Size-Based Termination for Non-Positive Types in Simply Typed Lambda-Calculus

Yuta Takahashi
Ochanomizu University, Tokyo, Japan

Abstract

So far, several typed lambda-calculus systems were combined with algebraic rewrite rules, and the termination (in other words, strong normalisation) problem of the combined systems was discussed. By the size-based approach, Blanqui formulated a termination criterion for simply typed lambda-calculus with algebraic rewrite rules which guarantees, in some specific cases, the termination of the rewrite relation induced by beta-reduction and algebraic rewrite rules on strictly or non-strictly positive inductive types. Using the inflationary fixed-point construction, we extend this termination criterion so that it is possible to show the termination of the rewrite relation induced by some rewrite rules on types which are called non-positive types. In addition, we note that a condition in Blanqui's proof can be dropped, and this improves the criterion also for non-strictly positive inductive types.

2012 ACM Subject Classification Theory of computation → Equational logic and rewriting; Theory of computation → Type theory

Keywords and phrases termination, higher-order rewriting, non-positive types, inductive types

Digital Object Identifier 10.4230/LIPIcs.TYPES.2021.12

Funding *Yuta Takahashi*: This work is supported by JSPS KAKENHI Grant Number JP21K12822.

Acknowledgements I want to thank Frédéric Blanqui and Ralph Matthes for valuable discussions on size-based termination and non-strictly positive inductive types, respectively. I also thank the anonymous reviewers for their comments which improved the earlier version of this paper.

1 Introduction

1.1 Background

Since the works [12, 21], several typed λ-calculus systems were combined with algebraic rewrite rules, and the termination (i.e., strong normalisation) problem of the combined systems was discussed: for instance, simply typed λ-calculus [11, 15, 18, 9], polymorphic λ-calculus [12, 21, 17], $\lambda\Pi$-calculus [10], the Calculus of Constructions [25, 7, 8], λ-cube [4], pure type systems [5, 6]. Rewrite rules can make each of these systems more expressive and efficient, and a termination criterion for a combined system provides a sufficient condition for the termination of the rewrite relation (i.e., the reduction relation) in this system. Of course, there are several non-terminating and interesting combined systems, but here we are interested in terminating systems only.

The performance of a combined system depends on not only its type discipline but also the range of rewrite rules whose termination is guaranteed. For instance, while Jouannaud-Okada's work [17] handles polymorphic λ-calculus and Blanqui-Jouannaud-Okada's work [11] does not, the termination criterion in the latter shows the termination of the recursion principle for the Brouwer ordinal type, which cannot be shown by the criterion in the former. The Brouwer ordinal type is a type of well founded trees and a typical example of strictly positive inductive types. Later, Blanqui ([9]) extended the criterion in [11] so that non-strictly positive inductive types can be dealt with. Though the setting of [11, 9] is simply typed λ-calculus, the termination criteria in [11, 9] are powerful enough to deal with several inductive types which are discussed in the literature on type theory.

1.2 Aim

We extend the termination criterion in [9] further by making it possible to verify the termination of the rewrite relation induced by some rewrite rules on types which are called *non-positive types*. When we denote arrow types by $T \Rightarrow U$, a non-positive type in simply typed λ-calculus means a sort (i.e., a basic type) B with a constructor $\mathsf{c} : T_1 \Rightarrow \cdots \Rightarrow T_n \Rightarrow \mathsf{B}$ such that B occurs in some T_i negatively. As shown in [20, 22, 7], there are some non-positive types such that their recursion principles induce non-termination. This indicates the difficulty in finding a terminating example of recursion principles for non-positive types. However, if one considers rewrite rules which are different from recursion principles, one can think of some rewrite rules on non-positive types whose rewrite relation should be shown to be terminating. It is desirable to extend the criterion in [9] in this respect.

1.3 Approach

The approach of [9] to a termination criterion uses computability predicates with size annotations. Roughly speaking, its termination criterion is formulated in the following way: first, an interpretation $\mathbb{I}$ of sorts is defined, and a *computability predicate* is assigned to each type T by extending this interpretation. A computability predicate is a set of terms which satisfies several desirable properties for the purpose of termination proofs. In this first step, the most crucial task is the construction of $\mathbb{I}$. For any sort B, $\mathbb{I}(\mathsf{B})$ is defined by using computability predicates annotated by ordinals: $\mathbb{I}(\mathsf{B})$ is equal to $\sup\{\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}} \mid \mathfrak{a} < \mathfrak{h}\}$ for some limit ordinal $\mathfrak{h}$ and some ordinal-indexed family $(\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}})_{\mathfrak{a}<\mathfrak{h}}$ of computability predicates, where $\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}} \subseteq \mathcal{S}^{\mathsf{B}}_{\mathfrak{b}}$ holds for any $\mathfrak{a}, \mathfrak{b}$ with $\mathfrak{a} \leq \mathfrak{b}$. This kind of ordinal-indexed family of computability predicates is called a *stratification*, and the ordinal $\mathfrak{a}$ in $\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}}$ represents the size of terms in $\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}}$. The interpretation $\mathbb{I}$ is extended to all types by defining $\mathbb{I}^*(\mathsf{B}) := \mathbb{I}(\mathsf{B})$ and

$$\mathbb{I}^*(T \Rightarrow U) := \mathbb{I}^*(T) \Rightarrow^* \mathbb{I}^*(U) := \{t \mid ts \in \mathbb{I}^*(U) \text{ for any } s \in \mathbb{I}^*(T)\}.$$

