Lambda-Definable Order-3 Tree Functions are Well-Quasi-Ordered

Kazuyuki Asada
Tohoku University, Sendai, Japan
asada@riec.tohoku.ac.jp
https://orcid.org/0000-0001-8782-2119

Naoki Kobayashi
The University of Tokyo, Tokyo, Japan
koba@is.s.u-tokyo.ac.jp

Abstract

Asada and Kobayashi [ICALP 2017] conjectured a higher-order version of Kruskal's tree theorem, and proved a pumping lemma for higher-order languages modulo the conjecture. The conjecture has been proved up to order-2, which implies that Asada and Kobayashi's pumping lemma holds for order-2 tree languages, but remains open for order-3 or higher. In this paper, we prove a variation of the conjecture for order-3. This is sufficient for proving that a variation of the pumping lemma holds for order-3 tree languages (equivalently, for order-4 word languages).

2012 ACM Subject Classification Theory of computation → Lambda calculus

Keywords and phrases higher-order grammar, pumping lemma, Kruskal's tree theorem, well-quasi-ordering, simply-typed lambda calculus

1 Introduction

Kruskal's tree theorem [8] says that the homeomorphic embedding relation $\preceq^{\mathrm{he}}$ on finite trees is a well-quasi-ordering, i.e., for every infinite sequence of trees $\pi_0, \pi_1, \pi_2, \ldots$, there exist $i < j$ such that $\pi_i \preceq^{\mathrm{he}} \pi_j$. Here, $\pi \preceq^{\mathrm{he}} \pi'$ means that there exists an embedding of the nodes of π to those of π', preserving the labels and the ancestor/descendant relation. Asada and Kobayashi [2] considered a higher-order version $\preceq^{\mathrm{he}}_{\kappa}$ of $\preceq^{\mathrm{he}}$ on simply-typed λ-terms of type κ, and conjectured that $\preceq^{\mathrm{he}}_{\kappa}$ is also a well-quasi-ordering, for every simple type κ. Under the assumption that the conjecture (which we call AK-conjecture) is true, they proved a pumping lemma for higher-order languages (a la higher-order languages in Damm's IO hierarchy [4]), which says that for any order-k tree grammar that generates an infinite language L, there exists a strictly increasing infinite sequence $\pi_0 \prec^{\mathrm{he}} \pi_1 \prec^{\mathrm{he}} \pi_2 \prec^{\mathrm{he}} \cdots$ such that $\pi_i \in L$ and $|\pi_i| \leq \mathbf{exp}_k(ci + d)$, where $\prec^{\mathrm{he}}$ is the strict version of the homeomorphic embedding, c and d are constants that depend on the grammar, and $\mathbf{exp}_k(x)$ is defined by $\mathbf{exp}_0(x) = x$ and $\mathbf{exp}_{k+1}(x) = 2^{\mathbf{exp}_k(x)}$. The pumping lemma can be used to prove that a certain language does not belong to the class of order-k languages. They also proved that the conjecture is true up to order-2 types, and hence also the pumping lemma for order-2 tree languages and (by the correspondence between tree/word languages [1, 4]) order-3 word languages. The AK-conjecture is still open for order-3 or higher.

In the present paper, we consider a variation of the AK-conjecture (which we call nAK-conjecture), where the homeomorphic embedding relation is replaced by $\preceq^{\#}$, defined by $\pi_1 \preceq^{\#} \pi_2$ if and only if, for every tree constructor a, $\#_a(\pi_1) \leq \#_a(\pi_2)$; here $\#_a(\pi)$ denotes the number of occurrences of a in π. The correctness of the nAK-conjecture would imply the following variation of the pumping lemma: for any order-k tree grammar that generates an infinite language L, there exists a strictly increasing infinite sequence $\pi_0 \prec^{\#} \pi_1 \prec^{\#} \pi_2 \prec^{\#} \cdots$

such that $\pi_i \in L$ and $|\pi_i| \leq \mathbf{exp}_k(ci+d)$. We prove that the nAK-conjecture is true for the order-3 case, i.e., that $\preceq^{\#}_{\kappa}$ (the logical relation on simply-typed λ-terms of type κ, obtained from $\preceq^{\#}$) is a well-quasi-ordering for any type κ of order up to 3. The variation of the pumping lemma above is thus obtained for order-3 tree languages and order-4 word languages. To our knowledge, pumping lemmas were known only for tree (word, resp.) languages of order up to 2 (3, resp.) [2].

To prove the order-3 nAK-conjecture, we define a transformation $(\cdot)^{\natural}$ from order-3 λ-terms to order-2 numeric functions (that are also represented by λ-terms), and prove (i) the transformation reflects the quasi-orderings, i.e., $t_1 \preceq^{\#}_{\kappa} t_2$ if ${t_1}^{\natural} \preceq^{\mathbb{N}} {t_2}^{\natural}$ for a certain quasi-ordering $\preceq^{\mathbb{N}}$ on numeric functions, and (ii) $\preceq^{\mathbb{N}}$ is a well-quasi-ordering.

Related work. We are not aware of directly related work, besides our own previous work [2]. Our reduction from the well-quasi-orderedness of order-3 λ-terms to that of order-2 numeric functions relies on the inexpressiveness of simply-typed λ-terms as (higher-order) tree functions. Zaionc [13, 14, 15] studied the expressive power of simply-typed λ-terms. Pumping lemmas for higher-order languages have been known to be difficult. After Hayashi [6] proved a pumping lemma for indexed languages (i.e. order-2 word languages), it was only in 2017 that a pumping lemma for order-3 word languages was proved [2]. We have further improved the result to obtain a pumping lemma for order-4 word (or, order-3 tree) languages.

The rest of the paper is structured as follows. Section 2 introduces basic definitions. Section 3 explains the nAK-conjecture and the pumping lemma. Section 4 proves the nAK-conjecture up to order-3. Section 5 concludes the paper.

2 Preliminaries

We give basic definitions on λ-terms and quasi-orderings.

2.1 λ-terms and higher-order languages

▶ **Definition 1** (types and terms). The set of *simple types*, ranged over by κ, is given by: $\kappa ::= \mathtt{o} \mid \kappa_1 \to \kappa_2$. The order[1] of a simple type κ, written $\mathtt{order}(\kappa)$ is defined by $\mathtt{order}(\mathtt{o}) = 0$ and $\mathtt{order}(\kappa_1 \to \kappa_2) = \max(\mathtt{order}(\kappa_1)+1, \mathtt{order}(\kappa_2))$. The type $\mathtt{o}$ describes trees, and $\kappa_1 \to \kappa_2$ describes functions from κ_1 to κ_2. A *(ranked) alphabet* Σ is a map from a finite set of constants (that represent tree constructors) to the set of natural numbers called *arities*. The set of λY^{nd}*-terms*, ranged over by s, t, u, v, is defined by:

$$t ::= x \mid a\, t_1 \cdots t_k \mid t_1\, t_2 \mid \lambda x:\kappa.t \mid Y_{\kappa}\, t \mid t_1 \oplus t_2$$

Here, $x, y, \ldots$ ranges over variables, and a over $dom(\Sigma)$. The term $a\, t_1 \cdots t_k$ (where we require $\Sigma(a) = k$) constructs a tree that has a as the root and (the values of) $t_1, \ldots, t_k$ as children. Y_{κ} and $\oplus$ represent a fixed-point combinator and a non-deterministic choice, respectively. We often omit the type annotation and just write $\lambda x.t$ and $Y\, t$ for $\lambda x:\kappa.t$ and $Y_{\kappa}\, t$. A λY^{nd}-term is called: (i) a $\lambda^{\to,\mathrm{nd}}$*-term* if it does not contain Y; (ii) a $\lambda^{\to}$*-term* if it contains neither Y nor $\oplus$; and (iii) an *applicative term* if it contains none of λ-abstractions, Y, and $\oplus$. We often call a $\lambda^{\to}$-term just a *term*. As usual, we identify λY^{nd}-terms up to the α-equivalence, and implicitly apply α-conversions.

A *type environment* Γ is a sequence of type bindings of the form $x:\kappa$ such that Γ contains at most one binding for each variable x. A λY^{nd}-term t has type κ under Γ if $\Gamma \vdash_{\mathtt{ST}} t : \kappa$ is

[1] For clarity, we use the word *order* for this notion, and *ordering* for relations such as $\leq$, $\preceq^{\mathrm{he}}$, etc.

derivable from the following typing rules.

$$\frac{}{\Gamma,\, x:\kappa,\, \Gamma' \vdash_{\mathtt{ST}} x:\kappa} \qquad \frac{\Sigma(a)=k \qquad \Gamma \vdash_{\mathtt{ST}} t_i : \mathtt{o}\ \ \text{(for each } i\in\{1,\dots,k\})}{\Gamma \vdash_{\mathtt{ST}} a\, t_1 \cdots t_k : \mathtt{o}} \qquad \frac{\Gamma \vdash_{\mathtt{ST}} t:\kappa\to\kappa}{\Gamma \vdash_{\mathtt{ST}} Y_\kappa\, t:\kappa}$$

$$\frac{\Gamma \vdash_{\mathtt{ST}} t_1:\kappa_2\to\kappa \qquad \Gamma \vdash_{\mathtt{ST}} t_2:\kappa_2}{\Gamma \vdash_{\mathtt{ST}} t_1\, t_2:\kappa} \qquad \frac{\Gamma,\, x:\kappa_1 \vdash_{\mathtt{ST}} t:\kappa_2}{\Gamma \vdash_{\mathtt{ST}} \lambda x:\kappa_1.t:\kappa_1\to\kappa_2} \qquad \frac{\Gamma \vdash_{\mathtt{ST}} t_1:\mathtt{o} \qquad \Gamma \vdash_{\mathtt{ST}} t_2:\mathtt{o}}{\Gamma \vdash_{\mathtt{ST}} t_1\oplus t_2:\mathtt{o}}$$

We consider below only well-typed λY^{nd}-terms. Note that given Γ and t, there exists at most one type κ such that $\Gamma \vdash_{\mathtt{ST}} t:\kappa$. We call κ the type of t (with respect to Γ). We often omit "with respect to Γ" if Γ is clear from context. Given a judgment $\Gamma \vdash t:\kappa$, we define $\lambda\Gamma.t$ by: $\lambda\emptyset.t := t$ and $\lambda(\Gamma, x:\kappa').t := \lambda\Gamma.\lambda x.t$. Also we define $\Gamma\to\kappa$ by: $\emptyset\to\kappa := \kappa$ and $(\Gamma, x:\kappa')\to\kappa := \Gamma\to(\kappa'\to\kappa)$; thus we have $\vdash \lambda\Gamma.t:\Gamma\to\kappa$ if $\Gamma\vdash t:\kappa$. Given an alphabet Σ, we write Λ^Σ for the set of $\lambda^\to$-terms whose constants are taken from Σ. Also we define $\Lambda^\Sigma_{\Gamma,\kappa} := \{t\in\Lambda^\Sigma \mid \Gamma\vdash t:\kappa\}$ and $\Lambda^\Sigma_\kappa := \Lambda^\Sigma_{\emptyset,\kappa}$.

For a λY^{nd}-term t with a type environment Γ, the *(internal) order* of t (with respect to Γ), written $\mathtt{order}_\Gamma(t)$, is the largest order of the types of subterms of $\lambda\Gamma.t$, and the *external order* of t (with respect to Γ), written $\mathtt{eorder}_\Gamma(t)$, is the order of the type of t with respect to Γ. We often omit Γ when it is clear from context. For example, for $t = (\lambda x:\mathtt{o}.x)\mathtt{e}$, $\mathtt{order}_\emptyset(t) = 1$ and $\mathtt{eorder}_\emptyset(t) = 0$. We define the *size* $|t|$ of a λY^{nd}-term t by: $|x| := 1$, $|a\, t_1 \cdots, t_k| := 1+|t_1|+\cdots+|t_k|$, $|s\, t| := |s|+|t|+1$, $|\lambda x.t| := |t|+1$, $|Y_\kappa\, t| := |t|+1$ and $|s\oplus t| := |s|+|t|+1$. We call a λY^{nd}-term t *ground* (with respect to Γ) if $\Gamma\vdash_{\mathtt{ST}} t:\mathtt{o}$. We call t a (finite, Σ-ranked) *tree* if t is a ground closed applicative term (consisting of only constants). We write $\mathbf{Tree}_\Sigma$ for the set of Σ-ranked trees, and use the meta-variable π for a tree. We often write $\overrightarrow{\cdot}$ to denote a sequence (possibly with a condition on the range of the sequence in the superscript). For example, $\overrightarrow{t_i}^{i\le m}$ denotes the sequence $t_1,\dots,t_m$ of terms, and $[\overrightarrow{t_i/x_i}^{i\le m}]$ denotes the substitution $[t_1/x_1,\dots,t_m/x_m]$.

We sometimes identify a ranked alphabet $\Sigma = \{a_1\mapsto r_1,\dots,a_k\mapsto r_k\}$ with the first-order environment $\Sigma = \{a_1:\mathtt{o}^{r_1}\to\mathtt{o},\dots,a_k:\mathtt{o}^{r_k}\to\mathtt{o}\}$ (assuming an arbitrary fixed linear ordering on Σ).

▶ **Definition 2** (reduction and language)**.** The set of *(call-by-name) evaluation contexts* is defined by:

$$E ::= [\,]\, t_1\cdots t_k \mid a\,\pi_1\cdots\pi_i\, E\, t_1\cdots t_k$$

and the *call-by-name reduction* for (possibly open) ground λY^{nd}-terms is defined by:

$$E[(\lambda x.t)t'] \longrightarrow E[t[t'/x]] \qquad E[Y\,t] \longrightarrow E[t\,(Y\,t)] \qquad E[t_1\oplus t_2] \longrightarrow E[t_i] \quad (i=1,2)$$

where $t[t'/x]$ is the usual capture-avoiding substitution. We write $\longrightarrow^*$ for the reflexive transitive closure of $\longrightarrow$. A *call-by-name normal form* is a ground λY^{nd}-term t such that $t \not\longrightarrow t'$ for any t'. For a ground closed λY^{nd}-term t, we define the *tree language* $\mathcal{L}(t)$ *generated by* t by $\mathcal{L}(t) := \{\pi \mid t\longrightarrow^* \pi\}$. For a ground closed $\lambda^\to$-term t, $\mathcal{L}(t)$ is a singleton set $\{\pi\}$; we write $\mathcal{T}(t)$ for such π and call it *the tree of* t.

In the previous paper [2] we stated the pumping lemma for the notion of a *higher-order grammar*; in this paper, following [9, 10], we use only the formalism by λY^{nd}-terms for simplicity. Since there exist well-known order-preserving and language-preserving transformations between higher-order grammars and ground closed λY^{nd}-terms, we obtain corresponding results on higher-order grammars immediately.

The notion of a word can be seen as a special case of that of a tree:

▶ **Definition 3** (word alphabet)**.** We call a ranked alphabet Σ a *word alphabet* if it has a special nullary constant $\mathtt{e}$ and all the other constants have arity 1. For a tree $\pi = a_1(\cdots(a_n\,\mathtt{e})\cdots)$ of a word alphabet, we define $\mathbf{word}(\pi) := a_1 \cdots a_n$, and we define $\mathbf{utree}$ as the inverse function of $\mathbf{word}$, i.e., $\mathbf{utree}(a_1 \cdots a_n) := a_1(\cdots(a_n \mathtt{e}))$. The *word language* generated by a ground closed λY^{nd}-term t over a word alphabet, written $\mathcal{L}_{\mathtt{w}}(t)$, is defined as $\{\mathbf{word}(\pi) \mid \pi \in \mathcal{L}(t)\}$.

A tree language (word language, resp.) over an alphabet (word alphabet, resp.) Σ is called *order-n* if it is generated by some order-n ground closed λY^{nd}-term of Σ; we note that the classes of order-0, order-1, and order-2 word languages coincide with those of regular, context-free, and indexed languages, respectively [12].

2.2 Some quasi-orderings and their logical relation extension

▶ **Definition 4** ((well-)quasi-ordering)**.** A *quasi-ordering* (a.k.a. preorder) on a set A is a binary relation on A that is reflexive and transitive. A *well-quasi-ordering* (*wqo* for short) on a set S is a quasi-ordering $\leq$ on S such that for any infinite sequence $(s_i)_i$ of elements in S there exist j and k such that $j < k$ and $s_j \leq s_k$.

As a general notation, for a quasi-ordering denoted by $\preceq$, we write $\approx$ for the induced equivalence relation (i.e., $x \approx y$ if $x \preceq y$ and $y \preceq x$), and write $\prec$ for the strict version (i.e., $x \prec y$ if $x \preceq y$ and $y \not\preceq x$). Also, for a quasi-ordering denoted by $\leq$, we write $\sim$ for the induced equivalence relation and $<$ for the strict version. We apply these conventions also to notations with superscript/subscript such as $\preceq^a$, $\preceq_b$, $\preceq^a_b$, $\leq^a$, $\leq_b$, and $\leq^a_b$. Further, for any quasi-ordering on the set of trees of a word alphabet, we use the same notation also for the quasi-ordering on the set of words induced through $\mathbf{utree}$.

▶ **Definition 5** (logical relation extension)**.** Let Σ be a ranked alphabet. We call $\leq$ a *base quasi-ordering* (with respect to Σ) if $\leq$ is a quasi-ordering on the set $\Lambda^{\Sigma}_{\mathtt{o}}$ modulo $\beta\eta$-equivalence and every constant in Σ is monotonic on $\leq$. We define the *logical relation extension of* $\leq$ as the family $(\leq_\kappa)_\kappa$ of relations $\leq_\kappa$ on the set $\Lambda^{\Sigma}_{\kappa}$ modulo $\beta\eta$-equivalence indexed by simple types κ where $\leq_\kappa$'s are defined by induction on κ as follows:

$$\begin{aligned} t_1 \leq_{\mathtt{o}} t_2 \quad &\text{if} \quad t_1 \leq t_2 \\ t_1 \leq_{\kappa \to \kappa'} t_2 \quad &\text{if} \quad \text{for any } t'_1, t'_2, \quad t'_1 \leq_\kappa t'_2 \implies t_1\, t'_1 \leq_{\kappa'} t_2\, t'_2. \end{aligned}$$

Furthermore we extend the relation to open terms: for $t_1, t_2 \in \Lambda^{\Sigma}_{\Gamma,\kappa}$, we define $t_1 \leq_{\Gamma,\kappa} t_2$ if $\lambda\Gamma.t_1 \leq_{\Gamma\to\kappa} \lambda\Gamma.t_2$. We omit the subscripts of $\leq_\kappa$ and $\leq_{\Gamma,\kappa}$ if there is no confusion.

The next lemma follows immediately from the basic lemma (a.k.a. the abstraction theorem) of logical relations (see Appendix A for details).

▶ **Lemma 6.** *Let $\leq$ be a base quasi-ordering. Each component $\leq_\kappa$ of the logical relation extension of $\leq$ is a quasi-ordering. Further, $\leq_\kappa$ is the point-wise quasi-ordering:*

$$t_1 \leq_{\kappa\to\kappa'} t_2 \quad \textit{if and only if} \quad \textit{for any } t' \in \Lambda^{\Sigma}_{\kappa}, \quad t_1\, t' \leq_{\kappa'} t_2\, t'.$$

Every quasi-ordering for higher-order terms used in this paper is a logical relation extension (of some base quasi-ordering). The next ordering is used in the previous paper [2].

▶ **Definition 7** (homeomorphic embedding)**.** Let Σ be a ranked alphabet. The *homeomorphic embedding* ordering $\preceq^{\mathrm{he},\Sigma}$ between Σ-ranked trees[2] is inductively defined by the following rules:

[2] In the usual definition, a quasi-ordering on labels (tree constructors) is assumed. Here we fix the quasi-order on labels to the identity relation.

$$\frac{\pi_i \preceq^{\mathrm{he},\Sigma} \pi'_i \quad (\text{for all } i \leq k) \qquad k = \Sigma(a)}{a\,\pi_1 \cdots \pi_k \preceq^{\mathrm{he},\Sigma} a\,\pi'_1 \cdots \pi'_k} \qquad \frac{\pi \preceq^{\mathrm{he},\Sigma} \pi_i \qquad k = \Sigma(a) > 0 \qquad 1 \leq i \leq k}{\pi \preceq^{\mathrm{he},\Sigma} a\,\pi_1 \cdots \pi_k}$$

We extend the above ordering to a base ordering by: $t_1 \preceq^{\mathrm{he},\Sigma} t_2$ if $\mathcal{T}(t_1) \preceq^{\mathrm{he},\Sigma} \mathcal{T}(t_2)$.

For example, $\mathtt{br}\,\mathtt{a}\,\mathtt{b} \preceq^{\mathrm{he}} \mathtt{br}\,(\mathtt{br}\,\mathtt{a}\,\mathtt{c})\,\mathtt{b}$. The homeomorphic embedding on words is nothing but the (scattered) subsequence ordering. The following is a fundamental result on the homeomorphic embedding:

▶ **Proposition 8** (Kruskal's tree theorem [8])**.** *For any (finite) ranked alphabet Σ, the homeomorphic embedding $\preceq^{\mathrm{he}}$ on Σ-ranked trees is a well-quasi-ordering.*

Also, we often use the Dickson's theorem [7] which says that the product quasi-ordering (component-wise quasi-ordering) of a finite number of wqo's is a wqo.

The next is the quasi-ordering that is used in the theorems in this paper.

▶ **Definition 9** (occurrence-number quasi-ordering)**.** Let Σ be a ranked alphabet. For $a \in \Sigma$ and a Σ-tree π, we define $\#_a(\pi)$ as the number of occurrences of a in π, and extend this to a ground closed $\lambda^{\to}$-term t by $\#_a(t) := \#_a(\mathcal{T}(t))$. Then we define a base quasi-ordering $\preceq^{\#,\Sigma,a}$ by:

$$t_1 \preceq^{\#,\Sigma,a} t_2 \qquad \text{if} \qquad \#_a(t_1) \leq \#_a(t_2).$$

Also we define a base quasi-ordering $\preceq^{\#,\Sigma}$ by:

$$t_1 \preceq^{\#,\Sigma} t_2 \qquad \text{if} \qquad \text{for every } a \in \Sigma, \quad t_1 \preceq^{\#,\Sigma,a} t_2.$$

Note that $\pi \preceq^{\mathrm{he}} \pi'$ implies $\pi \preceq^{\#,\Sigma} \pi'$, shown by induction on the rule of $\preceq^{\mathrm{he}}$; and further $\pi \preceq^{\mathrm{he}}_{\kappa} \pi'$ implies $\pi \preceq^{\#,\Sigma}_{\kappa} \pi'$ for any κ since $\preceq^{\mathrm{he}}_{\kappa}$ and $\preceq^{\#,\Sigma}_{\kappa}$ are point-wise quasi-ordering. Also note that $\preceq^{\#,\Sigma}_{\kappa} = \bigcap_{a\in\Sigma}(\preceq^{\#,\Sigma,a}_{\kappa})$ for any κ.

