_n^m}^m$ is the set of good contexts that match the second-order context $\llbracket k \rrbracket_n^m$.

The bijection is then stated as the following lemma.

Lemma 3.8.

$$\mathcal{T}_n(\Sigma) \cong \coprod_{E \in \mathcal{E}_n^m} \prod_{i=1}^{\text{shn}(E)} \mathcal{U}_{E.i}^m. \quad (3.3)$$

The rest of this subsection is devoted to a proof of the lemma above; readers may wish to skip the rest of this subsection upon the first reading.

Lemma 3.9. *If $E \in \mathcal{E}_n^m$, then $|E| = n$.*

Proof. Suppose $E \in \mathcal{E}_n^m$. Then $(E, P) = \Phi_m^\bullet(T)$ for some $T \in \mathcal{T}_n(\Sigma)$ and P . By Lemma 3.6(4), $|E| = |T| = n$. \square

For a second-order context E , and $m \geq 1$, we define the set \mathcal{P}_E^m of sequences of affine contexts:

$$\mathcal{P}_E^m \triangleq \{P \mid (E, P) = \Phi_m^\bullet(T) \text{ for some } T \in \mathcal{T}(\Sigma)\}.$$

The set \mathcal{P}_E^m consists of sequences P of affine contexts that match E and are obtained by the decomposition function Φ_m^\bullet . In the rest of this subsection, we prove the bijection in Lemma 3.8 in two steps. We first show $\mathcal{T}_n(\Sigma) \cong \coprod_{E \in \mathcal{E}_n^m} \mathcal{P}_E^m$ (Lemma 3.10, called ‘‘coproduct lemma’’), and then show $\mathcal{P}_E^m = \prod_{j=1}^{\text{shn}(E)} \mathcal{U}_{E.j}^m$ (Lemma 3.10, called ‘‘product lemma’’).

Lemma 3.10 (Coproduct Lemma (for Grammar-Independent Decomposition)). *For any $n \geq 1$ and $m \geq 1$, there exists a bijection*

$$\mathcal{T}_n(\Sigma) \cong \coprod_{E \in \mathcal{E}_n^m} \mathcal{P}_E^m$$

that maps each element (E, P) of the set $\coprod_{E \in \mathcal{E}_n^m} \mathcal{P}_E^m$ to $E[P] \in \mathcal{T}_n(\Sigma)$.

Proof. We define a function

$$f : \coprod_{E \in \mathcal{E}_n^m} \mathcal{P}_E^m \longrightarrow \mathcal{T}_n(\Sigma)$$

by $f(E, P) \triangleq E[P]$, and a function

$$g : \mathcal{T}_n(\Sigma) \longrightarrow \coprod_{E \in \mathcal{E}_n^m} \mathcal{P}_E^m$$

by $g(T) \triangleq \Phi_m^\bullet(T)$.

Let us check that these are functions into the codomains:

- $f(E, P) \in \mathcal{T}_n(\Sigma)$: Since $P \in \mathcal{P}_E^m$, there exists $T \in \mathcal{T}(\Sigma)$ such that $(E, P) = \Phi_m^\bullet(T)$. By Lemma 3.6(1), we have $f(E, P) = E[P] = T \in \mathcal{T}(\Sigma)$. By the condition $E \in \mathcal{E}_n^m$ and by Lemmas 3.4 and 3.9, $|E[P]| = |E| = n$. Thus, $f(E, P) \in \mathcal{T}_n(\Sigma)$ as required.

- $g(T) \in \coprod_{E \in \mathcal{E}_n^m} \mathcal{P}_E^m$: Obvious from the definitions of \mathcal{E}_n^m and \mathcal{P}_E^m .

We have $f(g(T)) = T$ by Lemmas 3.6(1). Let $(E, P) \in \coprod_{E \in \mathcal{E}_n^m} \mathcal{P}_E^m$. By definition, there exists $T \in \mathcal{T}(\Sigma)$ such that $(E, P) = \Phi_m^\bullet(T)$. By using Lemmas 3.4 and 3.9 again, we have $|T| = |E[P]| = |E| = n$. Thus, $(E, P) = g(T)$. Then

$$g(f(E, P)) = g(f(g(T))) = g(T) = (E, P).$$

□

It remains to show the product lemma: $\mathcal{P}_E^m = \prod_{j=1}^{\text{shn}(E)} \mathcal{U}_{E,j}^m$. To this end, we prove a few more properties about the auxiliary decomposition function Φ_m .

Lemma 3.11. *If $|T| \geq m$ and $\Phi_m(T) = (U, E, P)$, then $|E| \geq m$.*

Proof. Straightforward induction on T . □

The following lemma states that, given $\Phi_m(U[T]) = (U, E, P)$, P is determined only by T (U does not matter); this is because the decomposition is performed in a bottom-up manner.

Lemma 3.12. *For $m \geq 1$, E, P, T , and a linear context U with $|T| \geq m$ and $|U| < m$, we have $\Phi_m(T) = ([], E, P)$ if and only if $\Phi_m(U[T]) = (U, E, P)$.*

Proof. The proof proceeds by induction on $|U|$. If $U = []$ the claim trivially holds. If $U \neq []$, U is of the form $a(T_1, \dots, U_i, \dots, T_{\Sigma(a)})$. Since $|U| < m$, we have $|U_i| < m$ and $|T_j| < m$ for every $j \neq i$.

Assume that $\Phi_m(T) = ([], E, P)$. By the induction hypothesis, we have $\Phi_m(U_i[T]) = (U_i, E, P)$. Since $|U| = |a(T_1, \dots, T_{i-1}, U_i, T_{i+1}, \dots, T_{\Sigma(a)})| < m$, we should apply the third case of Equation (3.2) to compute $\Phi_m(U[T])$. Hence, we have

$$\Phi_m(U[T]) = (a(T_1, \dots, T_{i-1}, U_i, T_{i+1}, \dots, T_{\Sigma(a)}), E, P) = (U, E, P).$$

Conversely, assume that

$$\Phi_m(U[T]) = (U, E, P) = (a(T_1, \dots, T_{i-1}, U_i, T_{i+1}, \dots, T_{\Sigma(a)}), E, P).$$

Let $(U'_i, E'_i, P'_i) = \Phi_m(U_i[T])$ and $(U'_j, E'_j, P'_j) = \Phi_m(T_j)$ for each $j \neq i$. Then the final step in the computation of $\Phi_m(U[T])$ must be the third case; otherwise $U = []$, a contradiction. By the position of the unique hole in U , it must be the case that $U = a(T_1, \dots, T_{i-1}, U'_i, T_{i+1}, \dots, T_{\Sigma(a)})$, $E = E'_i$ and $P = P'_i$. So $(U_i, E, P) = \Phi_m(U_i[T])$. By the induction hypothesis, $\Phi_m(T) = ([], E, P)$. □

The following is the key lemma for the product lemma, which says that if $(U, E, P) = \Phi_m(T)$, the decomposition is actually independent of the P -part.

Lemma 3.13. *If $(U, E, P) = \Phi_m(T)$, then $\Phi_m(U[E][P']) = (U, E, P')$ for any $P' \in \prod_{i=1}^{\text{shn}(E)} \mathcal{U}_{E,i}^m$.*

Proof. The proof proceeds by induction on $|T| (= |U[E]|)$. If $|T| < m$, then $E = T$, hence $\text{shn}(E) = 0$. Thus, $P = P' = \epsilon$, which implies $\Phi_m(U[E][P']) = \Phi_m(T) = (U, E, P) = (U, E, P')$.

For the case $|T| \geq m$, we proceed by the case analysis on which rule of Equation (3.2) was used to compute $(U, E, P) = \Phi_m(T)$. Assume that $T = a(T_1, \dots, T_{\Sigma(a)})$ and $(U_i, E_i, P_i) = \Phi_m(T_i)$ for each $i = 1, \dots, \Sigma(a)$. By Lemma 3.6, we have $P_i : E_i$ for each i .

- The first case of Equation (3.2): We have $|T_j|, |T_{j'}| \geq m$ for some $1 \leq j < j' \leq \Sigma(a)$, and:

$$U = [] \quad E = a(U_1[E_1], \dots, U_{\Sigma(a)}[E_{\Sigma(a)}]) \quad P = P_1 \cdots P_{\Sigma(a)}.$$

Since $P, P' : E$, we can split $P' = P'_1 \cdots P'_{\Sigma(a)}$ so that $P'_i : E_i$ for each i . By the induction hypothesis,

$$\Phi_m(U_i[E_i][P'_i]) = (U_i, E_i, P'_i) \quad (\text{for each } i = 1, \dots, \Sigma(a)).$$

Since $|U_j[E_j][P'_j]| = |U_j[E_j]| = |T_j| \geq m$ and $|U_{j'}[E_{j'}][P'_{j'}]| = |U_{j'}[E_{j'}]| = |T_{j'}| \geq m$ for some $1 \leq j < j' \leq \Sigma(a)$, we have

$$\begin{aligned} \Phi_m(U[E][P']) &= \Phi_m(a(U_1[E_1][P'_1], \dots, U_{\Sigma(a)}[E_{\Sigma(a)}[P'_{\Sigma(a)}]])) \\ &= ([], a(U_1[E_1], \dots, U_{\Sigma(a)}[E_{\Sigma(a)}]), P'_1 \cdots P'_{\Sigma(a)}) \\ &= (U, E, P') \end{aligned}$$

as required.

- The second case of Equation (3.2): We have $|T_j| \geq m$ for a unique $j \in \{1, \dots, \Sigma(a)\}$ and:

$$n \triangleq |U_0| \geq m \quad U = [] \quad E = [\![\]_1^n[E_j]\!] \quad P = U_0 \cdot P_j$$

for $U_0 \triangleq a(T_1, \dots, T_{j-1}, U_j, T_{j+1}, \dots, T_{\Sigma(a)})$. Because $|T_j| \geq m$, we have $|E_j| \geq m$ by Lemma 3.11. Now we have $|E_j[P_j]| = |E_j| \geq m$, $|U_j| < m$, and $\Phi_m(U_j[E_j][P_j]) = \Phi_m(T_j) = (U_j, E_j, P_j)$; hence by Lemma 3.12, we have $\Phi_m(E_j[P_j]) = ([], E_j, P_j)$. Let $P' = U'_0 \cdot P''$. By the assumption, $U'_0 : [\![\]_1^n$ and thus U'_0 is a 1-context. Also, U'_0 is good for m ; thus $|U'_0| \geq m$ and U'_0 is of the form $a'(T'_1, \dots, T'_{j'-1}, U'_{j'}, T'_{j'+1}, \dots, T'_{\Sigma(a')})$ where $|U'_{j'}| < m$ and $|T'_i| < m$ for every $i \neq j'$. Now we have

- (1) $\Phi_m(U'_{j'}[E_j][P_j]) = (U'_{j'}, E_j, P_j)$ by Lemma 3.12 and $\Phi_m(E_j[P_j]) = ([], E_j, P_j)$;
- (2) $P'' \in \prod_{i=1}^{\text{shn}(E_j)} \mathcal{U}_{E_j, i}^m$ since $P' = U'_0 \cdot P'' \in \prod_{i=1}^{\text{shn}(E)} \mathcal{U}_{E, i}^m$ where $E = [\![\]_1^n[E_j]\!]$;
- (3) $|U'_{j'}[E_j]| < |U'_0[E_j]| = n + |E_j| = |E| = |U[E]|$.

Therefore, we can apply the induction hypothesis to $U'_{j'}[E_j]$, resulting in

$$\Phi_m(U'_{j'}[E_j][P'']) = (U'_{j'}, E_j, P'').$$

We have $|U'_{j'}[E_j]| \geq |E_j| \geq m$ and $|T'_i| < m$ for every $i \neq j'$. Furthermore

$$|a'(T'_1, \dots, T'_{j'-1}, U'_{j'}, T'_{j'+1}, \dots, T'_{\Sigma(a')})| = |U'_0| = n \geq m.$$

Hence we have

$$\begin{aligned} \Phi_m(U[E][P']) &= \Phi_m(U'_0[E_j][P'']) \\ &= \Phi_m(a'(T'_1, \dots, T'_{j'-1}, U'_{j'}, T'_{j'+1}, \dots, T'_{\Sigma(a')})[E_j][P'']) \\ &= \Phi_m(a'(T'_1, \dots, T'_{j'-1}, U'_{j'}[E_j][P''], T'_{j'+1}, \dots, T'_{\Sigma(a')})) \\ &= ([], [\![\]_1^n[E_j]\!], U'_0 \cdot P'') \\ &= (U, E, P') \end{aligned}$$

- The third case of Equation (3.2): We have $|T_j| = |U_j[E_j]| \geq m$ for a unique $j \in \{1, \dots, \Sigma(a)\}$ (and thus $|T_i| < m$ for every $i \neq j$) and:

$$U = a(T_1, \dots, T_{j-1}, U_j, T_{j+1}, \dots, T_{\Sigma(a)}) \quad |U| < m \quad E = E_j \quad P = P_j.$$

By the induction hypothesis,

$$\Phi_m(U_j[E_j][P']) = (U_j, E_j, P').$$

Since $|T_i| < m$ for every $i \neq j$ and $|a(T_1, \dots, T_{j-1}, U_j, T_{j+1}, \dots, T_{\Sigma(a)})| < m$, we have

$$\begin{aligned} \Phi_m(U[E][P']) &= \Phi_m(a(T_1, \dots, T_{j-1}, U_j[E_j][P'], T_{j+1}, \dots, T_{\Sigma(a)})) \\ &= (a(T_1, \dots, T_{j-1}, U_j, T_{j+1}, \dots, T_{\Sigma(a)}), E_j, P') \\ &= (U, E, P') \end{aligned}$$

- The fourth case of Equation (3.2): Let n be $|T|$; we have $n \geq m$ and:

$$U = [] \quad E = []_0^n \quad P = T.$$

Since $P' : E$, P' must be a singleton sequence consisting of a tree, say $T' = a'(T'_1, \dots, T'_{\Sigma(a')})$.

By the assumption, T' is good for m . Hence $|T'| \geq m$ and $|T'_i| < m$ for every $i = 1, \dots, \Sigma(a')$. So

$$\Phi_m(U[E][P']) = \Phi_m(T') = ([], []_0^n, T') = (U, E, P').$$

□

Corollary 3.14. *If $(E, P) = \Phi_m^\bullet(T)$, then $\Phi_m^\bullet(E[P']) = (E, P')$ for any $P' \in \prod_{i=1}^{\text{shn}(E)} \mathcal{U}_{E.i}^m$.*

Proof. If $(E, P) = \Phi_m^\bullet(T)$, then $(U, E', P) = \Phi_m(T)$ and $U[E'] = E$ for some U, E' . Since $\prod_{i=1}^{\text{shn}(E)} \mathcal{U}_{E.i}^m = \prod_{i=1}^{\text{shn}(E')} \mathcal{U}_{E'.i}^m$, Lemma 3.13 implies $\Phi_m(E[P']) = \Phi_m(U[E'] [P']) = (U, E', P')$. Thus, we have $\Phi_m^\bullet(E[P']) = (U[E'], P') = (E, P')$ as required. □

We are now ready to prove the product lemma.

Lemma 3.15 (Product Lemma (for Grammar-Independent Decomposition)). *For E and $m \geq 1$, if \mathcal{P}_E^m is non-empty, then*

$$\mathcal{P}_E^m = \prod_{i=1}^{\text{shn}(E)} \mathcal{U}_{E.i}^m.$$

Proof. The direction \subseteq follows from the fact that $(E, P) = \Phi_m^\bullet(T)$ implies $P : E$ and $P.i$ is good for m (Lemma 3.6).

We show the other direction. Since $\mathcal{P}_E^m \neq \emptyset$, there exist P and T such that $\Phi_m^\bullet(T) = (E, P)$. Let $P' \in \prod_{i=1}^{\text{shn}(E)} \mathcal{U}_{E.i}^m$. By Corollary 3.14, we have $\Phi_m^\bullet(E[P']) = (E, P')$. This means that $P' \in \mathcal{P}_E^m$. □

Lemma 3.8 follows as an immediate corollary of Lemmas 3.10 and 3.15.

3.3. Grammars in Canonical Form. As a preparation for generalizing the decomposition of $\mathcal{T}_n(\Sigma)$ (Lemma 3.8) to that of $\mathcal{L}_n(\mathcal{G}, N)$, we first transform a given regular tree grammar into *canonical form*, which will be defined shortly (in Definition 3.16). We prove that the transformation preserves unambiguity and (a weaker version of) strong connectivity.

Definition 3.16 (Canonical Grammar). A rewriting rule of a regular tree grammar \mathcal{G} is in *canonical form* if it is of the form

$$N \longrightarrow_{\mathcal{G}} a(N_1, \dots, N_{\Sigma(a)}).$$

A grammar $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ is *canonical* if every rewriting rule is in canonical form.

We transform a given regular tree grammar $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ to an equivalent one in canonical form. The idea of the transformation is fairly simple: we replace a rewriting rule

$$N \longrightarrow a(T_1, \dots, T_n) \quad (T_1, \dots, T_n \text{ are } (\Sigma \cup \mathcal{N})\text{-trees})$$

such that $T_i \notin \mathcal{N}$ with rules

$$\{ N \longrightarrow a(T_1, \dots, T_{i-1}, N', T_{i+1}, \dots, T_n), \quad N' \longrightarrow T_i \}$$

where N' is a fresh nonterminal that does not appear in \mathcal{N} . After iteratively applying the above transformation, we next replace a rewriting rule of the form $N \longrightarrow N'$ with rules

$$\{ N_0 \longrightarrow a(\dots, N', \dots) \mid (N_0 \longrightarrow a(\dots, N, \dots)) \in \mathcal{R} \}.$$

Again by iteratively applying this transformation, we finally obtain a grammar in canonical form.

The transformation, however, does not preserve strong connectivity. For example, consider the grammar $\mathcal{G} = (\{a \mapsto 2, b \mapsto 1, c \mapsto 0\}, \{N\}, \mathcal{R})$ where

$$\mathcal{R} = \{ N \longrightarrow b(c), \quad N \longrightarrow a(N, N) \}.$$

Then the above transformation introduces a nonterminal N' as well as rules

$$N \longrightarrow b(N') \quad \text{and} \quad N' \longrightarrow c.$$

Then N is not reachable from N' .

Observe that the problem above was caused by a newly introduced nonterminal that generates a single finite tree. To overcome the problem, we introduce a weaker version of strong connectivity called *essential strong-connectivity*. It requires strong connectivity only for nonterminals generating infinite languages; hence, it is preserved by the above transformation.

Definition 3.17 (Essential Strong-connectivity). Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be a regular tree grammar. We say that \mathcal{G} is *essentially strongly-connected* if for any nonterminals $N_1, N_2 \in \mathcal{N}$ with $\#(\mathcal{L}(\mathcal{G}, N_1)) = \#(\mathcal{L}(\mathcal{G}, N_2)) = \infty$, N_2 is reachable from N_1 .

Note that by the definition, every strongly-connected grammar is also essentially-strongly-connected. In the definition above, as well as in the arguments below, nonterminals N with $\#(\mathcal{L}(\mathcal{G}, N)) = \infty$ play an important role. We write \mathcal{N}^{inf} for the subset of \mathcal{N} consisting of such nonterminals.

Remark 3.18. A regular tree grammar that is essentially strongly-connected can be easily transformed into a strongly-connected grammar, hence the terminology. Let \mathcal{G} be an essentially strongly-connected grammar and $N \in \mathcal{N}^{\text{inf}}$. We say that a nonterminal is *inessential* if it generates a finite language. Let N_0 be an inessential nonterminal of a grammar \mathcal{G} such that $\mathcal{L}(\mathcal{G}, N_0) = \{T_1, \dots, T_n\}$. Then by replacing each rule

$$N_1 \longrightarrow C[N_0, \dots, N_0]$$

(where C is a k -context possibly having nonterminals other than N_0) with rules

$$\{ N_1 \longrightarrow C[T_{i_1}, \dots, T_{i_k}] \mid i_1, \dots, i_k \in \{1, \dots, n\} \},$$

one can remove the inessential nonterminal N_0 from the grammar. A grammar \mathcal{G} is essentially strongly-connected if and only if the grammar \mathcal{G}' obtained by removing all inessential nonterminals is strongly-connected. This transformation preserves the language in the following sense: writing $\mathcal{G}' = (\Sigma, \mathcal{N}^{\text{inf}}, \mathcal{R}')$ for the resulting grammar, we have

$\mathcal{L}(\mathcal{G}, N) = \mathcal{L}(\mathcal{G}', N)$ for each $N \in \mathcal{N}^{\text{inf}}$. Note that the process of erasing inessential non-terminals breaks canonicity; in fact, the class of languages generated by strongly-connected canonical regular tree grammars is a *proper* subset of that of essentially strongly-connected canonical regular tree grammars.

Recall that the second main theorem (Theorem 2.13) takes a family $(S_n)_{n \in \mathbb{N}}$ from $\mathcal{S}(\mathcal{G}) = \bigcup_{N, N' \in \mathcal{N}} \mathcal{L}(\mathcal{G}, N \Rightarrow N')$. In order to restate the theorem for essentially strongly connected grammars, we need to replace $\mathcal{S}(\mathcal{G})$ with the “essential” version, namely,

$$\mathcal{S}^{\text{inf}}(\mathcal{G}) \triangleq \bigcup_{N, N' \in \mathcal{N}^{\text{inf}}} \mathcal{L}(\mathcal{G}, N \Rightarrow N').$$

Lemma 3.19 (Canonical Form). *Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be a regular tree grammar that is unambiguous and strongly connected. Then one can (effectively) construct a grammar $\mathcal{G}' = (\Sigma, \mathcal{N}', \mathcal{R}')$ and a family $(\mathcal{I}_N)_{N \in \mathcal{N}}$ of subsets $\mathcal{I}_N \subseteq \mathcal{N}'$ that satisfy the following conditions:*

- \mathcal{G}' is canonical, unambiguous and essentially strongly-connected.
- $\mathcal{L}(\mathcal{G}, N) = \biguplus_{N' \in \mathcal{I}_N} \mathcal{L}(\mathcal{G}', N')$ for every $N \in \mathcal{N}$.
- If $\#(\mathcal{L}(\mathcal{G})) = \infty$, then $\mathcal{S}(\mathcal{G}) \subseteq \mathcal{S}^{\text{inf}}(\mathcal{G}')$.