Next, a termination criterion is presented, and it is shown that if a given rewrite system satisfies the termination criterion, then any term t of type T belongs to the computability predicate $\mathbb{I}^*(T)$ assigned to T; this implies that every rewrite sequence from t terminates, hence the correctness of the termination criterion is verified.

As explained in [9], the above notion of size is useful for showing the termination of subtraction and division on the natural number type N:

$$\begin{array}{lll} \mathsf{sub}\ x\ \mathsf{0} \to x & \mathsf{sub}\ \mathsf{0}\ y \to \mathsf{0} & \mathsf{sub}\ (\mathsf{s}\ x)(\mathsf{s}\ y) \to \mathsf{sub}\ x\ y \\ \mathsf{div}\ \mathsf{0}\ (\mathsf{s}\ y) \to \mathsf{0} & \mathsf{div}\ (\mathsf{s}\ x)\ (\mathsf{s}\ y) \to \mathsf{s}\ (\mathsf{div}\ (\mathsf{sub}\ x\ y)\ (\mathsf{s}\ y)) & \end{array}$$

The termination of the rewrite relation induced by these rules is not straightforward: it is not obvious to guarantee that the argument $\mathsf{sub}\ x\ y$ of the function call $\mathsf{div}\ (\mathsf{sub}\ x\ y)\ (\mathsf{s}\ y)$ is "smaller than" $\mathsf{s}\ x$ of $\mathsf{div}\ (\mathsf{s}\ x)\ (\mathsf{s}\ y)$. But the stratification $(\mathcal{S}^{\mathsf{N}}_{\mathfrak{a}})_{\mathfrak{a}<\mathfrak{h}}$ with size annotation to constructors and function symbols enables to assign sizes to terms so that the size of $\mathsf{sub}\ x\ y$ is not greater than the size of x, and the size of $\mathsf{s}\ x$ is greater than the size of x by one.

The main obstacle in extending this approach in [9] to non-positive types is as follows. Recall that, roughly speaking, a positive inductive type is a sort B which occurs in the types of arguments of its constructors only positively. For any interpretation $\mathbb{I}$ of sorts and any type T, let $[\mathsf{B} : \mathcal{X}, \mathbb{I}]^*T$ be the interpretation of T obtained from the sort-interpretation $\mathbb{I}'$ defined as follows: $\mathbb{I}'(\mathsf{C}) := \mathcal{X}$ if $\mathsf{C} = \mathsf{B}$, otherwise $\mathbb{I}'(\mathsf{C}) := \mathbb{I}(\mathsf{C})$. Then, a crucial fact for the method of [9] is that if B occurs in T only positively, then $[\mathsf{B} : \mathcal{X}_1, \mathbb{I}]^*T \subseteq [\mathsf{B} : \mathcal{X}_2, \mathbb{I}]^*T$ holds

whenever $\mathcal{X}_1 \subseteq \mathcal{X}_2$ holds. This monotonicity property enables one to define a stratification $\mathcal{S}_0 \subseteq \mathcal{S}_1 \subseteq \cdots \subseteq \mathcal{S}_\mathfrak{a} \subseteq \cdots$ in the bottom-up way, but this property does not always hold if B occurs in T negatively.

We remove this obstacle by using the inflationary fixed-point construction ([24]), which does not assume the monotonicity of operators for fixed points as explained in [1]. This construction provides the following obvious monotonicity to any ordinal-indexed family $(\mathcal{S}_\mathfrak{c})_{\mathfrak{c}<\mathfrak{h}}$ of computability predicates: if $\mathfrak{a} \leq \mathfrak{b}$ holds then $\bigcup_{\mathfrak{c}\leq\mathfrak{a}}([\mathsf{B} : \mathcal{S}_\mathfrak{c}, \mathbb{I}]^* T) \subseteq \bigcup_{\mathfrak{c}\leq\mathfrak{b}}([\mathsf{B} : \mathcal{S}_\mathfrak{c}, \mathbb{I}]^* T)$ holds, where B may occur in T negatively. A trade-off is that, for non-positive types, we need to reformulate a size-based termination argument with pre-fixed points only.