The next quasi-ordering is used just in proofs. We write $\Sigma_{\mathbb{N}}$ for the ranked alphabet $\{0 \mapsto 0, 1 \mapsto 0, + \mapsto 2, \times \mapsto 2\}$; we write $+\,t\,t'$ as $t + t'$ and $\times\,t\,t'$ as $t \times t'$. We define a set-theoretical denotational interpretation $[\![-]\!]$ of $\Lambda^{\Sigma_{\mathbb{N}}}$ by: $[\![\mathsf{o}]\!] := \mathbb{N}$, $[\![\kappa \to \kappa']\!]$ is the set of functions from $[\![\kappa]\!]$ to $[\![\kappa']\!]$, $[\![0]\!] := 0$, $[\![1]\!] := 1$, $[\![+]\!](n)(m) := n + m$, and $[\![\times]\!](n)(m) := n \times m$. For $t_1, t_2 \in \Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma,\kappa}$, we write $t_1 =^{[\![]\!]}_{\Gamma,\kappa} t_2$ (or $t_1 =^{[\![]\!]} t_2$) if $[\![t_1]\!] = [\![t_2]\!]$.

▶ **Definition 10** (natural number quasi-ordering)**.** We define a base quasi-ordering $\preceq^{\mathbb{N}}$ on the set $\Lambda^{\Sigma_{\mathbb{N}}}_{\mathsf{o}}$ by:

$$t_1 \preceq^{\mathbb{N}} t_2 \qquad \text{if} \qquad [\![t_1]\!] \leq [\![t_2]\!].$$

3 Numeric Pumping Lemma for Higher-order Tree Languages

Here we explain the nAK-conjecture and the pumping lemma for higher-order tree languages with respect to $\preceq^{\#,\Sigma}$.

▶ **Conjecture 11** (nAK-conjecture)**.** *For any Σ and κ, $\preceq^{\#,\Sigma}_{\kappa}$ is a well quasi-ordering.*

Our main theorem (Theorem 14) is to show the above conjecture for κ of order up to 3. The above conjecture (and Theorem 14) can be used for the following pumping lemma:

▶ **Theorem 12** (pumping lemma)**.** *Assume that Conjecture 11 holds. Then, for any order-n ground closed λY^{nd}-term t of a ranked alphabet Σ such that $\mathcal{L}(t)$ is infinite, there exist an infinite sequence of trees $\pi_0, \pi_1, \pi_2, \ldots \in \mathcal{L}(t)$, and constants c, d such that: (i) $\pi_0 \prec^{\#,\Sigma} \pi_1 \prec^{\#,\Sigma} \pi_2 \prec^{\#,\Sigma} \cdots$, and (ii) $|\pi_i| \leq \mathbf{exp}_n(ci + d)$ for each $i \geq 0$. Furthermore, we can drop the assumption on Conjecture 11 when $n \leq 3$.*

The proof of the above theorem is obtained as a simple modification of the proof of the pumping lemma in [2]: see Appendix B.

▶ Remark. The theorem we prove in Appendix B is actually slightly stronger than Theorem 12 above, in the following three points (see Theorem 39 in Appendix B.2 for details): (i) As in [2], we relax the assumption of nAK conjecture, so that $\preceq_{\kappa}^{\#,\Sigma}$ need not be the logical relation; any higher-order extension of the base quasi-ordering that is closed under application suffices. (ii) As in [2], we use actually a weaker conjecture, called the *periodicity*, which requires that, for any $\vdash_{\mathtt{ST}} t : \kappa \to \kappa$ and $\vdash_{\mathtt{ST}} s : \kappa$, there exist $i, j > 0$ such that $t^i\, s \preceq_{\kappa}^{\#,\Sigma} t^{i+j}\, s \preceq_{\kappa}^{\#,\Sigma} t^{i+2j}\, s \preceq_{\kappa}^{\#,\Sigma} \cdots$. (iii) Whilst Theorem 12 states a pumping lemma on $\preceq^{\#,\Sigma}$, the generalized theorem states a pumping lemma on arbitrary base quasi-ordering with certain conditions, which includes $\preceq^{\#,\Sigma}$ and $\preceq^{\mathrm{he}}$ as instances.

By the correspondence between order-n tree grammars and order-$(n+1)$ word grammars [4, 1], we also have:

▶ **Corollary 13** (pumping lemma for word languages)**.** *Assume that Conjecture 11 holds. Then, for any order-n ground closed λY^{nd}-term t of a word alphabet Σ (where $n \geq 1$) such that $\mathcal{L}_{\mathtt{w}}(t)$ is infinite, there exist an infinite sequence of words $w_0, w_1, w_2, \ldots \in \mathcal{L}_{\mathtt{w}}(t)$, and constants c, d such that: (i) $w_0 \prec^{\#,\Sigma} w_1 \prec^{\#,\Sigma} w_2 \prec^{\#,\Sigma} \cdots$, and (ii) $|w_i| \leq \mathbf{exp}_{n-1}(ci + d)$ for each $i \geq 0$. Furthermore, we can drop the assumption on Conjecture 11 when $n \leq 4$.*

4 Numeric Version of Order-3 Kruskal's Tree Theorem

Here we prove the main theorem (Theorem 14 below), which states that the nAK-conjecture (Conjecture 11) holds for order-3 types. In this whole section, by a *term*, we mean a $\lambda^{\to}$-term, and we never consider a fixed-point combinator nor non-determinism.

4.1 Main theorem

▶ **Theorem 14.** *For any alphabet Σ and any type κ of order up to 3, $\preceq_{\kappa}^{\#,\Sigma}$ on $\Lambda_{\kappa}^{\Sigma}$ is a wqo.*

The theorem above is obtained as a corollary of the following lemma.

▶ **Lemma 15.** *For any alphabet Σ, any $a \in \Sigma$, and any order-2 type environment Γ (i.e., a type environment whose codomain consists of types of order up to 2), the quasi-ordering $\preceq_{\Gamma,\mathtt{o}}^{\#,\Sigma,a}$ on $\Lambda_{\Gamma,\mathtt{o}}^{\emptyset}$ is a wqo.*

Proof sketch of Theorem 14

- For Theorem 14, it is sufficient that $\preceq_{\kappa}^{\#,\Sigma,a}$ on $\Lambda_{\kappa}^{\Sigma}$ is a wqo for every $a \in \Sigma$ and κ with $\mathtt{order}(\kappa) \leq 3$, because $\preceq_{\kappa}^{\#,\Sigma} = \cap_{a\in\Sigma}(\preceq_{\kappa}^{\#,\Sigma,a})$ and well-quasi-orderings are closed under finite intersection.
- For $\preceq_{\kappa}^{\#,\Sigma,a}$ to be a wqo for every order-3 type κ, it is sufficient that the restriction of $\preceq_{\kappa}^{\#,\Sigma,a}$ to $\Lambda_{\kappa}^{\emptyset}$ (i.e. $\preceq_{\kappa}^{\#,\Sigma,a} \cap (\Lambda_{\kappa}^{\emptyset} \times \Lambda_{\kappa}^{\emptyset})$) is a wqo for every order-3 type κ, because $t_1 \preceq_{\kappa}^{\#,\Sigma,a} t_2$ holds if $\lambda\Sigma.t_1 (\preceq_{\Sigma\to\kappa}^{\#,\Sigma,a} \cap (\Lambda_{\Sigma\to\kappa}^{\emptyset} \times \Lambda_{\Sigma\to\kappa}^{\emptyset})) \lambda\Sigma.t_2$, and $\mathtt{order}(\Sigma \to \kappa) \leq 3$.
- For $\preceq_{\kappa}^{\#,\Sigma,a} \cap (\Lambda_{\kappa}^{\emptyset} \times \Lambda_{\kappa}^{\emptyset}))$ to be a wqo, Lemma 15 is sufficient, because $t_1 (\preceq_{\kappa}^{\#,\Sigma,a} \cap (\Lambda_{\kappa}^{\emptyset} \times \Lambda_{\kappa}^{\emptyset})) t_2$ holds if $t_1\, z_1 \cdots z_k \preceq_{\Gamma,\mathtt{o}}^{\#,\Sigma,a} t_2\, z_1 \cdots z_k$, where $\kappa = \kappa_1 \to \cdots \to \kappa_k \to \mathtt{o}$ and $\Gamma = z_1 : \kappa_1, \ldots, z_k : \kappa_k$.

See Appendix C for details. ☐

Henceforth, we fix arbitrary $a_{\mathrm{fix}} \in \Sigma$, and show Lemma 15 for $a = a_{\mathrm{fix}}$. We prove this lemma in two steps: First we give a transformation $(\cdot)^{\natural}$ from order-3 terms in $\Lambda_{\Gamma,\mathtt{o}}^{\emptyset}$ (and their type environment Γ) to order-2 terms in $\Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma^\natural,\mathtt{o}}$ (and to $\Gamma^\natural$) so that it reflects quasi-orderings: $t^\natural \preceq^{\mathbb{N}}_{\Gamma^\natural,\mathtt{o}} t'^\natural$ implies $t \preceq^{\#,\Sigma,a_{\mathrm{fix}}}_{\Gamma,\mathtt{o}} t'$ (Lemma 18). Then we show that $\preceq^{\mathbb{N}}_{\Gamma^\natural,\mathtt{o}}$ on $\Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma^\natural,\mathtt{o}}$ is a wqo (Lemma 19). From these two results, Lemma 15 follows immediately.

4.2 Transformation from order-3 terms to order-2 terms

The key observation behind the transformation $(\cdot)^\natural$ is as follows. Let s be a closed term of type $\mathtt{o}^m \to \mathtt{o}$ and $t_1, \dots, t_m$ be closed terms of type $\mathtt{o}$. Then, we have:

$$\#_a(s\, t_1 \cdots t_m) = c_1 \times \#_a(t_1) + \cdots + c_m \times \#_a(t_m) + d$$

for some numbers $c_1, \dots, c_m, d$ that do not depend on $t_1, \dots, t_m$. This is because the order-1 function s representable as a $\lambda^{\to}$-term can copy only arguments, and the number of copies cannot depend on the arguments. Thus, if we are interested only in the number of occurrences of a constant, information about an order-1 function can be represented by a tuple $(c_1, \dots, c_m, d)$ of numbers (order-0 values, in other words). By lifting this representation to order-3 terms in $\Lambda^{\emptyset}_{\Gamma,\mathtt{o}}$, we obtain order-2 terms in $\Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma^\natural,\mathtt{o}}$.

The actual transformation is non-trivial. Let us first fix $\Gamma = \varphi_1 : \kappa_1, \dots, \varphi_m : \kappa_m, f_1 : \mathtt{o}^{q_1} \to \mathtt{o}, \dots, f_\ell : \mathtt{o}^{q_\ell} \to \mathtt{o}$. Here, φ_i's are order-2 variables and f_j's are variables of order up to 1. Every element of $\Lambda^{\emptyset}_{\Gamma,\mathtt{o}}$ can be normalized to a term generated by the following syntax (which we call an *order-3 normal form*):

$$t ::= y \mid f_j \mid t_1\, t_2 \mid \varphi_i\, t_1 \cdots t_k \mid \lambda y.t.$$

Here, y is a local variable of order 0. We require that the order of $\varphi\, t_1 \cdots t_k$ is at most 1. For example, $\varphi : (\mathtt{o} \to \mathtt{o}) \to \mathtt{o} \to \mathtt{o} \to \mathtt{o},\ f : \mathtt{o} \to \mathtt{o} \to \mathtt{o},\ x : \mathtt{o} \vdash \lambda y : \mathtt{o}.\ \varphi\,(f\,x)\,((\lambda y' : \mathtt{o}.\ f\,y'\,y')\,y) : \mathtt{o} \to \mathtt{o} \to \mathtt{o}$ is an order-3 normal form. It can be checked by induction that for any order-3 normal form t, $\mathtt{eorder}_\Gamma(t) \le 1$ (with a suitable environment Γ). Since any long $\beta\eta$-normal form in $\Lambda^{\emptyset}_{\Gamma,\mathtt{o}}$ with $\mathtt{order}(\Gamma \to \mathtt{o}) = 3$ is an order-3 normal form, considering only order-3 normal forms does not lose generality. In the rest of this section, we use the meta-variable t for order-3 normal forms.

We now define the transformation for order-3 normal forms. Given a term $t_0 \in \Lambda^{\emptyset}_{\Gamma,\mathtt{o}}$, we transform the term in a compositional manner, by transforming each subterm t typed by:

$$\varphi_1 : \kappa_1, \dots, \varphi_m : \kappa_m, f_1 : \mathtt{o}^{q_1} \to \mathtt{o}, \dots, f_\ell : \mathtt{o}^{q_\ell} \to \mathtt{o}; y_1 : \mathtt{o}, \dots, y_n : \mathtt{o} \vdash t : \mathtt{o}^r \to \mathtt{o}$$

to a term e with some suitable type environment. Here, $y_1, \dots, y_n$ are order-0 variables that are bound inside t_0 (rather than t), $\mathtt{order}(\kappa_i) = 2$ for $i \le m$, and $q_i \ge 0$ for $i \le \ell$. We call f_i and φ_i *external variables* and y_i an *internal variable*. Note that an external variable f_i can be order-0.

We first explain how variables and environments are transformed.

- The variables $y_1, \dots, y_n$ will just disappear after the transformation.
- For each order-1 variable f_i of type $\mathtt{o}^{q_i} \to \mathtt{o}$, we prepare a tuple of variables $(c_{f_i,1}, \dots, c_{f_i,q_i}, d_{f_i})$. Each $c_{f_i,j}$ expresses how often f_i copies the j-th argument, and d_{f_i} expresses how often a_{fix} occurs in the value of f_i, so that the number of a_{fix} in $f_i\, t_1, \dots, t_{q_i}$ can be represented by $c_{f_i,1} \times \#_{a_{\mathrm{fix}}}(t_1) + \cdots + c_{f_i,q_i} \times \#_{a_{\mathrm{fix}}}(t_{q_i}) + d_{f_i}$ (recall the observation given at the beginning of this subsection).
- For each order-2 variable φ_i of type $\kappa_i = (\mathtt{o}^{q_1} \to \mathtt{o}) \to \cdots \to (\mathtt{o}^{q_k} \to \mathtt{o}) \to (\mathtt{o}^{q} \to \mathtt{o})$ (where $q_k > 0$), we prepare a tuple of order-1 variables $(g_{\varphi_i,1}, \dots, g_{\varphi_i,q}, h_{\varphi_i}, \hat{h}_{\varphi_i})$. Basically, $g_{\varphi_i,j}$ and h_{φ_i} are analogous to $c_{f_i,j}$ and d_{f_i}, respectively. Given order-1 functions $t_1, \dots, t_k$

whose values are $\vec{u}_1, \dots, \vec{u}_k$ (where each $\vec{u}_\ell$ is a tuple of size $q_\ell + 1$), for each $j \leq q$, the function $\varphi_i\, t_1 \cdots t_k$ copies the j-th order-0 argument $g_{\varphi_i,j}(\vec{u}_1, \dots, \vec{u}_k)$ times, and creates $h_{\varphi_i}(\vec{u}_1, \dots, \vec{u}_k)$ copies of the constant a_{fix}. The other function variable $\hat{h}_{\varphi_i}$ is similar to h_{φ_i} but used for counting an internal variable y_j rather than a_{fix}.

For a type environment

$$\Gamma = \varphi_1 : \kappa_1, \dots, \varphi_m : \kappa_m, f_1 : \mathsf{o}^{q_1} \to \mathsf{o}, \dots, f_\ell : \mathsf{o}^{q_\ell} \to \mathsf{o}$$

where $\kappa_i = (\mathsf{o}^{q_1^i} \to \mathsf{o}) \to \cdots \to (\mathsf{o}^{q_{k_i}^i} \to \mathsf{o}) \to (\mathsf{o}^{q^i} \to \mathsf{o})$ $(q_{k_i}^i > 0,\ i = 1, \dots, k)$, we define:

$$\Gamma^\natural := \overrightarrow{\overrightarrow{g_{\varphi_i,j}}^{j \leq q^i}, h_{\varphi_i}, \hat{h}_{\varphi_i} : \mathsf{o}^{q_1^i+1} \to \dots \to \mathsf{o}^{q_{k_i}^i+1} \to \mathsf{o}}^{i \leq m}, \overrightarrow{\overrightarrow{c_{f_i,j}}^{j \leq q_i}, d_{f_i} : \mathsf{o}}^{i \leq \ell}$$

We now define the transformation of terms. A term t such that

$$\varphi_1 : \kappa_1, \dots, \varphi_m : \kappa_m, f_1 : \mathsf{o}^{q_1} \to \mathsf{o}, \dots, f_\ell : \mathsf{o}^{q_\ell} \to \mathsf{o}; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t : \mathsf{o}^r \to \mathsf{o}$$

is transformed to a tuple $(v_1, \dots, v_n; w_1, \dots, w_r; e)$, using the transformation relation

$$\varphi_1{:}\kappa_1, \dots, \varphi_m{:}\kappa_m, f_1{:}\mathsf{o}^{q_1} \to \mathsf{o}, \dots, f_\ell{:}\mathsf{o}^{q_\ell} \to \mathsf{o}; y_1{:}\mathsf{o}, \dots, y_n{:}\mathsf{o} \vdash t \rhd (v_1, \dots, v_n; w_1, \dots, w_r; e)$$

defined below. Here, each component is constructed from variables $c_{f_i,j}, d_{f_i}, g_{\varphi_i,j}, h_{\varphi_i}, \hat{h}_{\varphi_i}$ above and $\times, +, 0, 1$. The output of the transformation consists of three parts, separated by semicolons: a (possibly empty) sequence $v_1, \dots, v_n$, a (possibly empty) sequence $w_1, \dots, w_r$, and a single element e. The term v_j represents how often y_j is copied, w_j represents how often the j-th argument of t is copied, and e represents how often the constant a_{fix} is copied. The terms v_j and w_j are auxiliary ones for this transformation, and e plays the role of $t^\natural$ explained in Section 4.1.

The transformation relation is defined by the following rules, where $\Gamma = \varphi_1 : \kappa_1, \dots, \varphi_m : \kappa_m, f_1 : \mathsf{o}^{q_1} \to \mathsf{o}, \dots, f_\ell : \mathsf{o}^{q_\ell} \to \mathsf{o}$ is fixed.

$$\frac{}{\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash y_j \rhd (\underbrace{0, \dots, 0}_{j-1}, 1, \underbrace{0, \dots, 0}_{n-j};; 0)} \quad (\textsc{IVar})$$

$$\frac{}{\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash f_i \rhd (\underbrace{0, \dots, 0}_{n}; c_{f_i,1}, \dots, c_{f_i,q_i}; d_{f_i})} \quad (\textsc{Var})$$

$$\frac{\begin{array}{c}\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t_1 \rhd (v_1, \dots, v_n; w_1, \dots, w_r; e) \qquad r \geq 1 \\ \Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t_2 \rhd (v'_1, \dots, v'_n;; e')\end{array}}{\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t_1 t_2 \rhd (v_1 + w_1 v'_1, \dots, v_n + w_1 v'_n; w_2, \dots, w_r; e + w_1 e')} \quad (\textsc{App0})$$

$$\frac{\begin{array}{c}\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t_j \rhd (\vec{v}_j; \vec{w}_j; e_j) \qquad \vec{u}_j = (\vec{w}_j; e_j) \qquad (\text{for each } j \in \{1, \dots, k\}) \\ \vec{u'}_{j,j'} = (\vec{w}_j; v_{j,j'}) \qquad (\text{for each } j \in \{1, \dots, k\} \text{ and } j' \in \{1, \dots, n\}) \\ k \geq 1 \text{ and the type of } t_k \text{ is order-1}\end{array}}{\begin{array}{l}\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash \varphi_i\, t_1 \cdots t_k \rhd \\ (\hat{h}_{\varphi_i}(\vec{u'}_{1,1}, \dots, \vec{u'}_{k,1}) \dots, \hat{h}_{\varphi_i}(\vec{u'}_{1,n}, \dots, \vec{u'}_{k,n}); \\ \ g_{\varphi_i,1}(\vec{u}_1, \dots, \vec{u}_k), \dots, g_{\varphi_i,q_i}(\vec{u}_1, \dots, \vec{u}_k);\ h_{\varphi_i}(\vec{u}_1, \dots, \vec{u}_k))\end{array}} \quad (\textsc{App1})$$

$$\frac{\Gamma; y_1:\mathsf{o},\dots,y_n:\mathsf{o},y_{n+1}:\mathsf{o}\vdash t \rhd (v_1,\dots,v_n,v_{n+1};w_1,\dots,w_r;e)}{\Gamma; y_1:\mathsf{o},\dots,y_n:\mathsf{o}\vdash \lambda y_{n+1}.t \rhd (v_1,\dots,v_n;v_{n+1},w_1,\dots,w_r;e)} \quad (\textsc{Lam})$$

Rules (IVar) (for internal variables of type o) (Var) (for order-1 variables), and (Lam) should be obvious from the intuition on the tuple and the translation of an environment. Rules (App0) and (App1) are for applications of order-1 and order-2 functions respectively. (Note however that in (App0), t_1 itself may be an application of order-2 function, of the form $\varphi\, t_{1,1}\cdots t_{1,k}$.) In (App0), note that $t_1 t_2$ creates w_1 copies of (the value of) t_2, so that the number of copies of y_i can be calculated by $v_i + w_1 v_i'$, where v_i and v_i' are the numbers of copies created by t_1 and t_2 respectively. Rule (App1) is based on the intuition explained above about the translation of order-2 variables. Note that the same function $\hat{h}_{\varphi_i}$ is used for counting $y_1,\dots,y_n$; this is because φ_i does not know y_j (in other words, φ_i cannot be instantiated to a term containing y_j as a free variable), so that the information for counting y_j can only be passed through arguments $\vec{u'}_{j,j'}$.

It should be clear that if $\Gamma; y_1:\mathsf{o},\dots,y_n:\mathsf{o}\vdash t\rhd(v_1,\dots,v_n;w_1,\dots,w_r;e)$ then $v_j, w_{j'}, e\in \Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma^\natural,\mathsf{o}}$ and the order of $\Gamma^\natural\to\mathsf{o}$ is no greater than 2.