Proof. See Appendix B. □

3.4. Decomposition of Regular Tree Languages. This subsection generalizes the decomposition of $\mathcal{T}_n(\Sigma)$ in Section 3.2:

$$\mathcal{T}_n(\Sigma) \cong \prod_{E \in \mathcal{E}_n^m} \prod_{i=1}^{\text{shn}(E)} \mathcal{U}_{E.i}^m$$

to that of $\mathcal{L}_n(\mathcal{G}, N)$, and proves a bijection of the following form:

$$\mathcal{L}_n(\mathcal{G}, N) \cong \prod_{\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)} \prod_{i=1}^{\text{shn}(\tilde{E})} \tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}).$$

Here, \tilde{E} denotes a *typed* second-order context (which will be defined shortly in Section 3.4.1), each of whose second-order holes $\tilde{E}.i = \llbracket \cdot \rrbracket_{\kappa}^n$ carries not only the size n but the *context type* κ of a context to be substituted for the hole. Accordingly, we have replaced $\mathcal{U}_{E.i}^m$ with $\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G})$, which denotes the set of contexts that respect the context type specified by $\tilde{E}.i$.

In the rest of this subsection, we first define the notion of typed second-order contexts in Section 3.4.1, extend the decomposition function accordingly in Section 3.4.2, and use it to prove the above bijection in Section 3.4.3. Throughout this subsection (i.e., Section 3.4), we assume that \mathcal{G} is a canonical and unambiguous grammar. We emphasize here that the discussion in this subsection essentially relies on both unambiguity and canonicity of the grammar. The essential strong connectivity is not required for the results in this subsection; it will be used in Section 3.5, to show that each component $\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G})$ contains an affine context that has $S_{\lceil p \log n \rceil}$ as a subcontext (recall condition (i) of (T3) in Section 3.1).

Remark 3.20. Note that any deterministic bottom-up tree automaton (without any ϵ -rules) [21] can be considered an unambiguous canonical tree grammar, by regarding each transition rule $a(q_1, \dots, q_k) \longrightarrow q$ as a rewriting rule $N_q \longrightarrow a(N_{q_1}, \dots, N_{q_k})$. Thus, by

$$\begin{array}{c}
\frac{\vdash \tilde{E}_i : N_i \quad (\text{for each } i = 1, \dots, k)}{\vdash \llbracket \rrbracket_{N_1 \dots N_k \Rightarrow N}^n [\tilde{E}_1, \dots, \tilde{E}_k] : N} \quad (\text{SC-Hole}) \\
\\
\frac{(N \longrightarrow a(N_1, \dots, N_{\Sigma(a)})) \in \mathcal{R} \quad \vdash \tilde{E}_i : N_i \quad (\text{for each } i = 1, \dots, \Sigma(a))}{\vdash a(\tilde{E}_1, \dots, \tilde{E}_{\Sigma(a)}) : N} \quad (\text{SC-Term})
\end{array}$$

FIGURE 5. Typing rules for \tilde{E}

the equivalence between the class of tree languages generated by regular tree grammars and the class of those accepted by deterministic bottom-up tree automata, any regular tree grammar can be converted to an unambiguous canonical tree grammar.

3.4.1. Typed Second-Order Contexts. The set of \mathcal{G} -typed second-order contexts, ranged over by \tilde{E} , is defined by:

$$\tilde{E} ::= \llbracket \rrbracket_{N_1 \dots N_k \Rightarrow N}^n [\tilde{E}_1, \dots, \tilde{E}_k] \mid a(\tilde{E}_1, \dots, \tilde{E}_{\Sigma(a)}) \quad (a \in \text{Dom}(\Sigma)).$$

The subscript $\kappa = (N_1 \dots N_k \Rightarrow N)$ describes the type of first-order contexts that can be substituted for this second-order hole; the superscript n describes the size as before. Hence a (first-order) context C is suitable for filling a second-order hole $\llbracket \rrbracket_{\kappa}^n$ if $C : \kappa$ and $|C| = n$. We write $C : \llbracket \rrbracket_{\kappa}^n$ if $C : \kappa$ and $|C| = n$. The operations such as $\tilde{E}.i$, $\tilde{E}[C]$ and $\tilde{E}[P]$ are defined analogously. For a sequence of contexts $P = C_1 C_2 \dots C_{\ell}$, we write $P : \tilde{E}$ if $\#(P) (= \ell) = \text{shn}(\tilde{E})$ and $C_i : \tilde{E}.i$ for each $i \leq \text{shn}(\tilde{E})$.

We define the second-order context typing relation $\vdash \tilde{E} : N$ inductively by the rules in Figure 5. Intuitively, $\vdash \tilde{E} : N$ means that $\tilde{E}[P] \in \mathcal{L}(\mathcal{G}, N)$ holds for any P such that $P : \tilde{E}$ (as confirmed in Lemma 3.23 below). As in the case of untyped second-order contexts, we actually use only typed second-order contexts with holes of the form $\llbracket \rrbracket_{N_1 \dots N_k \Rightarrow N}^n$ where k is 0 or 1.

Example 3.21. Recall the second-order context $E = \mathbf{b}(\llbracket \rrbracket_1^3[\mathbf{a}(\llbracket \rrbracket_0^3, \mathbf{b}(\llbracket \rrbracket_0^5))])$ in Figure 4 and Example 3.2. Given the grammar consisting of the rules:

$$A \rightarrow \mathbf{a}(B, B) \quad B \rightarrow \mathbf{b}(A) \quad B \rightarrow \mathbf{b}(B) \quad B \rightarrow \mathbf{c},$$

the corresponding typed second-order context \tilde{E} is:

$$\mathbf{b}(\llbracket \rrbracket_{A \Rightarrow A}^3[\mathbf{a}(\llbracket \rrbracket_{\Rightarrow B}^3, \mathbf{b}(\llbracket \rrbracket_{\Rightarrow A}^5))])$$

and we have $\vdash \tilde{E} : B$. For $P = \mathbf{a}(\mathbf{b}(\llbracket \rrbracket, \mathbf{c}) \cdot \mathbf{b}(\mathbf{b}(\mathbf{c})) \cdot \mathbf{a}(\mathbf{b}(\mathbf{c}), \mathbf{b}(\mathbf{c})))$, we have $P : \tilde{E}$, and:

$$\tilde{E}[P] = \mathbf{b}(\mathbf{a}(\mathbf{b}(\mathbf{a}(\mathbf{b}(\mathbf{b}(\mathbf{c}))), \mathbf{b}(\mathbf{a}(\mathbf{b}(\mathbf{c}), \mathbf{b}(\mathbf{c}))))), \mathbf{c})) \in \mathcal{L}(\mathcal{G}, B).$$

Lemma 3.22. *The following rule is derivable:*

$$\frac{C \in \mathcal{L}(\mathcal{G}, N_1 \dots N_k \Rightarrow N) \quad \vdash \tilde{E}_i : N_i \quad (\text{for each } i = 1, \dots, k)}{\vdash C[\tilde{E}_1, \dots, \tilde{E}_k] : N} \quad (\text{SC-Ctx})$$

$$\frac{\tilde{E}_i \triangleleft E_i \quad (\text{for each } i = 1, \dots, k)}{\llbracket \rrbracket_{N_1 \dots N_k \Rightarrow N}^n[\tilde{E}_1, \dots, \tilde{E}_k] \triangleleft \llbracket \rrbracket_k^n[E_1, \dots, E_n]} \quad (\text{Ref-Hole})$$

$$\frac{\tilde{E}_i \triangleleft E_i \quad (\text{for each } i = 1, \dots, \Sigma(a))}{a(\tilde{E}_1, \dots, \tilde{E}_{\Sigma(a)}) \triangleleft a(E_1, \dots, E_n)} \quad (\text{Ref-Term})$$

FIGURE 6. Refinement relation

Proof. The proof proceeds by induction on C . Assume that the premises of the rule hold. If $C = []$, then $C[\tilde{E}_1, \dots, \tilde{E}_k] = \tilde{E}_1$ and $N = N_1$; thus the result follows immediately from the assumption. If $C = a(C_1, \dots, C_\ell)$, then by the assumption that the grammar is canonical and $C \in \mathcal{L}(\mathcal{G}, N_1 \dots N_k \Rightarrow N)$, we have $N \rightarrow a(N'_1, \dots, N'_\ell)$ with $C_i \in \mathcal{L}(\mathcal{G}, N_{k_{i-1}+1} \dots N_{k_i} \Rightarrow N'_i)$ for $i = 1, \dots, \ell$ where $k_0 = 0$ and $k_\ell = k$. By the induction hypothesis, we have $\vdash C_i[\tilde{E}_{k_{i-1}+1}, \dots, \tilde{E}_{k_i}] : N'_i$. Thus, by using rule (SC-Term), we have $\vdash C[\tilde{E}_1, \dots, \tilde{E}_k] : N$ as required. \square

Lemma 3.23. *Assume that $\vdash \tilde{E} : N$.*

- (1) *If $\text{shn}(\tilde{E}) = 0$, then \tilde{E} is a tree and $\tilde{E} \in \mathcal{L}(\mathcal{G}, N)$.*
- (2) *If $\text{shn}(\tilde{E}) \geq 1$ and $C : \tilde{E}.1$, then $\vdash \tilde{E}[C] : N$.*
- (3) *If $P : \tilde{E}$, then $\tilde{E}[P] \in \mathcal{L}(\mathcal{G}, N)$.*

Proof. (1) By induction on the structure of $\vdash \tilde{E} : N$. (2) By induction on the structure of $\vdash \tilde{E} : N$. The base case is when $\tilde{E} = \llbracket \rrbracket_{N_1 \dots N_k \Rightarrow N}^n[\tilde{E}_1, \dots, \tilde{E}_k]$. Since $\vdash \tilde{E} : N$, we have $\vdash \tilde{E}_i : N_i$ for each i . Then by the derived rule (SC-CTX), we have $\vdash C[\tilde{E}_1, \dots, \tilde{E}_k] : N$. (3) By induction on $\text{shn}(\tilde{E})$ (using (1) and (2)). \square

3.4.2. Grammar-Respecting Decomposition Function $\tilde{\Phi}_m^{\bullet, \mathcal{G}}$. We have defined in Section 3.2.2 the function Φ_m^\bullet that decomposes a tree T and returns a pair (E, P) of a second-order context E and a sufficiently long sequence P of (first-order) contexts. The aim here is to extend Φ_m^\bullet to a grammar-respecting one $\tilde{\Phi}_m^{\bullet, \mathcal{G}}$ that takes a pair (T, N) such that $T \in \mathcal{L}(\mathcal{G}, N)$ as an input, and returns a pair (\tilde{E}, P) , which is the same as $\Phi_m^\bullet(T) = (E, P)$, except that \tilde{E} is a “type-annotated” version of E . For example, for the tree in Figure 4 and the grammar in Example 3.21, we expect that $\tilde{\Phi}_m^{\bullet, \mathcal{G}}(T) = (\tilde{E}, P)$ where:

$$T = \mathbf{b}(\mathbf{a}(\mathbf{b}(\mathbf{a}(\mathbf{b}(\mathbf{b}(\mathbf{c}))), \mathbf{b}(\mathbf{a}(\mathbf{b}(\mathbf{c}), \mathbf{b}(\mathbf{c}))))), \mathbf{c}))$$

$$\tilde{E} = \mathbf{b}(\llbracket \rrbracket_{A \Rightarrow A}^3[\mathbf{a}(\llbracket \rrbracket_{\Rightarrow B}^3, \mathbf{b}(\llbracket \rrbracket_{\Rightarrow A}^5))]) \quad P = \mathbf{a}(\mathbf{b}([], \mathbf{c}) \cdot \mathbf{b}(\mathbf{b}(\mathbf{c})) \cdot \mathbf{a}(\mathbf{b}(\mathbf{c}), \mathbf{b}(\mathbf{c}))).$$

We say that \tilde{E} *refines* E , written $\tilde{E} \triangleleft E$, if E is obtained by simply forgetting type annotations, i.e., replacing $\llbracket \rrbracket_{N_1 \dots N_k \Rightarrow N}^n$ in \tilde{E} with $\llbracket \rrbracket_k^n$. This relation is formally defined by induction on the structures of E and \tilde{E} by the rules in Figure 6. The following lemma is obtained by straightforward induction on $\tilde{E} \triangleleft E$.

Lemma 3.24. *Assume $\tilde{E} \triangleleft E$.*

- (1) $|\tilde{E}| = |E|$.

- (2) If $P : \tilde{E}$, then $P : E$.
 (3) If $P : \tilde{E}$, then $\tilde{E} \llbracket P \rrbracket = E \llbracket P \rrbracket$.

Given $T \in \mathcal{L}(\mathcal{G}, N)$ and $(E, P) = \Phi_m^\bullet(T)$, the value of $\tilde{\Phi}_m^{\bullet, \mathcal{G}}(T, N)$ should be (\tilde{E}, P) where \tilde{E} is a \mathcal{G} -typed second-order context that satisfies the following conditions:

$$\tilde{E} \triangleleft E, \quad \vdash \tilde{E} : N \quad \text{and} \quad P : \tilde{E}.$$

We prove that, for every $T \in \mathcal{L}(\mathcal{G}, N)$, there exists exactly one typed second-order context \tilde{E} that satisfies the above condition. Hence the above constraints defines the function $\tilde{\Phi}_m^{\bullet, \mathcal{G}}(T, N)$.

We first state and prove a similar result for first-order contexts (in Lemma 3.25), and then prove the second-order version (in Lemma 3.26), using the former.

Lemma 3.25. *Let C be a (first-order) k -context and T_1, \dots, T_k be trees. Assume that $C[T_1, \dots, T_k] \in \mathcal{L}(\mathcal{G}, N)$. Then there exists a unique family $(N_i)_{i \in \{1, \dots, k\}}$ such that $C \in \mathcal{L}(\mathcal{G}, N_1 \dots N_k \Rightarrow N)$ and $T_i \in \mathcal{L}(\mathcal{G}, N_i)$ for every $i = 1, \dots, k$.*

Proof. By induction on C .

- Case $C = []$: Then $k = 1$. The existence follows from $[] \in \mathcal{L}(\mathcal{G}, N \Rightarrow N)$ (as $N \longrightarrow^* N$) and $T_1 = C[T_1] \in \mathcal{L}(\mathcal{G}, N)$. The uniqueness follows from the fact that $[] \in \mathcal{L}(\mathcal{G}, N_1 \Rightarrow N)$ implies $N_1 = N$.
- Case $C = a(C_1, \dots, C_{\Sigma(a)})$: Let $\ell = \Sigma(a)$, $k_i = \text{hn}(C_i)$ for each $i = 1, \dots, \ell$ and

$$(T_{1,1}, \dots, T_{1,k_1}, T_{2,1}, \dots, T_{2,k_2}, \dots, T_{\ell,1}, \dots, T_{\ell,k_\ell}) = (T_1, \dots, T_k).$$

Then $C[T_1, \dots, T_k] = a(C_1[T_{1,1}, \dots, T_{1,k_1}], \dots, C_\ell[T_{\ell,1}, \dots, T_{\ell,k_\ell}])$. Since $C[T_1, \dots, T_k] \in \mathcal{L}(\mathcal{G}, N)$, there exists a rewriting sequence

$$N \longrightarrow a(N_1, \dots, N_\ell) \longrightarrow^* a(C_1[T_{1,1}, \dots, T_{1,k_1}], \dots, C_\ell[T_{\ell,1}, \dots, T_{\ell,k_\ell}]).$$

Thus $N_i \longrightarrow^* C_i[T_{i,1}, \dots, T_{i,k_i}]$, i.e., $C_i[T_{i,1}, \dots, T_{i,k_i}] \in \mathcal{L}(\mathcal{G}, N_i)$, for each $i = 1, \dots, \ell$. By the induction hypothesis, there exist $N_{i,1}, \dots, N_{i,k_i}$ such that $C_i \in \mathcal{L}(\mathcal{G}, N_{i,1} \dots N_{i,k_i} \Rightarrow N_i)$ and $T_{i,j} \in \mathcal{L}(\mathcal{G}, N_{i,j})$ for each $j = 1, \dots, k_i$. Then (the rearrangement of) the family $(N_{i,j})_{i \leq \ell, j \leq k_i}$ satisfies the requirement.

We prove the uniqueness. Assume that both $(N_{i,j}^1)_{i,j}$ and $(N_{i,j}^2)_{i,j}$ satisfy the requirement. Then, for each $m = 1, 2$, there exist N_i^m ($i = 1, \dots, \ell$) such that

$$\begin{aligned} N &\longrightarrow a(N_1^m, \dots, N_\ell^m) \longrightarrow^* a(C_1[N_{1,1}^m, \dots, N_{1,k_1}^m], \dots, C_\ell[N_{\ell,1}^m, \dots, N_{\ell,k_\ell}^m]) \\ &\longrightarrow^* a(C_1[T_{1,1}, \dots, T_{1,k_1}], \dots, C_\ell[T_{\ell,1}, \dots, T_{\ell,k_\ell}]). \end{aligned}$$

Since \mathcal{G} is unambiguous, $N_i^1 = N_i^2$ for each $i = 1, \dots, \ell$. By the induction hypothesis, we have $N_{i,j}^1 = N_{i,j}^2$ for each i and j . \square

Lemma 3.26. *Let \mathcal{G} be a canonical unambiguous regular tree grammar and $T \in \mathcal{L}(\mathcal{G}, N)$. Given E and P , assume that $P : E$ and $E \llbracket P \rrbracket = T$. Then E has a unique refinement $\tilde{E} \triangleleft E$ such that $\vdash \tilde{E} : N$ and $P : \tilde{E}$.*

Proof. We prove by induction on E .

- Case $E = \llbracket \rrbracket_k^n[E_1, \dots, E_k]$: The sequence P can be decomposed as $P = C \cdot P_1 \dots P_k$ so that $P_i : E_i$ for each $i = 1, \dots, k$. Furthermore $\text{hn}(C) = k$ and $|C| = n$. We have $E \llbracket P \rrbracket = C[E_1 \llbracket P_1 \rrbracket, \dots, E_k \llbracket P_k \rrbracket] \in \mathcal{L}(\mathcal{G}, N)$.

We prove the existence. By Lemma 3.25, there exists a family $(N_i)_{i=1,\dots,k}$ such that $C \in \mathcal{L}(\mathcal{G}, N_1 \dots N_k \Rightarrow N)$ (i.e., $C : \llbracket \rrbracket_{N_1 \dots N_k \Rightarrow N}^n$) and $E_i \llbracket P_i \rrbracket \in \mathcal{L}(\mathcal{G}, N_i)$ for each $i = 1, \dots, k$. By the induction hypothesis, for each $i = 1, \dots, k$, there exists $\tilde{E}_i \triangleleft E_i$ such that $\vdash \tilde{E}_i : N_i$ and $P_i : \tilde{E}_i$. Let $\tilde{E} \triangleq \llbracket \rrbracket_{N_1 \dots N_k \Rightarrow N}^n[\tilde{E}_1, \dots, \tilde{E}_k]$. Then $\tilde{E} \triangleleft E$, $\vdash \tilde{E} : N$, and $P : \tilde{E}$.

We prove the uniqueness. Assume \tilde{E}^1 and \tilde{E}^2 satisfy that $\tilde{E}^j \triangleleft E$, $\vdash \tilde{E}^j : N$ and $P : \tilde{E}^j$ for $j = 1, 2$. Since $\tilde{E}^j \triangleleft E$ and $\vdash \tilde{E}^j : N$, \tilde{E}^j must be of the form: $\tilde{E}^j = \llbracket \rrbracket_{N_1^j \dots N_k^j \Rightarrow N}^n[\tilde{E}_1^j, \dots, \tilde{E}_k^j]$ with $\tilde{E}_i^j \triangleleft E_i$ and $\vdash \tilde{E}_i^j : N_i^j$. Since $P : \tilde{E}^j$, we have $C \in \mathcal{L}(\mathcal{G}, N_1^j \dots N_k^j \Rightarrow N)$ and $P_i : \tilde{E}_i^j$ for $i = 1, \dots, k$. By Lemma 3.23, $\tilde{E}_i^j \llbracket P_i \rrbracket \in \mathcal{L}(\mathcal{G}, N_i^j)$ for each $i = 1, \dots, k$ and $j = 1, 2$. By Lemma 3.24(3), $E \llbracket P \rrbracket = \tilde{E}^j \llbracket P \rrbracket = C[\tilde{E}_1^j \llbracket P_1 \rrbracket, \dots, \tilde{E}_k^j \llbracket P_k \rrbracket] \in \mathcal{L}(\mathcal{G}, N)$. Hence by Lemma 3.25, $N_i^1 = N_i^2$ for each $i = 1, \dots, k$. By the induction hypothesis, $\tilde{E}_i^1 = \tilde{E}_i^2$ for each i . Hence $\tilde{E}^1 = \tilde{E}^2$.

- Case $E = a(E_1, \dots, E_{\Sigma(a)})$: The sequence P can be decomposed as $P = P_1 \dots P_{\Sigma(a)}$ so that $P_i : E_i$ for each $i = 1, \dots, \Sigma(a)$. Then $E \llbracket P \rrbracket = a(E_1 \llbracket P_1 \rrbracket, \dots, E_{\Sigma(a)} \llbracket P_{\Sigma(a)} \rrbracket)$.

We prove the existence. Since $N \longrightarrow^* a(E_1 \llbracket P_1 \rrbracket, \dots, E_{\Sigma(a)} \llbracket P_{\Sigma(a)} \rrbracket)$, there exists a rule $N \longrightarrow a(N_1, \dots, N_{\Sigma(a)})$ such that

$$N \longrightarrow a(N_1, \dots, N_{\Sigma(a)}) \longrightarrow^* a(E_1 \llbracket P_1 \rrbracket, \dots, E_{\Sigma(a)} \llbracket P_{\Sigma(a)} \rrbracket).$$

So $N_i \longrightarrow^* E_i \llbracket P_i \rrbracket$ for each $i = 1, \dots, \Sigma(a)$. By the induction hypothesis, there exists $\tilde{E}_i \triangleleft E_i$ such that $\vdash \tilde{E}_i : N_i$ and $P_i : \tilde{E}_i$. Let $\tilde{E} \triangleq a(\tilde{E}_1, \dots, \tilde{E}_{\Sigma(a)})$. Then $\tilde{E} \triangleleft E$, $\vdash \tilde{E} : N$, and $P : \tilde{E}$.