Our construction of computability predicates for non-positive types enables us to extend the *accessibility* condition of the termination criterion in [9]. The extended accessibility condition can be explained as follows: let B be a non-positive type with a constructor $\mathsf{c} : T_1 \Rightarrow \cdots \Rightarrow T_n \Rightarrow \mathsf{B}$, and x be a variable of type T_i in which B occurs negatively. In addition, suppose that x also occurs in the right-hand side r of some rewrite rule $\mathsf{f}\, l_1 \cdots l_n \to r$. Then, the extended accessibility says that x must occur in some l_j $(1 \leq j \leq n)$ and the path from the position of x in l_j to the position of l_j consists of finitely many full applications of some constructors, where an n-ary constructor c is said to be fully applied if c takes n arguments $t_1, \ldots, t_n$. For instance, a variable g satisfies the extended accessibility if $l_j = \mathsf{c}\, g$ holds with $g : \mathsf{B} \Rightarrow \mathsf{B}$ and $\mathsf{c} : (\mathsf{B} \Rightarrow \mathsf{B}) \Rightarrow \mathsf{B}$, because we encounter only the full application of c in the path from the position of g in l_j to the position of l_j. It is crucial that, in this example, our accessibility condition permits the type $\mathsf{B} \Rightarrow \mathsf{B}$ of g to include a negative occurrence of B, which is not permitted by the accessibility condition in [9]. Together with one more revision of the termination criterion in [9], our accessibility condition provides the difference between this criterion and our termination criterion.

In addition, we note that a condition in Blanqui's proof can be dropped, and this improves the criterion with regard to non-strictly positive inductive types such as the one appearing in Hofmann's extract function for the breadth-first traversal of trees (see, e.g., [19]). Specifically, we drop a condition on a typing rule for the computability closure in [9]. This enables us to guarantee the termination of Hofmann's extract function, while it is, to the best of our knowledge, an open question whether the criterion in [9] guarantees the termination of Hofmann's extract function.

To sum up, our contributions are twofold: first, we extend the termination criterion in [9] by means of the inflationary fixed-point construction so that it is possible to show the termination of the rewrite relation induced by some rewrite rules on non-positive types. Second, we also improve this criterion with regard to non-strictly positive inductive types by verifying that a condition of a typing rule for the computability closure in [9] can be dropped.

1.4 Outline

In Section 2, we provide several preliminary definitions, and recall the facts needed in the later sections. Next, in Section 3, computability predicates with size annotations are defined. Finally, in Section 4, we formulate a termination criterion and prove the computability of typed terms with rewrite rules satisfying this criterion.

2 Preliminaries

For any finite sequence $\vec{e}$ of some elements, we denote the length of $\vec{e}$ by $|\vec{e}|$. The empty sequence is denoted by ϵ. Given a non-empty and countable set $\mathbb{S}$ of sorts, we define the set $\mathbb{T}$ of *types* by induction: (1) $\mathbb{S} \subseteq \mathbb{T}$, and (2) if $T, U \in \mathbb{T}$ holds then $T \Rightarrow U \in \mathbb{T}$ holds. The

arrow symbol $\Rightarrow$ is treated as right associative, and we often abbreviate $T_1 \Rightarrow \cdots \Rightarrow T_n \Rightarrow U$ by $\vec{T} \Rightarrow U$ with $|\vec{T}| = n$. The set $\mathbb{L}$ of *terms* is defined as follows: let $\mathbb{V}$ be a countably infinite set of variables, $\mathbb{C}$ be a countable set of constructors, and $\mathbb{F}$ be a countable set of function symbols such that $\mathbb{V}, \mathbb{C}, \mathbb{F}$ are pairwise disjoint. Then, (1) $\mathbb{V} \cup \mathbb{C} \cup \mathbb{F} \subseteq \mathbb{L}$, (2) if $T \in \mathbb{T}$, $x \in \mathbb{V}$ and $t \in \mathbb{L}$ hold then $\lambda x^T t \in \mathbb{L}$ holds, and (3) if $t, u \in \mathbb{L}$ holds then $tu \in \mathbb{L}$ holds. Below we identify two α-equivalent terms. We also adopt Barendregt's variable condition: no variable occurs both as a free one and as a bound one in a term, and all bound variables in a term are distinct. The set of all free variables in a term t is denoted by $\mathrm{FV}(t)$. We treat the term application tu as left associative, and often abbreviate $tu_1 \cdots u_n$ as $t\vec{u}$ with $|\vec{u}| = n$. When $\mathcal{X}_1$ and $\mathcal{X}_2$ are sets of terms, we define $\mathcal{X}_1 \Rightarrow^* \mathcal{X}_2 := \{s \in \mathbb{L} \mid st \in \mathcal{X}_2 \text{ for any } t \in \mathcal{X}_1\}$. The powerset of $\mathbb{L}$ is denoted by $\wp(\mathbb{L})$. As usual, a *position* in an expression such as a term is a string of positive integers (see, e.g., [3]). The subexpression of e at position p is denoted by $e|_p$, and we denote by $e[e']_p$ the expression obtained by replacing the subexpression of e at position p with e'. In addition, we denote by $\mathrm{Pos}(e, e')$ the set of all positions p in e' with $e'|_p = e$.