▶ **Example 16.** Let $\Gamma = \varphi:(\mathsf{o}\to\mathsf{o})\to\mathsf{o}\to\mathsf{o}, f:\mathsf{o}\to\mathsf{o}$. Then, we have

$$\Gamma^\natural \;=\; g_{\varphi,1},h_\varphi,\hat{h}_\varphi:\mathsf{o}^2\to\mathsf{o},\; c_{f,1},d_f:\mathsf{o}$$

and $t := \lambda y.\varphi(\varphi\, f)\, y$ is transformed to

$$t^\natural \;=\; h_\varphi\big(g_{\varphi,1}(c_{f,1},d_f),h_\varphi(c_{f,1},d_f)\big)+g_{\varphi,1}\big(g_{\varphi,1}(c_{f,1},d_f),h_\varphi(c_{f,1},d_f)\big)\times 0$$

by the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma;y:\mathsf{o}\vdash f\rhd(0;c_{f,1};d_f)}(\textsc{Var})}{\Gamma;y:\mathsf{o}\vdash\varphi\, f\rhd(\hat{h}_\varphi(c_{f,1},0);g_{\varphi,1}(c_{f,1},d_f);h_\varphi(c_{f,1},d_f))}(\textsc{App1})}{\Gamma;y:\mathsf{o}\vdash\varphi(\varphi\, f)\rhd(\hat{h}_\varphi(\vec{u}');g_{\varphi,1}(\vec{u});h_\varphi(\vec{u}))}(\textsc{App1})\quad \dfrac{}{\Gamma;y:\mathsf{o}\vdash y\rhd(1;;0)}(\textsc{IVar})}{\dfrac{\Gamma;y:\mathsf{o}\vdash\varphi(\varphi\, f)\, y\rhd(\hat{h}_\varphi(\vec{u}')+g_{\varphi,1}(\vec{u})\times 1;\;;h_\varphi(\vec{u})+g_{\varphi,1}(\vec{u})\times 0)}{\Gamma;\vdash\lambda y.\varphi(\varphi\, f)\, y\rhd(;\hat{h}_\varphi(\vec{u}')+g_{\varphi,1}(\vec{u})\times 1;h_\varphi(\vec{u})+g_{\varphi,1}(\vec{u})\times 0)}(\textsc{Lam})}(\textsc{App0})$$

where $\vec{u} = g_{\varphi,1}(c_{f,1},d_f),h_\varphi(c_{f,1},d_f)$ and $\vec{u}' = g_{\varphi,1}(c_{f,1},d_f),\hat{h}_\varphi(c_{f,1},0)$. The terms in the bottom line of the derivation, $\hat{h}_\varphi(\vec{u}')+g_{\varphi,1}(\vec{u})\times 1$ and $t^\natural = h_\varphi(\vec{u})+g_{\varphi,1}(\vec{u})\times 0$, have type o under the environment $\Gamma^\natural$, and $\mathtt{eorder}(\lambda\Gamma^\natural.t^\natural) = \mathtt{order}(\Gamma^\natural\to\mathsf{o}) = 2$.

The next example is a slightly modified one involving an external variable $x:\mathsf{o}$ instead of the internal variable $y:\mathsf{o}$. We have

$$(\Gamma,x:\mathsf{o})^\natural \;=\; \Gamma^\natural,\; d_x:\mathsf{o}$$

and $t' := \varphi(\varphi\, f)\, x$ is transformed to

$$t'^\natural \;=\; h_\varphi\big(g_{\varphi,1}(c_{f,1},d_f),h_\varphi(c_{f,1},d_f)\big)+g_{\varphi,1}\big(g_{\varphi,1}(c_{f,1},d_f),h_\varphi(c_{f,1},d_f)\big)\times d_x$$

by the following derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma,x:\mathsf{o};\vdash f\rhd(0;c_{f,1};d_f)}(\textsc{Var})}{\Gamma,x:\mathsf{o};\vdash\varphi\, f\rhd(\hat{h}_\varphi(c_{f,1},0);g_{\varphi,1}(c_{f,1},d_f);h_\varphi(c_{f,1},d_f))}(\textsc{App1})}{\Gamma,x:\mathsf{o};\vdash\varphi(\varphi\, f)\rhd(\hat{h}_\varphi(\vec{u}');g_{\varphi,1}(\vec{u});h_\varphi(\vec{u}))}(\textsc{App1})\quad \dfrac{}{\Gamma,x:\mathsf{o};\vdash x\rhd(0;;d_x)}(\textsc{Var})}{\Gamma,x:\mathsf{o};\vdash\varphi(\varphi\, f)\, x\rhd(\hat{h}_\varphi(\vec{u}')+g_{\varphi,1}(\vec{u})\times 0;\;;h_\varphi(\vec{u})+g_{\varphi,1}(\vec{u})\times d_x)}(\textsc{App0})$$

where $\vec{u}$ and $\vec{u}'$ are the same as above. ◻

Lemma 17 below says that the transformation preserves the meaning of ground terms. Here we regard constants in Σ as variables of up to order 1, and we define a substitution $\theta_\Sigma^{a_{\rm fix}}$ by:

$$\theta_\Sigma^{a_{\rm fix}} := [\overrightarrow{1/c_{a,i}}^{a\in\Sigma, i\le \mathtt{ar}(a)},\ 1/d_{a_{\rm fix}},\ \overrightarrow{0/d_a}^{a\in\Sigma\setminus\{a_{\rm fix}\}}].$$

(Recall that $a_{\rm fix}\in\Sigma$ above is the constant arbitrarily fixed at the end of Section 4.1.)

▶ **Lemma 17** (preservation of meaning). *If $\Sigma;\vdash t \triangleright (;;e)$, then we have $\#_{a_{\rm fix}}(t) = [\![e\theta_\Sigma^{a_{\rm fix}}]\!]$.*

The above lemma follows from a usual substitution lemma (on internal variables) and a subject reduction property; see Appendix D for the proof.

The correctness of the transformation is stated as the following lemma.

▶ **Lemma 18** (ordering reflection). *Let: Σ be an alphabet; $a_{\rm fix}\in\Sigma$; Γ be an environment of the form*

$$\Gamma = \varphi_1:\kappa_1,\dots,\varphi_m:\kappa_m, f_1:\mathsf{o}^{q_1}\to\mathsf{o},\dots,f_\ell:\mathsf{o}^{q_\ell}\to\mathsf{o}$$

where $\mathtt{order}(\kappa_i)=2$ and $q_i\ge 0$; $t,t'\in\Lambda^{\emptyset}_{\Gamma,\mathsf{o}}$; and

$$\Gamma;\vdash t\triangleright(;;e) \qquad \Gamma;\vdash t'\triangleright(;;e').$$

Then we have:

$$t \preceq^{\#,\Sigma,a_{\rm fix}}_{\Gamma,\mathsf{o}} t' \qquad \text{if} \qquad e\preceq^{\mathbb{N}}_{\Gamma^\natural,\mathsf{o}} e'.$$

The proof of the above lemma is given in Appendix D, where we use Lemma 17 and substitution lemmas on external variables.

4.3 $\preceq^{\mathbb{N}}$ on order-2 terms is a wqo

The main goal of this subsection is to prove the following lemma.

▶ **Lemma 19** ($\preceq^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ on order-2 terms is wqo). *For $\Gamma = f_1:\mathsf{o}^{q_1}\to\mathsf{o},\dots,f_n:\mathsf{o}^{q_n}\to\mathsf{o}$, the quasi-ordering $\preceq^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ on $\Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma,\mathsf{o}}$ is a wqo.*

Lemma 15 follows as a corollary of Lemma 19 above and Lemma 18 in the previous subsection:

Proof of Lemma 15. Let $t_0,t_1,\ldots\in\Lambda^{\emptyset}_{\Gamma,\mathsf{o}}$ be an infinite sequence. We have the infinite sequence $e_0,e_1,\ldots\in\Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma^\natural,\mathsf{o}}$ such that $\Gamma;\vdash t_i\triangleright(;;e_i)$, and by Lemma 18, $t_i\preceq^{\#,\Sigma,a_{\rm fix}}_{\Gamma,\mathsf{o}} t_j$ if $e_i\preceq^{\mathbb{N}}_{\Gamma^\natural,\mathsf{o}} e_j$. By Lemma 19, there indeed exist $i,j\ (i<j)$ such that $e_i\preceq^{\mathbb{N}}_{\Gamma^\natural,\mathsf{o}} e_j$. Thus, we have $t_i\preceq^{\#,\Sigma,a_{\rm fix}}_{\Gamma,\mathsf{o}} t_j$ as required. ◻

To prove Lemma 19, we restrict (without loss of generality) $\Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma,\mathsf{o}}$ to the set of β-normal forms (which we call *order-2 polynomials*), generated by the following grammar:

$$P ::= 0 \mid 1 \mid P_1+P_2 \mid P_1\times P_2 \mid f\,P_1\,\cdots\,P_q$$

Here, in $f\,P_1\cdots P_q$, f should have type $\mathsf{o}^q\to\mathsf{o}$. We write $\mathrm{P}^{\mathbb{N}}_2$ for the set of all order-2 polynomials, and write $\mathrm{P}^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ for $\Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma,\mathsf{o}}\cap\mathrm{P}^{\mathbb{N}}_2$. Note that the arity of f may be 0, so that, for example, $f_1(f_2\times(f_2+1))\in\mathrm{P}^{\mathbb{N}}_{f_1:\mathsf{o}\to\mathsf{o},f_2:\mathsf{o},\mathsf{o}}$. Thus, for Lemma 19, the following suffices:

▶ **Lemma 20** ($\preceq^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ on order-2 polynomials is wqo)**.** *For $\Gamma = f_1 : \mathsf{o}^{q_1} \to \mathsf{o}, \ldots, f_n : \mathsf{o}^{q_n} \to \mathsf{o}$, the quasi-ordering $\preceq^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ on $\mathrm{P}^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ is a wqo.*

The idea for proving this lemma is as follows:

- An order-2 polynomial is regarded as a tree. Thus, by Kruskal's tree theorem (Proposition 8), the set $\mathrm{P}^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ is well-quasi-ordered with respect to the homeomorphic embedding $\preceq^{\mathrm{he},\Sigma_{\mathbb{N}}\cup\Gamma}_{\mathsf{o}}$. Unfortunately, however, the relation $P_1 \preceq^{\mathrm{he},\Sigma_{\mathbb{N}}\cup\Gamma}_{\mathsf{o}} P_2$ does not necessarily imply $\preceq^{\mathbb{N}}_{\Gamma,\mathsf{o}}$; for example, if $P_1 = 1$ and $P_2 = f_1(1)$, then $P_1 \preceq^{\mathrm{he},\Sigma_{\mathbb{N}}\cup\Gamma}_{\mathsf{o}} P_2$ holds but $P_1 \preceq^{\mathbb{N}}_{\Gamma,\mathsf{o}} P_2$ does not, because f_1 may be instantiated to $\lambda x.0$. Similarly for $P_1 = f_2$ and $P_2 = f_2 \times 0$.
- To address the problem above, we classify the values of $f \in \mathrm{P}^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ (i.e. elements of $\Lambda^{\Sigma_{\mathbb{N}}}_{\mathsf{o}^q\to\mathsf{o}}$) into a finite number of equivalence classes $A^{(1)}, \ldots, A^{(\ell)}$, and use the classification to further normalize order-2 polynomials, so that $P_1 \preceq^{\mathrm{he},\Sigma_{\mathbb{N}}\cup\Gamma}_{\mathsf{o}} P_2$ implies $P_1 \preceq^{\mathbb{N}}_{\Gamma,\mathsf{o}} P_2$ on the normalized polynomials. For example, in the case of $P_1 = 1$ and $P_2 = f_1(1)$ above, the values of f_1 are classified to (i) those that use the argument, (ii) those that return a positive constant without using the argument, and (iii) those that always return 0. We can then normalize $P_2 = f_1(1)$ to $f_1(1)$ (in case (i)), $f_1(0)$ (in case (ii)), and 0 (in case (iii)), respectively. (In case (ii), any argument is replaced with 0, because the argument is irrelevant.) Thus, we can indeed deduce $P_1 \preceq^{\mathbb{N}}_{\Gamma,\mathsf{o}} P_2$ from $P_1 \preceq^{\mathrm{he},\Sigma_{\mathbb{N}}\cup\Gamma}_{\mathsf{o}} P_2$ when the value of f_1 is restricted to just those in (i); and the same holds also for (ii) and (iii). It follows that the restriction of the relation $\preceq^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ to each classification of the values of $f_1, \ldots, f_\ell \in \mathit{dom}(\Gamma)$ is a wqo. Since the number of classifications is finite, by Dickson's theorem (recall the sentence below Proposition 8), $\preceq^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ (which is the intersection of the restrictions of $\preceq^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ to the finite number of classifications) is also a wqo.

We first formalize and justify the reasoning in the last part (using Dickson's theorem).

▶ **Definition 21** (finite case analysis)**.** For $\Gamma = f_1 : \kappa_1, \ldots, f_n : \kappa_n$, we call a *finite case analysis of* Γ a family $(A_i^j)_{i\le n, j\in J_i}$ of sets such that $\Lambda^{\Sigma_{\mathbb{N}}}_{\kappa_i} = \cup_{j\le J_i} A_i^j$ for each $i \le n$. For $(A_i)_{i\le n}$ such that $A_i \subseteq \Lambda^{\Sigma_{\mathbb{N}}}_{\kappa_i}$, we define a quasi-ordering $\preceq^{\mathbb{N}}_{\Gamma,(A_i)_i}$ on $\Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma,\mathsf{o}}$ as follows:

$$t \preceq^{\mathbb{N}}_{\Gamma,(A_i)_i} t' \qquad\Longleftrightarrow\qquad \forall t_1 \in A_1, \ldots, t_n \in A_n.\ [\![t[t_i/f_i]_i]\!] \le [\![t'[t_i/f_i]_i]\!]$$

We often omit the subscript Γ of $\preceq^{\mathbb{N}}_{\Gamma,(A_i)_i}$ and write $\preceq^{\mathbb{N}}_{(A_i)_i}$.

The following lemma follows immediately from the fact that the intersection of a finite number of wqo's is a wqo (which is in turn an immediate corollary of Dickson's theorem). (see Appendix E for omitted proofs in the rest of this section).

▶ **Lemma 22.** *For $\Gamma = f_1 : \kappa_1, \ldots, f_n : \kappa_n$ and a finite case analysis $(A_i^j)_{i\le n, j\in J_i}$ of Γ, if $\preceq^{\mathbb{N}}_{(A_i^{j_i})_i}$ on $\Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma,\mathsf{o}}$ is a wqo for any "case" $(j_i)_{i\le n} \in \prod_{i\le n} J_i$, then so is $\preceq^{\mathbb{N}}$ on $\Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma,\mathsf{o}}$.*

Thus, to prove Lemma 20, it remains to find an appropriate decomposition $\Lambda^{\Sigma_{\mathbb{N}}}_{\kappa_i} = \cup_{j\le J_i} A_i^j$ (where κ_i is an order-1 type $\mathsf{o}^q \to \mathsf{o}$), and prove that $\preceq^{\mathbb{N}}_{(A_i^{j_i})_i}$ is a wqo.

Henceforth we identify an element of $\Lambda^{\Sigma_{\mathbb{N}}}_{\mathsf{o}^q\to\mathsf{o}}$ with the corresponding element of the polynomial semi-ring $\mathbb{N}[x_1, \ldots, x_q]$. For example, $\lambda x_1.\lambda x_2.((\lambda y.y)x_1) + x_2 \times x_2$ is identified with the polynomial $x_1 + x_2^2$ (which is obtained by normalizing and omitting λ-abstractions, assuming a fixed ordering of the bound variables). For $t \in \Lambda^{\Sigma_{\mathbb{N}}}_{\mathsf{o}^q\to\mathsf{o}}$ we write $\mathtt{poly}(t)$ for the corresponding polynomial.

We define the equivalence relation $\sim$ as the least semi-ring congruence relation on $\mathbb{N}[x_1, \ldots, x_q]$ that satisfies (i) $a \sim 1$ if $a > 0$ and (ii) $x_i^j \sim x_i$ if $j > 0$. For example, $2x_1^2x_2 + 3x_1x_2^2 + x_1 + 4 \sim x_1x_2 + x_1 + 1$, and the quotient set $\mathbb{N}[x_1]/\sim$ consists of:

$[0]_\sim, [1]_\sim, [x_1]_\sim, [x_1 + 1]_\sim,$

and $\mathbb{N}[x_1, x_2]/\sim$ consists of

$[0]_\sim, [1]_\sim, [x_1]_\sim, [x_2]_\sim, [x_1x_2]_\sim, [1+x_1]_\sim, [1+x_2]_\sim, [1+x_1x_2]_\sim, [x_1+x_2]_\sim, \ldots, [1+x_1+x_2+x_1x_2]_\sim.$

In general, $\mathcal{P}(\mathcal{P}([q]))$ (where $[q]$ denotes $\{1, \ldots, q\}$ and $\mathcal{P}(X)$ denotes the powerset of X) gives a complete representation of the quotient set $\mathbb{N}[x_1, \ldots, x_q]/\sim$, i.e.,

$$\mathbb{N}[x_1, \ldots, x_q]/\sim \;=\; \left\{ \Big[\sum_{\{p_1 < \cdots < p_r\} \in \Phi} x_{p_1} \cdots x_{p_r} \Big]_\sim \;\middle|\; \Phi \in \mathcal{P}(\mathcal{P}([q])) \right\}.$$

Through $\mathtt{poly} : \Lambda^{\Sigma_\mathbb{N}}_{\mathsf{o}^q \to \mathsf{o}} \to \mathbb{N}[x_1, \ldots, x_q]$, we can induce an equivalence relation on $\Lambda^{\Sigma_\mathbb{N}}_{\mathsf{o}^q \to \mathsf{o}}$ from $\sim$ on $\mathbb{N}[x_1, \ldots, x_q]$, and let A^Φ_q be the equivalence class corresponding to Φ, i.e.,

$$A^\Phi_q := \left\{ t \in \Lambda^{\Sigma_\mathbb{N}}_{\mathsf{o}^q \to \mathsf{o}} \;\middle|\; \mathtt{poly}(t) \sim \sum_{\{p_1 < \cdots < p_r\} \in \Phi} x_{p_1} \cdots x_{p_r} \right\}. \tag{1}$$

Then we have $\Lambda^{\Sigma_\mathbb{N}}_{\mathsf{o}^q \to \mathsf{o}} = \sqcup_{\Phi \in \mathcal{P}(\mathcal{P}([q]))} A^\Phi_q$. Now, given $\Gamma = f_1 : \mathsf{o}^{q_1} \to \mathsf{o}, \ldots, f_n : \mathsf{o}^{q_n} \to \mathsf{o}$, we have obtained a finite case analysis of Γ as $(A^\Phi_{q_i})_{i \le n, \Phi \in \mathcal{P}(\mathcal{P}([q_i]))}$; for $(\Phi_i)_i \in \prod_{i \le n} \mathcal{P}(\mathcal{P}([q_i]))$, we write $\preceq^\mathbb{N}_{(\Phi_i)_i}$ for $\preceq^\mathbb{N}_{(A^{\Phi_i}_{q_i})_i}$. Thus it remains to show that $\preceq^\mathbb{N}_{(\Phi_i)_i}$ on $\mathrm{P}^\mathbb{N}_{\Gamma, \mathsf{o}}$ is a wqo for each $(\Phi_i)_i \in \prod_{i \le n} \mathcal{P}(\mathcal{P}([q_i]))$.

The following lemma justifies the partition of polynomials based on $\sim$.

▶ **Lemma 23** (zero/positive). *For any $\Gamma = f_1 : \mathsf{o}^{q_1} \to \mathsf{o}, \ldots, f_n : \mathsf{o}^{q_n} \to \mathsf{o}$, $(\Phi_i)_i \in \prod_{i \le n} \mathcal{P}(\mathcal{P}([q_i]))$, and $\Gamma \vdash P : \mathsf{o}$, we have either $P \preceq^\mathbb{N}_{(\Phi_i)_i} 0$ or $1 \preceq^\mathbb{N}_{(\Phi_i)_i} P$.*

In other words, the lemma above says that, given an order-2 polynomial P, whether $P[t_1/f_1, \ldots, t_n/f_n]$ evaluates to 0 or not is solely determined by the equivalence classes $t_1, \ldots, t_n$ belong to.

▶ **Example 24.** Let $\Gamma := f : \mathsf{o}^2 \to \mathsf{o}$, and $\Phi := \{\emptyset, \{1, 2\}\} \in \mathcal{P}(\mathcal{P}([2]))$, which denotes the equivalence class $[1 + x_1x_2]_\sim$. We have $1 \preceq^\mathbb{N}_\Phi f\, P_1\, P_2$ for any P_1 and P_2, since any element of the equivalence class is of the form $a + \cdots$ for some natural number $a \ge 1$.

Based on the property above, we define the rewriting relation $\longrightarrow_{(\Phi_i)_i}$, to simplify order-2 polynomials by replacing (i) subterms that always evaluate to 0, and (ii) arguments of a function that are irrelevant, with 0.

▶ **Definition 25** (rewriting relation and $(\Phi_i)_i$-normal form). For $\Gamma = f_1 : \mathsf{o}^{q_1} \to \mathsf{o}, \ldots, f_n : \mathsf{o}^{q_n} \to \mathsf{o}$ and $(\Phi_i)_i \in \prod_{i \le n} \mathcal{P}(\mathcal{P}([q_i]))$, we define the relation $\longrightarrow^\circ_{(\Phi_i)_i}$ by the following two rules.

- $P \longrightarrow^\circ_{(\Phi_i)_i} 0$ if $P \preceq^\mathbb{N}_{(\Phi_i)_i} 0$ and $P \ne 0$.
- $f_\ell\, P_1 \cdots P_{q_\ell} \longrightarrow^\circ_{(\Phi_i)_i} f_\ell\, P_1 \cdots P_{k-1}\, 0\, P_{k+1} \cdots P_{q_\ell}$ if (i) $P_k \ne 0$ and (ii) for all $\phi \in \Phi_\ell$ such that $k \in \phi$, there exists $p \in \phi$ such that $P_p \preceq^\mathbb{N}_{(\Phi_i)_i} 0$.