We prove the uniqueness. Assume \tilde{E}^1 and \tilde{E}^2 satisfy that $\tilde{E}^j \triangleleft E$, $\vdash \tilde{E}^j : N$ and $P : \tilde{E}^j$ for $j = 1, 2$. Since $\tilde{E}^j \triangleleft E$, \tilde{E}^j must be of the form: $\tilde{E}^j = a(\tilde{E}_1^j, \dots, \tilde{E}_{\Sigma(a)}^j)$ with $\tilde{E}_i^j \triangleleft E_i$. By $\vdash \tilde{E}^j : N$, there exists a rule $N \longrightarrow a(N_1^j, \dots, N_{\Sigma(a)}^j)$ such that $\vdash \tilde{E}_i^j : N_i^j$ for each i . Since $P : \tilde{E}^j$, we have $P_i : \tilde{E}_i^j$ for each i . By Lemmas 3.23 and 3.24(3), we have $\tilde{E}_i^j \llbracket P_i \rrbracket = E_i \llbracket P_i \rrbracket \in \mathcal{L}(\mathcal{G}, N_i^j)$. Now we have

$$N \longrightarrow a(N_1^j, \dots, N_{\Sigma(a)}^j) \longrightarrow^* a(E_1 \llbracket P_1 \rrbracket, \dots, E_{\Sigma(a)} \llbracket P_{\Sigma(a)} \rrbracket)$$

for $j = 1, 2$. Since \mathcal{G} is unambiguous, $N_i^1 = N_i^2$ for each $i = 1, \dots, \Sigma(a)$. By the induction hypothesis, $\tilde{E}_i^1 = \tilde{E}_i^2$ for each i . Hence $\tilde{E}^1 = \tilde{E}^2$. \square

3.4.3. Decomposition of $\mathcal{L}_n(\mathcal{G}, N)$. To formally state the decomposition lemma, we prepare some definitions. For a canonical unambiguous grammar $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$, $N \in \mathcal{N}$, $n \geq 1$, and $m \geq 1$, we define $\tilde{\mathcal{E}}_n^m(\mathcal{G}, N)$, $\tilde{\mathcal{P}}_E^m(\mathcal{G}, N)$, and $\tilde{\mathcal{U}}_{\llbracket \rrbracket_\kappa^n}^m(\mathcal{G})$ by:

$$\tilde{\mathcal{E}}_n^m(\mathcal{G}, N) \triangleq \{\tilde{E} \mid (\tilde{E}, P) = \tilde{\Phi}_m^{\bullet, \mathcal{G}}(T, N) \text{ for some } T \in \mathcal{L}_n(\mathcal{G}, N) \text{ and } P\}.$$

$$\tilde{\mathcal{P}}_E^m(\mathcal{G}, N) \triangleq \{P \mid (\tilde{E}, P) = \tilde{\Phi}_m^{\bullet, \mathcal{G}}(T, N) \text{ for some } T \in \mathcal{L}(\mathcal{G}, N)\}.$$

$$\tilde{\mathcal{U}}_{\llbracket \rrbracket_\kappa^n}^m(\mathcal{G}) \triangleq \{U \mid U : \llbracket \rrbracket_\kappa^n, U \text{ is good for } m\}.$$

The set $\tilde{\mathcal{E}}_n^m(\mathcal{G}, N)$ consists of second-order contexts that are obtained by decomposing trees of size n , and $\tilde{\mathcal{P}}_E^m(\mathcal{G}, N)$ consists of affine context sequences that match \tilde{E} . The set $\tilde{\mathcal{P}}_E^m(\mathcal{G}, N)$ is the set of contexts that match the hole $\llbracket \cdot \rrbracket_\kappa^n$.

The following is the main result of this subsection.

Lemma 3.27.

$$\mathcal{L}_n(\mathcal{G}, N) \cong \coprod_{\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)} \prod_{i=1}^{\text{shn}(\tilde{E})} \tilde{\mathcal{U}}_{\tilde{E}, i}^m(\mathcal{G}).$$

The lemma above is a direct consequence of typed versions of coproduct and product lemmas (Lemmas 3.28 and 3.30 below). The following coproduct lemma can be shown in a manner similar to Lemma 3.10:

Lemma 3.28 (Coproduct Lemma (for Grammar-Respecting Decomposition)). *For any $n \geq 1$ and $m \geq 1$, there exists a bijection*

$$\mathcal{L}_n(\mathcal{G}, N) \cong \coprod_{\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)} \tilde{\mathcal{P}}_E^m(\mathcal{G}, N).$$

such that (\tilde{E}, P) in the right hand side is mapped to $\tilde{E}[\![P]\!]$.

Proof. We define a function

$$f : \coprod_{\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)} \tilde{\mathcal{P}}_E^m(\mathcal{G}, N) \longrightarrow \mathcal{L}_n(\mathcal{G}, N)$$

by $f(\tilde{E}, P) \triangleq \tilde{E}[\![P]\!]$, and a function

$$g : \mathcal{L}_n(\mathcal{G}, N) \longrightarrow \coprod_{\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)} \tilde{\mathcal{P}}_E^m(\mathcal{G}, N)$$

by $g(T) \triangleq \tilde{\Phi}_m^{\bullet, \mathcal{G}}(T, N)$.

Let us check that these are functions into the codomains:

- $f(\tilde{E}, P) \in \mathcal{L}_n(\mathcal{G}, N)$: Since $P \in \tilde{\mathcal{P}}_E^m(\mathcal{G}, N)$, there exists $T \in \mathcal{L}_n(\mathcal{G}, N)$ such that $(\tilde{E}, P) = \tilde{\Phi}_m^{\bullet, \mathcal{G}}(T, N)$. By Lemmas 3.6 and 3.24, we have $f(\tilde{E}, P) = \tilde{E}[\![P]\!] = T \in \mathcal{L}_n(\mathcal{G}, N)$.
- $g(T) \in \coprod_{\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)} \tilde{\mathcal{P}}_E^m(\mathcal{G}, N)$: Obvious from the definitions of $\tilde{\mathcal{E}}_n^m(\mathcal{G}, N)$ and $\tilde{\mathcal{P}}_E^m(\mathcal{G}, N)$.

We have $f(g(T)) = T$ by Lemmas 3.6(1) and 3.24. Let $(\tilde{E}, P) \in \coprod_{\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)} \tilde{\mathcal{P}}_E^m(\mathcal{G}, N)$. By definition, there exists $T \in \mathcal{L}_n(\mathcal{G}, N)$ such that $(\tilde{E}, P) = \tilde{\Phi}_m^{\bullet, \mathcal{G}}(T, N) = g(T)$. Then

$$g(f(\tilde{E}, P)) = g(f(g(T))) = g(T) = (\tilde{E}, P).$$

□

The following is a key lemma used for proving a typed version of the product lemma.

Lemma 3.29. *For a nonterminal N , $n \geq 1$, $m \geq 1$ and $\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)$, let E be the unique second-order context such that $\tilde{E} \triangleleft E$. Then we have*

$$\tilde{\mathcal{P}}_{\tilde{E}}^m(\mathcal{G}, N) = \mathcal{P}_E^m \cap \{P \mid P : \tilde{E}\}.$$

Proof. The direction \subseteq is clear. We prove the converse.

Let P be in the right hand side. Since $\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)$, there exist T' and P' such that

$$\tilde{\Phi}_m^{\bullet, \mathcal{G}}(T', N) = (\tilde{E}, P').$$

By the definition of $\tilde{\Phi}_m^{\bullet, \mathcal{G}}(T', N)$, we have $\vdash \tilde{E} : N$. Since $P \in \mathcal{P}_E^m$, there exists T such that $(E, P) = \Phi_m^\bullet(T)$ and thus, by Lemma 3.6,

$$\Phi_m^\bullet(E[P]) = (E, P).$$

Since $\Phi_m^\bullet(E[P]) = (E, P)$, $\tilde{E} \triangleleft E$, $\vdash \tilde{E} : N$, and $P : \tilde{E}$, by the definition of $\tilde{\Phi}_m^{\bullet, \mathcal{G}}$, we have $\tilde{\Phi}_m^{\bullet, \mathcal{G}}(E[P], N) = (\tilde{E}, P)$. By Lemmas 3.23(3) and 3.24(3), $E[P] = \tilde{E}[P] \in \mathcal{L}(\mathcal{G}, N)$. Hence $P \in \tilde{\mathcal{P}}_{\tilde{E}}^m(\mathcal{G}, N)$. \square

The following is the typed version of the product lemma, which follows from Lemmas 3.15 and 3.29.

Lemma 3.30 (Product Lemma (for Grammar-Respecting Decomposition)). *For any non-terminal N , $n \geq 1$, $m \geq 1$ and $\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)$, we have*

$$\tilde{\mathcal{P}}_{\tilde{E}}^m(\mathcal{G}, N) = \prod_{i=1}^{\text{shn}(\tilde{E})} \tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}).$$

Proof. Let E be the unique second-order context such that $\tilde{E} \triangleleft E$. By Lemmas 3.15 and 3.29, we have:

$$\begin{aligned} \tilde{\mathcal{P}}_{\tilde{E}}^m(\mathcal{G}, N) &= \mathcal{P}_E^m \cap \{P \mid P : \tilde{E}\} \\ &= (\prod_{i=1}^{\text{shn}(E)} \mathcal{U}_{E.i}^m) \cap (\prod_{i=1}^{\text{shn}(\tilde{E})} \{U \mid U : \tilde{E}.i\}) \\ &= \prod_{i=1}^{\text{shn}(\tilde{E})} (\mathcal{U}_{E.i}^m \cap \{U \mid U : \tilde{E}.i\}) \\ &= \prod_{i=1}^{\text{shn}(\tilde{E})} \tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}). \end{aligned}$$

\square

Lemma 3.27 is an immediate corollary of Lemmas 3.28 and 3.30.

3.5. Each Component Contains the Subcontext of Interest. In Section 3.4, we have shown that the set $\mathcal{L}_n(\mathcal{G}, N)$ of trees can be decomposed as:

$$\mathcal{L}_n(\mathcal{G}, N) \cong \prod_{\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)} \prod_{i=1}^{\text{shn}(\tilde{E})} \tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}),$$

assuming that \mathcal{G} is canonical and unambiguous. In this subsection, we further assume that \mathcal{G} is essentially strongly connected, and prove that, for each tree context $S \in \mathcal{S}^{\text{inf}}(\mathcal{G})$, every component “contains” S , i.e., there exists $U \in \tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G})$ such that $S \preceq U$ if m is sufficiently

large, say $m \geq m_0$ (where m_0 depends on $|S|$). More precisely, the goal of this subsection is to prove the following lemma.

Lemma 3.31. *Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be an unambiguous, essentially strongly-connected grammar in canonical form and $(S_n)_{n \in \mathbb{N}}$ be a family of linear contexts in $\mathcal{S}^{\text{inf}}(\mathcal{G})$ such that $|S_n| = O(n)$. Then there exist integers $b, c \geq 1$ that satisfy the following: For any $N \in \mathcal{N}^{\text{inf}}$, $n \geq 1$, $m \geq b$, $\tilde{E} \in \tilde{\mathcal{E}}_n^{\text{cm}}(\mathcal{G}, N)$ and $i \in \{1, \dots, \text{shn}(\tilde{E})\}$, there exists $U \in \tilde{\mathcal{U}}_{\tilde{E}, i}^{\text{cm}}(\mathcal{G})$ such that $S_m \preceq U$.*

The rest of this subsection is devoted to a proof of the lemma above. The idea of the proof is as follows. Assume $S \in \mathcal{L}(\mathcal{G}, N_1 \Rightarrow N_2)$ ($N_1, N_2 \in \mathcal{N}^{\text{inf}}$) and $\llbracket \cdot \rrbracket_\kappa^n = \tilde{E}.i$. Recall that

$$\tilde{\mathcal{U}}_{\llbracket \cdot \rrbracket_\kappa^n}^m(\mathcal{G}) = \{U \mid U \in \mathcal{L}(\mathcal{G}, \kappa), |U| = n, \text{ and } U \text{ is good for } m\}.$$

It is not difficult to find a context U that satisfies both $U \in \mathcal{L}(\mathcal{G}, \kappa)$ and $S \preceq U$. For example, assume that $\kappa = N_0 \Rightarrow N_3$ ($N_0, N_3 \in \mathcal{N}^{\text{inf}}$). Then, since the grammar is assumed to be essentially strongly-connected, there exist $S_{0,1} \in \mathcal{L}(\mathcal{G}, N_0 \Rightarrow N_1)$ and $S_{2,3} \in \mathcal{L}(\mathcal{G}, N_2 \Rightarrow N_3)$ and then $U \triangleq S_{2,3}[S[S_{0,1}]]$ satisfies $U \in \mathcal{L}(\mathcal{G}, \kappa)$ and $S \preceq U$. What is relatively difficult is to show that $S_{2,3}$ and $S_{0,1}$ can be chosen so that they meet the required size constraints (i.e., $|U| = n$ and U is good for m).

The following is a key lemma, which states that any essentially strongly connected grammar \mathcal{G} is *periodic* in the sense that there is a constant c (that depends on \mathcal{G}) and a family of constants $d_{N,N'}$ such that, for each $N, N' \in \mathcal{N}^{\text{inf}}$, and sufficiently large n , $\mathcal{L}_n(\mathcal{G}, N \Rightarrow N') \neq \emptyset$ if and only if $n \equiv d_{N,N'} \pmod{c}$.

Lemma 3.32. *Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be a regular tree grammar. Assume that \mathcal{G} is essentially strongly-connected and $\#(\mathcal{L}(\mathcal{G})) = \infty$. Then there exist constants $n_0, c > 0$ and a family $(d_{N,N'})_{N,N' \in \mathcal{N}^{\text{inf}}}$ of natural numbers $0 \leq d_{N,N'} < c$ that satisfy the following conditions:*

- (1) *For every $N, N' \in \mathcal{N}^{\text{inf}}$, if $\mathcal{L}_n(\mathcal{G}, N \Rightarrow N') \neq \emptyset$, then $n \equiv d_{N,N'} \pmod{c}$.*
- (2) *The converse of (1) holds for sufficiently large n : for any $N, N' \in \mathcal{N}^{\text{inf}}$ and $n \geq n_0$, if $n \equiv d_{N,N'} \pmod{c}$ then $\mathcal{L}_n(\mathcal{G}, N \Rightarrow N') \neq \emptyset$.*
- (3) *$d_{N,N} = 0$ for every $N \in \mathcal{N}^{\text{inf}}$.*
- (4) *$d_{N,N'} + d_{N',N''} \equiv d_{N,N''} \pmod{c}$ for every $N, N', N'' \in \mathcal{N}^{\text{inf}}$.*

The proof of the above lemma is rather involved; we defer it to Appendix C. We give some examples below, to clarify what the lemma means.

Example 3.33. Consider the grammar \mathcal{G}_1 consisting of the following rewriting rules:

$$A \longrightarrow \mathbf{a}(B) \quad B \longrightarrow \mathbf{b}(A) \quad A \longrightarrow \mathbf{c}(C) \quad C \longrightarrow \mathbf{c}$$

Then $\mathcal{N}^{\text{inf}} = \{A, B\}$ and the conditions of the lemma above hold for:

$$n_0 = 0 \quad c = 2 \quad d_{A,A} = d_{B,B} = 0 \quad d_{A,B} = d_{B,A} = 1.$$

In fact, $\mathcal{L}(\mathcal{G}_1, A \Rightarrow A) = \{(\mathbf{ab})^k[] \mid k \geq 0\}$, $\mathcal{L}(\mathcal{G}_1, B \Rightarrow B) = \{(\mathbf{ba})^k[] \mid k \geq 0\}$, $\mathcal{L}(\mathcal{G}_1, B \Rightarrow A) = \{(\mathbf{ab})^k \mathbf{a}[] \mid k \geq 0\}$, and $\mathcal{L}(\mathcal{G}_1, A \Rightarrow B) = \{(\mathbf{ba})^k \mathbf{b}[] \mid k \geq 0\}$. Here, since the arities of \mathbf{a} and \mathbf{b} are 1, we have used regular expressions to denote linear contexts.

Consider the grammar \mathcal{G}_2 , obtained by adding the following rules to the grammar above.

$$A \longrightarrow \mathbf{a}(A_1) \quad A_1 \longrightarrow \mathbf{a}(A_2) \quad A_2 \longrightarrow \mathbf{a}(A).$$

Then, the conditions of the lemma above holds for $n = 6$, $c = 1$, $d_{N,N'} = 0$ for $N, N' \in \{A, A_1, A_2, B\}$. Note that $\mathcal{L}(\mathcal{G}_2, A_2 \Rightarrow A_1) = \{\mathbf{a}^2(\mathbf{a}^3|\mathbf{ab})^k \mathbf{a}^2[] \mid k \geq 0\}$. \square

Remark 3.34. With some additional assumptions on a grammar, Lemma 3.32 above can be easily proved. For example, consider a canonical, unambiguous and essentially strongly-connected grammar \mathcal{G} and assume that (i) \mathcal{G} is N -aperiodic for some N and (ii) there exist a 2-context C and nonterminals $N, N_1, N_2 \in \mathcal{N}^{\text{inf}}$ such that $N \rightarrow_{\mathcal{G}}^* C[N_1, N_2]$. Then Lemma 3.32 for the grammar \mathcal{G} trivially holds with $c = 1$. In fact, this simpler approach is essentially what we adopted in the conference version [18] of this article.

Using the lemma above, we prove that for every $S \in \mathcal{S}^{\text{inf}}(\mathcal{G})$ and any sufficiently large n such that $\mathcal{L}_n(\mathcal{G}, \kappa) \neq \emptyset$, we can find a context $U \in \mathcal{L}_n(\mathcal{G}, \kappa)$ such that $S \preceq U$ (Lemma 3.35 below).

Lemma 3.35. *Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be a regular tree grammar in canonical form. Assume that \mathcal{G} is unambiguous and essentially strongly-connected and $\#(\mathcal{L}(\mathcal{G})) = \infty$. Then there exists a constant $n_0 \in \mathbb{N}$ that satisfies the following condition: For every*

- $S \in \mathcal{S}^{\text{inf}}(\mathcal{G})$,
 - $n \geq n_0 + |S|$, and
 - $\kappa = (\Rightarrow N')$ or $(N \Rightarrow N')$ where $N, N' \in \mathcal{N}^{\text{inf}}$,
- if $\mathcal{L}_n(\mathcal{G}, \kappa) \neq \emptyset$, then there exists $U \in \mathcal{L}_n(\mathcal{G}, \kappa)$ with $S \preceq U$.*

Proof. First, let us choose the following constants:

- $m_0 \in \mathbb{N}$ such that, for every $N, N' \in \mathcal{N}^{\text{inf}}$, there exists $S_{N, N'} \in \mathcal{L}(\mathcal{G}, N \Rightarrow N')$ with $|S_{N, N'}| \leq m_0$. The existence of m_0 is a consequence of essential strong-connectivity and of finiteness of \mathcal{N} .
- m_1 which is the constant n_0 of Lemma 3.32.
- $m_2 \in \mathbb{N}$ such that, for every $N \in \mathcal{N} \setminus \mathcal{N}^{\text{inf}}$ and every $T \in \mathcal{L}(\mathcal{G}, N)$, we have $|T| < m_2$. The existence of m_2 follows from the fact that $\bigcup_{N \in \mathcal{N} \setminus \mathcal{N}^{\text{inf}}} \mathcal{L}(\mathcal{G}, N)$ is a finite set.

Let c and $(d_{N, N'})_{N, N' \in \mathcal{N}^{\text{inf}}}$ be the constant and the family obtained by Lemma 3.32.

We define $n_1 \triangleq m_0 + m_1$ and $n_2 \triangleq m_0 + m_1 + rm_2 + 1$, where $r \triangleq \max(\text{Im}(\Sigma))$. Below we shall show: (i) the current lemma for the case $\kappa = (N \Rightarrow N')$, by setting $n_0 \triangleq n_1$ and then (ii) the lemma for the case $\kappa = (\Rightarrow N')$, by setting $n_0 \triangleq n_2$; we use (i) to show (ii). The whole lemma then follows immediately from (i) and (ii) with $n_0 \triangleq \max\{n_1, n_2\} (= n_2)$.

- Case (i): We define $n_0 \triangleq n_1 = m_0 + m_1$. Assume that: $S \in \mathcal{L}(\mathcal{G}, N_1 \Rightarrow N_2)$ where $N_1, N_2 \in \mathcal{N}^{\text{inf}}$; $n \geq n_0 + |S|$; $\kappa = (N \Rightarrow N')$ where $N, N' \in \mathcal{N}^{\text{inf}}$; and $\mathcal{L}_n(\mathcal{G}, \kappa) \neq \emptyset$. Let $S_1 \in \mathcal{L}(\mathcal{G}, N_2 \Rightarrow N')$ with $|S_1| \leq m_0$. Then $S_1[S] \in \mathcal{L}(\mathcal{G}, N_1 \Rightarrow N')$. It suffices to show that $\mathcal{L}_{n-|S_1[S]|}(\mathcal{G}, N \Rightarrow N_1) \neq \emptyset$. By Lemma 3.32(1)(4),

$$\begin{aligned} n &\equiv d_{N, N'} \equiv d_{N, N_1} + d_{N_1, N_2} + d_{N_2, N'} \pmod{c} & (\because \mathcal{L}_n(\mathcal{G}, \kappa) = \mathcal{L}_n(\mathcal{G}, N \Rightarrow N') \neq \emptyset) \\ |S_1| &\equiv d_{N_2, N'} \pmod{c} & (\because (S_1 \in) \mathcal{L}_{|S_1|}(\mathcal{G}, N_2 \Rightarrow N') \neq \emptyset) \\ |S| &\equiv d_{N_1, N_2} \pmod{c} & (\because (S \in) \mathcal{L}_{|S|}(\mathcal{G}, N_1 \Rightarrow N_2) \neq \emptyset) \end{aligned}$$

and thus

$$n - |S_1[S]| \equiv d_{N, N_1} \pmod{c}.$$

Since $n \geq m_0 + m_1 + |S|$, we have $n - |S_1[S]| \geq m_1$, and hence by Lemma 3.32(2), we have $\mathcal{L}_{n-|S_1[S]|}(\mathcal{G}, N \Rightarrow N_1) \neq \emptyset$.

- Case (ii): We define $n_0 \triangleq n_2 = m_0 + m_1 + rm_2 + 1$. Assume that: $S \in \mathcal{L}(\mathcal{G}, N_1 \Rightarrow N_2)$ where $N_1, N_2 \in \mathcal{N}^{\text{inf}}$; $n \geq n_0 + |S|$; $\kappa = (\Rightarrow N')$ where $N' \in \mathcal{N}^{\text{inf}}$; and $\mathcal{L}_n(\mathcal{G}, \kappa) \neq \emptyset$. Let $T \in \mathcal{L}_n(\mathcal{G}, \kappa)$. Since the arity of terminal symbols are bounded by r and $|T| \geq rm_2 + 1$,

there exists a subtree $T_0 \preceq T$ such that $m_2 \leq |T_0| \leq rm_2 + 1$, which can be shown by induction on tree T . Let U be a linear context such that $T = U[T_0]$. Since \mathcal{G} is canonical and unambiguous, by Lemma 3.25, $U \in \mathcal{L}(\mathcal{G}, N \Rightarrow N')$ and $T_0 \in \mathcal{L}(\mathcal{G}, N)$ for some $N \in \mathcal{N}$. Since $m_2 \leq |T_0| \leq rm_2 + 1$, we have $N \in \mathcal{N}^{\text{inf}}$ and $|U| \geq m_0 + m_1 + |S|$. By using the case (1), since $(U \in) \mathcal{L}_{|U|}(\mathcal{G}, N \Rightarrow N') \neq \emptyset$, there exists $U' \in \mathcal{L}_{|U|}(\mathcal{G}, N \Rightarrow N')$ with $S \preceq U'$. Let $T' \triangleq U'[T_0]$; then $T' \in \mathcal{L}_n(\mathcal{G}, \Rightarrow N')$ and $S \preceq T'$. \square

We prepare another lemma.