Mappings from $\mathbb{C} \cup \mathbb{F}$ to $\mathbb{T}$ are denoted by Θ and treated as sets of pairs. We often write $s : T$ whenever $(s, T) \in \Theta$, i.e., $\Theta(s) = T$ holds for a given Θ. For any $s \in \mathbb{C} \cup \mathbb{F}$ with $\Theta(s) = T_1 \Rightarrow \cdots \Rightarrow T_n \Rightarrow \mathsf{B}$ for some sort B, we put $r^s := n$. A *typing environment* is a mapping from a finite set of variables to a set of types. Typing environments are denoted by Γ, Δ and treated as sets of pairs. In this paper, *typing rules* are the ones of simply typed λ-calculus:

$$\frac{(s, T) \in \Theta \cup \Gamma}{\Gamma \vdash s : T} \qquad \frac{\Gamma \vdash s : T \Rightarrow U \quad \Gamma \vdash t : T}{\Gamma \vdash st : U} \qquad \frac{\Gamma \cup (x, T) \vdash u : U}{\Gamma \vdash \lambda x^T u : T \Rightarrow U}$$

A *substitution* is a mapping θ from $\mathbb{V}$ to $\mathbb{L}$ such that $\mathrm{dom}\,\theta := \{x \in \mathbb{V} \mid \theta(x) \neq x\}$ is finite. Define $\mathrm{FV}(\theta) := \bigcup\{\mathrm{FV}(\theta(x)) \mid x \in \mathrm{dom}\,\theta\}$. Any substitution θ is extended to $\mathbb{L}$ by stipulating $\theta(tu) := \theta(t)\theta(u)$ and $\theta(\lambda x^T u) := \lambda x^T \theta(u)$. We write $\theta(t)$ as $t\theta$, and always assume that no bound variable in t belongs to $\mathrm{dom}\,\theta \cup \mathrm{FV}(\theta)$, by using α-conversion if necessary.

We say that a pair (l, r) of terms is a *rewrite rule* and write it as $l \to r$ if there are a function symbol $\mathsf{f} \in \mathbb{F}$, a finite sequence $\vec{l}$ of terms, a typing environment Δ and a type T such that

- $l = \mathsf{f}\,\vec{l}$, $\mathrm{FV}(r) \subseteq \mathrm{FV}(l)$ and $\Delta \vdash l : T$ hold,
- (Subject Reduction) for any Γ and any U, if $\Gamma \vdash l : U$ holds then $\Gamma \vdash r : U$ holds.

If $\mathcal{R}$ is a set of rewrite rules, we define the *rewrite relation* $\to_{\mathcal{R}}$ on $\mathbb{L}$ as follows: $t \to_{\mathcal{R}} s$ holds if and only if $t = u[l\theta]_p$ and $u[r\theta]_p = s$ holds for some term u, some substitution θ, some position p in t and some $l \to r \in \mathcal{R}$. We define $\to := \to_{\mathcal{R}} \cup \to_\beta$, where $t \to_\beta s$ holds if and only if $t = u_0[(\lambda x^T u_1)u_2]_p$ and $u_0[u_1\{(x, u_2)\}]_p = s$ hold for some terms u_0, u_1, u_2 and some position p in u_0. For any term t and any set $\mathcal{X}$ of terms, define $\to(t) := \{u \in \mathbb{L} \mid t \to u\}$ and $\to(\mathcal{X}) := \bigcup\{\to(t) \mid t \in \mathcal{X}\}$. We say that $\to$ is *finitely branching* if $\to(t)$ is finite for any $t \in \mathbb{L}$. A term t is *normal* if there is no term t' such that $t \to t'$ holds. We denote by SN the set of all terms t which have no infinite rewrite sequence $t \to t_1 \to t_2 \to \cdots$.

When $\leq$ is a quasi ordering, we write $e_1 \leq e_2 \,\&\, e_2 \leq e_1$ as $e_1 \cong e_2$. Let $R, R_1, \ldots, R_n$ be relations. We write $\vec{x} R_{\mathrm{prod}} \vec{y}$ if and only if $|\vec{x}| = |\vec{y}|$ holds and there is an integer i with $x_i R y_i$ and $x_j = y_j$ for any $j \neq i$. We write $\vec{x}(R_1, \ldots, R_n)_{\mathrm{lex}}\vec{y}$ if and only if $|\vec{x}|, |\vec{y}| \geq n$ and there is an integer i such that $x_i R_i y_i$ and $x_j = y_j$ holds for any $j < i$.