We write $P_0 \longrightarrow_{(\Phi_i)_i} P_1$ if $P_i = E[P'_i]$ and $P'_0 \longrightarrow^\circ_{(\Phi_i)_i} P'_1$ for some E, P'_0 and P'_1, where the evaluation context E is defined by:

$$E ::= [\,] \mid E + P \mid P + E \mid E \times P \mid P \times E \mid f\, P_1\, \ldots\, P_{i-1}\, E\, P_{i+1}\, \ldots\, P_q.$$

We call a normal form of $\longrightarrow_{(\Phi_i)_i}$ a $(\Phi_i)_i$-*normal form.*

Intuitively, the condition (ii) in the second rule says that whenever the k-th argument P_k is used by f_ℓ, it occurs only in the form of $P_k \times P_p \times \cdots$ (up to equivalence) and P_p always evaluates to 0; thus, the value of P_k is actually irrelevant.

▶ **Example 26.** We continue Example 24. Recall $\Gamma = f : \mathsf{o}^2 \to \mathsf{o}$ and $\Phi = \{\emptyset, \{1,2\}\}$. Consider the order-2 polynomial $f\,1\,(1 \times 0)$. It can be rewritten to $f\,1\,0$ by using the first rule (and the evaluation context $E = f\,1\,[\,]$). We can further apply the second rule to obtain $f\,1\,0 \longrightarrow_\Phi f\,0\,0$, because $k = 1$ satisfies the conditions ((i) and) (ii). In fact, if $1 \in \phi \in \Phi$, then $\phi = \{1,2\}$; hence, the required condition holds for $p = 2$. Note that $f\,0\,0$ is a Φ-normal form; the first rule is not applicable, as $f\,0\,0 \not\preceq^{\mathbb{N}}_{\Phi} 0$ by the discussion in Example 24.

The following lemma guarantees that any order-2 polynomial can be transformed to at least one equivalent $(\Phi_i)_i$-*normal form.*

▶ **Lemma 27** (existence of normal form)**.**

1. $\longrightarrow_{(\Phi_i)_i}$ *is strongly normalizing.*
2. *If* $P \longrightarrow_{(\Phi_i)_i} P'$ *then* $P \approx^{\mathbb{N}}_{(\Phi_i)_i} P'$.

We can reduce the wqoness of $\preceq^{\mathbb{N}}_{(\Phi_i)_i}$ to that of $\preceq^{\mathrm{he},\Sigma_{\mathbb{N}}\cup\Gamma}_{\mathsf{o}}$ by the following lemma:

▶ **Lemma 28.** *For* $\Gamma = f_1 : \mathsf{o}^{q_1} \to \mathsf{o}, \dots, f_n : \mathsf{o}^{q_n} \to \mathsf{o}$, $(\Phi_i)_i \in \prod_{i\le n} \mathcal{P}(\mathcal{P}([q_i]))$, *and* $(\Phi_i)_i$-*normal forms* $\Gamma \vdash P', P : \mathsf{o}$, *if* $P' \preceq^{\mathrm{he},\Sigma_{\mathbb{N}}\cup\Gamma}_{\mathsf{o}} P$ *then* $P' \preceq^{\mathbb{N}}_{(\Phi_i)_i} P$.

The proof is given by a simple calculation using Lemma 23 and that the given $(\Phi_i)_i$-normal forms P', P do not satisfy the condition for the rewriting $\longrightarrow_{(\Phi_i)_i}$.

Now we are ready to prove Lemma 20.

Proof of Lemma 20. By Lemma 22, it suffices to show that $\preceq^{\mathbb{N}}_{(\Phi_i)_i}$ on $\mathrm{P}^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ is a wqo for each $(\Phi_i)_i \in \prod_{i\le n} \mathcal{P}(\mathcal{P}([q_i]))$. By the Kruskal's tree theorem, $\preceq^{\mathrm{he},\Sigma_{\mathbb{N}}\cup\Gamma}_{\mathsf{o}}$ on $\mathrm{P}^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ is a wqo, and hence the sub-ordering $\preceq^{\mathrm{he},\Sigma_{\mathbb{N}}\cup\Gamma}_{\mathsf{o}}$ on the subset

$$\{P \in \mathrm{P}^{\mathbb{N}}_{\Gamma,\mathsf{o}} \mid P \text{ is a } (\Phi_i)_i\text{-normal form}\} \subseteq \mathrm{P}^{\mathbb{N}}_{\Gamma,\mathsf{o}}$$

is a wqo. Therefore by Lemma 28, $\preceq^{\mathbb{N}}_{(\Phi_i)_i}$ on $\{P \in \mathrm{P}^{\mathbb{N}}_{\Gamma,\mathsf{o}} \mid P \text{ is a } (\Phi_i)_i\text{-normal form}\}$ is a wqo. By Lemma 27, $\{P \in \mathrm{P}^{\mathbb{N}}_{\Gamma,\mathsf{o}} \mid P \text{ is a } (\Phi_i)_i\text{-normal form}\}$ and $\mathrm{P}^{\mathbb{N}}_{\Gamma,\mathsf{o}}$—both modulo $\beta\eta$-equivalence—are isomorphic (with respect to $\preceq^{\mathbb{N}}_{(\Phi_i)_i}$ and $\preceq^{\mathbb{N}}_{(\Phi_i)_i}$); hence $\preceq^{\mathbb{N}}_{(\Phi_i)_i}$ on $\mathrm{P}^{\mathbb{N}}_{\Gamma,\mathsf{o}}$ is a wqo. ◻

5 Conclusion

W have introduced the nAK-conjecture, a weaker version of the AK-conjecture in [2], and proved it up to order 3. We have also proved a pumping lemma for higher-order grammars (which is slightly weaker than the pumping lemma conjectured in [2]) under the assumption that the nAK-conjecture holds. Obvious future work is to show the nAK-conjecture or the original AK-conjecture for arbitrary orders. Finding other applications of the two conjectures (cf. an application of Kruskal's tree theorem to program termination [5]) is also left for future work.

Acknowledgments

We would like to thank anonymous referees for useful comments. This work was supported by JSPS Kakenhi 15H05706 and 18K11156.

References

1 Kazuyuki Asada and Naoki Kobayashi. On Word and Frontier Languages of Unsafe Higher-Order Grammars. In *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *LIPIcs*, pages 111:1–111:13. Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016.

2 Kazuyuki Asada and Naoki Kobayashi. Pumping lemma for higher-order languages. In Ioannis Chatzigiannakis, Piotr Indyk, Fabian Kuhn, and Anca Muscholl, editors, *44th International Colloquium on Automata, Languages, and Programming, ICALP 2017, July 10-14, 2017, Warsaw, Poland*, volume 80 of *LIPIcs*, pages 97:1–97:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. URL: `https://doi.org/10.4230/LIPIcs.ICALP.2017.97`, `doi:10.4230/LIPIcs.ICALP.2017.97`.

3 Kazuyuki Asada and Naoki Kobayashi. Pumping Lemma for Higher-order Languages. *ArXiv e-prints, 1705.10699*, May 2017. URL: `https://arxiv.org/abs/1705.10699`, `arXiv:1705.10699`.

4 Werner Damm. The IO- and OI-hierarchies. *Theor. Comput. Sci.*, 20:95–207, 1982.

5 Nachum Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17(3):279 – 301, 1982. URL: `http://www.sciencedirect.com/science/article/pii/0304397582900263`, `doi:https://doi.org/10.1016/0304-3975(82)90026-3`.

6 Takeshi Hayashi. On derivation trees of indexed grammars –an extension of the uvwxy-theorem–. *Publ. RIMS, Kyoto Univ.*, pages 61–92, 1973.

7 Graham Higman. Ordering by divisibility in abstract algebras. *Proceedings of the London Mathematical Society*, 3(1):326–336, 1952.

8 Joseph B. Kruskal. Well-quasi-ordering, the tree theorem, and vazsonyi's conjecture. *Transactions of the American Mathematical Society*, 95(2):210–225, 1960. URL: `http://www.jstor.org/stable/1993287`.

9 Pawel Parys. Intersection types and counting. In Naoki Kobayashi, editor, *Proceedings Eighth Workshop on Intersection Types and Related Systems, ITRS 2016, Porto, Portugal, 26th June 2016.*, volume 242 of *EPTCS*, pages 48–63, 2016. URL: `https://doi.org/10.4204/EPTCS.242.6`, `doi:10.4204/EPTCS.242.6`.

10 Pawel Parys. The complexity of the diagonal problem for recursion schemes. In Satya V. Lokam and R. Ramanujam, editors, *37th IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2017, December 11-15, 2017, Kanpur, India*, volume 93 of *LIPIcs*, pages 45:1–45:14. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2017. URL: `https://doi.org/10.4230/LIPIcs.FSTTCS.2017.45`, `doi:10.4230/LIPIcs.FSTTCS.2017.45`.

11 Kazushige Terui. 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*, pages 323–338. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

12 Mitchell Wand. An algebraic formulation of the Chomsky hierarchy. In *Category Theory Applied to Computation and Control*, volume 25 of *LNCS*, pages 209–213. Springer, 1974.

13 Marek Zaionc. Word operation definable in the typed lambda-calculus. *Theor. Comput. Sci.*, 52:1–14, 1987. URL: `http://dx.doi.org/10.1016/0304-3975(87)90077-6`, `doi:10.1016/0304-3975(87)90077-6`.

14 Marek Zaionc. On the "lambda"-definable tree operations. In *Algebraic Logic and Universal Algebra in Computer Science, Conference, Ames, Iowa, USA, June 1-4, 1988, Proceedings*, volume 425 of *Lecture Notes in Computer Science*, pages 279–292, 1990.

15 Marek Zaionc. Lambda representation of operations between different term algebras. In Leszek Pacholski and Jerzy Tiuryn, editors, *Computer Science Logic, 8th International Workshop, CSL '94, Kazimierz, Poland, September 25-30, 1994, Selected Papers*, volume

933 of *Lecture Notes in Computer Science*, pages 91–105. Springer, 1994. URL: https://doi.org/10.1007/BFb0022249, doi:10.1007/BFb0022249.

A Preliminaries for Appendix

▶ **Definition 29** (contexts)**.** The set of *contexts*, ranged over by C, D, G, H, is defined by $C ::= [\,] \mid C\,t \mid t\,C \mid \lambda x.C$ where t ranges over $\lambda^{\to,\mathrm{nd}}$-terms. We write $C[t]$ for the $\lambda^{\to,\mathrm{nd}}$-term obtained from C by replacing $[\,]$ with t. Note that the replacement may capture variables; e.g., $(\lambda x.[\,])[x]$ is $\lambda x.x$. We call C a (Γ',κ')-(Γ,κ)-*context* if $\Gamma \vdash_{\mathsf{ST}} C : \kappa$ is derived by using axiom $\Gamma' \vdash_{\mathsf{ST}} [\,] : \kappa'$. We also call a $(\emptyset,\kappa')$-$(\emptyset,\kappa)$-context a κ'-κ-*context*. The (internal) *order* of a (Γ',κ')-(Γ,κ)-context, is the largest order of the types occurring in the derivation of $\Gamma \vdash_{\mathsf{ST}} C : \kappa$. A context is called a $\lambda^{\to}$-*context* if it does not contain $\oplus$. The size $|C|$ of a context C is defined similarly to that of λY^{nd}-terms, with $|[\,]| := 0$.

▶ **Definition 30** (linear term / pair of contexts)**.** For a $\lambda^{\to}$-term $x : \kappa \vdash_{\mathsf{ST}} t : \mathsf{o}$, we call t *linear* (with respect to x) if x occurs exactly once in the call-by-name normal form of t. A pair of $\lambda^{\to}$-contexts $[\,] : \kappa \vdash_{\mathsf{ST}} C : \mathsf{o}$ and $[\,] : \kappa \vdash_{\mathsf{ST}} D : \kappa$ is called *linear* if $x : \kappa \vdash_{\mathsf{ST}} C[D^i[x]] : \mathsf{o}$ is linear for any $i \geq 0$ where x is a fresh variable that is not captured by the context applications.

▶ **Definition 31** (higher-order grammar)**.** A *higher-order grammar* (or *grammar* for short) is a quadruple $(\Sigma, \mathcal{N}, \mathcal{R}, S)$, where (i) Σ is a ranked alphabet; (ii) $\mathcal{N}$ is a map from a finite set of *non-terminals* to their types; (iii) $\mathcal{R}$ is a finite set of *rewriting rules* of the form $A \to \lambda x_1.\cdots\lambda x_\ell.t$, where $\mathcal{N}(A) = \kappa_1 \to \cdots \to \kappa_\ell \to \mathsf{o}$, t is an applicative term, and $\mathcal{N}, x_1 : \kappa_1, \ldots, x_\ell : \kappa_\ell \vdash_{\mathsf{ST}} t : \mathsf{o}$ holds; (iv) S is a non-terminal called the *start symbol*, and $\mathcal{N}(S) = \mathsf{o}$. The *order* of a grammar $\mathcal{G}$ is the largest order of the types of non-terminals. We sometimes write $\Sigma_{\mathcal{G}}, \mathcal{N}_{\mathcal{G}}, \mathcal{R}_{\mathcal{G}}, S_{\mathcal{G}}$ for the four components of $\mathcal{G}$. We often write $A\,x_1\,\cdots\,x_k \to t$ for the rule $A \to \lambda x_1.\cdots\lambda x_k.t$.

For a grammar $\mathcal{G} = (\Sigma, \mathcal{N}, \mathcal{R}, S)$, the rewriting relation $\longrightarrow_{\mathcal{G}}$ is defined by:

$$\frac{(A \to \lambda x_1.\cdots\lambda x_k.t) \in \mathcal{R}}{A\,t_1\,\cdots\,t_k \longrightarrow_{\mathcal{G}} t[t_1/x_1,\ldots,t_k/x_k]} \qquad \frac{t_i \longrightarrow_{\mathcal{G}} t_i' \qquad i \in \{1,\ldots,k\} \qquad \Sigma(a) = k}{a\,t_1\,\cdots\,t_k \longrightarrow_{\mathcal{G}} a\,t_1\,\cdots\,t_{i-1}\,t_i'\,t_{i+1}\,\cdots\,t_k}$$

We write $\longrightarrow_{\mathcal{G}}^*$ for the reflexive transitive closure of $\longrightarrow_{\mathcal{G}}$. The *tree language generated by* $\mathcal{G}$, written $\mathcal{L}(\mathcal{G})$, is the set $\{\pi \mid S \longrightarrow_{\mathcal{G}}^* \pi\}$. We call a grammar $\mathcal{G}$ a *word grammar* if its alphabet is a word alphabet.

As explained after Definition 2, we have:

$$\{\mathcal{L}(\mathcal{G}) \mid \mathcal{G} \text{ is an order-}n \text{ tree grammar}\} = \{\mathcal{L}(t) \mid t \text{ is an order-}n \text{ ground closed } \lambda Y^{\mathrm{nd}}\text{-term}\}$$

for any n.

▶ **Definition 32** (frontier language / $\mathtt{br}$-alphabet)**.** The *frontier word* of a tree π, written $\mathbf{leaves}(\pi)$, is the sequence of symbols in the leaves of π. It is defined inductively by: $\mathbf{leaves}(a) = a$ when $\Sigma(a) = 0$, and $\mathbf{leaves}(a\,\pi_1\,\cdots\,\pi_k) = \mathbf{leaves}(\pi_1)\cdots\mathbf{leaves}(\pi_k)$ when $\Sigma(a) = k > 0$. The *frontier language* generated by t, written $\mathcal{L}_{\mathtt{leaf}}(t)$, is the set: $\{\mathbf{leaves}(\pi) \mid \pi \in \mathcal{L}_{\mathtt{leaf}}(t)\}$. A $\mathtt{br}$-*alphabet* is a ranked alphabet such that it has a special binary constant $\mathtt{br}$ and a special nullary constant $\mathtt{e}$ and the other constants are nullary. We consider $\mathtt{e}$ as the empty word ε: for a ground closed λY^{nd}-term with a $\mathtt{br}$-alphabet, we also define $\mathcal{L}^{\varepsilon}_{\mathtt{leaf}}(t) := (\mathcal{L}_{\mathtt{leaf}}(t) \setminus \{\mathtt{e}\}) \cup \{\varepsilon \mid \mathtt{e} \in \mathcal{L}_{\mathtt{leaf}}(t)\}$. We call a tree π an $\mathtt{e}$-*free* $\mathtt{br}$-*tree* if it is a tree of some $\mathtt{br}$-alphabet but does not contain $\mathtt{e}$.

Proof of Lemma 6. Since every constant is monotonic on $\leq$ by definition, the basic lemma of logical relation holds, which exactly means that $\leq_\kappa$ is reflexive, from which the transitivity and being point-wise follow. ◻

B Proof of the Pumping Lemma

As explained after Theorem 12, the proof of this theorem is just a simple modification of the proof of the pumping lemma in [2]; the (essentially) modified points are Lemmas 34 and 43 below. We first gives some lemmas which are independent of orderings. Then, as explained in Remark 3, we show some generalized pumping lemma, where we generalize $\preceq^{\#,\Sigma}$ to an arbitrary quasi-ordering with certain conditions. Lastly, we show that Theorem 12 is an instance of the generalized pumping lemma.

B.1 Lemmas independent of ordering

The next lemma is exactly the same as Lemma 11 in [2].

▶ **Lemma 33** ([2]). *Given an order-n grammar $\mathcal{G}$ such that $\mathcal{L}(\mathcal{G})$ is infinite, there exist order-n linear $\lambda^{\rightarrow}$-contexts C, D, and an order-n closed $\lambda^{\rightarrow}$-term t such that*

1. $\{\mathcal{T}(C[D^k[t]]) \mid k \geq 1\} \subseteq \mathcal{L}(\mathcal{G})$
2. $\{\mathcal{T}(C[D^{\ell_k}[t]]) \mid k \geq 1\}$ *is infinite for any strictly increasing sequence* $(\ell_k)_k$.

In [2], we also used a similar lemma to the above, which reasons about the length of a particular path of a tree. We slightly modify this lemma as follows.

We annotate each constructor a as $a^{\langle i\rangle}$ as follows: We write Σ_{d} for the alphabet $\{a^{\langle i\rangle} \mapsto k \mid (a \mapsto k) \in \Sigma,\ 0 \leq i \leq k\}$. Then we call i of $a^{\langle i\rangle}$ a *direction* and call a λY^{nd}-term / context of alphabet Σ_{d} *direction-annotated*.

We define a function **path** which extracts the path from a direction-annotated tree: we write Σ_{p} for the word alphabet $\{a_i \mapsto 1 \mid (a \mapsto k) \in \Sigma,\ 1 \leq i \leq k\} \cup \{\mathtt{e} \mapsto 0\}$, and define the function **path** from trees of Σ_{d} to trees of Σ_{p}, by the following induction: $\mathbf{path}(a^{\langle i\rangle}\,\pi_1 \cdots \pi_\ell) := a_i\,\mathbf{path}(\pi_i)$ if $i > 0$ and $\mathbf{path}(a^{\langle 0\rangle}\,\pi_1 \cdots \pi_\ell) := \mathtt{e}$. We also define **rmdir** as the function that removes all the direction annotations.

We call a direction annotated tree π *0-annotated* if all the directions in π are 0; then we define a direction annotated tree π to be *(direction-)well-annotated* by the following induction: $a^{\langle 0\rangle}\pi_1 \cdots \pi_k$ is well-annotated if π_j is 0-annotated for any j; and $a^{\langle i\rangle}\pi_1 \cdots \pi_k$ $(i > 0)$ is well-annotated if π_j is 0-annotated for any $j \neq i$ and π_i is well-annotated.

The next lemma is modified from Lemma 12 in [2]; we just added the natural condition of well-annotatedness (condition 3 below).

▶ **Lemma 34.** *For any order-n linear $\lambda^{\rightarrow}$-contexts C, D and any order-n closed $\lambda^{\rightarrow}$-term t such that $\{\mathcal{T}(C[D^k[t]]) \mid k \geq 1\}$ is infinite, there exist direction-annotated order-n linear $\lambda^{\rightarrow}$-contexts G, H, a direction-annotated order-n closed $\lambda^{\rightarrow}$-term u, and $p, q > 0$ such that*

1. $\mathbf{rmdir}(\mathcal{T}(G[H^k[u]])) = \mathcal{T}(C[D^{pk+q}[t]])$ *for any* $k \geq 1$
2. $\{|\mathbf{path}(\mathcal{T}(G[H^{\ell_k}[u]]))| \mid k \geq 1\}$ *is infinite for any strictly increasing sequence* $(\ell_k)_k$.
3. $\mathcal{T}(G[H^k[u]])$ *is well-annotated for any* $k \geq 1$.

Proof. We replace the rule (PDT-Const) used in Appendix A.5 of [3] with the following one:

$$\frac{\begin{array}{c}\mathtt{ar}(a) = k \\ \Theta_i \vdash^{\mathtt{dir}}_n t_j : (F_j, M_j, \mathtt{o}) \triangleright c_j \Rightarrow s_j \text{ for each } j \in \{1, \dots, k\} \\ M = M_1 \uplus \cdots \uplus M_k \qquad 0 \notin \bigcup_{j \in \{1,\dots,k\}\setminus\{i\}} M_j \\ 0 \in M_i \text{ if } i > 0 \\ \mathit{Comp}_n(\{(\{0\}, 0), (F_1, c_1), \dots, (F_k, c_k)\}, M) = (F, c)\end{array}}{\Theta_1 + \cdots + \Theta_k \vdash^{\mathtt{dir}}_n a^{\langle i \rangle}\, t_1 \, \cdots \, t_k : (F, M, \mathtt{o}) \triangleright c \Rightarrow a^{\langle i \rangle}\, s_1 \, \cdots \, s_k} \quad \text{(PDT-Const)}$$

Here, we allow i to range over $\{0, 1, \dots, k\}$. The new condition "$0 \in M_i$ if $i > 0$" ensures directions to be well-annotated. All the remaining parts of the proof are completely analogous to those in [3]. ☐

The next lemma is exactly the same as Lemma 13 in [2].