Lemma 3.36. *If $\Phi_m(T) = (U, E, P)$, $\ell \in \{1, \dots, \text{shn}(E)\}$, $P.\ell$ is a linear context, and the second-order hole $E.\ell$ occurs in E in the form $(E.\ell)[E']$, then $|E'| \geq m$.*

Proof. By straightforward induction on $|T|$ and case analysis of Equation (3.2) in the definition of Φ_m . We can use Lemma 3.11 in the second case of Equation (3.2) when $\ell = 1$; all the other cases immediately follow from induction hypothesis. \square

We are now ready to prove the main lemma of this subsection.

Proof of Lemma 3.31. Since $S_0 \in \mathcal{S}^{\text{inf}}(\mathcal{G})$, we have $\mathcal{N}^{\text{inf}} \neq \emptyset$ and hence $\#(\mathcal{L}(\mathcal{G})) = \infty$. Let n_0 be the constant of Lemma 3.35. Let c_1 be a positive integer such that $|S_n| \leq c_1 n$ for every $n \in \mathbb{N}$. We define $c \triangleq (2r + 1)c_1$ (recall that r is the largest arity of Σ) and choose b so that $b \geq n_0$ and $b > |T|$ for any $T \in \bigcup_{N \in \mathcal{N} \setminus \mathcal{N}^{\text{inf}}} \mathcal{L}(\mathcal{G}, N)$.

Assume that $N \in \mathcal{N}^{\text{inf}}$, $n \geq 1$, $m \geq b$, $\tilde{E} \in \tilde{\mathcal{E}}_n^{cm}(\mathcal{G}, N)$ and $i \in \{1, \dots, \text{shn}(\tilde{E})\}$. Let $\tilde{E}.i = \llbracket \rrbracket_{\kappa'}^{n'}$. We need to show that there exists $U \in \tilde{\mathcal{U}}_{\llbracket \rrbracket_{\kappa'}^{n'}}^{cm}(\mathcal{G})$ such that $S_m \preceq U$.

Since $\tilde{E} \in \tilde{\mathcal{E}}_n^{cm}(\mathcal{G}, N)$, there exist $T \in \mathcal{L}_n(\mathcal{G}, N)$ and P such that $(\tilde{E}, P) = \tilde{\Phi}_{cm}^{\bullet, \mathcal{G}}(T, N)$; hence $P.i \in \tilde{\mathcal{U}}_{\tilde{E}.i}^{cm}(\mathcal{G}) = \tilde{\mathcal{U}}_{\llbracket \rrbracket_{\kappa'}^{n'}}^{cm}(\mathcal{G})$. The affine context $P.i$ must be of the form $a(U_1, \dots, U_{\Sigma(a)})$.

Since $P.i$ is good for cm , we have $|P.i| \geq cm = (2r + 1)c_1 m \geq 2rc_1 m + 1$. Since $\Sigma(a) \leq r$, there exists $j \leq \Sigma(a)$ such that $|U_j| \geq 2c_1 m$. We have $\kappa' = (N' \Rightarrow N'')$ or $(\Rightarrow N'')$ for some $N', N'' \in \mathcal{N}$. Since $P.i = a(U_1, \dots, U_{\Sigma(a)}) \in \mathcal{L}(\mathcal{G}, \kappa')$, there exist $N_1, \dots, N_{\Sigma(a)} \in \mathcal{N}$ such that

$$N'' \longrightarrow a(N_1, \dots, N_{\Sigma(a)}) \longrightarrow^* T'$$

where $T' = P.i[N']$ if $\kappa' = (N' \Rightarrow N'')$, and $T' = P.i$ if $\kappa' = (\Rightarrow N'')$. Let κ be $N' \Rightarrow N_j$ if U_j is a linear context, and $\Rightarrow N_j$ otherwise. Then $U_j \in \mathcal{L}_{|U_j|}(\mathcal{G}, \kappa) \neq \emptyset$. In order to apply Lemma 3.35 (for $S = S_m$ and $n = |U_j|$), we need to check the conditions: (i) $|U_j| \geq n_0 + |S_m|$ and (ii) κ consists of only nonterminals in \mathcal{N}^{inf} . Condition (i) follows immediately from $|U_j| \geq 2c_1 m \geq m + c_1 m \geq n_0 + |S_m|$. As for (ii), it suffices to check that $\mathcal{L}(\mathcal{G}, N_j)$ and $\mathcal{L}(\mathcal{G}, N')$ contain a tree whose size is no less than b (where the condition on $\mathcal{L}(\mathcal{G}, N')$ is required only if U_j is a linear context). The condition on $\mathcal{L}(\mathcal{G}, N_j)$ follows from $|U_j| \geq 2c_1 m \geq m \geq b$. If U_j is a linear context, by Lemma 3.36, there exist S and T'' such that:

$$T = S[(P.i)[T'']] \quad |T''| \geq cm \geq b \quad T'' \in \mathcal{L}(\mathcal{G}, N')$$

as required.

Thus, we can apply Lemma 3.35 and obtain $U'_j \in \mathcal{L}(\mathcal{G}, \kappa)$ such that $S_m \preceq U'_j$ and $|U'_j| = |U_j|$. Since $P.i = a(U_1, \dots, U_{\Sigma(a)})$ is good for cm , $U \triangleq a(U_1, \dots, U_{j-1}, U'_j, U_{j+1}, \dots, U_{\Sigma(a)})$ is also good for cm . Obviously $|U| = |P.i|$ and thus $U \in \tilde{\mathcal{U}}_{\llbracket \rrbracket_{\kappa'}^{n'}}^{cm}(\mathcal{G})$. Since $S_m \preceq U'_j$ and $U'_j \preceq U$, we have $S_m \preceq U$ as required. \square

3.6. Main Proof. Here, we give a proof of Theorem 2.13. Before the proof, we prepare a simple lemma. We write $\mathcal{U}_{\leq n}(\Sigma)$ for the set of affine contexts over Σ of size at most n . Lemma 3.37 below gives an upper bound of $\#(\mathcal{U}_{\leq n}(\Sigma))$. A more precise bound can be obtained by using a technique of analytic combinatorics such as Drmota–Lalley–Woods theorem (cf. [5], Theorem VII.6), but the rough bound provided by the lemma below is sufficient for our purpose.

Lemma 3.37. *For every ranked alphabet Σ , there exists a real constant $\gamma > 1$ such that*

$$\#(\mathcal{U}_{\leq n}(\Sigma)) \leq \gamma^n$$

for every $n \geq 0$.

Proof. Let A be the set of symbols: $\{\dot{a}, \acute{a} \mid a \in \text{Dom}(\Sigma)\} \cup \{\square\}$. Intuitively \dot{a} and \acute{a} are opening and closing tags of an XML-like language.

We can transform an affine context U to its XML-like string representation $U^\dagger \in A^*$ by:

$$\begin{aligned} (a(U_1, \dots, U_{\Sigma(a)}))^\dagger &\triangleq \dot{a} U_1^\dagger \cdots U_{\Sigma(a)}^\dagger \acute{a} \\ [\]^\dagger &\triangleq \square. \end{aligned}$$

Obviously, $(\cdot)^\dagger$ is injective. Furthermore, $|U^\dagger| = 2|U|$ if U is a 0-context (i.e., a tree), and $|U^\dagger| = 2|U| + 1$ if U is a linear context (note that the size of the hole $[\]$ is zero, but its word representation is of length 1). Thus, for $n > 0$, we have

$$\#(\mathcal{U}_{\leq n}(\Sigma)) \leq \sum_{i=0}^{2n+1} (\#(A))^i \leq \sum_{i=0}^{3n} (\#(A))^i \leq (\#(A) + 1)^{3n} = ((\#(A) + 1)^3)^n.$$

If $n = 0$, then $\#(\mathcal{U}_{\leq n}(\Sigma)) = 1 = ((\#(A) + 1)^3)^n$, as $\mathcal{U}_{\leq n}(\Sigma)$ is the singleton set $\{[\]\}$. Thus, the required result holds for $\gamma = (\#(A) + 1)^3$. \square

The following lemma is a variant of Theorem 2.13, specialized to a canonical grammar.

Lemma 3.38. *Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be a canonical, unambiguous, and essentially strongly-connected regular tree grammar such that $\#(\mathcal{L}(\mathcal{G})) = \infty$, and $(S_n)_{n \in \mathbb{N}}$ be a family of linear contexts in $\mathcal{S}^{\text{inf}}(\mathcal{G})$ such that $|S_n| = O(n)$. Then there exists a real constant $p > 0$ such that for any $N \in \mathcal{N}^{\text{inf}}$,*

$$\lim_{n \rightarrow \infty} \frac{\#(\{T \in \mathcal{L}_n(\mathcal{G}, N) \mid S_{\lfloor p \log n \rfloor} \preceq T\})}{\#(\mathcal{L}_n(\mathcal{G}, N))} = 1.$$

Proof. The overall structure of the proof is the same as that of Proposition 3.1. Let Z_n be

$$1 - \frac{\#(\{T \in \mathcal{L}_n(\mathcal{G}, N) \mid S_{\lfloor p \log n \rfloor} \preceq T\})}{\#(\mathcal{L}_n(\mathcal{G}, N))} = \frac{\#(\{T \in \mathcal{L}_n(\mathcal{G}, N) \mid S_{\lfloor p \log n \rfloor} \not\preceq T\})}{\#(\mathcal{L}_n(\mathcal{G}, N))}.$$

It suffices to show that Z_n converges to 0.

By Lemma 3.27, we have

$$\mathcal{L}_n(\mathcal{G}, N) \cong \coprod_{\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)} \prod_{i=1}^{\text{shn}(\tilde{E})} \tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G})$$

for any $m > 0$. Thus, we have

$$Z_n \leq \frac{\sum_{\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)} \prod_{i=1}^{\text{shn}(\tilde{E})} \# \left(\{U \in \tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \mid S_{\lceil p \log n \rceil} \not\leq U\} \right)}{\sum_{\tilde{E} \in \tilde{\mathcal{E}}_n^{m(n)}(\mathcal{G}, N)} \prod_{i=1}^{\text{shn}(\tilde{E})} \# \left(\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \right)}. \quad (3.4)$$

Let $b, c \geq 1$ be the numbers in Lemma 3.31. Then, by the lemma, for any n and p such that $\lceil p \log n \rceil \geq b$, each $\tilde{\mathcal{U}}_{\tilde{E}.i}^{c \lceil p \log n \rceil}(\mathcal{G})$ contains at least one U that has $S_{\lceil p \log n \rceil}$ as a subcontext.

Thus, for $m = c \lceil p \log n \rceil$, $\# \left(\{U \in \tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \mid S_{\lceil p \log n \rceil} \not\leq U\} \right)$ is bounded above by:

$$\# \left(\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \right) - 1 = \left(1 - \frac{1}{\# \left(\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \right)} \right) \# \left(\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \right) \leq \left(1 - \frac{1}{\gamma^{rm}} \right) \# \left(\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \right).$$

Here γ is the constant (that only depends on Σ) of Lemma 3.37, and r is the largest arity of Σ . In the last inequality, we have used the fact that $\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \subseteq \mathcal{U}_{\leq rm}(\Sigma)$.

By using the upper-bound above, Equation (3.4), and $\text{shn}(\tilde{E}) \geq \frac{n}{2rm}$ (Lemma 3.7), we have:

$$\begin{aligned} Z_n &\leq \frac{\sum_{\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N)} \prod_{i=1}^{\text{shn}(\tilde{E})} \left(1 - \frac{1}{\gamma^{rm}} \right) \# \left(\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \right)}{\sum_{\tilde{E} \in \tilde{\mathcal{E}}_n^{m(n)}(\mathcal{G}, N)} \prod_{i=1}^{\text{shn}(\tilde{E})} \# \left(\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \right)} \\ &= \frac{\sum_{\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N) \left(1 - \frac{1}{\gamma^{rm}} \right)^{\text{shn}(\tilde{E})} \prod_{i=1}^{\text{shn}(\tilde{E})} \# \left(\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \right)}{\sum_{\tilde{E} \in \tilde{\mathcal{E}}_n^{m(n)}(\mathcal{G}, N)} \prod_{i=1}^{\text{shn}(\tilde{E})} \# \left(\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \right)} \\ &\leq \frac{\sum_{\tilde{E} \in \tilde{\mathcal{E}}_n^m(\mathcal{G}, N) \left(1 - \frac{1}{\gamma^{rm}} \right)^{\frac{n}{2rm}} \prod_{i=1}^{\text{shn}(\tilde{E})} \# \left(\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \right)}{\sum_{\tilde{E} \in \tilde{\mathcal{E}}_n^{m(n)}(\mathcal{G}, N)} \prod_{i=1}^{\text{shn}(\tilde{E})} \# \left(\tilde{\mathcal{U}}_{\tilde{E}.i}^m(\mathcal{G}) \right)} = \left(1 - \frac{1}{\gamma^{rm}} \right)^{\frac{n}{2rm}} \end{aligned}$$

for any n and p such that $\lceil p \log n \rceil \geq b$ and $m = c \lceil p \log n \rceil$.

It remains to choose p so that $\left(1 - \frac{1}{\gamma^{rm}} \right)^{\frac{n}{2rm}} = \left(1 - \frac{1}{\gamma^{rc \lceil p \log n \rceil}} \right)^{\frac{n}{2rc \lceil p \log n \rceil}}$ converges to 0. Let us choose positive real numbers a, p , and q so that p and q satisfy the following conditions for every $n \geq a$:

$$p \log n \geq b \quad (3.5)$$

$$\gamma^{rc \lceil p \log n \rceil} \leq n^q \quad (3.6)$$

$$q < 1. \quad (3.7)$$

For example, we can choose a, p, q as follows:

$$a = \max \{ \gamma^{b(rc+2)}, \gamma^{rc(rc+2)} \} \quad p = \frac{1}{(rc+2) \log \gamma} \quad q = \frac{rc+1}{rc+2}$$

In fact, condition (3.6) follows from:

$$\begin{aligned} rc[p \log n] - \log_\gamma n^q &\leq rc\left(\frac{1}{(rc+2)\log \gamma} \log n + 1\right) - \frac{rc+1}{rc+2} \frac{\log n}{\log \gamma} \\ &= \frac{1}{(rc+2)\log \gamma} (rc \log n + rc(rc+2) \log \gamma - (rc+1) \log n) \\ &= \frac{1}{(rc+2)\log \gamma} (rc(rc+2) \log \gamma - \log n) \\ &\leq \frac{1}{(rc+2)\log \gamma} (rc(rc+2) \log \gamma - \log a) \leq 0. \end{aligned}$$

Thus, for $n \geq a$, we have:

$$\left(1 - \frac{1}{\gamma^{rc[p \log n]}}\right)^{\frac{n}{2rc[p \log n]}} \leq \left(1 - \frac{1}{n^q}\right)^{\frac{n}{2rc[p \log n]}} \leq \left(\left(1 - \frac{1}{n^q}\right)^{\frac{n}{\log n}}\right)^{\frac{1}{3rcp}}.$$

Since $\lim_{n \rightarrow \infty} \left(1 - \frac{1}{n^q}\right)^{\frac{n}{\log n}} = 0$, we have $\liminf_{n \rightarrow \infty} Z_n = 0$ as required. \square

Remark 3.39. In the proof above, we used the fact that if $0 < q < 1$ then

$$\lim_{n \rightarrow \infty} \left(1 - \frac{1}{n^q}\right)^{\frac{n}{\log n}} = 0.$$

We also remark that if $q \geq 1$ then

$$\lim_{n \rightarrow \infty} \left(1 - \frac{1}{n^q}\right)^{\frac{n}{\log n}} = 1.$$

Thus, p should be chosen to be sufficiently small so that Equation (3.6) in the proof holds for some $q < 1$.

We are now ready to prove Theorem 2.13. We restate the theorem.

Theorem 2.13 (Parameterized Infinite Monkey Theorem for Regular Tree Languages). *Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be an unambiguous and strongly-connected regular tree grammar such that $\#(\mathcal{L}(\mathcal{G})) = \infty$, and $(S_n)_{n \in \mathbb{N}}$ be a family of linear contexts in $\mathcal{S}(\mathcal{G})$ such that $|S_n| = O(n)$. Then there exists a real constant $p > 0$ such that for any $N \in \mathcal{N}$ the following equation holds:*

$$\liminf_{n \rightarrow \infty} \frac{\#(\{T \in \mathcal{L}_n(\mathcal{G}, N) \mid S_{[p \log n]} \preceq T\})}{\#(\mathcal{L}_n(\mathcal{G}, N))} = 1.$$

Proof. By Lemma 3.19, there exists a canonical, unambiguous and essentially strongly-connected grammar $\mathcal{G}' = (\Sigma, \mathcal{N}', \mathcal{R}')$ and a family $(\mathcal{I}_N)_{N \in \mathcal{N}}$ of subsets $\mathcal{I}_N \subseteq \mathcal{N}'$ such that $\mathcal{L}(\mathcal{G}, N) = \biguplus_{N' \in \mathcal{I}_N} \mathcal{L}(\mathcal{G}', N')$ for every $N \in \mathcal{N}$ and $\mathcal{S}(\mathcal{G}) \subseteq \mathcal{S}^{\text{inf}}(\mathcal{G}')$. Let $\mathcal{I}'_N \triangleq \mathcal{I}_N \cap \mathcal{N}'^{\text{inf}} = \{N' \in \mathcal{I}_N \mid \mathcal{L}(\mathcal{G}', N') = \infty\}$. For any $N \in \mathcal{N}$, since $\mathcal{L}(\mathcal{G}) = \infty$ and \mathcal{G} is strongly connected, we have $\mathcal{L}(\mathcal{G}, N) = \infty$, and hence $\mathcal{I}'_N \neq \emptyset$. By Lemma 3.38, there exists a real constant $p > 0$ such that, for each $N' \in \mathcal{I}'_N$,

$$\liminf_{n \rightarrow \infty} \frac{\#(\{T \in \mathcal{L}_n(\mathcal{G}', N') \mid S_{[p \log n]} \preceq T\})}{\#(\mathcal{L}_n(\mathcal{G}', N'))} = 1. \quad (3.8)$$

Thus, we have

$$\begin{aligned}
& \limdef_{n \rightarrow \infty} \frac{\#(\{T \in \mathcal{L}_n(\mathcal{G}, N) \mid S_{[p \log n]} \preceq T\})}{\#(\mathcal{L}_n(\mathcal{G}, N))} \\
&= \limdef_{n \rightarrow \infty} \frac{\sum_{N' \in \mathcal{I}_N} \#(\{T \in \mathcal{L}_n(\mathcal{G}', N') \mid S_{[p \log n]} \preceq T\})}{\sum_{N' \in \mathcal{I}_N} \#(\mathcal{L}_n(\mathcal{G}', N'))} \\
&= \limdef_{n \rightarrow \infty} \frac{\sum_{N' \in \mathcal{I}'_N} \#(\{T \in \mathcal{L}_n(\mathcal{G}', N') \mid S_{[p \log n]} \preceq T\})}{\sum_{N' \in \mathcal{I}'_N} \#(\mathcal{L}_n(\mathcal{G}', N'))} \\
&= 1
\end{aligned}$$

as required; we have used Lemma 2.12 and Equation (3.8) in the last step. \square

4. PROOF OF THE MAIN THEOREM ON λ -CALCULUS

This section proves our main theorem (Theorem 2.5). We first prepare a regular tree grammar that generates the set of tree representations of elements of $\Lambda(\delta, \iota, \xi)$ in Section 4.1, and then apply Corollary 2.17 to obtain Theorem 2.5, where $(T_n)_{n \in \mathbb{N}}$ in the corollary are set to (the tree representations of) the terms in the introduction that have long β -reduction sequences.

4.1. Regular Tree Grammar $\mathcal{G}(\delta, \iota, \xi)$ of $\Lambda(\delta, \iota, \xi)$. Recall that $\Lambda(\delta, \iota, \xi)$ is the set of (α -equivalence classes of) closed well-typed terms, whose order, internal arity, and number of variables are bounded above by δ , ι , and ξ (consult Definition 2.4 for the precise definition). The set $\Lambda(\delta, \iota, \xi)$ can be generated by the following grammar (up to isomorphism).

Definition 4.1 (grammar of $\Lambda(\delta, \iota, \xi)$). Let $\delta, \iota, \xi \geq 0$ be integers and $X_\xi = \{x_1, \dots, x_\xi\}$ 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:

$$\begin{aligned}
\Sigma(\delta, \iota, \xi) &\triangleq \{x \mapsto 0 \mid x \in X_\xi\} \cup \{ @ \mapsto 2 \} \\
&\quad \cup \{ \lambda \bar{x}^\tau \mapsto 1 \mid \bar{x} \in \{*\} \cup X_\xi, \tau \in \text{Types}(\delta - 1, \iota) \} \\
\mathcal{N}(\delta, \iota, \xi) &\triangleq \{ N_{\langle \Gamma; \tau \rangle} \mid \tau \in \text{Types}(\delta, \iota), \text{Dom}(\Gamma) \subseteq X_\xi, \text{Im}(\Gamma) \subseteq \text{Types}(\delta - 1, \iota), \\
&\quad \Lambda(\langle \Gamma; \tau \rangle, \delta, \iota, \xi) \neq \emptyset \} \\
\mathcal{R}(\delta, \iota, \xi) &\triangleq \{ N_{\langle \{x_i; \tau\}; \tau \rangle} \longrightarrow x_i \} \cup \{ N_{\langle \Gamma; \sigma \rightarrow \tau \rangle} \longrightarrow \lambda *^\sigma (N_{\langle \Gamma; \tau \rangle}) \} \\
&\quad \cup \{ N_{\langle \Gamma; \sigma \rightarrow \tau \rangle} \longrightarrow \lambda x_i^\sigma (N_{\langle \Gamma \cup \{x_i; \sigma\}; \tau \rangle}) \mid i = \min\{j \mid x_j \in X_\xi \setminus \text{Dom}(\Gamma)\} \} \\
&\quad \cup \{ N_{\langle \Gamma; \tau \rangle} \longrightarrow @ (N_{\langle \Gamma_1; \sigma \rightarrow \tau \rangle}, N_{\langle \Gamma_2; \sigma \rangle}) \mid \Gamma = \Gamma_1 \cup \Gamma_2 \}
\end{aligned}$$

The grammar above generates the tree representations of elements of $\Lambda(\delta, \iota, \xi)$, where a variable x , a lambda-abstraction, and an application are represented respectively as the nullary tree constructor x , unary tree constructor $\lambda \bar{x}$, and binary tree constructor $@$. The nonterminal $N_{\langle \Gamma; \tau \rangle}$ is used to generate (the tree representations of) the elements of $\Lambda(\langle \Gamma; \tau \rangle, \delta, \iota, \xi)$; the condition $\Lambda(\langle \Gamma; \tau \rangle, \delta, \iota, \xi) \neq \emptyset$ on nonterminal $N_{\langle \Gamma; \tau \rangle}$ ensures that every nonterminal generates at least one tree. To guarantee that the grammar generates at most one tree for each α -equivalence class $[t]_\alpha$, (i) variables are chosen from the fixed set X_ξ , and (ii) in the rule for generating a λ -abstraction, a variable is chosen in a deterministic manner.