Hereafter, we suppose that the following is given:

- a non-empty and countable set $\mathbb{S}$ of sorts,
- a countably infinite set $\mathbb{V}$ of variables, a countable set $\mathbb{C}$ of constructors, a countable set $\mathbb{F}$ of functional symbols and a mapping $\Theta : \mathbb{C} \cup \mathbb{F} \to \mathbb{T}$,
- a set $\mathcal{R}$ of rewrite rules.

▶ **Definition 1** (Interpretations of Types). *Let $\mathbb{I} : \mathbb{S} \rightharpoonup \wp(\mathbb{L})$ be a partial function from $\mathbb{S}$ to $\wp(\mathbb{L})$. We define a partial function $\mathbb{I}^* : \mathbb{T} \rightharpoonup \wp(\mathbb{L})$ as follows:*

- *$\mathbb{I}^*(\mathsf{B}) := \mathbb{I}(\mathsf{B})$ if $\mathbb{I}(\mathsf{B})$ is defined, otherwise $\mathbb{I}^*(\mathsf{B})$ is undefined.*
- *$\mathbb{I}^*(T \Rightarrow U) := \mathbb{I}^*(T) \Rightarrow^* \mathbb{I}^*(U)$ if both $\mathbb{I}^*(T)$ and $\mathbb{I}^*(U)$ are defined, otherwise $\mathbb{I}^*(T \Rightarrow U)$ is undefined.*

We write $\mathbb{I}_1^*(T) = \mathbb{I}_2^*(S)$ if and only if both $\mathbb{I}_1^*(T)$ and $\mathbb{I}_2^*(S)$ are defined and equal.

For any partial function $\mathbb{I} : \mathbb{S} \rightharpoonup \wp(\mathbb{L})$, we denote by $[\mathsf{B} : \mathcal{X}, \mathbb{I}]$ the partial function $\mathbb{I}' : \mathbb{S} \rightharpoonup \wp(\mathbb{L})$ such that

$$\mathbb{I}'(\mathsf{C}) = \begin{cases} \mathcal{X}, & \text{if } \mathsf{C} = \mathsf{B}, \\ \mathbb{I}(\mathsf{C}), & \text{if } \mathsf{C} \neq \mathsf{B} \text{ and } \mathbb{I}(\mathsf{C}) \text{ is defined}, \\ \text{undefined}, & \text{else.} \end{cases}$$

For the purpose of this paper, the distinction of *positive* positions and *negative* positions in a type is crucial. We denote the set of all positions in a type T by $\mathrm{Pos}(T)$.

▶ **Definition 2.** *For any $T \in \mathbb{T}$, we define the sets $\mathrm{Pos}^+(T)$ and $\mathrm{Pos}^-(T)$ by induction:*

- *$\mathrm{Pos}^+(\mathsf{B}) := \{\epsilon\}$, $\mathrm{Pos}^-(\mathsf{B}) := \emptyset$.*
- *$\mathrm{Pos}^s(T \Rightarrow U) := \{1p \mid p \in \mathrm{Pos}^{-s}(T)\} \cup \{2p \mid p \in \mathrm{Pos}^s(U)\}$ for each $s \in \{+, -\}$ with $-+ := -$ and $-- := +$.*

We call $\mathrm{Pos}^+(T)$ the set of all positive positions in T, and $\mathrm{Pos}^-(T)$ the set of all negative positions in T. Moreover, for any $\mathsf{B} \in \mathbb{S}$ and any $T \in \mathbb{T}$, we define $\mathrm{Pos}^s(\mathsf{B}, T) := \mathrm{Pos}(\mathsf{B}, T) \cap \mathrm{Pos}^s(T)$ for each $s \in \{+, -\}$.

A *non-positive* type is a sort B with a constructor $\mathsf{c} : T_1 \Rightarrow \cdots \Rightarrow T_n \Rightarrow \mathsf{B}$ such that for some i $(1 \leq i \leq n)$, $\mathrm{Pos}^-(\mathsf{B}, T_i)$ is non-empty.