▶ **Lemma 35** ([2]). *Given order-n $\lambda^{\to}$-contexts C, D, and an order-n closed $\lambda^{\to}$-term t such that*

- *the constants in C, D, t are in a word alphabet Σ,*
- *$\{\mathcal{T}(C[D^{\ell_i}[t]]) \mid i \geq 0\}$ is infinite for any strictly increasing sequence $(\ell_i)_i$, and*
- *C and D are linear,*

there exist order-$(n-1)$ $\lambda^{\to}$-contexts G, H, order-$(n-1)$ closed $\lambda^{\to}$-term u, and some constant numbers c, $d \geq 1$ such that

- *the constants in G, H, u are in the $\mathtt{br}$-alphabet $\{a \mapsto 0 \mid a \in \mathit{dom}(\Sigma) \setminus \{\mathtt{e}\}\} \cup \{\mathtt{e} \mapsto 0, \mathtt{br} \mapsto 2\}$.*
- *for $i \geq 0$, $\mathcal{T}(G[H^i[u]])$ is either an $\mathtt{e}$-free $\mathtt{br}$-tree or $\mathtt{e}$, and*

$$\mathbf{word}(\mathcal{T}(C[D^{ci+d}[t]])) = \begin{cases} \varepsilon & (\mathcal{T}(G[H^i[u]]) = \mathtt{e}) \\ \mathbf{leaves}(\mathcal{T}(G[H^i[u]])) & (\mathcal{T}(G[H^i[u]]) \neq \mathtt{e}) \end{cases}$$

- *G and H are linear.*

B.2 Generalized pumping lemma

Here we give the generalized pumping lemma. First we abstract the nAK-conjecture and the AK-conjecture, using the following notion: we call a *parametrized quasi-ordering* a family of quasi-orderings $(\leq^{\Sigma}_{\kappa})_{\Sigma,\kappa}$ parametrized by a ranked alphabet Σ and a simple type κ where $\leq^{\Sigma}_{\kappa}$ is a quasi-ordering on the set of closed $\lambda^{\to}$-terms of type κ and of alphabet Σ modulo $\beta\eta$-equivalence.

▶ **Condition 36** (Parametrized well-quasi-ordering). Let $(\leq^{\Sigma}_{\kappa})_{\Sigma,\kappa}$ be a parametrized quasi-ordering.

- $\leq^{\Sigma}_{\kappa}$ is a well-quasi-ordering; i.e., for an infinite sequence $t_1, t_2, \dots$ of closed $\lambda^{\to}$-terms of type κ, there exist $i < j$ such that $t_i \leq^{\Sigma}_{\kappa} t_j$.
- $(\leq^{\Sigma}_{\kappa})_{\kappa}$ is closed under application, i.e., if $t \leq^{\Sigma}_{\kappa_1 \to \kappa_2} t'$ and $s \leq^{\Sigma}_{\kappa_1} s'$ then $t\, s \leq^{\Sigma}_{\kappa_2} t'\, s'$.

The nAK-conjecture (the AK-conjecture, resp.) says that $(\preceq^{\#,\Sigma}_{\kappa})_{\Sigma,\kappa}$ ($(\preceq^{\mathrm{he},\Sigma}_{\kappa})_{\Sigma,\kappa}$, resp.) satisfies Condition 36; note that the second condition on closedness under application is satisfied by any logical relation extension. Also note that the AK-conjecture implies the nAK-conjecture, since $(\preceq^{\mathrm{he},\Sigma}_{\kappa}) \subseteq (\preceq^{\#,\Sigma}_{\kappa})$ as explained in Section 2.2.

Similarly to [2], for the pumping lemma in the current paper, the following weaker property called the *periodicity* is sufficient.

▶ **Condition 37** (Periodicity). Let $(\leq_\kappa^\Sigma)_{\Sigma,\kappa}$ be a parametrized quasi-ordering.

- For any $\vdash_{\mathtt{ST}} t : \kappa \to \kappa$ and $\vdash_{\mathtt{ST}} s : \kappa$, there exist $i, j > 0$ such that

$$t^i\, s \leq_\kappa^\Sigma t^{i+j}\, s \leq_\kappa^\Sigma t^{i+2j}\, s \leq_\kappa^\Sigma \cdots .$$

- $(\leq_\kappa^\Sigma)_\kappa$ is closed under application.

Condition 36 implies Condition 37 (cf. [2]), and we use the latter in the proof of the generalized pumping lemma (Theorem 39).

In the generalized pumping lemma, we consider an arbitrary parametrized quasi-ordering that satisfies (Condition 37 and) the following condition, which is satisfied by $(\preceq_\kappa^{\#,\Sigma})_{\Sigma,\kappa}$ as shown in Section B.3. The next condition is essentially a condition on $(\leq_{\mathtt{o}}^\Sigma)_\Sigma$ rather than $(\leq_\kappa^\Sigma)_{\Sigma,\kappa}$.

▶ **Condition 38.** Let $(\leq_\kappa^\Sigma)_{\Sigma,\kappa}$ be a parametrized quasi-ordering.

1. For any order-0 linear $\lambda^\to$-contexts C, D and any tree π such that $D \neq [\,]$, we have $C[D[\pi]] \not\leq_{\mathtt{o}}^\Sigma C[\pi]$.
2. Let Σ_{ur}, Σ_{br}, and Σ_{wd} be an unranked alphabet, the $\mathtt{br}$-alphabet of Σ_{ur}, and the word alphabet of Σ_{ur}, respectively. For $\mathtt{e}$-free $\mathtt{br}$-trees π and π' of Σ_{br}, if $\pi <_{\mathtt{o}}^{\Sigma_{\mathrm{br}}} \pi'$ then $\mathbf{utree}(\mathbf{leaves}(\pi)) <_{\mathtt{o}}^{\Sigma_{\mathrm{wd}}} \mathbf{utree}(\mathbf{leaves}(\pi'))$.
3. Let Σ be a ranked alphabet. For well-annotated trees π and π', if $\pi <_{\mathtt{o}}^{\Sigma_{\mathrm{d}}} \pi'$ then $\mathbf{rmdir}(\pi) <_{\mathtt{o}}^{\Sigma} \mathbf{rmdir}(\pi')$.
4. Let Σ be a ranked alphabet. For well-annotated trees π and π', if $\pi \sim_{\mathtt{o}}^{\Sigma_{\mathrm{d}}} \pi'$ then $\mathbf{path}(\pi) \sim_{\mathtt{o}}^{\Sigma_{\mathrm{p}}} \mathbf{path}(\pi')$.

On Condition 38-1, note that a linear pair of order-0 $\lambda^\to$-contexts is nothing but a pair of trees except that each of them contains just one context-hole $[\,]$ of type $\mathtt{o}$. In [2], we used that $\preceq_{\mathtt{o}}^{\mathrm{he},\Sigma}$ satisfies the above four conditions. Conditions 38-2 and 38-3 corresponds to Lemmas 14 and 15 in [2], while Conditions 38-1 and 38-4 are trivial and hence implicit in [2]; note that Condition 38-4 is trivial if $\leq_{\mathtt{o}}^\Sigma$ is ordering (i.e. antisymmetric) for any Σ, as $\preceq_{\mathtt{o}}^{\mathrm{he},\Sigma}$ is.

Now we give the generalized pumping lemma:

▶ **Theorem 39** (generalized pumping lemma). *Let $(\leq_\kappa^\Sigma)_{\Sigma,\kappa}$ be a parametrized quasi-ordering that satisfies Conditions 37 and 38.*

Then, for any order-n grammar $\mathcal{G}$ such that $\mathcal{L}(\mathcal{G})$ is infinite, there exist an infinite sequence of trees $\pi_0, \pi_1, \pi_2, \ldots \in \mathcal{L}(\mathcal{G})$, and constants c, d such that: (i) $\pi_0 \leq_{\mathtt{o}}^\Sigma \pi_1 \leq_{\mathtt{o}}^\Sigma \pi_2 \leq_{\mathtt{o}}^\Sigma \cdots$, and (ii) $|\pi_i| \leq \mathbf{exp}_n(ci+d)$ for each $i \geq 0$.

Of course, we also have the generalized version of Corollary 13. The above theorem follows immediately from Lemma 33 and the next lemma. The proof of the next lemma is essentially the same as that of Lemma 16 in [2] except for the explicit use of Condition 38-4.

▶ **Lemma 40.** *Let $(\leq_\kappa^\Sigma)_{\Sigma,\kappa}$ be a parametrized quasi-ordering that satisfies Conditions 37 and 38. Let n be a natural number and Σ be an alphabet. For any order-n linear $\lambda^\to$-contexts C, D and any order-n $\lambda^\to$-term t such that $\{\mathcal{T}(C[D^i[t]]) \mid i \geq 1\}$ is infinite, there exist c, d, j, $k \geq 1$ such that*

- $\mathcal{T}(C[D^j[t]]) <_{\mathtt{o}}^\Sigma \mathcal{T}(C[D^{j+k}[t]]) <_{\mathtt{o}}^\Sigma \mathcal{T}(C[D^{j+2k}[t]]) <_{\mathtt{o}}^\Sigma \cdots$
- $|\mathcal{T}(C[D^{j+ik}[t]])| \leq \mathbf{exp}_n(ci+d) \qquad (i = 0, 1, \ldots)$

Proof. The proof proceeds by induction on n. The case where $n = 0$ follows immediately from Conditions 37 and 38-1. We discuss the case $n > 0$ below.

By Lemma 34, from C, D, and t, we obtain direction-annotated order-n linear $\lambda^{\to}$-contexts G, H, a direction-annotated order-n $\lambda^{\to}$-term u, and j_0, $k_0 > 0$ such that

$$\mathbf{rmdir}(\mathcal{T}(G[H^i[u]])) = \mathcal{T}(C[D^{j_0+ik_0}[t]]) \text{ for any } i \geq 1 \tag{2}$$

$$\{|\mathbf{path}(\mathcal{T}(G[H^{\ell_i}[u]]))| \mid i \geq 1\} \text{ is infinite for any strictly increasing sequence } (\ell_i)_i. \tag{3}$$

$$\mathcal{T}(G[H^i[u]]) \text{ is well-annotated for any } i \geq 1. \tag{4}$$

Next we transform G, H, and u by choosing a path according to directions as **path** does, i.e., we define G_{p}, H_{p}, and u_{p} as the $\lambda^{\to}$-contexts/term of Σ_{p} obtained from G, H, and u by replacing each $a^{\langle i\rangle}$ with: (i) $\lambda x_1 \ldots x_\ell.a_i x_i$ if $i > 0$ or (ii) $\lambda x_1 \ldots x_\ell.\mathtt{e}$ if $i = 0$, where $\ell = \Sigma(a)$. For any $i \geq 0$, we have

$$\mathbf{path}(\mathcal{T}(G[H^i[u]])) = \mathcal{T}(G_{\mathrm{p}}[{H_{\mathrm{p}}}^i[u_{\mathrm{p}}]]). \tag{5}$$

By (3) and (5), $\{\mathcal{T}(G_{\mathrm{p}}[{H_{\mathrm{p}}}^{\ell_i}[u_{\mathrm{p}}]]) \mid i \geq 0\}$ is infinite for any strictly increasing sequence $(\ell_i)_i$. Also, the transformation from G, H to G_{p}, H_{p} preserves the linearity, because: let N be the normal form of $G[H^i[x]]$ where x is fresh, and N_{p} be the $\lambda^{\to}$-term obtained by applying this transformation to N; then $G_{\mathrm{p}}[{H_{\mathrm{p}}}^i[x]] \longrightarrow^* N_{\mathrm{p}}$, and by the infiniteness of $\{\mathcal{T}(G_{\mathrm{p}}[{H_{\mathrm{p}}}^i[u_{\mathrm{p}}]]) \mid i \geq 0\}$, N_{p} must contain x, which implies N_{p} is a linear normal form.

Now we decrease the order for induction. We write Σ_l for the `br`-alphabet version of Σ_{p}, i.e., $\Sigma_l := \{a_i \mapsto 0 \mid (a \mapsto k) \in \Sigma,\ 1 \leq i \leq k\} \cup \{\mathtt{e} \mapsto 0, \mathtt{br} \mapsto 2\}$. By applying Lemma 35 to G_{p}, H_{p}, and u_{p}, there exist order-$(n-1)$ linear $\lambda^{\to}$-contexts G_l, H_l, an order-$(n-1)$ $\lambda^{\to}$-term u_l of alphabet Σ_l, and some constant numbers c', $d' \geq 1$ such that, for any $i \geq 0$, $\mathcal{T}(G_l[{H_l}^i[u_l]])$ is either an `e`-free `br`-tree or `e`, and

$$\mathbf{word}(\mathcal{T}(G_{\mathrm{p}}[{H_{\mathrm{p}}}^{c'i+d'}[u_{\mathrm{p}}]])) = \begin{cases} \varepsilon & (\mathcal{T}(G_l[{H_l}^i[u_l]]) = \mathtt{e}) \\ \mathbf{leaves}(\mathcal{T}(G_l[{H_l}^i[u_l]])) & (\mathcal{T}(G_l[{H_l}^i[u_l]]) \neq \mathtt{e}). \end{cases} \tag{6}$$

By (3), (5), and (6), $\{\mathcal{T}(G_l[{H_l}^i[u_l]]) \mid i \geq 1\}$ is also infinite.

By the induction hypothesis, there exist j_1 and k_1 such that

$$\mathcal{T}(G_l[{H_l}^{j_1}[u_l]]) <_{\mathsf{o}}^{\Sigma_l} \mathcal{T}(G_l[{H_l}^{j_1+k_1}[u_l]]) <_{\mathsf{o}}^{\Sigma_l} \mathcal{T}(G_l[{H_l}^{j_1+2k_1}[u_l]]) <_{\mathsf{o}}^{\Sigma_l} \cdots .$$

By Condition 38-2 and (6), we have

$$\mathcal{T}(G_{\mathrm{p}}[{H_{\mathrm{p}}}^{c'j_1+d'}[u_{\mathrm{p}}]]) <_{\mathsf{o}}^{\Sigma_{\mathrm{p}}} \mathcal{T}(G_{\mathrm{p}}[{H_{\mathrm{p}}}^{c'(j_1+k_1)+d'}[u_{\mathrm{p}}]]) <_{\mathsf{o}}^{\Sigma_{\mathrm{p}}} \mathcal{T}(G_{\mathrm{p}}[{H_{\mathrm{p}}}^{c'(j_1+2k_1)+d'}[u_{\mathrm{p}}]]) <_{\mathsf{o}}^{\Sigma_{\mathrm{p}}} \cdots .$$

Let $j'_1 = c'j_1 + d'$ and $k'_1 = c'k_1$; then

$$\mathcal{T}(G_{\mathrm{p}}[{H_{\mathrm{p}}}^{j'_1}[u_{\mathrm{p}}]]) <_{\mathsf{o}}^{\Sigma_{\mathrm{p}}} \mathcal{T}(G_{\mathrm{p}}[{H_{\mathrm{p}}}^{j'_1+k'_1}[u_{\mathrm{p}}]]) <_{\mathsf{o}}^{\Sigma_{\mathrm{p}}} \mathcal{T}(G_{\mathrm{p}}[{H_{\mathrm{p}}}^{j'_1+2k'_1}[u_{\mathrm{p}}]]) <_{\mathsf{o}}^{\Sigma_{\mathrm{p}}} \cdots . \tag{7}$$

Now, by Condition 37, there exist $j_2 \geq 0$ and $k_2 > 0$ such that

$$H^{j_2}[u] \leq_{\kappa}^{\Sigma_{\mathrm{d}}} H^{j_2+k_2}[u] \leq_{\kappa}^{\Sigma_{\mathrm{d}}} H^{j_2+2k_2}[u] \leq_{\kappa}^{\Sigma_{\mathrm{d}}} \cdots . \tag{8}$$

Let j_3 be the least j_3 such that $j_3 = j'_1 + i_3\, k'_1 = j_2 + m_0$ for some i_3 and m_0, and k_3 be the least common multiple of k'_1 and k_2, whence $k_3 = m_1 k'_1 = m_2 k_2$ for some m_1 and m_2. Then since the mapping $s \mapsto \mathcal{T}(G[H^{m_0}[s]])$ is monotonic, from (8) we have:

$$\mathcal{T}(G[H^{j_3}[u]]) \leq_{\mathsf{o}}^{\Sigma_{\mathrm{d}}} \mathcal{T}(G[H^{j_3+k_2}[u]]) \leq_{\mathsf{o}}^{\Sigma_{\mathrm{d}}} \mathcal{T}(G[H^{j_3+2k_2}[u]]) \leq_{\mathsf{o}}^{\Sigma_{\mathrm{d}}} \cdots .$$

Since $j_3 + ik_3 = j_3 + (im_2)k_2$, we have

$$\mathcal{T}(G[H^{j_3}[u]]) \leq_{\circ}^{\Sigma_{\rm d}} \mathcal{T}(G[H^{j_3+k_3}[u]]) \leq_{\circ}^{\Sigma_{\rm d}} \mathcal{T}(G[H^{j_3+2k_3}[u]]) \leq_{\circ}^{\Sigma_{\rm d}} \cdots . \tag{9}$$

Also, since $j_3 + ik_3 = j_1' + (i_3 + im_1)k_1'$, from (7) we have

$$\mathcal{T}(G_{\rm p}[{H_{\rm p}}^{j_3}[u_{\rm p}]]) <_{\circ}^{\Sigma_{\rm p}} \mathcal{T}(G_{\rm p}[{H_{\rm p}}^{j_3+k_3}[u_{\rm p}]]) <_{\circ}^{\Sigma_{\rm p}} \mathcal{T}(G_{\rm p}[{H_{\rm p}}^{j_3+2k_3}[u_{\rm p}]]) <_{\circ}^{\Sigma_{\rm p}} \cdots . \tag{10}$$

Thus, by Condition 38-4, (5), (9), and (10), we obtain

$$\mathcal{T}(G[H^{j_3}[u]]) <_{\circ}^{\Sigma_{\rm d}} \mathcal{T}(G[H^{j_3+k_3}[u]]) <_{\circ}^{\Sigma_{\rm d}} \mathcal{T}(G[H^{j_3+2k_3}[u]]) <_{\circ}^{\Sigma_{\rm d}} \cdots . \tag{11}$$

By Condition 38-3, (2), and (11), we have

$$\mathcal{T}(C[D^{j_0+j_3k_0}[t]]) <_{\circ}^{\Sigma} \mathcal{T}(C[D^{j_0+(j_3+k_3)k_0}[t]]) <_{\circ}^{\Sigma} \mathcal{T}(C[D^{j_0+(j_3+2k_3)k_0}[t]]) <_{\circ}^{\Sigma} \cdots . \tag{12}$$

We define $j = j_0 + k_0 j_3$ and $k = k_0 k_3$; then we obtain

$$\mathcal{T}(C[D^{j}[t]]) <_{\circ}^{\Sigma} \mathcal{T}(C[D^{j+k}[t]]) <_{\circ}^{\Sigma} \mathcal{T}(C[D^{j+2k}[t]]) <_{\circ}^{\Sigma} \cdots .$$

Finally, we show that $|\mathcal{T}(C[D^{j+ik}[t]])| \leq \mathbf{exp}_n(ci+d)$ for some c and d. Since C and D are single-hole contexts, $|C[D^{j+ik}[t]]| = |C|+(j+ik)|D|+|t|$. Let $c = k|D|$ and $d = |C|+j|D|+|t|$; then $|C[D^{j+ik}[t]]| = ci + d$. It is well-known that, for an order-n $\lambda^{\to}$-term s, we have $|\mathcal{T}(s)| \leq \mathbf{exp}_n(|s|)$ (see, e.g., [11, Lemma 3]). Thus, we have $|\mathcal{T}(C[D^{j+ik}[t]])| \leq \mathbf{exp}_n(ci+d)$. ◻

B.3 Proof of the pumping lemma on $\preceq^{\#}$

Finally, Theorem 12 follows from Theorems 14, 39 and the next three lemmas, which just say that $(\preceq_{\kappa}^{\#,\Sigma})_{\Sigma,\kappa}$ satisfies Conditions 38-2, 38-3, 38-4 (Condition 38-1 is clear).