Note that $\Sigma(\delta, \iota, \xi)$, $\mathcal{N}(\delta, \iota, \xi)$ and $\mathcal{R}(\delta, \iota, \xi)$ are finite. The finiteness of $\mathcal{N}(\delta, \iota, \xi)$ follows from that of X_ξ , $\text{Types}(\delta - 1, \iota)$, and $\{\Gamma \mid \text{Dom}(\Gamma) \subseteq X_\xi, \text{Im}(\Gamma) \subseteq \text{Types}(\delta - 1, \iota)\}$. The finiteness of $\mathcal{R}(\delta, \iota, \xi)$ also follows immediately from that of $\mathcal{N}(\delta, \iota, \xi)$.

Example 4.2. Let us consider the case where $\delta = \iota = \xi = 1$. The grammar $\mathcal{G}(1, 1, 1)$ consists of the following components.

$$\begin{aligned} \Sigma(1, 1, 1) &= \{x_1, @, \lambda x_1^\circ, \lambda *^\circ\} & \mathcal{N}(1, 1, 1) &= \{N_{\langle \emptyset; \sigma \rightarrow \sigma \rangle}, N_{\langle \{x_1 : \sigma\}; \sigma \rangle}, N_{\langle \{x_1 : \sigma\}; \sigma \rightarrow \sigma \rangle}\} \\ \mathcal{R}(1, 1, 1) &= \begin{cases} N_{\langle \emptyset; \sigma \rightarrow \sigma \rangle} \longrightarrow \lambda x_1^\circ(N_{\langle \{x_1 : \sigma\}; \sigma \rangle}) \\ N_{\langle \{x_1 : \sigma\}; \sigma \rangle} \longrightarrow x_1 \mid @ (N_{\langle \{x_1 : \sigma\}; \sigma \rightarrow \sigma \rangle}, N_{\langle \{x_1 : \sigma\}; \sigma \rangle}) \mid @ (N_{\langle \emptyset; \sigma \rightarrow \sigma \rangle}, N_{\langle \{x_1 : \sigma\}; \sigma \rangle}) \\ N_{\langle \{x_1 : \sigma\}; \sigma \rightarrow \sigma \rangle} \longrightarrow \lambda *^\circ(N_{\langle \{x_1 : \sigma\}; \sigma \rangle}). \end{cases} \end{aligned}$$

There is an obvious embedding $e^{(\delta, \iota, \xi)}$ (e for short) from trees in $\mathcal{T}(\Sigma(\delta, \iota, \xi))$ into (not necessarily well-typed) λ -terms. For $N_{\langle \Gamma; \tau \rangle} \in \mathcal{N}(\delta, \iota, \xi)$ we define

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

where $[-]_\alpha$ maps a term to its α -equivalence class. We sometimes omit the superscript and/or the subscript and may write just π for $\pi_{\langle \Gamma; \tau \rangle}^{(\delta, \iota, \xi)}$.

The following lemma says that $\mathcal{G}(\delta, \iota, \xi)$ gives a complete representation system of the α -equivalence classes.

Lemma 4.3. For $\delta, \iota, \xi \geq 0$, $\pi_{\langle \Gamma; \tau \rangle}^{(\delta, \iota, \xi)} : \mathcal{L}(\mathcal{G}(\delta, \iota, \xi), N_{\langle \Gamma; \tau \rangle}) \rightarrow \Lambda(\langle \Gamma; \tau \rangle, \delta, \iota, \xi)$ is a size-preserving bijection.

Proof. It is trivial that the image of $\pi_{\langle \Gamma; \tau \rangle}$ is contained in $\Lambda(\langle \Gamma; \tau \rangle, \delta, \iota, \xi)$ and π preserves the size.

The injectivity, i.e., $e(T) \sim_\alpha e(T')$ implies $T = T'$ for $T, T' \in \mathcal{L}(N_{\langle \Gamma; \tau \rangle})$, is shown by induction on the length of the leftmost rewriting sequence $N_{\langle \Gamma; \tau \rangle} \longrightarrow^* T$ and by case analysis of the rewriting rule used in the first step of the reduction sequence. In the case analysis below, we use the fact that if $T \in \mathcal{L}(N_{\langle \Gamma; \tau \rangle})$, then $\mathbf{FV}(e(T)) = \text{Dom}(\Gamma)$, which can be proved by straightforward induction.

- Case for the rule $N_{\langle \{x_i : \tau\}; \tau \rangle} \longrightarrow x_i$: In this case, $T = x_i$. By the assumption $e(T) \sim_\alpha e(T')$, $T = T'$ follows immediately.
- Case for the rule $N_{\langle \Gamma; \sigma \rightarrow \tau \rangle} \longrightarrow \lambda *^\sigma(N_{\langle \Gamma; \tau \rangle})$: In this case, $T = \lambda *^\sigma(T_1)$ with $N_{\langle \Gamma; \tau \rangle} \longrightarrow^* T_1$. By the assumption $e(T) \sim_\alpha e(T')$, T' must be of the form $\lambda \bar{x}^\sigma(T'_1)$ where (i) \bar{x} does not occur free in $e(T'_1)$, and (ii) $e(T_1) \sim_\alpha e(T'_1)$. Then $\bar{x} = *$ and hence we have $N_{\langle \Gamma; \sigma \rightarrow \tau \rangle} \longrightarrow \lambda *^\sigma(N_{\langle \Gamma; \tau \rangle})$ with $N_{\langle \Gamma; \tau \rangle} \longrightarrow^* T'_1$, because, if $\bar{x} \neq *$, we have $N_{\langle \Gamma; \sigma \rightarrow \tau \rangle} \longrightarrow \lambda \bar{x}^\sigma(N_{\langle \Gamma \cup \{\bar{x} : \sigma\}; \tau \rangle})$ with $N_{\langle \Gamma \cup \{\bar{x} : \sigma\}; \tau \rangle} \longrightarrow^* T'_1$, and hence $\mathbf{FV}(e(T'_1)) = \text{Dom}(\Gamma \cup \{\bar{x} : \sigma\})$, which contradicts the condition (i). Therefore, by the induction hypothesis, we have $T_1 = T'_1$, which implies $T = T'$ as required.
- Case for the rule: $N_{\langle \Gamma; \sigma \rightarrow \tau \rangle} \longrightarrow \lambda x_i^\sigma(N_{\langle \Gamma \cup \{x_i : \sigma\}; \tau \rangle})$, where $i = \min\{j \mid x_j \in X_\xi \setminus \text{Dom}(\Gamma)\}$. By the assumption $e(T) \sim_\alpha e(T')$, T' must be of the form $\lambda x^\sigma(T'_1)$ where x occurs free in $e(T'_1)$, and $[x'/x_i]e(T_1) \sim_\alpha [x'/x]e(T'_1)$ for a fresh variable x' . Thus, by the definition of $\mathcal{R}(\delta, \iota, \xi)$, T' must have also been generated by the same rule, i.e., $T' = \lambda x_i^\sigma(T'_1)$ with $N_{\langle \Gamma \cup \{x_i : \sigma\}; \tau \rangle} \longrightarrow^* T'_1$. By the induction hypothesis, we have $T_1 = T'_1$, which also implies $T = T'$ as required.

- Case for the rule: $N_{\langle\Gamma;\tau\rangle} \longrightarrow @ (N_{\langle\Gamma_1;\sigma\rightarrow\tau\rangle}, N_{\langle\Gamma_2;\sigma\rangle})$ where $\Gamma = \Gamma_1 \cup \Gamma_2$. Then $T = @(T_1, T_2)$ with $N_{\langle\Gamma_1;\sigma\rightarrow\tau\rangle} \longrightarrow^* T_1$ and $N_{\langle\Gamma_2;\sigma\rangle} \longrightarrow^* T_2$. By the assumption $e(T) \sim_\alpha e(T')$, T' must also be of the form $@(T'_1, T'_2)$, with $e(T_1) \sim_\alpha e(T'_1)$ and $e(T_2) \sim_\alpha e(T'_2)$. Therefore, T' must also have been generated by a rule for applications; hence $N_{\langle\Gamma'_1;\sigma'\rightarrow\tau\rangle} \longrightarrow^* T'_1$ and $N_{\langle\Gamma'_2;\sigma'\rangle} \longrightarrow^* T'_2$ for some $\Gamma'_1, \Gamma'_2, \sigma'$ such that $\Gamma = \Gamma'_1 \cup \Gamma'_2$. By the condition $e(T_i) \sim_\alpha e(T'_i)$, $\text{Dom}(\Gamma_i) = \mathbf{FV}(e(T_i)) = \mathbf{FV}(e(T'_i)) = \text{Dom}(\Gamma'_i)$ for each $i \in \{1, 2\}$. Thus, since $\Gamma_1 \cup \Gamma_2 = \Gamma'_1 \cup \Gamma'_2$, we have $\Gamma_i = \Gamma'_i$, which also implies $\sigma = \sigma'$ (since the type of a simply-typed λ -term is uniquely determined by the term and the type environment). Therefore, by the induction hypothesis, we have $T_i = T'_i$ for each $i \in \{1, 2\}$, which implies $T = T'$ as required.

Next we show the surjectivity, i.e., for any $[t]_\alpha \in \Lambda(\langle\Gamma;\tau\rangle, \delta, \iota, \xi)$ there exists $T \in \mathcal{L}(N_{\langle\Gamma;\tau\rangle})$ such that $e(T) \sim_\alpha t$. For δ, ι, ξ with $X_\xi = \{x_1, \dots, x_\xi\}$ and $N_{\langle\Gamma;\tau\rangle} \in \mathcal{N}(\delta, \iota, \xi)$, we define a “renaming” function $\rho_{\langle\Gamma;\tau\rangle}^{(\delta, \iota, \xi)}$ ($\rho_{\langle\Gamma;\tau\rangle}$ or ρ for short) from $\{t \mid [t]_\alpha \in \Lambda(\langle\Gamma;\tau\rangle, \delta, \iota, \xi)\}$ to $\mathcal{L}(N_{\langle\Gamma;\tau\rangle})$ by induction on the size of t , so that $\pi(\rho(t)) = [t]_\alpha$ holds.

$$\begin{aligned}
\rho_{\langle\{x:\tau\};\tau\rangle}(x) &\triangleq x \\
\rho_{\langle\Gamma;\sigma\rightarrow\tau\rangle}(\lambda^{\sigma}.t) &\triangleq \lambda^{\sigma}(\rho_{\langle\Gamma;\tau\rangle}(t)) \\
\rho_{\langle\Gamma;\sigma\rightarrow\tau\rangle}(\lambda x^{\sigma}.t) &\triangleq \lambda^{\sigma}(\rho_{\langle\Gamma;\tau\rangle}(t)) && (\text{if } x \notin \mathbf{FV}(t)) \\
\rho_{\langle\Gamma;\sigma\rightarrow\tau\rangle}(\lambda x^{\sigma}.t) &\triangleq \lambda x_i^{\sigma}(\rho_{\langle\Gamma \cup \{x_i:\sigma\};\tau\rangle}([x_i \leftrightarrow x]t)) && \text{where } i \triangleq \min\{j \mid x_j \in X_\xi \setminus \text{Dom}(\Gamma)\} \\
&&& (\text{if } x \in \mathbf{FV}(t)) \\
\rho_{\langle\Gamma_1 \cup \Gamma_2;\tau\rangle}(t_1 t_2) &\triangleq @(\rho_{\langle\Gamma_1;\sigma\rightarrow\tau\rangle}(t_1), \rho_{\langle\Gamma_2;\sigma\rangle}(t_2)) && (\text{if } \Gamma_1 \vdash t_1 : \sigma \rightarrow \tau \text{ and } \Gamma_2 \vdash t_2 : \sigma)
\end{aligned}$$

Here, $[x \leftrightarrow y]t$ represents the term obtained by swapping every occurrence of x with that of y ; for example, $[x \leftrightarrow y](\lambda x.x y) = \lambda y.y x$. Note that in the last clause, if $\Gamma \vdash t_1 t_2 : \tau$, then there exists a unique triple $(\Gamma_1, \Gamma_2, \sigma)$ such that $\Gamma = \Gamma_1 \cup \Gamma_2$, $\Gamma_1 \vdash t_1 : \sigma \rightarrow \tau$, and $\Gamma_2 \vdash t_2 : \sigma$. We can prove that $\pi(\rho(t)) = [t]_\alpha$ holds for every λ -term t such that $[t]_\alpha \in \Lambda(\langle\Gamma;\tau\rangle, \delta, \iota, \xi)$, by straightforward induction on the size of t . \square

Lemma 4.4. *For $\delta, \iota, \xi \geq 0$, $\mathcal{G}(\delta, \iota, \xi)$ is unambiguous.*

Proof. The proof is similar to that of the injectivity of Lemma 4.3. We show that, for any $N_{\langle\Gamma;\tau\rangle} \in \mathcal{N}(\delta, \iota, \xi)$, $T \in \mathcal{T}(\Sigma(\delta, \iota, \xi))$, and two leftmost rewriting sequences of the form

$$\begin{aligned}
N_{\langle\Gamma;\tau\rangle} &\longrightarrow T^{(1)} \longrightarrow \dots \longrightarrow T^{(n)} = T && (n \geq 1) \\
N_{\langle\Gamma;\tau\rangle} &\longrightarrow T'^{(1)} \longrightarrow \dots \longrightarrow T'^{(n')} = T && (n' \geq 1),
\end{aligned}$$

$n = n'$ and $T^{(k)} = T'^{(k)}$ hold for $1 \leq k < n$. The proof proceeds by induction on n , with case analysis on the rule $N_{\langle\Gamma;\tau\rangle} \longrightarrow T^{(1)}$. As in the proof of Lemma 4.3, we use the fact that if $T \in \mathcal{L}(N_{\langle\Gamma;\tau\rangle})$, then $\mathbf{FV}(e(T)) = \text{Dom}(\Gamma)$.

- Case for the rule $N_{\langle\{x_i:\tau\};\tau\rangle} \longrightarrow x_i$: In this case, $T = x_i$ and $n = 1$. Since the root of $T'^{(1)}$ must be x_i , we have $T'^{(1)} = T$ and $n' = 1 = n$.
- Case for the rule $N_{\langle\Gamma;\sigma\rightarrow\tau\rangle} \longrightarrow \lambda^{\sigma}(N_{\langle\Gamma;\tau\rangle})$: In this case, $n \geq 2$, $T^{(k)} = \lambda^{\sigma}(T_1^{(k)})$ ($k \leq n$), $T = \lambda^{\sigma}(T_1)$, and we have the following leftmost rewriting sequences:

$$N_{\langle\Gamma;\tau\rangle} = T_1^{(1)} \longrightarrow \dots \longrightarrow T_1^{(n)} = T_1.$$

Since the root of $T'^{(k)}$ is the same as that of T , $T'^{(k)} = \lambda *^\sigma(T_1'^{(k)})$ for $k \leq n'$, and we have the following leftmost rewriting sequence:

$$T_1'^{(1)} \longrightarrow \dots \longrightarrow T_1'^{(n')} = T_1.$$

By the definition of rules, $T'^{(1)}$ must be of the form $\lambda *^\sigma(N_{\langle \Gamma; \tau \rangle})$ and so $T_1'^{(1)} = N_{\langle \Gamma; \tau \rangle}$. By the induction hypothesis for $N_{\langle \Gamma; \tau \rangle} \longrightarrow T_1^{(2)}$, we have $n = n'$ and $T_1^{(k)} = T_1'^{(k)}$ for $2 \leq k < n$. Thus we have also $T^{(k)} = T'^{(k)}$ for $1 \leq k < n$, as required.

- Case for the rule: $N_{\langle \Gamma; \sigma \rightarrow \tau' \rangle} \longrightarrow \lambda x_i^\sigma(N_{\langle \Gamma \cup \{x_i: \sigma\}; \tau' \rangle})$, where $i = \min\{j \mid x_j \in X_\xi \setminus \text{Dom}(\Gamma)\}$. In this case, $n \geq 2$, $T^{(k)} = \lambda x_i^\sigma(T_1^{(k)})$ ($k \leq n$), $T = \lambda x_i^\sigma(T_1)$, and we have the following leftmost rewriting sequences:

$$N_{\langle \Gamma \cup \{x_i: \sigma\}; \tau' \rangle} = T_1^{(1)} \longrightarrow \dots \longrightarrow T_1^{(n)} = T_1.$$

Since the root of $T'^{(k)}$ are the same as that of T , $T'^{(k)} = \lambda x_i^\sigma(T_1'^{(k)})$ ($k \leq n'$), and we have the leftmost rewriting sequence:

$$T_1'^{(1)} \longrightarrow \dots \longrightarrow T_1'^{(n')} = T_1.$$

By the definition of rules, $N_{\langle \Gamma; \sigma \rightarrow \tau' \rangle} \longrightarrow \lambda x_i^\sigma(T_1'^{(1)})$ implies that $T_1'^{(1)} = N_{\langle \Gamma \cup \{x_i: \sigma\}; \tau' \rangle}$. Hence by the induction hypothesis for $N_{\langle \Gamma \cup \{x_i: \sigma\}; \tau' \rangle} \longrightarrow T_1^{(2)}$, we have $n = n'$ and $T_1^{(k)} = T_1'^{(k)}$ for $2 \leq k < n$. Thus we also have $T^{(k)} = T'^{(k)}$ for $1 \leq k < n$.

- Case for the rule: $N_{\langle \Gamma; \tau \rangle} \longrightarrow @ (N_{\langle \Gamma_1; \sigma \rightarrow \tau \rangle}, N_{\langle \Gamma_2; \sigma \rangle})$ where $\Gamma = \Gamma_1 \cup \Gamma_2$. In this case, $n = n_1 + n_2 - 1$, $n_1, n_2 \geq 2$, $T = @ (T_1, T_2)$,

$$\begin{aligned} T^{(k)} &= @ (T_1^{(k)}, N_{\langle \Gamma_2; \sigma \rangle}) & (1 \leq k \leq n_1) \\ T^{(n_1-1+k)} &= @ (T_1, T_2^{(k)}) & (1 \leq k \leq n_2), \end{aligned}$$

and we have the following leftmost rewriting sequences:

$$\begin{aligned} N_{\langle \Gamma_1; \sigma \rightarrow \tau \rangle} &= T_1^{(1)} \longrightarrow \dots \longrightarrow T_1^{(n_1)} = T_1 \\ N_{\langle \Gamma_2; \sigma \rangle} &= T_2^{(1)} \longrightarrow \dots \longrightarrow T_2^{(n_2)} = T_2. \end{aligned}$$

Since the root of $T'^{(1)}$ is $@$, $T'^{(1)}$ must be of the form $@ (N_{\langle \Gamma'_1; \sigma' \rightarrow \tau \rangle}, N_{\langle \Gamma'_2; \sigma' \rangle})$ with $\Gamma'_1 \cup \Gamma'_2 = \Gamma$. Also let us consider $T'^{(k)}$: we have $n = n'_1 + n'_2 - 1$, $n'_1, n'_2 \geq 2$,

$$\begin{aligned} T'^{(k)} &= @ (T_1'^{(k)}, N_{\langle \Gamma'_2; \sigma' \rangle}) & (1 \leq k \leq n'_1) \\ T'^{(n'_1-1+k)} &= @ (T_1, T_2'^{(k)}) & (1 \leq k \leq n'_2), \end{aligned}$$

and we have the following leftmost rewriting sequences:

$$\begin{aligned} N_{\langle \Gamma'_1; \sigma' \rightarrow \tau \rangle} &= T_1'^{(1)} \longrightarrow \dots \longrightarrow T_1'^{(n'_1)} = T_1 \\ N_{\langle \Gamma'_2; \sigma' \rangle} &= T_2'^{(1)} \longrightarrow \dots \longrightarrow T_2'^{(n'_2)} = T_2. \end{aligned}$$

Since $\text{Dom}(\Gamma_i) = \mathbf{FV}(e(T_i)) = \text{Dom}(\Gamma'_i)$ for each $i \in \{1, 2\}$ and $\Gamma_1 \cup \Gamma_2 = \Gamma'_1 \cup \Gamma'_2$, we have $\Gamma_i = \Gamma'_i$, which also implies $\sigma = \sigma'$. Hence by the induction hypothesis for $N_{\langle \Gamma_1; \sigma \rightarrow \tau \rangle} \longrightarrow T_1^{(2)}$ and for $N_{\langle \Gamma_2; \sigma \rangle} \longrightarrow T_2^{(2)}$, we have $n_i = n'_i$ and $T_i^{(k)} = T_i'^{(k)}$ for each $i \in \{1, 2\}$ and $2 \leq k < n_i$. Thus we also have $T^{(k)} = T'^{(k)}$ for $1 \leq k < n$. \square

4.2. Strong Connectivity and Aperiodicity. In this section, we restrict the grammar $\mathcal{G}(\delta, \iota, \xi)$ to $\mathcal{G}^\emptyset(\delta, \iota, \xi)$ by removing unnecessary nonterminals, and show the strong connectivity and aperiodicity of $\mathcal{G}^\emptyset(\delta, \iota, \xi)$ for $\delta, \iota, \xi \geq 2$ (Lemma 4.8 below). Recall that the strong connectivity and aperiodicity is required to apply Corollary 2.17 and Remark 2.18, respectively.

We define the restricted grammar $\mathcal{G}^\emptyset(\delta, \iota, \xi)$ by:

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

$$\mathcal{R}^\emptyset(\delta, \iota, \xi) \triangleq \{N \rightarrow T \in \mathcal{R}(\delta, \iota, \xi) \mid N \in \mathcal{N}^\emptyset(\delta, \iota, \xi)\}$$

$$\mathcal{G}^\emptyset(\delta, \iota, \xi) \triangleq (\Sigma(\delta, \iota, \xi), \mathcal{N}^\emptyset(\delta, \iota, \xi), \mathcal{R}^\emptyset(\delta, \iota, \xi)).$$

For $N \in \mathcal{N}^\emptyset(\delta, \iota, \xi)$, clearly $\mathcal{L}(\mathcal{G}^\emptyset(\delta, \iota, \xi), N) = \mathcal{L}(\mathcal{G}(\delta, \iota, \xi), N)$. 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 $\Lambda(\delta, \iota, \xi)$ whose type derivation contains a type judgment of the form $\Gamma \vdash t : \tau$.