The following fact is known:

▶ **Proposition 3** ([7]). *For any sort B, if $\mathbb{I}$ is defined for any sort occurring in T except B and $\mathrm{Pos}(\mathsf{B}, T) \subseteq \mathrm{Pos}^+(\mathsf{B}, T)$ holds, then $[\mathsf{B} : \mathcal{X}, \mathbb{I}]^*(T) : \wp(\mathbb{L}) \to \wp(\mathbb{L})$ is monotone with respect to $\mathcal{X}$, that is, if $\mathcal{X}_1 \subseteq \mathcal{X}_2$ holds then $[\mathsf{B} : \mathcal{X}_1, \mathbb{I}]^*(T) \subseteq [\mathsf{B} : \mathcal{X}_2, \mathbb{I}]^*(T)$ holds.*

The notion of computability predicate we use is a standard one, as the definition below shows. A term t is *neutral* if t has one of the following forms: (1) $x\vec{s}$, (2) $(\lambda x.t)u\vec{s}$, (3) $\mathsf{f}\,\vec{t}$, where $|\vec{t}| \geq \max\{|\vec{l}| \mid \mathsf{f}\,\vec{l} \to r \in \mathcal{R} \text{ for some } r\}$ holds.

▶ **Definition 4** (Computability Predicates). *A computability predicate is a set $\mathcal{S}$ of terms satisfying*

- *$\mathcal{S} \subseteq \mathrm{SN}$,*
- *$\to(\mathcal{S}) \subseteq \mathcal{S}$,*
- *if t is neutral and $\to(t) \subseteq \mathcal{S}$ holds, then $t \in \mathcal{S}$ holds.*

Note that for any computability predicates $\mathcal{X}_1$ and $\mathcal{X}_2$, $\mathcal{X}_1 \Rightarrow^* \mathcal{X}_2$ is a computability predicate. In Section 3, we will use the following lemma (for a proof, see [9, Lemma 1]) to define computability predicates with size annotations.

▶ **Lemma 5.** *If $\to$ is finitely branching and $\mathbb{Q}$ is a non-empty set of computability predicates with $(\mathbb{Q}, \subset)$ well ordered, then $\bigcup \mathbb{Q}$ is a computability predicate.*

The definition of computability predicates with size annotations will proceed along the hierarchy of ordinals; for this purpose, we use the notion of stratification defined below. We denote ordinals by $\mathfrak{a}, \mathfrak{b}, \mathfrak{c}, \mathfrak{d}$, and the first uncountable cardinal by $\mathfrak{h}$.

▶ **Definition 6** (Stratifications). *A stratification is an ordinal-indexed family $(\mathcal{S}_\mathfrak{a})_{\mathfrak{a}<\mathfrak{c}}$ of sets of terms for some $\mathfrak{c} \leq \mathfrak{h}$. For any term $t \in \bigcup_{\mathfrak{a}<\mathfrak{c}} \mathcal{S}_\mathfrak{a}$, we define the ordinal $o_\mathcal{S}(t)$ as the least ordinal $\mathfrak{b}$ such that $t \in \mathcal{S}_\mathfrak{b}$ holds. We say that a stratification $\mathcal{S} = (\mathcal{S}_\mathfrak{a})_{\mathfrak{a}<\mathfrak{c}}$ is monotone if and only if $\mathcal{S}_\mathfrak{a} \subseteq \mathcal{S}_\mathfrak{b}$ holds for any $\mathfrak{a}, \mathfrak{b}$ with $\mathfrak{a} \leq \mathfrak{b} < \mathfrak{c}$.*

Let $\mathcal{S} = (\mathcal{S}_\mathfrak{a})_{\mathfrak{a}<\mathfrak{h}}$ be a monotone stratification. Since any set of terms is countable, there is a countable ordinal $\mathfrak{a}$ such that $\mathcal{S}_\mathfrak{a} = \mathcal{S}_\mathfrak{c}$ holds for any $\mathfrak{c} > \mathfrak{a}$. We denote the least ordinal satisfying this property by $\mathfrak{m}(\mathcal{S})$. Moreover, let $\mathcal{S} = (\mathcal{S}_\mathfrak{a})_{\mathfrak{a}<\mathfrak{c}}$ be a monotone stratification, and suppose that $\mathbb{I}$ is defined for any sort occurring in T except B with $\mathrm{Pos}(\mathsf{B}, T) \subseteq \mathrm{Pos}^+(\mathsf{B}, T)$. Then, by Proposition 3, $([\mathsf{B} : \mathcal{S}_\mathfrak{a}, \mathbb{I}]^*(T))_{\mathfrak{a}<\mathfrak{c}}$ is a monotone stratification. We denote this monotone stratification by $[\mathsf{B} : \mathcal{S}, \mathbb{I}]^*(T)$.

One can prove the lemma below as in Lemma 3.(1) of [9].

▶ **Lemma 7.** *Let $\mathcal{S} = (\mathcal{S}_\mathfrak{a})_{\mathfrak{a}<\mathfrak{h}}$ be a monotone stratification such that every $\mathcal{S}_\mathfrak{a}$ is a computability predicate, and assume that $\to$ is finitely branching. If $t \in \mathcal{S}_{\mathfrak{m}(\mathcal{S})}$ and $t \to t'$ hold, then $t' \in \mathcal{S}_{\mathfrak{m}(\mathcal{S})}$ and $o_\mathcal{S}(t) \geq o_\mathcal{S}(t')$ hold.*

In the rest of this paper, we always assume that a given rewrite relation $\to$ is finitely branching.