▶ **Lemma 41.** *Let $\Sigma_{\rm ur}$, $\Sigma_{\rm br}$, and $\Sigma_{\rm wd}$ be an unranked alphabet, the* `br`*-alphabet of $\Sigma_{\rm ur}$, and the word alphabet of $\Sigma_{\rm ur}$, respectively. For* `e`*-free* `br`*-trees π and π' of $\Sigma_{\rm br}$, if $\pi \prec^{\#,\Sigma_{\rm br}} \pi'$ then $\mathbf{utree}(\mathbf{leaves}(\pi)) \prec^{\#,\Sigma_{\rm wd}} \mathbf{utree}(\mathbf{leaves}(\pi'))$.*

Proof. We can show that $\#_a(\pi) = \#_a(\mathbf{utree}(\mathbf{leaves}(\pi)))$ for any $a \in \Sigma_{\rm ur}$ and any `e`-free `br`-tree π. Hence we obtain $\mathbf{utree}(\mathbf{leaves}(\pi)) \preceq^{\#,\Sigma_{\rm wd}} \mathbf{utree}(\mathbf{leaves}(\pi'))$. We prove the remaining strictness by contradiction. Suppose that $\mathbf{utree}(\mathbf{leaves}(\pi)) \approx^{\#,\Sigma_{\rm wd}} \mathbf{utree}(\mathbf{leaves}(\pi'))$; then $\#_a(\pi) = \#_a(\pi')$ for any $a \in \Sigma_{\rm ur}$. Since we can show that $\sum_{a\in\Sigma_{\rm ur}} \#_a(\pi) = \#_{\tt br}(\pi) + 1$ for any `e`-free `br`-tree π, we also have $\#_{\tt br}(\pi') = \#_{\tt br}(\pi)$. Thus we have $\pi \approx^{\#,\Sigma_{\rm br}} \pi'$, which contradicts the assumption. ◻

▶ **Lemma 42.** *For trees π and π', if $\pi \prec^{\#,\Sigma_{\rm d}} \pi'$ then $\mathbf{rmdir}(\pi) \prec^{\#,\Sigma} \mathbf{rmdir}(\pi')$.*

Proof. We can show that $\#_a(\mathbf{rmdir}(\pi)) = \sum_{0\leq i\leq\Sigma(a)} \#_{a^{\langle i\rangle}}(\pi)$ for any direction annotated tree π. Hence we have $\mathbf{rmdir}(\pi) \preceq^{\#,\Sigma} \mathbf{rmdir}(\pi')$. By the assumption, we have $\#_{a^{\langle j\rangle}}(\pi) < \#_{a^{\langle j\rangle}}(\pi')$ for some $a \in \Sigma$ and $0 \leq j \leq \Sigma(a)$, and also $\#_{a^{\langle i\rangle}}(\pi) \leq \#_{a^{\langle i\rangle}}(\pi')$ for any $0 \leq i \leq \Sigma(a)$. Then we have

$$\#_a(\mathbf{rmdir}(\pi)) = \sum_{0\leq i\leq\Sigma(a)} \#_{a^{\langle i\rangle}}(\pi) < \sum_{0\leq i\leq\Sigma(a)} \#_{a^{\langle i\rangle}}(\pi') = \#_a(\mathbf{rmdir}(\pi')).$$

Therefore we have $\mathbf{rmdir}(\pi) \prec^{\#,\Sigma} \mathbf{rmdir}(\pi')$. ◻

▶ **Lemma 43.** *For well-annotated trees π and π', if $\pi \approx^{\#,\Sigma_{\mathrm{d}}} \pi'$ then $\mathbf{path}(\pi) \approx^{\#,\Sigma_{\mathrm{p}}} \mathbf{path}(\pi')$.*

Proof. Let $a \in \Sigma$ and $1 \le i \le \Sigma(a)$, and it suffices to show that $\#_{a^{\langle i \rangle}}(\pi) = \#_{a_i}(\mathbf{path}(\pi))$ for any well-annotated tree π, by induction on π. Let $\pi = b^{\langle j \rangle}\pi_1 \cdots \pi_k$. If $j = 0$, then $\#_{a^{\langle i \rangle}}(\pi) = 0 = \#_{a_i}(\mathbf{path}(\pi))$. If $j > 0$, then let $\delta := 1$ if $b^{\langle j \rangle} = a^{\langle i \rangle}$ and $\delta := 0$ otherwise; then $\#_{a^{\langle i \rangle}}(\pi) = \delta + \#_{a^{\langle i \rangle}}(\pi_j) = \delta + \#_{a_i}(\mathbf{path}(\pi_j)) = \#_{a_i}(\mathbf{path}(\pi))$. ◻

C Reduction from Theorem 14 to Lemma 15

We show Theorem 14 by Lemma 15, in three steps. First, we focus on each constant $a \in \Sigma$:

▶ **Lemma 44.** *For any alphabet Σ and any type κ of order up to 3, if $\preceq_\kappa^{\#,\Sigma,a}$ on Λ_κ^Σ is a wqo for each $a \in \Sigma$, then $\preceq_\kappa^{\#,\Sigma}$ on Λ_κ^Σ is a wqo.*

Proof. Since $\preceq_\kappa^{\#,\Sigma}$ and $\preceq_\kappa^{\#,\Sigma,a}$ are point-wise quasi-orderings for any function type κ, we have $t_1 \preceq^{\#,\Sigma} t_2$ iff for any $a \in \Sigma$, $t_1 \preceq^{\#,\Sigma,a} t_2$. Hence by Dickson's Theorem, if $\preceq^{\#,\Sigma,a}$ is a wqo for any $a \in \Sigma$, so is $\preceq^{\#,\Sigma}$. ◻

Next, it suffices to consider the subset of terms that contain no constant:

▶ **Lemma 45.** *For any alphabet Σ and any $a \in \Sigma$, if the sub-quasi-ordering $\preceq_\kappa^{\#,\Sigma,a}$ on $\Lambda_\kappa^\emptyset$ ($\subseteq \Lambda_\kappa^\Sigma$) is a wqo for any type κ of order up to 3, then $\preceq_\kappa^{\#,\Sigma,a}$ on Λ_κ^Σ is a wqo for any type κ of order up to 3.*

Proof. Let $\Sigma = \{a_1 \mapsto r_1, \ldots, a_k \mapsto r_k\}$. For given sequence $(u_i)_{i\in\mathbb{N}}$ of terms in Λ_κ^Σ, consider the sequence $(\lambda a_1, \ldots, a_k.u_i)_i$ in $\Lambda_{\Sigma\to\kappa}^\emptyset$. By the assumption, there exist $i < j$ such that $\lambda a_1, \ldots, a_k.u_i \preceq_{\Sigma\to\kappa}^{\#,\Sigma,a} \lambda a_1, \ldots, a_k.u_j$. Then we have

$$u_i =_\beta (\lambda a_1, \ldots, a_k.u_i)\,\overline{a_1} \cdots \overline{a_k} \preceq_\kappa^{\#,\Sigma,a} (\lambda a_1, \ldots, a_k.u_j)\,\overline{a_1} \cdots \overline{a_k} =_\beta u_j,$$

where for a constant a we write $\overline{a}$ for $\lambda x_1, \ldots, x_{\Sigma(a)}.a\,x_1 \cdots x_{\Sigma(a)}$. ◻

Finally we consider open ground terms rather than closed terms:

▶ **Lemma 46.** *For any alphabet Σ and any $a \in \Sigma$, if $\preceq_{\Gamma,\mathsf{o}}^{\#,\Sigma,a}$ on $\Lambda_{\Gamma,\mathsf{o}}^\emptyset$ is a wqo for any order-2 type environment Γ, then $\preceq_\kappa^{\#,\Sigma,a}$ on $\Lambda_\kappa^\emptyset$ is a wqo for any type κ of order up to 3.*

Proof. Let $\kappa = \kappa_1 \to \cdots \to \kappa_k \to \mathsf{o}$ be a type of order up to 3, and $\Gamma := z_1 : \kappa_1, \ldots, z_k : \kappa_k$. Then, for any term $t \in \Lambda_\kappa^\emptyset$, we have $t\,z_1 \cdots z_k \in \Lambda_{\Gamma,\mathsf{o}}^\emptyset$, and

$$t_1 \preceq_\kappa^{\#,\Sigma,a} t_2 \quad \text{if and only if} \quad t_1\,z_1 \cdots z_k \preceq_{\Gamma,\mathsf{o}}^{\#,\Sigma,a} t_2\,z_1 \cdots z_k$$

since $\preceq_\kappa^{\#,\Sigma,a}$ is closed under η-equality. Hence it follows from the assumption that $\preceq_\kappa^{\#,\Sigma,a}$ on $\Lambda_\kappa^\emptyset$ is a wqo. ◻

D Proofs for Section 4.2

The main goal of this section is to show (Lemma 17 and) Lemma 18; the former is used to show the latter. Below we write

$$\Gamma; y_1 : \mathsf{o}, \ldots, y_n : \mathsf{o} \vdash t \triangleright (\ldots) =^{[\![]\!]} (\overrightarrow{v}; \overrightarrow{w}; e)$$

if there exists $(\overrightarrow{v'}; \overrightarrow{w'}; e')$ such that

$$\Gamma; y_1 : \mathsf{o}, \ldots, y_n : \mathsf{o} \vdash t \triangleright (\overrightarrow{v'}; \overrightarrow{w'}; e') \quad \text{and} \quad (\overrightarrow{[\![v']\!]}; \overrightarrow{[\![w']\!]}; [\![e']\!]) = (\overrightarrow{[\![v]\!]}; \overrightarrow{[\![w]\!]}; [\![e]\!]).$$

D.1 Preservation of meaning of ground terms

Here we show Lemma 17.

▶ **Lemma 47** (weakening and exchange on internal variables). *If t contains no order-2 variable and*

$$\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t \triangleright (v_1, \dots, v_n; w_1, \dots, w_r; e)$$

then we have

$$\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o}, y_{n+1} : \mathsf{o} \vdash t \triangleright (v_1, \dots, v_n, 0; w_1, \dots, w_r; e)$$

for any y_{n+1}, and

$$\Gamma; y_1 : \mathsf{o}, \dots, y_i : \mathsf{o}, y_{i+2} : \mathsf{o}, y_{i+1} : \mathsf{o}, y_{i+3} : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t$$
$$\triangleright (v_1, \dots, v_i, v_{i+2}, v_{i+1}, v_{i+3} \dots, v_n; w_1, \dots, w_r; e)$$

for any i.

Proof. The proof is given by straightforward induction on t; note that (App1) never happens. ◻

▶ **Lemma 48** (strengthening on internal variables). *If t contains no order-2 variable and*

$$\Gamma; y : \mathsf{o}, y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t \triangleright (v, v_1, \dots, v_n; w_1, \dots, w_k; e) \qquad y \notin \mathbf{FV}(t),$$

then $v =^{[\![]\!]} 0$ and we have

$$\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t \triangleright (v_1, \dots, v_n; w_1, \dots, w_k; e).$$

Proof. The proof is given by straightforward induction on t; note that (App1) never happens. ◻

▶ **Lemma 49** (substitution lemma on internal variables). *If*

$$\Sigma; y' : \mathsf{o}, y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t \triangleright (v', v_1, \dots, v_n; w_1, \dots, w_r; e)$$
$$\Sigma; \vdash t' \triangleright (;; e')$$

then we have

$$\Sigma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t[t'/y'] \triangleright (\dots) =^{[\![]\!]} (v_1, \dots, v_n; w_1, \dots, w_r; e + v'e')$$

Proof. The proof is straightforward as usual substitution lemmas, and proceeds by induction on t and case analysis on the last rule used for $\Sigma; y' : \mathsf{o}, y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t \triangleright (v', v_1, \dots, v_n; w_1, \dots, w_r; e)$.

- Case of (IVar) where $t = y'$ follows from Lemma 47.
- Case of (IVar) where $t = y_i$ and case of (Var): In this case, $y' \notin \mathbf{FV}(t)$, and the goal follows from Lemma 48.

- Case of (App0): In this case, we have:

$$
\begin{aligned}
&t = t_1\, t_2\\
&\Sigma; y' : \mathtt{o}, y_1 : \mathtt{o}, \dots, y_n : \mathtt{o} \vdash t_1 \rhd (v'_1, v_{1,1}, \dots, v_{1,n}; w_0, w_1, \dots, w_r; e_1)\\
&\Sigma; y' : \mathtt{o}, y_1 : \mathtt{o}, \dots, y_n : \mathtt{o} \vdash t_2 \rhd (v'_2, v_{2,1}, \dots, v_{2,n}; ; e_2)\\
&v' = v'_1 + w_0 v'_2 \qquad v_j = v_{1,j} + w_0 v_{2,j} \quad (j = 1, \dots, n)\\
&e = e_1 + w_0 e_2
\end{aligned}
$$

By induction hypothesis, we have:

$$
\begin{aligned}
&\Sigma; y_1 : \mathtt{o}, \dots, y_n : \mathtt{o} \vdash t_1[t'/y'] \rhd (\dots) =^{[\![]\!]} (v_{1,1}, \dots, v_{1,n}; w_0, w_1, \dots, w_r; e_1 + v'_1 e')\\
&\Sigma; y_1 : \mathtt{o}, \dots, y_n : \mathtt{o} \vdash t_2[t'/y'] \rhd (\dots) =^{[\![]\!]} (v_{2,1}, \dots, v_{2,n}; ; e_2 + v'_2 e')
\end{aligned}
$$

and by (App0), we have

$$
\begin{aligned}
&\Sigma; y_1 : \mathtt{o}, \dots, y_n : \mathtt{o} \vdash t_1[t'/y']\, t_2[t'/y'] \rhd (\dots) =^{[\![]\!]}\\
&(v_{1,1} + w_0 v_{2,1}, \dots, v_{1,n} + w_0 v_{2,n}; w_1, \dots, w_r; (e_1 + v'_1 e') + w_0(e_2 + v'_2 e'))\\
&= (v_1, \dots, v_n; w_1, \dots, w_r; e + v' e')
\end{aligned}
$$

as required.

- Case of (App1): This case does not happen since t contains no order-2 variable.
- Case of (Lam): In this case, we have:

$$
\begin{aligned}
&t = \lambda y_{n+1}.s\\
&\Sigma; y' : \mathtt{o}, y_1 : \mathtt{o}, \dots, y_n : \mathtt{o}, y_{n+1} : \mathtt{o} \vdash s \rhd (v', v_1, \dots, v_n, w_1; w_2, \dots, w_r; e)
\end{aligned}
$$

By induction hypothesis, we have

$$
m\Sigma; y_1 : \mathtt{o}, \dots, y_n : \mathtt{o}, y_{n+1} : \mathtt{o} \vdash s[t'/y'] \rhd (\dots) =^{[\![]\!]} (v_1, \dots, v_n, w_1; w_2, \dots, w_r; e + v' e')
$$

and by (Lam),

$$
\Sigma; y_1 : \mathtt{o}, \dots, y_n : \mathtt{o} \vdash \lambda y_{n+1}.s[t'/y'] \rhd (\dots) =^{[\![]\!]} (v_1, \dots, v_n; w_1, w_2, \dots, w_r; e + v' e').
$$

as required. ☐

We write $\longrightarrow_{\mathrm{h}}$ for head reduction, i.e., $\longrightarrow_{\mathrm{h}}$ is defined by the following rules:

$$
\frac{}{(\lambda y.t)t'\, t_1 \cdots t_k \longrightarrow_{\mathrm{h}} t[t'/y]\, t_1 \cdots t_k} \qquad \frac{t_i \longrightarrow_{\mathrm{h}} t'_i}{z\, t_1 \cdots t_k \longrightarrow_{\mathrm{h}} z\, t_1 \cdots t'_i \cdots t_k}
$$

▶ **Lemma 50** (subject reduction). *If*

$$
\Sigma; \vdash t \rhd (; ; e) \qquad \textit{and} \qquad t \longrightarrow_{\mathrm{h}} t'
$$

then

$$
\Sigma; \vdash t' \rhd (\dots) =^{[\![]\!]} (; ; e).
$$

Proof. The proof proceeds by induction on order-3 normal form t. Now the head of t is either $a \in \Sigma$ or $\lambda y.t_0$. The former case is clear by induction hypothesis and we consider the latter case. Let $t = (\lambda y.t_0)\, t_1 \cdots t_k$, where $k \geq 1$. Since $t \longrightarrow_{\mathrm{h}} t'$, we have $t' = t_0[t_1/y]\, t_2 \cdots t_k$. By the derivation tree of $\Sigma; \vdash t \triangleright (;;e)$, there exist e_i and w_i such that:

$$\Sigma; y : \mathsf{o} \vdash t_0 \triangleright (w_1; w_2, \dots, w_k; e_0)$$
$$\Sigma; \vdash t_i \triangleright (;;e_i) \qquad (i = 1, \dots, k)$$
$$e = e_0 + w_1 e_1 + \cdots + w_k e_k.$$

By Lemma 49, we have

$$\Sigma; \vdash t_0[t_1/y] \triangleright (\dots) =^{[\![]\!]} (; w_2, \dots, w_k; e_0 + w_1 e_1)$$

and hence

$$\Sigma; \vdash t_0[t_1/y]\, t_2 \cdots t_k \triangleright (\dots) =^{[\![]\!]} (;; e_0 + w_1 e_1 + \cdots + w_k e_k)$$

i.e., we have $\Sigma; \vdash t' \triangleright (\dots) =^{[\![]\!]} (;;e)$ as required. ◻

Proof of Lemma 17. By Lemma 50, we can assume that t is β-normal, i.e., t is a tree consisting of constants in Σ (but recall that Σ is now regarded as an environment). The proof proceeds by induction on tree t. Let t be of the form $a\, t_1 \cdots t_k$, and let $\Sigma; \vdash t_i \triangleright (;;e_i)$ for each $i \leq k$. Then we have:

$$\begin{aligned}
[\![e\theta_{\Sigma}^{a_{\mathrm{fix}}}]\!] &= [\![(d_a + c_{a,1} e_1 + \cdots + c_{a,k} e_k)\theta_{\Sigma}^{a_{\mathrm{fix}}}]\!] \\
&= [\![(d_a\theta_{\Sigma}^{a_{\mathrm{fix}}}) + (e_1\theta_{\Sigma}^{a_{\mathrm{fix}}}) + \cdots + (e_k\theta_{\Sigma}^{a_{\mathrm{fix}}})]\!] \\
&= [\![d_a\theta_{\Sigma}^{a_{\mathrm{fix}}}]\!] + \#_{a_{\mathrm{fix}}}(t_1) + \cdots + \#_{a_{\mathrm{fix}}}(t_k) \\
&= \#_{a_{\mathrm{fix}}}(t).
\end{aligned}$$

◻

D.2 Substitution lemmas on external variables

To show Lemma 18, we also use substitution lemmas for external variables. We give two substitution lemmas, on order-1 variables and on order-2 variables. The following two lemmas are trivial.

▶ **Lemma 51** (weakening and exchange on external variables). *If*

$$z_1 : \kappa_1, \dots, z_k : \kappa_k; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t \triangleright (\overrightarrow{v}; \overrightarrow{w}; e)$$

then we have

$$z : \kappa, z_1 : \kappa_1, \dots, z_k : \kappa_k; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t \triangleright (\overrightarrow{v}; \overrightarrow{w}; e).$$

for any z and κ, and

$$z_1 : \kappa_1, \dots, z_i : \kappa_i, z_{i+2} : \kappa_{i+2}, z_{i+1} : \kappa_{i+1}, z_{i+3} : \kappa_{i+3}, \dots, z_k : \kappa_k; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t \triangleright (\overrightarrow{v}; \overrightarrow{w}; e)$$

for any i.

▶ **Lemma 52** (strengthening on external variables). *If*

$$z : \kappa, z_1 : \kappa_1, \dots, z_k : \kappa_k; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t \triangleright (\overrightarrow{v}; \overrightarrow{w}; e)$$

and $z \notin \mathbf{FV}(t)$, then

$$z_1 : \kappa_1, \dots, z_k : \kappa_k; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t \triangleright (\overrightarrow{v}; \overrightarrow{w}; e).$$

Now we show the substitution lemma on order-1 variables. We define $\Gamma' \leq \Gamma$ if Γ contains $x : \kappa$ whenever Γ' contains $x : \kappa$.

▶ **Lemma 53** (substitution lemma on order-1 variables). *Suppose*

$$f'_1 : \mathsf{o}^{q'_1} \to \mathsf{o}, \ldots, f'_k : \mathsf{o}^{q'_k} \to \mathsf{o}, \Gamma'; y_1 : \mathsf{o}, \ldots, y_n : \mathsf{o} \vdash s \triangleright (v'_1, \ldots, v'_n; w'_1, \ldots, w'_r; e')$$
Γ' *is order-1*
$$\Gamma; y_1 : \mathsf{o}, \ldots, y_n : \mathsf{o} \vdash t_i \triangleright (v_{i,1}, \ldots, v_{i,n}; w_{i,1}, \ldots, w_{i,q'_i}; e_i) \qquad (i \leq k)$$
$$\Gamma = \varphi_1 : \kappa_1, \ldots, \varphi_m : \kappa_m, f_1 : \mathsf{o}^{q_1} \to \mathsf{o}, \ldots, f_\ell : \mathsf{o}^{q_\ell} \to \mathsf{o} \qquad \Gamma' \leq \Gamma.$$

Let

$$\theta := \left[\overrightarrow{w_{i,j}/c_{f'_i,j}}^{i \leq k, j \leq q'_i}, \overrightarrow{e_i/d_{f'_i}}^{i \leq k}\right]$$
$$\theta^{(x_i)_{i \leq k}} := \left[\overrightarrow{w_{i,j}/c_{f'_i,j}}^{i \leq k, j \leq q'_i}, \overrightarrow{x_i/d_{f'_i}}^{i \leq k}, \overrightarrow{0/d_{f'}}^{f' \notin \{f'_i \mid i \leq k\}}\right]$$

Then we have

$$\Gamma; y_1 : \mathsf{o}, \ldots, y_n : \mathsf{o} \vdash s\left[\overrightarrow{t_i/f'_i}^{i \leq k}\right] \triangleright (\ldots) =^{[\![]\!]} \left(\overrightarrow{(v'_j\theta) + (e'\theta^{(v_{i,j})_i})}^{j \leq n}; \overrightarrow{w'_j\theta}^{j \leq r}; e'\theta\right).$$

▶ Remark. We state the above lemma in the form of simultaneous substitution because we cannot repeat the form of substitution in the lemma; observe that $s[t_1/f'_1]$ is not (in general) an order-2 normal form any more. The above lemma is used at two places below (in the proofs of Lemmas 54 and 18).

Proof. The proof proceeds by induction on s and case analysis on the last rule used for $f'_1 : \mathsf{o}^{q'_1} \to \mathsf{o}, \ldots, f'_k : \mathsf{o}^{q'_k} \to \mathsf{o}, \Gamma'; y_1 : \mathsf{o}, \ldots, y_n : \mathsf{o} \vdash s \triangleright (v'_1, \ldots, v'_n; w'_1, \ldots, w'_r; e')$.