The strong connectivity of $\mathcal{G}^\emptyset(\delta, \iota, \xi)$ follows from the following facts: (i) each $N_{\langle \Gamma; \tau \rangle} \in \mathcal{N}^\emptyset(\delta, \iota, \xi)$ is reachable from some $N_{\langle \emptyset; \tau' \rangle} \in \mathcal{N}^\emptyset(\delta, \iota, \xi)$ (by the definition of $\mathcal{G}^\emptyset(\delta, \iota, \xi)$ above), (ii) each $N_{\langle \emptyset; \tau \rangle} \in \mathcal{N}^\emptyset(\delta, \iota, \xi)$ is reachable from $N_{\langle \emptyset; \circ \rightarrow \circ \rangle}$ (Lemma 4.5 below), and (iii) $N_{\langle \emptyset; \circ \rightarrow \circ \rangle}$ is reachable from every $N_{\langle \Gamma; \tau \rangle} \in \mathcal{N}^\emptyset(\delta, \iota, \xi)$ (Lemma 4.6 below).

Lemma 4.5. *Let $\delta, \iota \geq 2$ and $\xi \geq 1$ be integers. Then for any nonterminal $N_{\langle \emptyset; \tau \rangle} \in \mathcal{N}^\emptyset(\delta, \iota, \xi)$, $N_{\langle \emptyset; \tau \rangle}$ is reachable from $N_{\langle \emptyset; \circ \rightarrow \circ \rangle}$.*

Proof. Let $\tau = \tau_1 \rightarrow \dots \rightarrow \tau_n \rightarrow \circ$ and $\tau_i = \tau_1^i \rightarrow \dots \rightarrow \tau_{\ell_i}^i \rightarrow \circ$ for $i = 1, \dots, n$. For $i = 1, \dots, n$, let $T_{\tau_i} \triangleq \lambda * \tau_1^i (\dots \lambda * \tau_{\ell_i}^i (x_1) \dots)$. Then, we have:

$$\begin{aligned} N_{\langle x_1 : \circ; \tau_i \rangle} &\longrightarrow \lambda * \tau_1^i (N_{\langle x_1 : \circ; \tau_2^i \rightarrow \dots \rightarrow \tau_{\ell_i}^i \rightarrow \circ \rangle}) \longrightarrow^* \lambda * \tau_1^i (\dots \lambda * \tau_{\ell_i}^i (N_{\langle x_1 : \circ; \circ \rangle}) \dots) \\ &\longrightarrow \lambda * \tau_1^i (\dots \lambda * \tau_{\ell_i}^i (x_1) \dots) = T_{\tau_i} \end{aligned}$$

and hence

$$\begin{aligned} N_{\langle \emptyset; \circ \rightarrow \circ \rangle} &\longrightarrow \lambda x_1^\circ (N_{\langle x_1 : \circ; \circ \rangle}) \\ &\longrightarrow \lambda x_1^\circ (@ (N_{\langle x_1 : \circ; \tau_n \rightarrow \circ \rangle}, N_{\langle x_1 : \circ; \tau_n \rangle})) \longrightarrow^* \lambda x_1^\circ (@ (N_{\langle x_1 : \circ; \tau_n \rightarrow \circ \rangle}, T_{\tau_n})) \\ &\longrightarrow^* \lambda x_1^\circ (@ (@ (\dots @ (N_{\langle x_1 : \circ; \tau_2 \rightarrow \dots \rightarrow \tau_n \rightarrow \circ \rangle}, T_{\tau_2}), \dots), T_{\tau_n})) \\ &\longrightarrow \lambda x_1^\circ (@ (@ (@ (\dots @ (@ (N_{\langle \emptyset; \tau_1 \rightarrow \tau_2 \rightarrow \dots \rightarrow \tau_n \rightarrow \circ \rangle}, N_{\langle x_1 : \circ; \tau_1 \rangle}), T_{\tau_2}), \dots), T_{\tau_n})) \\ &\longrightarrow^* \lambda x_1^\circ (@ (@ (@ (@ (N_{\langle \emptyset; \tau \rangle}, T_{\tau_1}), T_{\tau_2}), \dots), T_{\tau_n})). \end{aligned}$$

□

Lemma 4.6. *Let $\delta, \iota, \xi \geq 2$ be integers. Then for any nonterminal $N_{\langle \Gamma; \tau \rangle} \in \mathcal{N}^\emptyset(\delta, \iota, \xi)$, $N_{\langle \emptyset; \circ \rightarrow \circ \rangle}$ is reachable from $N_{\langle \Gamma; \tau \rangle}$.*

Proof. Suppose $N_{\langle \Gamma; \tau \rangle} \in \mathcal{N}^\emptyset(\delta, \iota, \xi)$. By the definition of $\mathcal{N}^\emptyset(\delta, \iota, \xi)$ (and $\mathcal{N}(\delta, \iota, \xi)$), there exists t such that $[t]_\alpha \in \Lambda(\langle \Gamma; \tau \rangle, \delta, \iota, \xi)$. Let $T \triangleq \pi^{-1}([t]_\alpha) \in \mathcal{L}(N_{\langle \Gamma; \tau \rangle})$. Now T contains at least one (possibly bound) variable, say x , and let σ be the type of x . Since $T \in \mathcal{L}(N_{\langle \Gamma; \tau \rangle})$, there exists a linear context C such that

$$N_{\langle \Gamma; \tau \rangle} \longrightarrow^* C[N_{\langle \{x : \sigma\}; \sigma \rangle}] \longrightarrow C[x] = T.$$

We show, by the induction on the structure of σ , that $N_{\langle\{x:\sigma\};\sigma\rangle} \longrightarrow^* C'[N_{\langle\{y:o\};y\rangle}]$ for some y and linear context C' . The case for $\sigma = o$ is obvious. If $\sigma = \sigma' \rightarrow \tau'$, then since $\xi \geq 2$, we obtain:

$$\begin{aligned} N_{\langle\{x:\sigma\};\sigma\rangle} &\longrightarrow \lambda x_i^{\sigma'} N_{\langle\{x:\sigma, x_i:\sigma'\};\tau'\rangle} \longrightarrow \lambda x_i^{\sigma'} (@ (N_{\langle\{x:\sigma\};\sigma\rangle}, N_{\langle\{x_i:\sigma'\};\sigma'\rangle})) \\ &\longrightarrow \lambda x_i^{\sigma'} (@ (x, N_{\langle\{x_i:\sigma'\};\sigma'\rangle})) \end{aligned}$$

by “ η -expansion”. By the induction hypothesis, we have $N_{\langle\{x_i:\sigma'\};\sigma'\rangle} \longrightarrow^* C''[N_{\langle\{y:o\};o\rangle}]$ for some y and 1-context C'' . Thus, the result holds for $C' = \lambda x_i^{\sigma'} (@ (x, C''))$.

By using the property above, we obtain

$$\begin{aligned} N_{\langle\Gamma;\tau\rangle} &\longrightarrow^* C[N_{\langle\{x:\sigma\};\sigma\rangle}] \longrightarrow^* C[C'[N_{\langle\{y:o\};o\rangle}]] \\ &\longrightarrow C[C'[@ (N_{\langle\emptyset;o \rightarrow o\rangle}, N_{\langle\{y:o\};o\rangle})]] \longrightarrow C[C'[@ (N_{\langle\emptyset;o \rightarrow o\rangle}, y)]]]. \end{aligned}$$

Thus, $N_{\langle\emptyset;o \rightarrow o\rangle}$ is reachable from $N_{\langle\Gamma;\tau\rangle}$. \square

Lemma 4.7. *Let $\delta, \iota \geq 2$ and $\xi \geq 1$ be integers. Then for any integer $n \geq 5$, the nonterminal $N_{\langle\emptyset;o \rightarrow o\rangle}$ of $\mathcal{G}^\emptyset(\delta, \iota, \xi)$ satisfies $\mathcal{L}_n(N_{\langle\emptyset;o \rightarrow o\rangle}) \neq \emptyset$.*

Proof. For simplicity of the presentation, here we identify trees with terms by the size-preserving bijection π . The proof proceeds by induction on n . For $n = 5, 6, 7$, the following terms

$$\begin{aligned} &\lambda x^\circ.(\lambda *^\circ.x)x \\ &\lambda x^\circ.(\lambda *^{\circ \rightarrow \circ}.x)(\lambda *^\circ.x) \\ &\lambda x^\circ.(\lambda *^{\circ \rightarrow \circ \rightarrow \circ}.x)\lambda *^\circ.\lambda *^\circ.x \end{aligned}$$

belong to $\mathcal{L}_n(N_{\langle\emptyset;o \rightarrow o\rangle})$, respectively. For $n \geq 8$, by the induction hypothesis, there exists $t \in \mathcal{L}_{n-3}(N_{\langle\emptyset;o \rightarrow o\rangle})$. Thus we have $\lambda x.tx \in \mathcal{L}_n(N_{\langle\emptyset;o \rightarrow o\rangle})$ as required. \square

The following is the main result of this subsection.

Lemma 4.8. *$\mathcal{G}^\emptyset(\delta, \iota, \xi)$ is strongly connected and aperiodic for any $\delta, \iota, \xi \geq 2$.*

Proof. Strong connectivity follows from Lemmas 4.5 and 4.6, and the definition of $\mathcal{G}^\emptyset(\delta, \iota, \xi)$ (as stated at the beginning of this subsection). The aperiodicity follows from Lemma 4.7 and strong connectivity. (Note that for every nonterminal N , there exists a linear context $U \in \mathcal{L}(\mathcal{G}, N_{\langle\emptyset;o \rightarrow o\rangle} \Rightarrow N)$. Thus, for any $n \geq |U| + 5$, $\mathcal{L}_n(\mathcal{G}, N) \supseteq \{U[T] \mid T \in \mathcal{L}_{n-|U|}(\mathcal{G}, N_{\langle\emptyset;o \rightarrow o\rangle})\} \neq \emptyset$.) \square

4.3. Explosive Terms. In this section, we define a family $(\text{Expl}_n^k)_n$ of λ -terms, which have long β -reduction sequences. They play the role of $(T_n)_n$ in Corollary 2.17.

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

$$\bar{2}_k \triangleq \lambda f^{\tau^{(k-1)}}. \lambda x^{\tau^{(k-2)}}. f(fx)$$

where $\tau^{(i)}$ is defined by $\tau^{(0)} \triangleq o$ and $\tau^{(i+1)} \triangleq \tau^{(i)} \rightarrow \tau^{(i)}$.

Definition 4.9 (explosive terms). Let $m \geq 1$ and $k \geq 2$ be integers. We define the *explosive term* Expl_m^k by:

$$\text{Expl}_m^k \triangleq \lambda x^\circ.((\bar{2}_k \uparrow^m \bar{2}_{k-1})\bar{2}_{k-2} \cdots \bar{2}_2 \text{Dup}(\text{Id } x)).$$

We state key properties of $Expl_m^k$ below.

- Lemma 4.10** (explosive). (1) $\emptyset \vdash Expl_m^k : \mathbf{o} \rightarrow \mathbf{o}$ is derivable.
(2) $\text{ord}(Expl_m^k) = k$, $\text{iar}(Expl_m^k) = k$ and $\#(\mathbf{V}(Expl_m^k)) = 2$.
(3) $|Expl_m^k| = 8m + 8k - 2$.
(4) $[Expl_m^k]_\alpha \in \Lambda(\langle \emptyset; \mathbf{o} \rightarrow \mathbf{o} \rangle, \delta, \iota, \xi)$ if $\delta, \iota \geq k$ and $\xi \geq 2$.
(5) If a term t satisfies $Expl_m^k \preceq t$, then $\beta(t) \geq \mathbf{exp}_k(m)$ holds.

Proof. First, observe that by the definition of $\bar{2}_k$, we have:

$$|\bar{2}_k| = 7, \quad \vdash \bar{2}_k : \tau^{(k)}, \quad \text{ord}(\bar{2}_k) = k, \quad \text{iar}(\bar{2}_k) = k.$$

(1) is obvious, and in the derivation of $\emptyset \vdash Expl_m^k : \mathbf{o} \rightarrow \mathbf{o}$, $\bar{2}_k$ is a subterm that has a type of the largest order and internal arity. Thus, (2) follows also immediately from $\text{ord}(\bar{2}_k) = \text{iar}(\bar{2}_k) = k$, and $\mathbf{V}(Expl_m^k) = \{f, x\}$.

(3) follows from the following calculation.

$$\begin{aligned} |Expl_m^k| &= 1 + |\bar{2}_k \uparrow^m \bar{2}_{k-1}| + \sum_{i=2}^{k-2} (1 + |\bar{2}_i|) + (1 + |Dup|) + (1 + |Id\ x|) \\ &\quad \text{(by } |t_0 t_1 \cdots t_n| = |t_0| + \sum_{i=1}^n (1 + |t_i|)) \\ &= 1 + ((|\bar{2}_k| + 1)m + |\bar{2}_{k-1}|) + \sum_{i=2}^{k-2} (1 + |\bar{2}_i|) + (1 + |Dup|) + (1 + |Id\ x|) \\ &= 1 + (8m + 7) + 8(k - 3) + 9 + 5 = 8m + 8k - 2. \end{aligned}$$

(4) follows from (1) and (2).

The proof for (5) is the same as the corresponding proof in [1]. \square

4.4. Proof of the Main Theorem. We are now ready to prove Theorem 2.5.

Proof of Theorem 2.5. We apply Corollary 2.17 with Remark 2.18 to the grammar $\mathcal{G}^\emptyset(\delta, \iota, \xi)$, which is unambiguous by Lemma 4.4, and aperiodic and strongly connected by Lemma 4.8. We define $T_n \triangleq \pi^{-1}([Expl_n^k]_\alpha)$; then by Lemma 4.10(3), we have $|T_n| = 8n + 8k - 2 = O(n)$. Thus, by Corollary 2.17 (with Remark 2.18, which allows us to replace $\lim_{n \rightarrow \infty}$ with $\lim_{n \rightarrow \infty}$), there exists $p > 0$ such that, for $\mathcal{I} \triangleq \{N_{\langle \emptyset; \tau \rangle} \mid \tau \in \text{Types}(\delta, \iota)\}$,

$$\lim_{n \rightarrow \infty} \frac{\#(\{(N, T) \in \coprod_{N \in \mathcal{I}} \mathcal{L}_n(\mathcal{G}^\emptyset(\delta, \iota, \xi), N) \mid T_{\lceil p \log n \rceil} \preceq T\})}{\#(\coprod_{N \in \mathcal{I}} \mathcal{L}_n(\mathcal{G}^\emptyset(\delta, \iota, \xi), N))} = 1. \quad (4.1)$$

For $n \in \mathbb{N}$, we have

$$\{[t]_\alpha \in \Lambda_n(\delta, \iota, \xi) \mid Expl_{\lceil p \log n \rceil}^k \preceq t\} \subseteq \{[t]_\alpha \in \Lambda_n(\delta, \iota, \xi) \mid \beta(t) \geq \mathbf{exp}_{k-1}(n^p)\} \quad (4.2)$$

by Lemma 4.10(5), and

$$\Lambda_n(\delta, \iota, \xi) = \bigsqcup_{\tau \in \text{Types}(\delta, \iota)} \Lambda_n(\langle \emptyset; \tau \rangle, \delta, \iota, \xi) \cong \coprod_{N \in \mathcal{I}} \mathcal{L}_n(\mathcal{G}^\emptyset(\delta, \iota, \xi), N) \quad (4.3)$$

by Lemma 4.3.

Therefore, we have:

$$\begin{aligned}
1 &\geq \frac{\#\left(\{[t]_\alpha \in \Lambda_n(\delta, \iota, \xi) \mid \beta(t) \geq \mathbf{exp}_{k-1}(n^p)\}\right)}{\#(\Lambda_n(\delta, \iota, \xi))} \\
&\geq \frac{\#\left(\{[t]_\alpha \in \Lambda_n(\delta, \iota, \xi) \mid \text{Expl}_{[p \log n]}^k \preceq t\}\right)}{\#(\Lambda_n(\delta, \iota, \xi))} && (\because \text{by (4.2)}) \\
&= \frac{\#\left(\{(N, T) \in \coprod_{N \in \mathcal{I}} \mathcal{L}_n(\mathcal{G}^\emptyset(\delta, \iota, \xi), N) \mid T_{[p \log n]} \preceq T\}\right)}{\#\left(\coprod_{N \in \mathcal{I}} \mathcal{L}_n(\mathcal{G}^\emptyset(\delta, \iota, \xi), N)\right)} && (\because \text{by (4.3)})
\end{aligned}$$

Since the right hand side converges to 1 by (4.1), we have

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

as required. \square

5. RELATED WORK

As mentioned in Section 1, there are several pieces of work on probabilistic properties of untyped λ -terms [2, 3, 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 (the latter result has later been generalized for arbitrary Turing complete combinator bases [24]). 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 Bruijn 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).

Choppy et al. [25] proposed a method to evaluate the average number of reduction steps for a restricted class of term rewriting systems called *regular* rewriting systems. In our context, the average number of reduction steps is not of much interest; note that, as the worst-case number of reduction steps is k -fold exponential for order- k terms, the average is also k -fold exponential, even if it were the case that the number of reduction steps is small for almost all terms.

In a larger context, our work may be viewed as an instance of the studies of average-case complexity [26, Chapter 10], which discusses “typical-case feasibility”. We are not aware of much work on the average-case complexity of problems with hyper-exponential complexity.

As a result related to our parameterized infinite monkey theorem for trees (Theorem 2.13), Steyaert and Flajolet [27] studied the probability that a given pattern (which is, in our terminology, a tree context of which every leaf is a hole) occurs at a randomly chosen node of a randomly chosen tree. Their result immediately yields a *non-parameterized* infinite monkey theorem for trees (which says that the probability that a given pattern occurs in a sufficiently large tree tends to 1), but their technique does not seem directly applicable to obtain our *parameterized* version.

6. CONCLUSION

We have shown that almost every simply-typed λ -term of order k has a β -reduction sequence as long as $(k - 1)$ -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. We obtained the result through the parameterized infinite monkey theorem for regular tree grammars, which may be of independent interest.

A lot of questions are left for future work, such as (i) whether our assumption (on the boundedness 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 [15] and/or the higher-order model checking problem [7]. To resolve the question (ii) above, it may be useful to conduct experiments to count the number of reduction steps for randomly generated terms.

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. *Journal of Symbolic Logic* **66**(3) (2001) 1277–1285
- [2] David, R., Grygiel, K., Kozik, J., Raffalli, C., Theyssier, G., Zaionc, M.: Asymptotically almost all λ -terms are strongly normalizing. *Logical Methods in Computer Science* **9**(1) (2013)
- [3] Grygiel, K., Lescanne, P.: Counting and generating lambda terms. *Journal of Functional Programming* **23**(05) (2013) 594–628
- [4] Bendkowski, M., Grygiel, K., Lescanne, P., Zaionc, M.: A Natural Counting of Lambda Terms. In: *SOFSEM 2016: Theory and Practice of Computer Science: 42nd International Conference on Current Trends in Theory and Practice of Computer Science, Harrachov, Czech Republic, January 23-28, 2016, Proceedings*. Springer Berlin Heidelberg, Berlin, Heidelberg (2016) 183–194
- [5] Flajolet, P., Sedgewick, R.: *Analytic Combinatorics*. 1 edn. Cambridge University Press, New York, NY, USA (2009)
- [6] Knapik, T., Niwinski, D., Urzyczyn, P.: Higher-order pushdown trees are easy. In: *Foundations of Software Science and Computation Structures, 5th International Conference, FOSSACS 2002. Volume 2303 of Lecture Notes in Computer Science.*, Springer (2002) 205–222
- [7] Ong, C.H.L.: On model-checking trees generated by higher-order recursion schemes. In: *21th IEEE Symposium on Logic in Computer Science (LICS 2006)*, IEEE Computer Society Press (2006) 81–90
- [8] Kobayashi, N.: Model checking higher-order programs. *Journal of the ACM* **60**(3) (2013)
- [9] Broadbent, C.H., Kobayashi, N.: Saturation-based model checking of higher-order recursion schemes. In: *Proceedings of CSL 2013. Volume 23 of LIPIcs.* (2013) 129–148
- [10] Ramsay, S., Neatherway, R., Ong, C.H.L.: An abstraction refinement approach to higher-order model checking. In: *Proceedings of ACM SIGPLAN/SIGACT Symposium on Principles of Programming Languages (POPL)*. (2014)

- [11] Kobayashi, N.: Types and higher-order recursion schemes for verification of higher-order programs. In: Proceedings of ACM SIGPLAN/SIGACT Symposium on Principles of Programming Languages (POPL), ACM Press (2009) 416–428
- [12] Kobayashi, N., Sato, R., Unno, H.: Predicate abstraction and CEGAR for higher-order model checking. In: Proceedings of ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI), ACM Press (2011) 222–233
- [13] Ong, C.H.L., Ramsay, S.: Verifying higher-order programs with pattern-matching algebraic data types. In: Proceedings of ACM SIGPLAN/SIGACT Symposium on Principles of Programming Languages (POPL), ACM Press (2011) 587–598
- [14] Sato, R., Unno, H., Kobayashi, N.: Towards a scalable software model checker for higher-order programs. In: Proceedings of the ACM SIGPLAN 2013 Workshop on Partial Evaluation and Program Manipulation, PEPM 2013, ACM Press (2013) 53–62
- [15] Terui, K.: Semantic evaluation, intersection types and complexity of simply typed lambda calculus. In: 23rd International Conference on Rewriting Techniques and Applications (RTA'12). Volume 15 of LIPIcs., Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik (2012) 323–338
- [16] Mairson, H.G.: Deciding ML typability is complete for deterministic exponential time. In: Proceedings of ACM SIGPLAN/SIGACT Symposium on Principles of Programming Languages (POPL), ACM Press (1990) 382–401
- [17] Kfoury, A.J., Tiuryn, J., Urzyczyn, P.: ML typability is DEXTIME-complete. In: Proceedings of CAAP '90. Volume 431 of Lecture Notes in Computer Science., Springer (1990) 206–220
- [18] Sin'ya, R., Asada, K., Kobayashi, N., Tsukada, T.: Almost every simply typed λ -term has a long β -reduction sequence. In: Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2017, Uppsala, Sweden, April 22–29, 2017, Proceedings. Volume 10203 of Lecture Notes in Computer Science. (2017) 53–68
- [19] Barendregt, H., Dekkers, W., Statman, R.: Lambda Calculus with Types. Cambridge University Press (2013)
- [20] Saoudi, A.: Rational and recognizable infinite tree sets. In Nivat, M., Podelski, A., eds.: Tree Automata and Languages. North-Holland, Amsterdam (1992) 225–234
- [21] Comon, H., Dauchet, M., Gilleron, R., Löding, C., Jacquemard, F., Lugiez, D., Tison, S., Tommasi, M.: Tree automata techniques and applications. Available on: <http://www.grappa.univ-lille3.fr/tata> (2007) release October, 12th 2007.
- [22] Heintze, N., McAllester, D.A.: Linear-time subtransitive control flow analysis. In: Proceedings of ACM SIGPLAN Conference on Programming Language Design and Implementation (PLDI). (1997) 261–272
- [23] Mureşan, M.: A concrete approach to classical analysis. CMS books in mathematics. Springer (2009)
- [24] Bendkowski, M., Grygiel, K., Zaionc, M.: On the likelihood of normalization in combinatory logic. J. Log. Comput. **27**(7) (2017) 2251–2269
- [25] Choppy, C., Kaplan, S., Soria, M.: Complexity analysis of term-rewriting systems. Theor. Comput. Sci. **67**(2&3) (1989) 261–282
- [26] Goldreich, O.: Computational Complexity: A Conceptual Perspective. Cambridge University Press (2008)
- [27] Steyaert, J.M., Flajolet, P.: Patterns and pattern-matching in trees: An analysis. Information and Control **58**(1) (1983) 19 – 58