3 Construction of Computability Predicates with Size Annotations

In this section, we define computability predicates with size annotations. Specifically, we first define the notion of size function (Definition 9), which controls the size-information of computability predicates. Next, given arbitrary size functions, we define stratifications by size functions (Definition 10). It is these stratifications that form a family of computability predicates with size annotations, and this family provides each sort with a computability predicate as its interpretation. Then, this interpretation of sorts is extended to all types in a straightforward way.

We fix an arbitrary well founded ordering $<_\mathbb{S}$ on $\mathbb{S}$, and denote the reflexive closure of $<_\mathbb{S}$ by $\leq_\mathbb{S}$. Therefore, $\leq_\mathbb{S}$ is a partial ordering: there is no pair $(\mathsf{B}_1, \mathsf{B}_2)$ of sorts such that $\mathsf{B}_1 \leq_\mathbb{S} \mathsf{B}_2$, $\mathsf{B}_2 \leq_\mathbb{S} \mathsf{B}_1$ and $\mathsf{B}_1 \neq \mathsf{B}_2$ hold. Since $\leq_\mathbb{S}$ is a partial ordering, we can adopt the following definition of inductive types and non-strictly positive inductive types: a sort B is an *inductive type* if for any constructor $\mathsf{c} : T_1 \Rightarrow \cdots \Rightarrow T_n \Rightarrow \mathsf{B}$ of B and any i with $1 \leq i \leq n$, $\mathrm{Pos}(\mathsf{B}, T_i) \subseteq \mathrm{Pos}^+(\mathsf{B}, T_i)$ holds. An inductive type B is *strictly positive* if for any constructor $\mathsf{c} : T_1 \Rightarrow \cdots \Rightarrow T_n \Rightarrow \mathsf{B}$ of B and any argument T_i of c with $T_i = T_{i,n_1} \Rightarrow \cdots \Rightarrow T_{i,n_j} \Rightarrow U_i$ $(j \geq 0)$, B does not occur in any of $T_{i,n_1}, \ldots, T_{i,n_j}$. A *non-strictly positive inductive type* is an inductive type not being strictly positive.

▶ **Definition 8** (Arguments of Constructors). *Let $<_\mathbb{S}$ be a well founded ordering on sorts, and T_i be the i-th argument of the constructor $\mathsf{c} : \vec{T} \Rightarrow \mathsf{B}$. Then,*

1. *T_i is recursive iff $\mathrm{Pos}(\mathsf{B}, T_i)$ is not empty,*
2. *T_i is negative iff $\mathrm{Pos}^-(\mathsf{B}, T_i)$ is not empty and for any C, either $\mathrm{Pos}(\mathsf{C}, T_i)$ is empty or $\mathsf{C} \leq_\mathbb{S} \mathsf{B}$ holds,*

3. *T_i is accessible iff* $\mathrm{Pos}(\mathsf{B}, T_i) \subseteq \mathrm{Pos}^+(\mathsf{B}, T_i)$ *holds and for any* C*, either* $\mathrm{Pos}(\mathsf{C}, T_i)$ *is empty or* $\mathsf{C} \leq_{\mathbb{S}} \mathsf{B}$ *holds.*

In order to make our stratifications well-defined (Definition 10 below), hereafter we assume that for any sort B, B has no constructor which has some non-negative and non-accessible argument. For instance, when $\mathsf{C} >_{\mathbb{S}} \mathsf{B}$ holds, a constructor $\mathsf{c}^{\#} : \mathsf{C} \Rightarrow \mathsf{B}$ of a sort B has a non-negative and non-accessible argument C. If we admit $\mathsf{c}^{\#}$ then there exists an argument T (i.e., C) of $\mathsf{c}^{\#}$ which includes an occurrence of a sort C with $\mathsf{C} >_{\mathbb{S}} \mathsf{B}$, and this breaks our definition of stratifications by size functions. Therefore, we may assume without loss of generality that for any constructor $\mathsf{c} : \vec{T} \Rightarrow \mathsf{B}$, there are natural numbers $n^{\mathsf{c}}, p^{\mathsf{c}}, q^{\mathsf{c}}$ ($n^{\mathsf{c}}, p^{\mathsf{c}}, q^{\mathsf{c}} \geq 0$) satisfying the following (if needed, we permute the arguments of c):

- for any $i \in \{1, \ldots, n^{\mathsf{c}}\}$, the i-th argument T_i of c is negative,
- for any $j \in \{n^{\mathsf{c}} + 1, \ldots, p^{\mathsf{c}}\}$, the j-th argument T_j of c is accessible and recursive,
- for any $k \in \{p^{\mathsf{c}} + 1, \ldots, q^{\mathsf{c}}\}$, the k-th argument T_k of c is accessible and non-recursive, and there is a sort occurring in T_k only positively,
- for any $l \in \{q^{\mathsf{c}} + 1, \ldots, r^{\mathsf{c}}\}$, the l-th argument T_l of c is accessible and non-recursive, and there is no sort occurring in T_l only positively.