- Cases of (IVAR) and (VAR) where $s \notin \{f'_i \mid i \leq k\}$: By using Lemmas 52 for f'_j and 51 for Γ' and Γ, we have

  $$\Gamma; y_1 : \mathsf{o}, \ldots, y_n : \mathsf{o} \vdash s \triangleright (v'_1, \ldots, v'_n; w'_1, \ldots, w'_r; e').$$

  Now v'_j, w'_j, and e' do not contain $c_{f'_i,j}$ nor $d_{f'_i}$; hence $(-)\theta$ does not affect v'_j, w'_j, and e'. Thus it suffices to show $e'\theta^{(v_{i,j})_i} = 0$. In the case of (IVAR), $e' = 0$. In the case of (VAR) where $s = f'$ ($\notin \{f'_i \mid i \leq k\}$), we have $e'\theta^{(v_{i,j})_i} = d_{f'}\theta^{(v_{i,j})_i} = 0$.
- Case of (VAR) where $s = f'_i$ and $i \leq k$: In this case, it suffices to show that $(v_{i,1}, \ldots, v_{i,n}; w_{i,1}, \ldots, w_{i,q'_i}; e_i)$ in the judgment of t_i in the assumption is equal to $\left(\overrightarrow{(v'_j\theta) + (e'\theta^{(v_{i,j})_i})}^{j \leq n}; \overrightarrow{w'_j\theta}^{j \leq r}; e'\theta\right)$ in the goal. This is routine; for example,

  $$(v'_j\theta) + (e'\theta^{(v_{i,j})_i}) = (0\theta) + (d_{f'_i}\theta^{(v_{i,j})_i}) = v_{i,j}.$$

- Case of (APP0): In this case, we have:

  $$s = s_1 s_2$$
  $$f'_1 : \mathsf{o}^{q'_1} \to \mathsf{o}, \ldots, f'_k : \mathsf{o}^{q'_k} \to \mathsf{o}, \Gamma'; y_1 : \mathsf{o}, \ldots, y_n : \mathsf{o} \vdash s_1 \triangleright (v'_{1,1}, \ldots, v'_{1,n}; w'_{1,1}, \ldots, w'_{1,r+1}; e'_1)$$
  $$f'_1 : \mathsf{o}^{q'_1} \to \mathsf{o}, \ldots, f'_k : \mathsf{o}^{q'_k} \to \mathsf{o}, \Gamma'; y_1 : \mathsf{o}, \ldots, y_n : \mathsf{o} \vdash s_2 \triangleright (v'_{2,1}, \ldots, v'_{2,n};; e'_2)$$
  $$v'_j = v'_{1,j} + w'_{1,1} v'_{2,j} \qquad (j \leq n)$$
  $$w'_j = w'_{1,j+1} \qquad (j \leq r)$$
  $$e' = e'_1 + w'_{1,1} e'_2.$$

By induction hypothesis, we have:

$$\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash s_1\left[\overrightarrow{t_i/f'_i}^{i \le k}\right] \rhd (\dots) =^{[\![]\!]} \left(\overrightarrow{(v'_{1,j}\theta) + (e'_1\theta^{(v_{i,j})_i})}^{j \le n}; \overrightarrow{w'_{1,j}\theta}^{j \le r+1}; e'_1\theta\right)$$
$$\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash s_2\left[\overrightarrow{t_i/f'_i}^{i \le k}\right] \rhd (\dots) =^{[\![]\!]} \left(\overrightarrow{(v'_{2,j}\theta) + (e'_2\theta^{(v_{i,j})_i})}^{j \le n}; ; e'_2\theta\right).$$

By applying (App0), we have

$$\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash s_1\left[\overrightarrow{t_i/f'_i}^{i \le k}\right] s_2\left[\overrightarrow{t_i/f'_i}^{i \le k}\right] \rhd (\dots) =^{[\![]\!]}$$
$$\left(\overrightarrow{(v'_{1,j}\theta) + (e'_1\theta^{(v_{i,j})_i}) + (w'_{1,1}\theta)\big((v'_{2,j}\theta) + (e'_2\theta^{(v_{i,j})_i})\big)}^{j \le n}; \overrightarrow{w'_{1,j}\theta}^{2 \le j \le r+1}; (e'_1\theta) + (w'_{1,1}\theta)(e'_2\theta)\right).$$

Then it suffices to show that the above tuple in the right hand side is equal (with respect to $=^{[\![]\!]}$) to $\left(\overrightarrow{(v'_j\theta) + (e'\theta^{(v_{i,j})_i})}^{j \le n}; \overrightarrow{w'_j\theta}^{j \le r}; e'\theta\right)$. It is clear on $w'_j\theta$ and $e'\theta$. On the first component, we have

$$\begin{aligned}
&(v'_j\theta) + (e'\theta^{(v_{i,j})_i}) \\
=^{[\![]\!]}\ &(v'_{1,j}\theta) + (w'_{1,1}\theta)(v'_{2,j}\theta) + (e'_1\theta^{(v_{i,j})_i}) + (w'_{1,1}\theta^{(v_{i,j})_i})(e'_2\theta^{(v_{i,j})_i})
\end{aligned}$$

and

$$\begin{aligned}
&(v'_{1,j}\theta) + (e'_1\theta^{(v_{i,j})_i}) + (w'_{1,1}\theta)\big((v'_{2,j}\theta) + (e'_2\theta^{(v_{i,j})_i})\big) \\
=^{[\![]\!]}\ &(v'_{1,j}\theta) + (e'_1\theta^{(v_{i,j})_i}) + (w'_{1,1}\theta)(v'_{2,j}\theta) + (w'_{1,1}\theta)(e'_2\theta^{(v_{i,j})_i}).
\end{aligned}$$

Hence it remains to show $w'_{1,1}\theta^{(v_{i,j})_i} =^{[\![]\!]} w'_{1,1}\theta$. This immediately follows from the fact that, if $\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash s \rhd (v'_1, \dots, v'_n; w'_1, \dots, w'_r; e')$ and Γ is order-1, then v'_j and w'_j do not contain any variable of the form d_{-}, which can be shown easily by induction on s.

- Case of (App1): This case does not happen since s contains no order-2 variable.
- Case of (Lam): In this case we have:

$$s = \lambda y_{n+1}.s'$$
$$f'_1 : \mathsf{o}^{q'_1} \to \mathsf{o}, \dots, f'_k : \mathsf{o}^{q'_k} \to \mathsf{o}, \Gamma'; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o}, y_{n+1} : \mathsf{o} \vdash s' \rhd (v'_1, \dots, v'_n, w'_1; w'_2, \dots, w'_r; e')$$

By Lemma 47 we have

$$\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o}, y_{n+1} : \mathsf{o} \vdash t_i \rhd (v_{i,1}, \dots, v_{i,n}, 0; w_{i,1}, \dots, w_{i,q'_i}; e_i) \qquad (i \le k)$$

and by induction hypothesis, we have

$$\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o}, y_{n+1} : \mathsf{o} \vdash s'\left[\overrightarrow{t_i/f'_i}^{i \le k}\right] \rhd (\dots) =^{[\![]\!]}$$
$$\left(\overrightarrow{(v'_j\theta) + (e'\theta^{(v_{i,j})_i})}^{j \le n}, (w'_1\theta) + (e'\theta^{(0)_i}); \overrightarrow{w'_j\theta}^{2 \le j \le r}; e'\theta\right).$$

Now the goal is

$$\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash \lambda y_{n+1}.s'\left[\overrightarrow{t_i/f'_i}^{i \le k}\right] \rhd (\dots) =^{[\![]\!]} \left(\overrightarrow{(v'_j\theta) + (e'\theta^{(v_{i,j})_i})}^{j \le n}; \overrightarrow{w'_j\theta}^{j \le r}; e'\theta\right)$$

which needs the following:

$$\Gamma; y_1 : \mathsf{o}, \ldots, y_n : \mathsf{o}, y_{n+1} : \mathsf{o} \vdash s'\Big[\overrightarrow{t_i/f'_i}^{i\le k}\Big] \rhd (\ldots) =^{[\![]\!]} \Big(\overrightarrow{(v'_j\theta) + (e'\theta^{(v_{i,j})_i})}^{j\le n}, w'_1\theta; \overrightarrow{w'_j\theta}^{2\le j\le r}; e'\theta\Big).$$

The remaining gap, $e'\theta^{(0)_i} =^{[\![]\!]} 0$, follows from the fact that, if $\Gamma; y_1 : \mathsf{o}, \ldots, y_n : \mathsf{o} \vdash s \rhd (v'_1, \ldots, v'_n; w'_1, \ldots, w'_r; e')$ and Γ is order-1, then $[\![e'\theta^{(0)_i}]\!] = 0$, which can be shown easily by induction on s. ☐

Next we show the substitution lemma on order-2 variables. We first define some special substitution for order-3 normal forms, which we use in the substitution lemma.

We call Γ *order-1* if Γ contains only variables of order up to 1. Also we call an order-3 normal form t an *order-2 normal form* if t contains no order-2 variable; we use the meta-variable s for order-2 normal forms in the rest of this section. For an environment $\Gamma = x_1 : \kappa_1, \ldots, x_n : \kappa_n$, we define:

$$\Gamma|_{>\ell} := x_{\ell+1} : \kappa_{\ell+1}, \ldots, x_n : \kappa_n \qquad \Gamma|_{\le\ell} := x_1 : \kappa_1, \ldots, x_\ell : \kappa_\ell.$$

For an environment $\Gamma = x_1 : \kappa_1, \ldots, x_n : \kappa_n$ and terms $u, u_1, \ldots, u_n$, we write $u[(u_i)_i/\Gamma]$ for the (usual) simultaneous substitution $u[u_1/x_1, \ldots, u_n/x_n]$.

For t and $\Gamma \vdash s : \mathsf{o}^r \to \mathsf{o}$ where Γ is order-1, we define *double substitution* $t[\lambda\Gamma.s/\!\!/z]$ by induction on t as follows, where the type of z must be $\Gamma \to \mathsf{o}^r \to \mathsf{o}$ and we write s' for $\lambda\Gamma.s$:

$$\begin{aligned}
f[s'/\!\!/z] &:= s' \text{ (if } f = z) \text{ or } f \text{ (if } f \ne z)\\
(t_1\, t_2)[s'/\!\!/z] &:= t_1[s'/\!\!/z]\, t_2[s'/\!\!/z]\\
(\varphi\, t_1 \cdots t_k)[s'/\!\!/z] &:= \begin{cases} \big(\lambda\Gamma|_{>\ell}.s\big[(t_i[s'/\!\!/z])_{i\le\ell}/\Gamma|_{\le\ell}\big]\big)\, (t_{\ell+1}[s'/\!\!/z]) \cdots (t_k[s'/\!\!/z]) & \\ \qquad\qquad (\varphi = z, \ell := \min\{|\Gamma|, k\}) & \\ \varphi(t_1[s'/\!\!/z]) \cdots (t_k[s'/\!\!/z]) \qquad (\varphi \ne z) & \end{cases}\\
(\lambda y.t)[s'/\!\!/z] &:= \lambda y.(t[s'/\!\!/z]) \text{ (we assume } y \ne z \text{ and } y \notin \mathbf{FV}(s') \text{ by } \alpha\text{-renaming)}
\end{aligned}$$

It is clear that $t[s'/\!\!/z] =_\beta t[s'/z]$. Also it can be easily shown that $t[s'/\!\!/z]$ is an order-3 normal form, by induction on t. For this note that: (i) it is easily shown that, if t and t' are order-3 normal forms and f is a variable of order up to 1, then $t[t'/f]$ is an order-3 normal form, by induction on t; (ii) in the case of $(\varphi\, t_1 \cdots t_k)[s'/\!\!/z]$ and when $\varphi = z$ and $|\Gamma| > k$, the environment $\Gamma|_{>\ell}$ ($= \Gamma|_{>k}$) consists of only the ground type o, because order-3 normal form $\varphi\, t_1 \cdots t_k$ has a type of the form $\mathsf{o}^{r'} \to \mathsf{o}$, which is now $\Gamma|_{>\ell} \to \mathsf{o}^r \to \mathsf{o}$; hence $\lambda\Gamma|_{>\ell}.s\big[(t_i[s'/\!\!/z])_{i\le\ell}/\Gamma|_{\le\ell}\big]$ is an order-3 normal form.

▶ **Lemma 54** (substitution lemma on order-2 variables). *Suppose*

$$\begin{aligned}
&\varphi : \kappa, \Gamma; y_1 : \mathsf{o}, \ldots, y_n : \mathsf{o} \vdash t \rhd (v_1, \ldots, v_n; w_1, \ldots, w_r; e)\\
&\Gamma = \varphi_1 : \kappa_1, \ldots, \varphi_m : \kappa_m, f_1 : \mathsf{o}^{q_1} \to \mathsf{o}, \ldots, f_\ell : \mathsf{o}^{q_\ell} \to \mathsf{o}\\
&\kappa = (\mathsf{o}^{q'_1} \to \mathsf{o}) \to \ldots \to (\mathsf{o}^{q'_k} \to \mathsf{o}) \to (\mathsf{o}^{q'} \to \mathsf{o}) \qquad q'_k > 0\\
&\Gamma_0, \Gamma'; \vdash s \rhd (; w'_1, \ldots, w'_{q'}; e')\\
&\Gamma' = f'_1 : \mathsf{o}^{q'_1} \to \mathsf{o}, \ldots, f'_k : \mathsf{o}^{q'_k} \to \mathsf{o} \qquad \Gamma_0 \textit{ is order-1} \qquad \Gamma_0 \le \Gamma.
\end{aligned}$$

Then we have

$$\Gamma; y_1 : \mathsf{o}, \ldots, y_n : \mathsf{o} \vdash t[\lambda\Gamma'.s/\!\!/\varphi] \rhd (\ldots) =^{[\![]\!]} (\vec{v}\theta; \vec{w}\theta; e\theta)$$

where

$$\theta := \left[\overrightarrow{\lambda\Gamma'^{\natural}.w'_j/g_{\varphi,j}}^{j\le q'},\quad \lambda\Gamma'^{\natural}.e'\Big/h_\varphi,\quad \lambda\Gamma'^{\natural}.e'\left[\overrightarrow{0/d_f}^{f\in dom(\Gamma_0)}\right]\Big/\hat{h}_\varphi\right]$$

and recall $\Gamma'^{\natural} := \left(\overrightarrow{\overrightarrow{c_{f'_i,j} : \mathsf{o}}^{j\le q'_i}, d_{f'_i} : \mathsf{o}}^{i\le k}\right)$.

Proof. The proof proceeds by induction on t and case analysis on the last rule used for $\varphi : \kappa, \Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t \rhd (v_1, \dots, v_n; w_1, \dots, w_r; e)$. (All the cases except for (App1) are in fact straightforward.)

- Cases of (IVar) and (Var): By using Lemma 52 for φ, we have

  $$\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t \rhd (v_1, \dots, v_n; w_1, \dots, w_r; e).$$

  Now v_j, w_j, and e do not contain $g_{\varphi,j}$, h_φ, nor $\hat{h}_\varphi$; hence θ does not affect v_j, w_j, and e. Thus we have the required judgment.
- Case of (App0): In this case, we have:

  $$\begin{aligned}
  &t = t_1 t_2\\
  &\varphi : \kappa, \Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t_1 \rhd (v_{1,1}, \dots, v_{1,n}; w_{1,1}, \dots, w_{1,r+1}; e_1)\\
  &\varphi : \kappa, \Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t_2 \rhd (v_{2,1}, \dots, v_{2,n};; e_2)\\
  &v_j = v_{1,j} + w_{1,1}v_{2,j} \qquad (j \le n)\\
  &w_j = w_{1,j+1} \qquad (j \le r)\\
  &e = e_1 + w_{1,1}e_2.
  \end{aligned}$$

  By induction hypothesis, we have

  $$\begin{aligned}
  &\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t_1[\lambda\Gamma'.s/\!\!/\varphi] \rhd (\dots) =^{[\![]\!]} (\overrightarrow{v_{1,j}}^{j\le n}\theta; \overrightarrow{w_{1,j}}^{j\le r+1}\theta; e_1\theta)\\
  &\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t_2[\lambda\Gamma'.s/\!\!/\varphi] \rhd (\dots) =^{[\![]\!]} (\overrightarrow{v_{2,j}}^{j\le n}\theta;; e_2\theta).
  \end{aligned}$$

  By rule (App0), we have

  $$\begin{aligned}
  &\Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash (t_1\, t_2)[\lambda\Gamma'.s/\!\!/\varphi] \rhd (\dots) =^{[\![]\!]}\\
  &(\overrightarrow{v_{1,j} + w_{1,1}v_{2,j}}^{j\le n}\theta; \overrightarrow{w_{1,j}}^{2\le j\le r+1}\theta; (e_1 + w_{1,1}e_2)\theta) \;=\; (\overrightarrow{v_j}^{j\le n}\theta; \overrightarrow{w_j}^{j\le r}\theta; e\theta).
  \end{aligned}$$

- Case of (App1): The case where the head variable of t is not φ is straightforward, so we concentrate on the case of φ. By the rule (App1), t is of the form $\varphi\, t_1 \cdots t_{k'}$. Since the type of $t_{k'}$ is order-1 and $q'_k > 0$, $k' = k$. Thus, we have:

  $$\begin{aligned}
  &t = \varphi\, t_1 \cdots t_k\\
  &\varphi : \kappa, \Gamma; y_1 : \mathsf{o}, \dots, y_n : \mathsf{o} \vdash t_i \rhd (\vec{v}_i; \vec{w}_i; e_i) \qquad \vec{u}_i = (\vec{w}_i; e_i) \qquad (\text{for each } i \in \{1, \dots, k\})\\
  &\vec{u'}_{i,i'} = (\vec{w}_i; v_{i,i'}) \qquad (\text{for each } i \in \{1, \dots, k\} \text{ and } i' \in \{1, \dots, n\})\\
  &\vec{v} = \hat{h}_\varphi(\vec{u'}_{1,1}, \dots, \vec{u'}_{k,1}), \dots, \hat{h}_\varphi(\vec{u'}_{1,n}, \dots, \vec{u'}_{k,n})\\
  &\vec{w} = g_{\varphi,1}(\vec{u}_1, \dots, \vec{u}_k), \dots, g_{\varphi,q'}(\vec{u}_1, \dots, \vec{u}_k)\\
  &e = h_\varphi(\vec{u}_1, \dots, \vec{u}_k)\\
  &r = q'.
  \end{aligned}$$

Then we have:

$$t[\lambda\Gamma'.s/\!\!/\varphi] \;=\; s\big[(t_i[\lambda\Gamma'.s/\!\!/\varphi])_{i\le k}/\Gamma'|_{\le k}\big] \;=\; s\Big[\overrightarrow{t_i[\lambda\Gamma'.s/\!\!/\varphi]/f'_i}^{\,i\le k}\Big]$$

By induction hypothesis, we have

$$\Gamma; y_1:\mathsf{o},\dots,y_n:\mathsf{o} \vdash t_i[\lambda\Gamma'.s/\!\!/\varphi] \rhd (\dots) =^{[\![]\!]} (\vec{v}_i\theta;\vec{w}_i\theta;e_i\theta) \qquad (i\le k)$$

and by Lemmas 47 and 51, we have

$$\Gamma',\Gamma_0; y_1:\mathsf{o},\dots,y_n:\mathsf{o} \vdash s \rhd (0,\dots,0;w'_1,\dots,w'_{q'};e').$$

Hence by Lemma 53, we have

$$\Gamma; y_1:\mathsf{o},\dots,y_n:\mathsf{o} \vdash s\Big[\overrightarrow{t_i[\lambda\Gamma'.s/\!\!/\varphi]/f'_i}^{\,i\le k}\Big] \rhd (\dots) =^{[\![]\!]} \Big(\overrightarrow{e'\theta_1^{(v_{i,i'}\theta)_{i\le k}}}^{\,i'\le n};\vec{w}'\theta_1;e'\theta_1\Big)$$

where

$$\begin{aligned}
\theta_1 &:= \Big[\overrightarrow{w_{i,j}\theta/c_{f'_i,j}}^{\,i\le k,j\le q'_i},\overrightarrow{e_i\theta/d_{f'_i}}^{\,i\le k}\Big]\\
\theta_1^{(x_i)_{i\le k}} &:= \Big[\overrightarrow{w_{i,j}\theta/c_{f'_i,j}}^{\,i\le k,j\le q'_i},\overrightarrow{x_i/d_{f'_i}}^{\,i\le k},\overrightarrow{0/d_f}^{\,f\notin\{f'_i|i\le k\}}\Big]
\end{aligned}$$

Thus it remains to show that

$$(\vec{v}\theta;\vec{w}\theta;e\theta) = \Big(\overrightarrow{e'\theta_1^{(v_{i,i'}\theta)_{i\le k}}}^{\,i'\le n};\vec{w}'\theta_1;e'\theta_1\Big).$$

For each $i'\le n$,

$$\begin{aligned}
v_{i'}\theta &= \hat{h}_\varphi(\vec{u}'_{1,i'},\dots,\vec{u}'_{k,i'})\theta\\
&= \Big(\lambda\Gamma'^{\natural}.e'\Big[\overrightarrow{0/d_f}^{\,f\in dom(\Gamma_0)}\Big]\Big)\,(\vec{w}_1\theta)\,(v_{1,i'}\theta)\,\cdots\,(\vec{w}_k\theta)\,(v_{k,i'}\theta)\\
&= e'\Big[\overrightarrow{w_{i,j}\theta/c_{f'_i,j}}^{\,i\le k,j\le q'_i},\overrightarrow{v_{i,i'}\theta/d_{f'_i}}^{\,i\le k},\overrightarrow{0/d_f}^{\,f\in dom(\Gamma_0)}\Big]
\end{aligned}$$

On the other hand,

$$\begin{aligned}
e'\theta_1^{(v_{i,i'}\theta)_{i\le k}} &= e'\Big[\overrightarrow{w_{i,j}\theta/c_{f'_i,j}}^{\,i\le k,j\le q'_i},\overrightarrow{v_{i,i'}\theta/d_{f'_i}}^{\,i\le k},\overrightarrow{0/d_f}^{\,f\notin\{f'_i|i\le k\}}\Big]\\
&= e'\Big[\overrightarrow{w_{i,j}\theta/c_{f'_i,j}}^{\,i\le k,j\le q'_i},\overrightarrow{v_{i,i'}\theta/d_{f'_i}}^{\,i\le k},\overrightarrow{0/d_f}^{\,f\in dom(\Gamma_0)}\Big]
\end{aligned}$$

where the last equation holds because we have:

$$(\Gamma_0,\Gamma')^{\natural} \vdash e':\mathsf{o}.$$

Next, for each $i'\le r$ $(=q')$,

$$\begin{aligned}
w_{i'}\theta &= \;(g_{\varphi,i'}(\vec{u}_1,\dots,\vec{u}_k))\,\theta\\
&= (\lambda\Gamma'^{\natural}.w'_{i'})\,(\vec{w}_1\theta)\,(e_1\theta)\,\dots\,(\vec{w}_k\theta)\,(e_k\theta)\\
&= w'_{i'}\Big[\overrightarrow{w_{i,j}\theta/c_{f'_i,j}}^{\,i\le k,j\le q'_i},\overrightarrow{e_i\theta/d_{f'_i}}^{\,i\le k}\Big]\\
&= w'_{i'}\theta_1.
\end{aligned}$$

Finally,

$$\begin{aligned} e\theta &= (h_\varphi(\vec{u}_1,\dots,\vec{u}_k))\theta \\ &= (\lambda\Gamma'^{\natural}.e')\,(\vec{w}_1\theta)\,(e_1\theta)\,\dots\,(\vec{w}_k\theta)\,(e_k\theta) \\ &= e'\Big[\overrightarrow{w_{i,j}\theta/c_{f'_i,j}}^{i\le k, j\le q'_i}, \overrightarrow{e_i\theta/d_{f'_i}}^{i\le k}\Big] \\ &= e'\theta_1. \end{aligned}$$

- Case of (Lam): Since (Lam) changes just the position of the first semicolon, the proof is given clearly by using induction hypothesis.