APPENDIX A. A LIST OF NOTATIONS

notation	explanation	introduced at:
$\#(X)$	the cardinality of set X , or the length of sequence X	Section 2.1
$s.i$	the i -th element of sequence s	Section 2.1
$\text{Dom}(f)$	the domain of function f	Section 2.1
$\text{Im}(f)$	the image of function f	Section 2.1
τ	a type (of a simply-typed λ -term);	Section 2.1
$\text{ord}(\tau)$	the order of type τ	Section 2.1
$\text{iar}(\tau)$	the internal arity of type τ	Section 2.1
$\text{Types}(\delta, \iota)$	the set of types whose order and internal arity are bounded by δ and ι respectively	Section 2.1
t	a λ -term	Section 2.1
$\beta(t)$	the maximal length of β -reduction sequences of t	Section 2.1
δ	a bound on the order of λ -terms	Section 2.1
ι	a bound on the arity of λ -terms	Section 2.1
ξ	a bound on the number of variables used in λ -terms	Section 2.1
$\Lambda(\langle \Gamma; \tau \rangle, \delta, \iota, \xi)$	the set of α -equivalence classes of terms t such that $\Gamma \vdash t : \tau$, with the given bounds	Section 2.1
$\Lambda_n(\langle \Gamma; \tau \rangle, \delta, \iota, \xi)$	Like $\Lambda(\langle \Gamma; \tau \rangle, \delta, \iota, \xi)$, but the term size is restricted to n	Section 2.1
$\Lambda(\delta, \iota, \xi)$	the set of α -equivalence classes of closed well-typed terms, with the given bounds	Section 2.1
$\Lambda_n(\delta, \iota, \xi)$	Like $\Lambda(\delta, \iota, \xi)$, but the term size is restricted to n	Section 2.1
T	a tree	Section 2.2
$ T $	the size of tree T	Section 2.2
C	a tree context	Section 2.2
$ C $	the size of context C	Section 2.2
$C \preceq C'$	C is a subcontext of C'	Section 2.2
S	a linear context	Section 2.2
U	an affine context	Section 2.2
N	a nonterminal	Section 2.2
a	a terminal symbol	Section 2.2
\mathcal{G}	a grammar	Section 2.2
κ	a context type	Section 2.2
$\mathcal{L}(\mathcal{G}, N)$	the set of trees generated by N	Section 2.2
$\mathcal{L}_n(\mathcal{G}, N)$	Like $\mathcal{L}(\mathcal{G}, N)$, but the tree size is restricted to n	Section 2.2
$\mathcal{L}(\mathcal{G}, \kappa)$	the set of contexts C such that $N \rightarrow^* C[N_1, \dots, N_k]$ where $\kappa = N_1 \cdots N_k \Rightarrow N$	Section 2.2
$\mathcal{L}_n(\mathcal{G}, \kappa)$	Like $\mathcal{L}(\mathcal{G}, \kappa)$, but the context size is restricted to n	Section 2.2

notation	explanation	introduced at:
$\mathcal{L}(\mathcal{G})$	the set of trees generated from some nonterminals	Section 2.2
$\mathcal{L}_n(\mathcal{G})$	Like $\mathcal{L}(\mathcal{G})$, but the tree size is restricted to n	Section 2.2
$\mathcal{S}(\mathcal{G})$	the set of linear contexts generated from some nonterminals	Section 2.2
$\mathcal{S}_n(\mathcal{G})$	Like $\mathcal{S}(\mathcal{G})$, but the size is restricted to n	Section 2.2
$\mathcal{U}(\mathcal{G})$	the set of affine contexts generated from some nonterminals	Section 2.2
$\mathcal{U}_n(\mathcal{G})$	Like $\mathcal{U}(\mathcal{G})$, but the size is restricted to n	Section 2.2
E	a second-order context	Section 3.2
$\text{shn}(E)$	the number of second-order holes of E	Section 3.2
P	a sequences of affine contexts	Section 3.2
$\#(P)$	the length of the sequence P of affine contexts	Section 3.2
r	the largest arity of symbols in Σ	Section 3.2.2
Φ_m^\bullet	grammar-independent decomposition function	Section 3.2.2
Φ_m	an auxiliary decomposition function	Section 3.2.2
\mathcal{E}_n^m	the set of second-order contexts obtained by decomposing trees of size n	Section 3.2.4
$\mathcal{U}_{\llbracket \rrbracket_k^n}^m$	the set of affine contexts that match the hole $\llbracket \rrbracket_k^n$	Section 3.2.4
\mathcal{P}_E^m	the set of sequence P of affine contexts that match the second-order context E	Section 3.2.4
\tilde{E}	a typed second order context	Section 3.4
$\llbracket \rrbracket_\kappa^n$	a second-order hole	Section 3.4
$\text{shn}(\tilde{E})$	the number of second-order holes in \tilde{E}	Section 3.4
$\tilde{E}.i$	the i -th second-order hole of \tilde{E}	Section 3.4
$\Phi_m^{\bullet, \mathcal{G}}$	grammar-respecting decomposition function for trees	Section 3.4.2
$\tilde{\mathcal{E}}_n^m(\mathcal{G}, N)$	typed-version of \mathcal{E}_n^m	Section 3.4.3
$\tilde{\mathcal{U}}_{\llbracket \rrbracket_k^n}^m(\mathcal{G})$	typed-version of $\mathcal{U}_{\llbracket \rrbracket_k^n}^m$	Section 3.4.3
$\tilde{\mathcal{P}}_{\tilde{E}}^m(\mathcal{G}, N)$	typed-version of \mathcal{P}_E^m	Section 3.4.3
$\Sigma(\delta, \iota, \xi)$	the set of terminals of the grammar for λ -terms	Section 4.1
$\mathcal{N}(\delta, \iota, \xi)$	the set of nonterminals of the grammar for λ -terms	Section 4.1
$\mathcal{R}(\delta, \iota, \xi)$	the set of rules of the grammar for λ -terms	Section 4.1
Expl_m^k	an explosive term	Section 4.3

APPENDIX B. PROOF OF CANONICAL FORM LEMMA

This section proves Lemma 3.19 in Section 3.3. We describe the required transformation as the composite of two transformations. In the first step, we transform a grammar to *semi-canonical* one:

Definition B.1 (Semi-Canonical Grammar). A rewriting rule is in *semi-canonical form* if it is in canonical form or of the form

$$N \longrightarrow_{\mathcal{G}} N'.$$

A grammar $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ is *semi-canonical* if every rewriting rule is in semi-canonical form.

Lemma 3.19 will be shown by the next two lemmas.

Lemma B.2. *Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be a regular tree grammar that is unambiguous and essentially strongly-connected. Then one can (effectively) construct a grammar $\mathcal{G}' = (\Sigma, \mathcal{N}', \mathcal{R}')$ with $\mathcal{N} \subseteq \mathcal{N}'$ that satisfies the following conditions:*

- \mathcal{G}' is semi-canonical, unambiguous and essentially strongly-connected.
- $\mathcal{L}(\mathcal{G}, N) = \mathcal{L}(\mathcal{G}', N)$ for every $N \in \mathcal{N}$.
- $\mathcal{S}^{\text{inf}}(\mathcal{G}) \subseteq \mathcal{S}^{\text{inf}}(\mathcal{G}')$.

Proof. Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be a grammar that satisfies the assumption. We can assume without loss of generality that $\mathcal{L}(\mathcal{G}, N_0) \neq \emptyset$ for every $N_0 \in \mathcal{N}$.

Assume \mathcal{G} has a rule of the form

$$N' \longrightarrow a(T_1, \dots, T_n) \quad (T_1, \dots, T_n \text{ are } (\Sigma \cup \mathcal{N})\text{-trees})$$

such that $T_i \notin \mathcal{N}$ for some i . (If \mathcal{G} does not have a rule of this form, then \mathcal{G} is already semi-canonical.) Let N'' be a fresh nonterminal not in \mathcal{N} . Consider the grammar $\mathcal{G}'' = (\Sigma, \mathcal{N} \cup \{N''\}, \mathcal{R}'')$ where

$$\begin{aligned} \mathcal{R}'' = & (\mathcal{R} \setminus \{ N' \longrightarrow a(T_1, \dots, T_n) \}) \\ & \cup \{ N' \longrightarrow a(T_1, \dots, T_{i-1}, N'', T_{i+1}, \dots, T_n), \quad N'' \longrightarrow T_i \}. \end{aligned}$$

Then we can show the following claim by induction on the length of rewriting sequences.

Claim For every $N \in \mathcal{N}$ and $(\Sigma \cup \mathcal{N})$ -tree T ,

- $N \longrightarrow_{\mathcal{G}}^* T$ if and only if $N \longrightarrow_{\mathcal{G}''}^* T$, and
- $N \longrightarrow_{\mathcal{G}, \ell}^* T$ if and only if $N \longrightarrow_{\mathcal{G}'', \ell}^* T$.

This claim implies that \mathcal{G}'' is unambiguous and $\mathcal{L}(\mathcal{G}, N) = \mathcal{L}(\mathcal{G}'', N)$ for every $N \in \mathcal{N}$.

We show that \mathcal{G}'' is essentially strongly-connected. Let $N_1, N_2 \in \mathcal{N} \cup \{N''\}$ and assume $\#(\mathcal{L}(\mathcal{G}'', N_1)) = \#(\mathcal{L}(\mathcal{G}'', N_2)) = \infty$. We should show that N_2 is reachable from N_1 . We can assume without loss of generality that $N_1 \neq N''$ and $N_2 \neq N''$ by the following argument:

- If $N_1 = N''$, then T_i must contain a nonterminal $N'_1 \in \mathcal{N}$ with $\#(\mathcal{L}(\mathcal{G}'', N'_1)) = \infty$. Since $N'' \longrightarrow_{\mathcal{G}''} T_i$ is the unique rule for N'' and $\#(\mathcal{L}(\mathcal{G}'', N'')) = \infty$, we have $\mathcal{L}(\mathcal{G}, N_0) \neq \emptyset$ for each nonterminal N_0 appearing in T_i . Hence $N'' \longrightarrow_{\mathcal{G}''} T_i \longrightarrow_{\mathcal{G}''}^* S[N'_1]$ for some linear context S over Σ by rewriting each occurrence of a nonterminal N_0 in T_i (except for one occurrence of N'_1) with an element of $\mathcal{L}(\mathcal{G}'', N_0)$. So N'_1 is reachable from N'' in \mathcal{G}'' . Since the reachability relation is transitive, it suffices to show that N_2 is reachable from $N'_1 \in \mathcal{N}$, which satisfies $\#(\mathcal{L}(\mathcal{G}'', N'_1)) = \infty$.
- If $N_2 = N''$, then $\mathcal{L}(\mathcal{G}'', N')$ is infinite and N'' is reachable from N' . (Here we use the assumption that $\mathcal{L}(\mathcal{G}, N_0) \neq \emptyset$ for every $N_0 \in \mathcal{N}$.) Hence it suffices to show that N' is reachable from N_1 .

Since $\mathcal{L}(\mathcal{G}'', N_i) = \mathcal{L}(\mathcal{G}, N_i)$ ($i = 1, 2$), both $\mathcal{L}(\mathcal{G}, N_1)$ and $\mathcal{L}(\mathcal{G}, N_2)$ are infinite. Since \mathcal{G} is essentially strongly-connected, there exists a 1-context S such that $N_1 \longrightarrow_{\mathcal{G}}^* S[N_2]$. Since

$S[N_2]$ is a $(\Sigma \cup \mathcal{N})$ -tree, by the above claim, $N_1 \rightarrow_{\mathcal{G}''}^* S[N_2]$ and thus N_2 is reachable from N_1 in \mathcal{G}'' .

We show that $\mathcal{S}^{\text{inf}}(\mathcal{G}) \subseteq \mathcal{S}^{\text{inf}}(\mathcal{G}'')$. Suppose that $S \in \mathcal{L}(\mathcal{G}, N \Rightarrow N')$ with $\#(\mathcal{L}(\mathcal{G}, N)) = \#(\mathcal{L}(\mathcal{G}, N')) = \infty$. By definition, $N' \rightarrow_{\mathcal{G}}^* S[N]$. By the above claim, $N' \rightarrow_{\mathcal{G}''}^* S[N]$. Hence $S \in \mathcal{L}(\mathcal{G}'', N \Rightarrow N')$. Since $\mathcal{L}(\mathcal{G}, N) = \mathcal{L}(\mathcal{G}'', N)$ and $\mathcal{L}(\mathcal{G}, N') = \mathcal{L}(\mathcal{G}'', N')$, we have $\#(\mathcal{L}(\mathcal{G}'', N)) = \#(\mathcal{L}(\mathcal{G}'', N')) = \infty$.

By applying the above transformation as much as needed, we obtain a grammar that satisfies the requirements. What remains is to show the termination of the iterated application of the above transformation. A termination measure can be given as follows: the measure of a rule $N \rightarrow C[N_1, \dots, N_k]$ is defined as $\max(|C| - 1, 0)$ and the measure of a grammar is the sum of the measures of rules. The measure is by definition a non-negative integer and decreases by 1 by the above transformation. \square

Lemma B.3. *Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be a regular tree grammar that is semi-canonical, unambiguous and essentially strongly-connected. Then one can (effectively) construct a grammar $\mathcal{G}' = (\Sigma, \mathcal{N}', \mathcal{R}')$ and a family $(\mathcal{I}_N)_{N \in \mathcal{N}}$ of subsets $\mathcal{I}_N \subseteq \mathcal{N}'$ that satisfy the following conditions:*

- \mathcal{G}' is canonical, unambiguous and essentially strongly-connected.
- $\mathcal{L}(\mathcal{G}, N) = \biguplus_{N' \in \mathcal{I}_N} \mathcal{L}(\mathcal{G}', N')$ for every $N \in \mathcal{N}$.
- $\mathcal{S}^{\text{inf}}(\mathcal{G}) \subseteq \mathcal{S}^{\text{inf}}(\mathcal{G}')$.

Proof. Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be a grammar that satisfies the assumption. Let \mathcal{N}' be the subset of \mathcal{N} defined by

$$N_0 \in \mathcal{N}' \Leftrightarrow (N_0 \rightarrow a(N_1, \dots, N_{\Sigma(a)})) \in \mathcal{R} \text{ for some } a, N_1, \dots, N_{\Sigma(a)}.$$

The rewriting rule \mathcal{R}' is defined by

$$\mathcal{R}' := \{N_0 \rightarrow a(N'_1, \dots, N'_{\Sigma(a)}) \mid N_0, N'_1, \dots, N'_{\Sigma(a)} \in \mathcal{N}', \\ N_0 \rightarrow_{\mathcal{G}} a(N_1, \dots, N_{\Sigma(a)}) \rightarrow_{\mathcal{G}}^* a(N'_1, \dots, N'_{\Sigma(a)})\}.$$

In other words, $N_0 \rightarrow a(N'_1, \dots, N'_{\Sigma(a)})$ is in \mathcal{R}' if and only if there exists a rule $N_0 \rightarrow a(N_1, \dots, N_{\Sigma(a)})$ in \mathcal{R} with $N_i \rightarrow_{\mathcal{G}}^* N'_i$ for each $i = 1, \dots, \Sigma(a)$. Given a nonterminal $N \in \mathcal{N}$, let $\mathcal{I}_N \subseteq \mathcal{N}'$ be the subset given by

$$\mathcal{I}_N \triangleq \{N' \in \mathcal{N}' \mid N \rightarrow_{\mathcal{G}}^* N'\}.$$

We show that the grammar $\mathcal{G}' \triangleq (\Sigma, \mathcal{N}', \mathcal{R}')$ together with $(\mathcal{I}_N)_{N \in \mathcal{N}}$ satisfies the requirement.

We can prove the following claims by induction on the length of rewriting sequences:

- (1) For $N \in \mathcal{N}$ and $T \in \mathcal{T}(\Sigma \cup \mathcal{N}')$, if $N \rightarrow_{\mathcal{G}}^* T$, then $N' \rightarrow_{\mathcal{G}'}^* T$ for some $N' \in \mathcal{I}_N$.
- (2) For $N' \in \mathcal{N}'$ and $T \in \mathcal{T}(\Sigma \cup \mathcal{N}')$, if $N' \rightarrow_{\mathcal{G}'}^* T$, then $N' \rightarrow_{\mathcal{G}}^* T$.

We show that $\mathcal{L}(\mathcal{G}, N) = \bigcup_{N' \in \mathcal{I}_N} \mathcal{L}(\mathcal{G}', N')$. The above (1) shows that $\mathcal{L}(\mathcal{G}, N) \subseteq \bigcup_{N' \in \mathcal{I}_N} \mathcal{L}(\mathcal{G}', N')$. The above (2) shows that $\mathcal{L}(\mathcal{G}', N') \subseteq \mathcal{L}(\mathcal{G}, N)$ for every $N' \in \mathcal{I}_N$. By definition of \mathcal{I}_N , we have $N \rightarrow_{\mathcal{G}}^* N'$ and thus $\mathcal{L}(\mathcal{G}, N') \subseteq \mathcal{L}(\mathcal{G}, N)$ for every $N' \in \mathcal{I}_N$. So $\bigcup_{N' \in \mathcal{I}_N} \mathcal{L}(\mathcal{G}', N') \subseteq \mathcal{L}(\mathcal{G}, N)$.

We prove by contraposition that, for $N_1, N_2 \in \mathcal{I}_N$, if $N_1 \neq N_2$, then $\mathcal{L}(\mathcal{G}', N_1) \cap \mathcal{L}(\mathcal{G}', N_2) = \emptyset$. Suppose that $T \in \mathcal{L}(\mathcal{G}', N_1) \cap \mathcal{L}(\mathcal{G}', N_2)$. Then we have

$$N_i \rightarrow_{\mathcal{G}'} a(N'_{i,1}, \dots, N'_{i,\Sigma(a)}) \rightarrow_{\mathcal{G}'}^* T \quad (i = 1, 2)$$

for some $N'_{i,j} \in \mathcal{N}'$. By the definition of \mathcal{R}' , we have

$$N_i \rightarrow_{\mathcal{G}} a(N_{i,1}, \dots, N_{i,\Sigma(a)}) \rightarrow_{\mathcal{G}}^* a(N'_{i,1}, \dots, N'_{i,\Sigma(a)}) \rightarrow_{\mathcal{G}'}^* T \quad (i = 1, 2).$$

Because $N_i \in \mathcal{I}_N$, we have $N \rightarrow_{\mathcal{G}}^* N_i$, for $i = 1, 2$. By the claim (2), we have

$$N \rightarrow_{\mathcal{G}}^* N_i \rightarrow_{\mathcal{G}} a(N_{i,1}, \dots, N_{i,\Sigma(a)}) \rightarrow_{\mathcal{G}}^* a(N'_{i,1}, \dots, N'_{i,\Sigma(a)}) \rightarrow_{\mathcal{G}}^* T \quad (i = 1, 2).$$

The above two rewriting sequences of \mathcal{G} induce the following two leftmost rewriting sequences:

$$N \rightarrow_{\mathcal{G},\ell}^* N_i \rightarrow_{\mathcal{G},\ell} a(N_{i,1}, \dots, N_{i,\Sigma(a)}) \rightarrow_{\mathcal{G},\ell}^* T \quad (i = 1, 2),$$

which are the same by the unambiguity of \mathcal{G} . Then the length of the sequence $N \rightarrow_{\mathcal{G},\ell}^* N_1$ and that of $N \rightarrow_{\mathcal{G},\ell}^* N_2$ are the same since just after them the root terminal a occurs, and hence $N_1 = N_2$, as required. Next we prove the unambiguity of \mathcal{G}' , by contradiction: we assume that \mathcal{G}' is not unambiguous, and show that \mathcal{G} is not unambiguous, which is a contradiction. Let $N'_0 \in \mathcal{N}'$ and $T \in \mathcal{T}(\Sigma)$ be a pair of a nonterminal and a tree such that there exist two leftmost rewriting sequences from N'_0 to T in \mathcal{G}' . We can assume without loss of generality that the leftmost rewriting sequences differ on the first step. Assume that the two rewriting sequences are

$$N'_0 \rightarrow_{\mathcal{G}',\ell} a(N'_{1,1}, \dots, N'_{1,\Sigma(a)}) \rightarrow_{\mathcal{G}',\ell}^* a(T_1, \dots, T_{\Sigma(a)}) = T$$

and

$$N'_0 \rightarrow_{\mathcal{G}',\ell} a(N'_{2,1}, \dots, N'_{2,\Sigma(a)}) \rightarrow_{\mathcal{G}',\ell}^* a(T_1, \dots, T_{\Sigma(a)}) = T.$$

We have $N'_{1,j} \neq N'_{2,j}$ for some j . By the definition of \mathcal{R}' , there are rules

$$N'_0 \rightarrow_{\mathcal{G}} a(N_{1,1}, \dots, N_{1,\Sigma(a)}) \quad \text{and} \quad N'_0 \rightarrow_{\mathcal{G}} a(N_{2,1}, \dots, N_{2,\Sigma(a)})$$

with $N_{i,j} \rightarrow_{\mathcal{G}}^* N'_{i,j}$ for every $(i, j) \in \{1, 2\} \times \{1, \dots, \Sigma(a)\}$. There are two cases:

- Case $N_{1,j} \neq N_{2,j}$ for some $j \in \{1, \dots, \Sigma(a)\}$: By the rewriting sequences above and the claim (2), we have two sequences

$$N'_0 \rightarrow_{\mathcal{G}} a(N_{i,1}, \dots, N_{i,\Sigma(a)}) \rightarrow_{\mathcal{G}}^* a(N'_{i,1}, \dots, N'_{i,\Sigma(a)}) \rightarrow_{\mathcal{G}}^* a(T_1, \dots, T_{\Sigma(a)})$$

for $i = 1, 2$. The corresponding leftmost rewriting sequences differ at the first step. So \mathcal{G} is unambiguous.