□

D.3 Ordering reflection

Finally we show Lemma 18:

Proof of Lemma 18. Let $\Sigma, \Gamma_i; \vdash s_i : \mathsf{o}^{r_i} \to \mathsf{o}$ $(i = 1,\dots,m)$ be order-2 normal forms such that $\Gamma_i \to \mathsf{o}^{r_i} \to \mathsf{o} = \kappa_i$ and the last type of Γ_i is not o. Let $s'_i = \lambda\Gamma_i.s_i$; thus $s'_i \in \Lambda^{\Sigma}_{\kappa_i}$. Also, let $\Sigma; \vdash \bar{s}_i : \mathsf{o}^{q_i} \to \mathsf{o}$ $(i = 1,\dots,k)$ be order-2 normal forms (in $\Lambda^{\Sigma}_{\mathsf{o}^{q_i}\to\mathsf{o}}$). It suffices to show that

$$t[s'_i /\!\!/ \varphi_i]_{i\le m}[\bar{s}_i/f_i]_{i\le k} \quad \preceq^{\#,\Sigma,a_{\mathrm{fix}}}_{\emptyset,\mathsf{o}} \quad t'[s'_i /\!\!/ \varphi_i]_{i\le m}[\bar{s}_i/f_i]_{i\le k}. \tag{13}$$

By Lemma 51 we have

$$\Gamma, \Sigma; \vdash t \rhd (;;e) \qquad \Gamma, \Sigma; \vdash t' \rhd (;;e'),$$

and also

$$\Sigma \le (\varphi_{i+1} : \kappa_{i+1}, \dots, \varphi_m : \kappa_m, f_1 : \mathsf{o}^{q_1} \to \mathsf{o}, \dots, f_\ell : \mathsf{o}^{q_\ell} \to \mathsf{o}, \Sigma)$$

for each $i \in \{1,\dots,m\}$; therefore, by using Lemma 54 repeatedly, we have

$$\begin{aligned} &f_1 : \mathsf{o}^{q_1} \to \mathsf{o}, \dots, f_\ell : \mathsf{o}^{q_\ell} \to \mathsf{o}, \Sigma; \vdash t[s'_i /\!\!/ \varphi_i]_{i\le m} \rhd (\dots) =^{[\![]\!]} (;;e\theta_1\cdots\theta_m) \\ &f_1 : \mathsf{o}^{q_1} \to \mathsf{o}, \dots, f_\ell : \mathsf{o}^{q_\ell} \to \mathsf{o}, \Sigma; \vdash t'[s'_i /\!\!/ \varphi_i]_{i\le m} \rhd (\dots) =^{[\![]\!]} (;;e'\theta_1\cdots\theta_m) \end{aligned}$$

where θ_i is the substitution given in Lemma 54 corresponding to $[s'_i /\!\!/ \varphi_i]$. Also, by Lemma 53, we have

$$\begin{aligned} &\Sigma; \vdash t[s'_i /\!\!/ \varphi_i]_{i\le m}[\bar{s}_i/f_i]_{i\le k} \rhd (\dots) =^{[\![]\!]} (;;e\theta_1\cdots\theta_m\theta) \\ &\Sigma; \vdash t'[s'_i /\!\!/ \varphi_i]_{i\le m}[\bar{s}_i/f_i]_{i\le k} \rhd (\dots) =^{[\![]\!]} (;;e'\theta_1\cdots\theta_m\theta) \end{aligned}$$

where θ is the substitution given in Lemma 53 corresponding to $[\bar{s}_i/f_i]_{i\le k}$. Here note that $\cup_i \mathbf{FV}(\theta_i) \cup \mathbf{FV}(\theta) \subseteq dom(\Sigma^\natural) = dom(\theta^{a_{\mathrm{fix}}}_\Sigma)$, where we define $\mathbf{FV}([\overrightarrow{t_j/x_j}^{j}]) := \cup_j \mathbf{FV}(t_j)$ in general. Then we have:

$$\begin{aligned} (13) \iff &\#_{a_{\mathrm{fix}}}(t[s'_i /\!\!/ \varphi_i]_{i\le m}[\bar{s}_i/f_i]_{i\le k}) \quad \le \quad \#_{a_{\mathrm{fix}}}(t'[s'_i /\!\!/ \varphi_i]_{i\le m}[\bar{s}_i/f_i]_{i\le k}) \\ \iff &[\![(e\theta_1\cdots\theta_m\theta)\theta^{a_{\mathrm{fix}}}_\Sigma]\!] \quad \le \quad [\![(e'\theta_1\cdots\theta_m\theta)\theta^{a_{\mathrm{fix}}}_\Sigma]\!] \qquad (\because \text{Lemma 17}) \\ \iff &[\![e(\theta_1\theta^{a_{\mathrm{fix}}}_\Sigma)\cdots(\theta_m\theta^{a_{\mathrm{fix}}}_\Sigma)(\theta\theta^{a_{\mathrm{fix}}}_\Sigma)]\!] \quad \le \quad [\![e'(\theta_1\theta^{a_{\mathrm{fix}}}_\Sigma)\cdots(\theta_m\theta^{a_{\mathrm{fix}}}_\Sigma)(\theta\theta^{a_{\mathrm{fix}}}_\Sigma)]\!] \end{aligned}$$

where we define $[\overrightarrow{t_j/x_j}^{j}]\theta' := [\overrightarrow{t_j\theta'/x_j}^{j}]$ in general. The last condition follows from the assumption $e \preceq^{\mathbb{N}}_{\Gamma^\natural,\mathsf{o}} e'$. □

$$\texttt{poly}(t) = \begin{array}{c|ccc} a^{\varepsilon}_{\varepsilon} & a^2_1 x_2 & a^2_2 x_2^2 & \cdots \\ \hline a^1_1 x_1 & a^{1,2}_{1,1} x_1 x_2 & a^{1,2}_{1,2} x_1^2 x_2 & \cdots \\ a^1_2 x_1^2 & a^{1,2}_{2,1} x_1 x_2^2 & a^{1,2}_{2,2} x_1^2 x_2^2 & \cdots \\ \vdots & \vdots & \vdots & \ddots \end{array} \qquad \mathcal{P}([q]) = \begin{array}{c|c} \emptyset & \{2\} \\ \hline \{1\} & \{1,2\} \end{array}$$

Figure 1 Partitioning of a polynomial (with q variables) into 2^q parts, when $q = 2$

E Proofs for Section 4.3

Here we give the omitted proofs in Section 4.3.

Proof of Lemma 22. Let $\Delta : \Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma,\mathsf{o}} \to \left(\Lambda^{\Sigma_{\mathbb{N}}}_{\Gamma,\mathsf{o}}\right)^{(\prod_{i \leq n}[m_i])}$ be the diagonal function defined by $t \mapsto (t, \ldots, t)$. Consider $\preceq^{\mathbb{N}}$ on the domain set and the product quasi-ordering of $\left(\preceq^{\mathbb{N}}_{(A^{j_i}_i)_i}\right)_{(j_i)_i \in \prod_{i \leq n}[m_i]}$ on the codomain set, which is a wqo by Dickson's theorem. This function reflects the quasi-orderings since $\Lambda^{\Sigma_{\mathbb{N}}}_{\kappa_i} = \cup_{j \leq m_i} A^j_i$ for each $i \leq n$; hence the domain is a wqo. ☐

On the decomposition $\Lambda^{\Sigma_{\mathbb{N}}}_{\mathsf{o}^q \to \mathsf{o}} = \sqcup_{\Phi \in \mathcal{P}(\mathcal{P}([q]))} A^{\Phi}_q$ introduced after Lemma 22, we explain two properties, which are used in the proofs below.

First, each $\sum_{\{p_1 < \cdots < p_r\} \in \Phi} x_{p_1} \cdots x_{p_r}$ in (1) is not just a representation of the equivalence class A^{Φ}_q but further the minimum one in A^{Φ}_q: i.e., for any $t \in A^{\Phi}_q$, we have

$$\sum_{\{p_1 < \cdots < p_r\} \in \Phi} x_{p_1} \cdots x_{p_r} \quad \preceq^{\mathbb{N}} \quad \texttt{poly}(t) \quad =^{[\![\,]\!]} \quad t. \tag{14}$$

Next, note first that for any $t \in \Lambda^{\Sigma_{\mathbb{N}}}_{\mathsf{o}^q \to \mathsf{o}}$, the polynomial $\texttt{poly}(t)$ can be represented as follows (see also Figure 1):

$$\texttt{poly}(t) = \sum_{\{p_1 < \cdots < p_r\} \in \mathcal{P}([q])} \Bigg(\sum_{i_1, \ldots, i_r \geq 1} a^{p_1, \ldots, p_r}_{i_1, \ldots, i_r} x^{i_1}_{p_1} \cdots x^{i_r}_{p_r} \Bigg).$$

Then, for each $\Phi \in \mathcal{P}(\mathcal{P}([q]))$ (i.e., $\Phi \subseteq \mathcal{P}([q])$), we have the following equation:

$$\begin{aligned} A^{\Phi}_q = \Big\{ t \in \Lambda^{\Sigma_{\mathbb{N}}}_{\mathsf{o}^q \to \mathsf{o}} \mid & \text{ there exists } (a^{p_1, \ldots, p_r}_{i_1, \ldots, i_r})_{\{p_1 < \cdots < p_r\} \in \Phi, i_1, \ldots, i_r \geq 1} \text{ such that:} \\ & \texttt{poly}(t) = \sum_{\{p_1 < \cdots < p_r\} \in \Phi} \Bigg(\sum_{i_1, \ldots, i_r \geq 1} a^{p_1, \ldots, p_r}_{i_1, \ldots, i_r} x^{i_1}_{p_1} \cdots x^{i_r}_{p_r} \Bigg), \text{ and} \\ & \text{for each } \{p_1 < \cdots < p_r\} \in \Phi \text{ there exist } i_1, \ldots, i_r \geq 1 \text{ such that } a^{p_1, \ldots, p_r}_{i_1, \ldots, i_r} \geq 1 \Big\}. \end{aligned} \tag{15}$$

Equation (15) can be shown as follows: let us write the right hand side as B^{Φ}_q; then we have $\Lambda^{\Sigma_{\mathbb{N}}}_{\mathsf{o}^q \to \mathsf{o}} = \sqcup_{\Phi \in \mathcal{P}(\mathcal{P}([q]))} B^{\Phi}_q$; hence it suffices to show that $B^{\Phi}_q \subseteq A^{\Phi}_q$ for each Φ, which is straightforward.

Proof of Lemma 23. The proof proceeds by induction on P. We consider only the case that P is of the form $f_\ell\, P_1 \, \cdots \, P_{q_\ell}$; the other cases are trivial. By induction hypothesis, for each $p \leq q_\ell$, we have either $P_p \preceq^{\mathbb{N}}_{(\Phi_i)_i} 0$ or $1 \preceq^{\mathbb{N}}_{(\Phi_i)_i} P_p$. From now we show that we have $1 \preceq^{\mathbb{N}}_{(\Phi_i)_i} P$

if there exists $\{p'_1 < \cdots < p'_{r'}\} \in \Phi_\ell$ such that $\{p'_1 < \cdots < p'_{r'}\} \subseteq \{p \leq q_\ell \mid 1 \preceq^{\mathbb{N}}_{(\Phi_i)_i} P_p\}$, and we have $P \preceq^{\mathbb{N}}_{(\Phi_i)_i} 0$ otherwise.

Suppose that there exists $\{p'_1 < \cdots < p'_{r'}\} \in \Phi_\ell$ such that $\{p'_1 < \cdots < p'_{r'}\} \subseteq \{p \leq q_\ell \mid 1 \preceq^{\mathbb{N}}_{(\Phi_i)_i} P_p\}$. Let $t_i \in A^{\Phi_i}_{q_i}$ $(i \leq n)$. We have

$$\begin{aligned}
[\![(f_\ell\, P_1\, \cdots\, P_{q_\ell})[t_i/f_i]_i]\!] &= [\![t_\ell]\!]\, [\![P_1[t_i/f_i]_i]\!]\, \cdots\, [\![P_{q_\ell}[t_i/f_i]_i]\!] \\
&\geq \sum_{\{p_1 < \cdots < p_r\} \in \Phi_\ell} [\![P_{p_1}[t_i/f_i]_i]\!] \cdots [\![P_{p_r}[t_i/f_i]_i]\!] && (\because (14)) \\
&\geq [\![P_{p'_1}[t_i/f_i]_i]\!] \cdots [\![P_{p'_{r'}}[t_i/f_i]_i]\!] && (\because \{p'_1 < \cdots < p'_{r'}\} \in \Phi_\ell) \\
&\geq 1 && (\because \{p'_1 < \cdots < p'_{r'}\} \subseteq \{p \leq q_\ell \mid 1 \preceq^{\mathbb{N}}_{(\Phi_i)_i} P_p\})
\end{aligned}$$

Next, suppose the other case that, for any $\phi = \{p_1 < \cdots < p_r\} \in \Phi_\ell$, there exist $j(\phi) \leq r$ such that $P_{p_{j(\phi)}} \preceq^{\mathbb{N}}_{(\Phi_i)_i} 0$. Let $t_i \in A^{\Phi_i}_{q_i}$ $(i \leq n)$. By (15), $\mathtt{poly}(t_\ell)$ is of the form

$$\sum_{\{p_1 < \cdots < p_r\} \in \Phi_\ell} \Big(\sum_{i_1, \ldots, i_r \geq 1} a^{p_1, \ldots, p_r}_{i_1, \ldots, i_r} x^{i_1}_{p_1} \cdots x^{i_r}_{p_r} \Big).$$

Then we have:

$$\begin{aligned}
&[\![(f_\ell\, P_1\, \cdots\, P_{q_\ell})[t_i/f_i]_i]\!] \\
&= [\![t_\ell]\!]\, [\![P_1[t_i/f_i]_i]\!]\, \cdots\, [\![P_{q_\ell}[t_i/f_i]_i]\!] \\
&= [\![\mathtt{poly}(t_\ell)]\!]\, [\![P_1[t_i/f_i]_i]\!]\, \cdots\, [\![P_{q_\ell}[t_i/f_i]_i]\!] \\
&= \sum_{\{p_1 < \cdots < p_r\} \in \Phi_\ell} \Big(\sum_{i_1, \ldots, i_r \geq 1} a^{p_1, \ldots, p_r}_{i_1, \ldots, i_r} [\![P_{p_1}[t_i/f_i]_i]\!]^{i_1} \cdots [\![P_{p_r}[t_i/f_i]_i]\!]^{i_r} \Big) \\
&= \sum_{\phi = \{p_1 < \cdots < p_r\} \in \Phi_\ell} \Big(\sum_{i_1, \ldots, i_r \geq 1} a^{p_1, \ldots, p_r}_{i_1, \ldots, i_r}\ 0^{i_{j(\phi)}} \prod_{j \in [r] \setminus \{j(\phi)\}} [\![P_{p_j}[t_i/f_i]_i]\!]^{i_j} \Big) \quad (\because P_{p_{j(\phi)}} \preceq^{\mathbb{N}}_{(\Phi_i)_i} 0) \\
&= 0 \quad (\because i_{j(\phi)} \geq 1).
\end{aligned}$$

◻

Proof of Lemma 27. On the former part: We define a function h from order-2 polynomials to natural numbers as follows:

$$\begin{aligned}
&h(0) := 0 \qquad h(1) := 1 \qquad h(P_1 + P_2) := h(P_1) + h(P_2) + 1 \\
&h(P_1 \times P_2) := h(P_1) + h(P_2) + 1 \qquad h(f\, P_1\, \cdots\, P_q) := h(P_1) + \cdots + h(P_q) + 1
\end{aligned}$$

Then $h(P) = 0$ iff $P = 0$, and we can easily show that $P \longrightarrow_{(\Phi_i)_i} P'$ implies $h(P) > h(P')$.

On the latter part: We show only the base case that $P \longrightarrow^{\circ}_{(\Phi_i)_i} P'$, and then the first rule of $P \longrightarrow^{\circ}_{(\Phi_i)_i} 0$ is clear. Let $P = f_\ell\, P_1 \cdots P_{q_\ell}$ and assume the two conditions (i) and (ii). Let $t_i \in A^{\Phi_i}_{q_i}$ and we show that

$$[\![(f_\ell\, P_1 \cdots P_{q_\ell})[t_i/f_i]_{i \leq n}]\!] = [\![(f_\ell\, P_1\, \cdots\, P_{k-1}, 0, P_{k+1}\, \cdots\, P_{q_\ell})[t_i/f_i]_{i \leq n}]\!]. \tag{16}$$

By (15), $\mathtt{poly}(t_\ell)$ is of the form:

$$\sum_{\{p_1 < \cdots < p_r\} \in \Phi_\ell} \Big(\sum_{i_1, \ldots, i_r \geq 1} a^{p_1, \ldots, p_r}_{i_1, \ldots, i_r} x^{i_1}_{p_1} \cdots x^{i_r}_{p_r} \Big).$$

Then for Equation (16), it suffices to show that for any $\phi = \{p_1 < \cdots < p_r\} \in \Phi_\ell$ such that $k = p_h$ for some h,

$$\begin{aligned}&\sum_{i_1,\ldots,i_r \geq 1} a^{p_1,\ldots,p_r}_{i_1,\ldots,i_r} [\![P_{p_1}[t_i/f_i]_i]\!]^{i_1} \cdots [\![P_{p_r}[t_i/f_i]_i]\!]^{i_r} \\ =& \sum_{i_1,\ldots,i_r \geq 1} a^{p_1,\ldots,p_r}_{i_1,\ldots,i_r} [\![P_{p_1}[t_i/f_i]_i]\!]^{i_1} \cdots 0^{i_h} \cdots [\![P_{p_r}[t_i/f_i]_i]\!]^{i_r}.\end{aligned}$$

Note that the right hand side is just 0 since $i_h \geq 1$. By condition (ii) in Definition 25, for any $\phi = \{p_1 < \cdots < p_r\} \in \Phi_\ell$ such that $k \in \phi$, there exists $j(\phi) \leq r$ such that $P_{p_{j(\phi)}} \preceq^{\mathbb{N}}_{(\Phi_i)_i} 0$, and hence the left hand side above is also 0. ☐

Proof of Lemma 28. The proof proceeds by induction on the derivation of $P' \preceq^{\text{he}}_{\Sigma_\mathbb{N} \cup \Gamma} P$ and case analysis on the last rule used for $P' \preceq^{\text{he}}_{\Sigma_\mathbb{N} \cup \Gamma} P$. The case of (HE-Pres) is straightforward. In the case of (HE-Skip), let $P' \preceq^{\text{he}}_{\Sigma_\mathbb{N} \cup \Gamma} P_k$ and $P = h\, P_1 \cdots P_q$, where h is a constant in $\Sigma_\mathbb{N}$ or a variable, and $q > 0$. By induction hypothesis, it suffices to show $P_k \preceq^{\mathbb{N}}_{(\Phi_i)_i} P$. We perform a case analysis on h.

- Cases where $h = 0$ or 1: This case does not happen since $q > 0$.
- Case where $h = +$: Clear.
- Case where $h = \times$: Since P $(\neq 0)$ is a $(\Phi_i)_i$-normal form, $P = P_1 \times P_2 \not\preceq^{\mathbb{N}}_{(\Phi_i)_i} 0$ (otherwise $P \longrightarrow^{\circ}_{(\Phi_i)_i} 0$ happens). Hence $P_i \not\preceq^{\mathbb{N}}_{(\Phi_i)_i} 0$ for both i, equivalently (by Lemma 23), $1 \preceq^{\mathbb{N}}_{(\Phi_i)_i} P_i$ for both i. Then we have $P_k \preceq^{\mathbb{N}}_{(\Phi_i)_i} P_1 \times P_2 = P$.
- Case where $h = f_\ell$ $(\ell \leq n)$: In this case $q = q_\ell$ and $P = f_\ell\, P_1 \cdots P_{q_\ell}$. The case where $P_k \preceq^{\mathbb{N}}_{(\Phi_i)_i} 0$ is trivial, and we consider the case where $P_k \not\preceq^{\mathbb{N}}_{(\Phi_i)_i} 0$. To show $P_k \preceq^{\mathbb{N}}_{(\Phi_i)_i} f_\ell\, P_1 \cdots P_{q_\ell}$, let $t_i \in A^{\Phi_i}_{q_i}$ $(i \leq n)$, and we shall show that $[\![P_k[t_i/f_i]_i]\!] \leq [\![(f_\ell\, P_1 \cdots P_{q_\ell})[t_i/f_i]_i]\!]$.

  Since $f_\ell\, P_1 \cdots P_{q_\ell}$ is $(\Phi_i)_i$-normal form and $P_k \not\preceq^{\mathbb{N}}_{(\Phi_i)_i} 0$, the condition (ii) in Definition 25 must not hold, i.e., there exists $\phi' \in \Phi_\ell$ such that $k \in \phi'$ and $\phi' \subseteq \{p \in [q_\ell] \mid 1 \preceq^{\mathbb{N}}_{(\Phi_i)_i} P_p\}$. By (15), `poly`$(t_\ell)$ is of the form:

  $$\sum_{\{p_1 < \cdots < p_r\} \in \Phi_\ell} \Big(\sum_{i_1,\ldots,i_r \geq 1} a^{p_1,\ldots,p_r}_{i_1,\ldots,i_r} x^{i_1}_{p_1} \cdots x^{i_r}_{p_r} \Big)$$

  and furthermore, for $\phi' = \{p'_1 < \cdots < p'_{r'}\}$ above, there exists $i'_1, \ldots, i'_{r'} \geq 1$ such that $a^{p'_1,\ldots,p'_{r'}}_{i'_1,\ldots,i'_{r'}} \geq 1$.

  Therefore we have:

  $$\begin{aligned}&[\![(f_\ell\, P_1 \cdots P_{q_\ell})[t_i/f_i]_i]\!] \\ &= [\![t_\ell]\!]\, [\![P_1[t_i/f_i]_i]\!] \cdots [\![P_{q_\ell}[t_i/f_i]_i]\!] \\ &= \sum_{\{p_1 < \cdots < p_r\} \in \Phi_\ell} \Big(\sum_{i_1,\ldots,i_r \geq 1} a^{p_1,\ldots,p_r}_{i_1,\ldots,i_r} [\![P_{p_1}[t_i/f_i]_i]\!]^{i_1} \cdots [\![P_{p_r}[t_i/f_i]_i]\!]^{i_r} \Big) \\ &\geq a^{p'_1,\ldots,p'_{r'}}_{i'_1,\ldots,i'_{r'}} [\![P_{p'_1}[t_i/f_i]_i]\!]^{i'_1} \cdots [\![P_{p'_{r'}}[t_i/f_i]_i]\!]^{i'_{r'}} \qquad (\because \text{the case of } \phi' \text{ and } i'_1, \ldots, i'_{r'}) \\ &\geq [\![P_k[t_i/f_i]_i]\!] \qquad (\because a^{p'_1,\ldots,p'_{r'}}_{i'_1,\ldots,i'_{r'}} \geq 1,\ \ 1 \preceq^{\mathbb{N}}_{(\Phi_i)_i} P_{p'_j}\ (j \leq r'),\ \ k \in \phi',\ \ i'_k \geq 1)\end{aligned}$$

☐