- Case $N_{1,j} = N_{2,j}$ for every $j \in \{1, \dots, \Sigma(a)\}$: Let j be an index such that $N'_{1,j} \neq N'_{2,j}$ and $N_j \triangleq N_{1,j} = N_{2,j}$. Then $N_j \rightarrow_{\mathcal{G}}^* N'_{1,j} \rightarrow_{\mathcal{G}'}^* T_j$ and $N_j \rightarrow_{\mathcal{G}}^* N'_{2,j} \rightarrow_{\mathcal{G}'}^* T_j$. By the definition of \mathcal{R}' and the claim (2), we have

$$N_j \rightarrow_{\mathcal{G}}^* N'_{i,j} \rightarrow_{\mathcal{G}} b(N''_{i,1}, \dots, N''_{i,\Sigma(b)}) \rightarrow_{\mathcal{G}}^* T_j \quad (i = 1, 2)$$

with $N'_{1,j} \neq N'_{2,j}$. Hence \mathcal{G} is not unambiguous.

We show that \mathcal{G}' is essentially strongly-connected. Let $N_1, N_2 \in \mathcal{N}'$ such that $\#(\mathcal{L}(\mathcal{G}', N_1)) = \#(\mathcal{L}(\mathcal{G}', N_2)) = \infty$. We show that N_2 is reachable from N_1 in \mathcal{G}' . By the claim (2), $\mathcal{L}(\mathcal{G}, N_2) \supseteq \mathcal{L}(\mathcal{G}', N_2)$ is infinite. Since $\mathcal{L}(\mathcal{G}', N_1)$ is infinite, we have a rule

$$N_1 \rightarrow_{\mathcal{G}'} a(N_{1,1}, \dots, N_{1,\Sigma(a)})$$

in \mathcal{R}' such that $\mathcal{L}(\mathcal{G}', N_{1,i})$ is infinite for some i and $\mathcal{L}(\mathcal{G}', N_{1,j}) \neq \emptyset$ for every $j = 1, \dots, \Sigma(a)$. Again, by the claim (2), $\mathcal{L}(\mathcal{G}, N_{1,i}) \supseteq \mathcal{L}(\mathcal{G}', N_{1,i})$ is infinite. Since \mathcal{G} is essentially strongly-connected, there exists a linear context S such that $N_{1,i} \rightarrow_{\mathcal{G}}^* S[N_2]$. By the claim (1), there

exists $N'_{1,i} \in \mathcal{I}_{N_{1,i}}$ such that $N'_{1,i} \rightarrow_{\mathcal{G}'}^* S[N_2]$. Since $N'_{1,i} \in \mathcal{I}_{N_{1,i}}$, we have $N_{1,i} \rightarrow_{\mathcal{G}}^* N'_{1,i}$, and hence by the definition of \mathcal{R}' , we find that

$$N_1 \rightarrow_{\mathcal{G}'} a(N_{1,1}, \dots, N_{1,i-1}, N'_{1,i}, N_{1,i+1}, \dots, N_{1,\Sigma(a)})$$

is in \mathcal{R}' . Let us pick $T_{1,j} \in \mathcal{L}(\mathcal{G}', N_{1,j})$ for each $j \neq i$; then we have:

$$\begin{aligned} N_1 &\rightarrow_{\mathcal{G}'} a(N_{1,1}, \dots, N_{1,i-1}, N'_{1,i}, N_{1,i+1}, \dots, N_{1,\Sigma(a)}) \\ &\rightarrow_{\mathcal{G}'}^* a(N_{1,1}, \dots, N_{1,i-1}, S[N_2], N_{1,i+1}, \dots, N_{1,\Sigma(a)}) \\ &\rightarrow_{\mathcal{G}'}^* a(T_{1,1}, \dots, T_{1,i-1}, S[N_2], T_{1,i+1}, \dots, T_{1,\Sigma(a)}). \end{aligned}$$

Thus N_2 is reachable from N_1 in \mathcal{G}' .

Lastly, we show that $\mathcal{S}^{\text{inf}}(\mathcal{G}) \subseteq \mathcal{S}^{\text{inf}}(\mathcal{G}')$. Assume that $S \in \mathcal{L}(\mathcal{G}, N_1 \Rightarrow N_2)$ and $\#(\mathcal{L}(\mathcal{G}, N_1)) = \#(\mathcal{L}(\mathcal{G}, N_2)) = \infty$. Since $\mathcal{L}(\mathcal{G}, N_1) = \bigcup_{N'_1 \in \mathcal{I}_{N_1}} \mathcal{L}(\mathcal{G}', N'_1)$, there exists $N'_1 \in \mathcal{I}_{N_1}$ such that $\#(\mathcal{L}(\mathcal{G}', N'_1)) = \infty$. Since $N_2 \rightarrow_{\mathcal{G}}^* S[N_1]$ and $N_1 \rightarrow_{\mathcal{G}}^* N'_1$, we have $N_2 \rightarrow_{\mathcal{G}}^* S[N'_1]$. By the claim (1), there exists $N'_2 \in \mathcal{I}_{N_2}$ such that $N'_2 \rightarrow_{\mathcal{G}'}^* S[N'_1]$. Hence $S \in \mathcal{L}(\mathcal{G}', N'_1 \Rightarrow N'_2)$. Since $\#(\mathcal{L}(\mathcal{G}', N'_1)) = \infty$, we have $\#(\mathcal{L}(\mathcal{G}', N'_2)) = \infty$ as well. \square

Proof of Lemma 3.19. Let \mathcal{G} be a regular tree grammar that is unambiguous and strongly connected. Then \mathcal{G} is essentially strongly-connected. So by applying Lemmas B.2 and B.3, we obtain a grammar \mathcal{G}' and a family $(\mathcal{I}_N)_{N \in \mathcal{N}}$ that satisfy the following conditions:

- \mathcal{G}' is canonical, unambiguous and essentially strongly-connected.
- $\mathcal{L}(\mathcal{G}, N) = \biguplus_{N' \in \mathcal{I}_N} \mathcal{L}(\mathcal{G}', N')$ for every $N \in \mathcal{N}$.
- $\mathcal{S}^{\text{inf}}(\mathcal{G}) \subseteq \mathcal{S}^{\text{inf}}(\mathcal{G}')$.

The grammar \mathcal{G}' and a family $(\mathcal{I}_N)_{N \in \mathcal{N}}$ satisfy the first and second requirements of Lemma 3.19. We show that $\#(\mathcal{L}(\mathcal{G})) = \infty$ implies $\mathcal{S}(\mathcal{G}) \subseteq \mathcal{S}^{\text{inf}}(\mathcal{G}')$.

Suppose that $\#(\mathcal{L}(\mathcal{G})) = \infty$. Since $\mathcal{L}(\mathcal{G}) = \bigcup_{N \in \mathcal{N}} \mathcal{L}(\mathcal{G}, N)$ and \mathcal{N} is finite, there exists $N_0 \in \mathcal{N}$ such that $\#(\mathcal{L}(\mathcal{G}, N_0)) = \infty$. By strong connectivity of \mathcal{G} , we have $\#(\mathcal{L}(\mathcal{G}, N)) = \infty$ for every $N \in \mathcal{N}$. Hence $\mathcal{N}^{\text{inf}} = \mathcal{N}$, and thus $\mathcal{S}(\mathcal{G}) = \mathcal{S}^{\text{inf}}(\mathcal{G})$. We have $\mathcal{S}(\mathcal{G}) = \mathcal{S}^{\text{inf}}(\mathcal{G}) \subseteq \mathcal{S}^{\text{inf}}(\mathcal{G}')$ as required. \square

APPENDIX C. PROOF OF LEMMA 3.32

We start from the formal definition of periodicity and its basic properties.

Definition C.1 (period, basic period). Let $X \subseteq \mathbb{N}$ be a subset of natural numbers. We say X is *periodic* with *period* c ($c \in \mathbb{N} \setminus \{0\}$) if

$$\exists n_0 \in \mathbb{N}. \forall n \geq n_0. \forall k \in \mathbb{N}. \quad n \in X \Leftrightarrow (n + ck) \in X.$$

If X is periodic, we call the minimum period the *basic period*.

Let \mathcal{G} be a regular tree grammar and $\kappa = (N_1 \dots N_k \Rightarrow N)$ be a context type of \mathcal{G} . We say κ is *periodic with period* c if so is the subset of natural numbers

$$\{n \in \mathbb{N} \mid \mathcal{L}_n(\mathcal{G}, \kappa) \neq \emptyset\},$$

and call the minimum period the *basic period*.

Lemma C.2. Let $X \subseteq \mathbb{N}$ be a subset of natural numbers and $c \in \mathbb{N} \setminus \{0\}$.

(1) Suppose that

$$\exists n_0 \in \mathbb{N}. \forall n \geq n_0. \quad n \in X \Rightarrow n + c \in X.$$

Then X is periodic with period c .

(2) Suppose that

$$\exists n_0 \in \mathbb{N}. \exists k_0 \in \mathbb{N}. \forall n \geq n_0. \forall k \geq k_0. \quad n \in X \Rightarrow n + ck \in X.$$

Then X is periodic with period c .

(3) Suppose that c and c' are periods of X . Then $\gcd(c, c')$ is also a period of X , where $\gcd(c, c')$ is the greatest common divisor.

Proof. (1) Let $n_0 \in \mathbb{N}$ be a witness of the assumption. Then for every $n \geq n_0$ and $k \geq 0$, we have $n \in X \Rightarrow (n + ck) \in X$, by induction on k . Let $Y \subseteq \mathbb{N}$ be the set of counterexamples of $n + c \in X \Rightarrow n \in X$, namely,

$$Y \triangleq \{n \in \mathbb{N} \mid n + c \in X \wedge n \notin X\}.$$

Then Y has at most $n_0 + c$ elements because, for every $n \geq n_0$ and $k \geq 1$,

$$n \in Y \Rightarrow n + c \in X \Rightarrow n + ck \in X \Rightarrow n + ck \notin Y.$$

Let $n'_0 \geq n_0$ be a natural number that is (strictly) greater than all elements of Y . Then for every $n \geq n'_0$,

$$n \in X \Leftrightarrow n + c \in X.$$

By induction on $k \geq 0$, we conclude that $n \in X \Leftrightarrow n + kd \in X$ for every $n \geq n'_0$.

(2) Let n_0 and k_0 be witnesses of the assumption and $Y \subseteq \mathbb{N}$ be the set of counterexamples of $n \in X \Rightarrow n + c \in X$, i.e.,

$$Y \triangleq \{n \in \mathbb{N} \mid n \in X \wedge n + c \notin X\}.$$

Then Y has at most $n_0 + ck_0$ elements because, for every $n \geq n_0$ and $k \geq k_0$,

$$n \in Y \Rightarrow n \in X \Rightarrow n + c(k+1) \in X \Rightarrow n + ck \notin Y.$$

Let n'_0 be a natural number that is (strictly) greater than all elements of Y . Then for every $n \geq n'_0$,

$$n \in X \Rightarrow n + c \in X.$$

Hence by item (1), X is periodic with period c .

(3) The claim trivially holds if $\gcd(c, c') = c$ or c' . We assume that $\gcd(c, c') < c, c'$. Let $d = \gcd(c, c')$. There exist integers a, b such that $ac + bc' = d$ (by Bézout's lemma). Since $0 < d < c, c'$, exactly one of a and b is negative. Suppose that $a > 0$ and $b < 0$.

Let n_0 be a constant such that

$$\forall n \geq n_0. \forall k \in \mathbb{N}. \quad n \in X \Leftrightarrow (n + ck) \in X$$

and n'_0 be a constant for a similar condition for c' . Let $n''_0 = \max(n_0, n'_0) - bc'$. Then, for every $n \geq n''_0$,

$$\begin{aligned} n \in X &\Leftrightarrow n + bc' \in X && (\text{since } n + bc' \geq n'_0 \text{ and } b < 0) \\ &\Leftrightarrow n + bc' + ac \in X && (\text{since } n + bc' \geq n_0 \text{ and } a > 0) \\ &\Leftrightarrow n + d \in X. \end{aligned}$$

By induction on $k \geq 0$, we conclude that $n \in X \Leftrightarrow n + kd \in X$ for every $n \geq n''_0$. \square

We show that $N \Rightarrow N'$ is periodic for every $N, N' \in \mathcal{N}^{\text{inf}}$ and its basic period is independent of the choice of N and N' . We call the basic period of $N \Rightarrow N'$, which is independent of N and N' , the *basic period of the grammar \mathcal{G}* . Further we show that $U, U' \in \mathcal{L}(\mathcal{G}, N \Rightarrow N')$ implies $|U| \equiv |U'| \pmod{c}$, where c is the basic period of the grammar \mathcal{G} .

Lemma C.3. *Let $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R})$ be a regular tree grammar. Assume that \mathcal{G} is essentially strongly-connected and $\#(\mathcal{L}(\mathcal{G})) = \infty$.*

- (1) $\mathcal{L}(\mathcal{G}, N \Rightarrow N')$ is infinite for every $N, N' \in \mathcal{N}^{\text{inf}}$.
- (2) $N \Rightarrow N$ is periodic for every $N \in \mathcal{N}^{\text{inf}}$.
- (3) Let $N \in \mathcal{N}^{\text{inf}}$ and c_N be the basic period of $N \Rightarrow N$. Then $\mathcal{L}_n(\mathcal{G}, N \Rightarrow N) \neq \emptyset$ implies $n \equiv 0 \pmod{c_N}$.
- (4) The basic period c_N of $N \Rightarrow N$ is independent of $N \in \mathcal{N}^{\text{inf}}$. Let $c = c_N$.
- (5) For every $N, N' \in \mathcal{N}^{\text{inf}}$, if $U_1, U_2 \in \mathcal{L}(\mathcal{G}, N \Rightarrow N')$, then $|U_1| \equiv |U_2| \pmod{c}$.
- (6) For every $N, N' \in \mathcal{N}^{\text{inf}}$, $N \Rightarrow N'$ has the basic period c .

Proof. (1) There exist $N_1, N_2 \in \mathcal{N}^{\text{inf}}$ such that $\mathcal{L}(\mathcal{G}, N_1 \Rightarrow N_2)$ contains a context of size greater than 0, because there exists $N_2 \in \mathcal{N}$ with $\mathcal{L}(\mathcal{G}, N_2) = \infty$ due to $\#(\mathcal{L}(\mathcal{G})) = \infty$. Let $U \in \mathcal{L}(\mathcal{G}, N_1 \Rightarrow N_2)$ be a context with $|U| > 0$. Let $N \in \mathcal{N}^{\text{inf}}$. By essential strong-connectivity there exist $S_N \in \mathcal{L}(\mathcal{G}, N \Rightarrow N_1)$ and $S'_N \in \mathcal{L}(\mathcal{G}, N_2 \Rightarrow N)$, and then $U_N \triangleq S'_N[U[S_N]] \in \mathcal{L}(\mathcal{G}, N \Rightarrow N)$ with $|U_N| > 0$. Let U_N^n be the context defined by $U_N^0 = []$ and $U_N^{n+1} = U_N[U_N^n]$. Then $U_N^n \in \mathcal{L}(\mathcal{G}, N \Rightarrow N)$ for every n and $|U_N^n| \geq n$; hence $\mathcal{L}(\mathcal{G}, N \Rightarrow N)$ is an infinite set. Let $N' \in \mathcal{N}^{\text{inf}}$. By essential strong-connectivity there exists $S_{N,N'} \in \mathcal{L}(\mathcal{G}, N \Rightarrow N')$, which implies $\mathcal{L}(\mathcal{G}, N \Rightarrow N')$ is also infinite.

(2) Let $N \in \mathcal{N}^{\text{inf}}$. By (1), $\mathcal{L}(\mathcal{G}, N \Rightarrow N)$ is an infinite set. In particular, there exists $U_N \in \mathcal{L}(\mathcal{G}, N \Rightarrow N)$ with $|U_N| > 0$. Let $n \in \mathbb{N}$ and assume that $\mathcal{L}_n(\mathcal{G}, N \Rightarrow N) \neq \emptyset$. Then $U \in \mathcal{L}_n(\mathcal{G}, N \Rightarrow N)$ for some U and thus $U[U_N] \in \mathcal{L}_{n+|U_N|}(\mathcal{G}, N \Rightarrow N)$. Therefore $\mathcal{L}_n(\mathcal{G}, N \Rightarrow N) \neq \emptyset$ implies $\mathcal{L}_{n+|U_N|}(\mathcal{G}, N \Rightarrow N) \neq \emptyset$. By applying Lemma C.2(1), $N \Rightarrow N$ is periodic with period $|U_N|$.

(3) Let $N \in \mathcal{N}^{\text{inf}}$ and c_N be the basic period of $N \Rightarrow N$, and suppose that $\mathcal{L}_n(\mathcal{G}, N \Rightarrow N) \neq \emptyset$. If $n = 0$, then the claim trivially holds. Suppose that $n > 0$ and let $U \in \mathcal{L}_n(\mathcal{G}, N \Rightarrow N)$. Then by the argument of the proof of (1), $N \Rightarrow N$ has period $|U| = n$. By Lemma C.2(3), $N \Rightarrow N$ has period $\gcd(c_N, n)$, which is smaller than or equal to c_N . Since c_N is basic, this implies $c_N = \gcd(c_N, n)$, i.e., $n \equiv 0 \pmod{c_N}$.

(4) Let $N_1, N_2 \in \mathcal{N}^{\text{inf}}$ and c_1 and c_2 be the basic periods of $N_1 \Rightarrow N_1$ and of $N_2 \Rightarrow N_2$, respectively. We show that $c_1 \equiv 0 \pmod{c_2}$. By essential strong-connectivity, there exist $S_{1,2} \in \mathcal{L}(\mathcal{G}, N_1 \Rightarrow N_2)$ and $S_{2,1} \in \mathcal{L}(\mathcal{G}, N_2 \Rightarrow N_1)$. Since $\mathcal{L}(\mathcal{G}, N_1 \Rightarrow N_1)$ is infinite (by (1)) and $N_1 \Rightarrow N_1$ has period c_1 , there exist contexts $U, U' \in \mathcal{L}(\mathcal{G}, N_1 \Rightarrow N_1)$ such that $|U'| = |U| + c_1$. Let $n = |S_{1,2}[U[S_{2,1}]]|$ and $n' = |S_{1,2}[U'[S_{2,1}]]|$; then $n' = n + c_1$. Then both $\mathcal{L}_n(\mathcal{G}, N_2 \Rightarrow N_2)$ and $\mathcal{L}_{n'}(\mathcal{G}, N_2 \Rightarrow N_2)$ are non-empty. Hence $n \equiv 0 \pmod{c_2}$ and $n' \equiv 0 \pmod{c_2}$ by (3). Therefore $c_1 \equiv n' - n \equiv 0 \pmod{c_2}$.

(5) Let $N, N' \in \mathcal{N}^{\text{inf}}$ and assume that $U_1, U_2 \in \mathcal{L}(\mathcal{G}, N \Rightarrow N')$. By essential strong-connectivity, there exists $S \in \mathcal{L}(\mathcal{G}, N' \Rightarrow N)$. Then $S[U_1], S[U_2] \in \mathcal{L}(\mathcal{G}, N \Rightarrow N)$. By (3),

$$|S| + |U_1| \equiv |S[U_1]| \equiv |S[U_2]| \equiv |S| + |U_2| \pmod{c}.$$

Hence $|U_1| \equiv |U_2| \pmod{c}$.

(6) Let $N, N' \in \mathcal{N}^{\text{inf}}$. Recall that c is the basic period of $N \Rightarrow N$. Since c is a period of $N \Rightarrow N$ and by (3), there exists $k_0 \geq 0$ such that, for every $k \geq k_0$, we have $\mathcal{L}_{ck}(\mathcal{G}, N \Rightarrow N) \neq \emptyset$. Let $(U_k)_{k \geq k_0}$ be a family such that $U_k \in \mathcal{L}_{ck}(\mathcal{G}, N \Rightarrow N)$. Now $\mathcal{L}_n(\mathcal{G}, N \Rightarrow N') \neq \emptyset$

implies $\mathcal{L}_{n+ck}(\mathcal{G}, N \Rightarrow N') \neq \emptyset$ for every $k \geq k_0$, since $U' \in \mathcal{L}_n(\mathcal{G}, N \Rightarrow N')$ implies $U'[U_k] \in \mathcal{L}_{n+ck}(\mathcal{G}, N \Rightarrow N')$. By Lemma C.2(2), $N \Rightarrow N'$ has period c .

Conversely assume that $N \Rightarrow N'$ has period c' . Since $\mathcal{L}(\mathcal{G}, N \Rightarrow N')$ is an infinite set by (1), we have $U_1, U_2 \in \mathcal{L}(\mathcal{G}, N \Rightarrow N')$ such that $|U_2| = |U_1| + c'$. By (5), we have $|U_1| \equiv |U_2| \pmod{c}$, which implies $c' \equiv 0 \pmod{c}$. \square

We are now ready to prove Lemma 3.32.

Proof of Lemma 3.32. Let c be the basic period of \mathcal{G} . We first define $d_{N,N'}$ for each $N, N' \in \mathcal{N}^{\text{inf}}$. Since $\mathcal{L}(\mathcal{G}, N \Rightarrow N')$ is infinite (Lemma C.3(1)), there exists $U_{N,N'} \in \mathcal{L}(\mathcal{G}, N \Rightarrow N')$. Let $d_{N,N'}$ be the unique natural number $0 \leq d_{N,N'} < c$ such that $|U_{N,N'}| \equiv d_{N,N'} \pmod{c}$. This definition of $d_{N,N'}$ does not depend on the choice of $U_{N,N'}$, by Lemma C.3(5).

We define n_0 . For every $N, N' \in \mathcal{N}^{\text{inf}}$, since c is a period of $\mathcal{L}(\mathcal{G}, N \Rightarrow N')$, there exists $n_0^{N,N'}$ such that:

$$\forall n \geq n_0^{N,N'}. \forall k \in \mathbb{N}. \quad \mathcal{L}_n(\mathcal{G}, N \Rightarrow N') \neq \emptyset \Leftrightarrow \mathcal{L}_{n+ck}(\mathcal{G}, N \Rightarrow N') \neq \emptyset.$$

Let $n_0 \triangleq \max\{n_0^{N,N'} \mid N, N' \in \mathcal{N}^{\text{inf}}\}$.

Then:

- (1) follows from Lemma C.3(5).
- (2) follows from the definition of n_0 , infinity of $\mathcal{L}(\mathcal{G}, N \Rightarrow N')$ (Lemma C.3(1)), and Lemma C.3(5) (given $n \geq n_0$, pick $U \in \mathcal{L}(\mathcal{G}, N \Rightarrow N')$ such that $|U| \geq n$; then $|U| = n + ck$ for some $k \geq 0$).
- (3) follows from Lemma C.3(3).
- (4) follows from $U_{N',N''}[U_{N,N'}] \in \mathcal{L}(\mathcal{G}, N \Rightarrow N'')$ and Lemma C.3(5). \square