When $\Theta(\mathsf{c}) = \vec{T} \Rightarrow \mathsf{B}$ holds, $\vec{T} \Rightarrow \mathsf{B}$ always has the following structure:

$$\underbrace{T_1 \Rightarrow \cdots \Rightarrow T_{n^{\mathsf{c}}}}_{\text{negative arguments}} \Rightarrow \underbrace{T_{n^{\mathsf{c}}+1} \Rightarrow \cdots \Rightarrow T_{p^{\mathsf{c}}}}_{\text{accessible and recursive arguments}} \Rightarrow$$
$$\underbrace{T_{p^{\mathsf{c}}+1} \Rightarrow \cdots \Rightarrow T_{q^{\mathsf{c}}} \Rightarrow T_{q^{\mathsf{c}}+1} \Rightarrow \cdots \Rightarrow T_{r^{\mathsf{c}}}}_{\text{accessible and non-recursive arguments}} \Rightarrow \mathsf{B}.$$

For instance, if $\mathsf{C} <_{\mathbb{S}} \mathsf{B}$ and $\Theta(\mathsf{c}) = (\mathsf{B} \Rightarrow \mathsf{C}) \Rightarrow (\mathsf{C} \Rightarrow \mathsf{B}) \Rightarrow \mathsf{C} \Rightarrow (\mathsf{C} \Rightarrow \mathsf{C}) \Rightarrow \mathsf{B}$ hold, then we have $n^{\mathsf{c}} = 1$, $p^{\mathsf{c}} = 2$, $q^{\mathsf{c}} = 3$ and $r^{\mathsf{c}} = 4$. On the other hand, if $\mathsf{C} <_{\mathbb{S}} \mathsf{B}$ and $\Theta(\mathsf{c}) = (\mathsf{C} \Rightarrow \mathsf{B}) \Rightarrow (\mathsf{C} \Rightarrow \mathsf{C}) \Rightarrow \mathsf{B}$ hold, then $n^{\mathsf{c}} = 0$, $p^{\mathsf{c}} = 1$, $q^{\mathsf{c}} = 1$ and $r^{\mathsf{c}} = 2$ hold. Notice that if $p^{\mathsf{c}} = 0$ holds then $n^{\mathsf{c}} = 0$ holds, and similar implications hold for q^{c} and r^{c}.

The size-information of stratifications is controlled by *size functions*, which compute an ordinal as the size of $\mathsf{c}\,\vec{t}$ from ordinals attached to $t_1, \ldots, t_{q^{\mathsf{c}}}$ as their sizes.

▶ **Definition 9** (Size Functions). *For any constructor* $\mathsf{c} : \vec{T} \Rightarrow \mathsf{B}$*, a size function* $(\Sigma^{\mathsf{c}}, \vec{\mathsf{B}^{\mathsf{c}}})$ *for* c *consists of a function* $\Sigma^{\mathsf{c}} : \mathfrak{h}^{q^{\mathsf{c}}} \to \mathfrak{h}$ *and sorts* $\mathsf{B}^{\mathsf{c}}_1, \ldots, \mathsf{B}^{\mathsf{c}}_{q^{\mathsf{c}}}$ *such that*

- *for any* $i \in \{1, \ldots, p^{\mathsf{c}}\}$, $\mathsf{B}^{\mathsf{c}}_i = \mathsf{B}$ *holds, and*
- *for any* $i \in \{p^{\mathsf{c}} + 1, \ldots, q^{\mathsf{c}}\}$, $\mathsf{B}^{\mathsf{c}}_i$ *occurs in* T_i *with* $\mathrm{Pos}(\mathsf{B}^{\mathsf{c}}_i, T_i) \subseteq \mathrm{Pos}^+(\mathsf{B}^{\mathsf{c}}_i, T_i)$ *and* $\mathsf{B}^{\mathsf{c}}_i <_{\mathbb{S}} \mathsf{B}$.

We often denote a size function $(\Sigma^{\mathsf{c}}, \vec{\mathsf{B}^{\mathsf{c}}})$ by Σ^{c}. If $\mathsf{B} \in \mathbb{S}$ holds, then we define

- $\mathbb{C}^{\mathsf{B}} := \{(\mathsf{c}, \vec{t}, \vec{T}) \mid \mathsf{c} \in \mathbb{C},\ \mathsf{c} : \vec{T} \Rightarrow \mathsf{B},\ |\vec{t}| = |\vec{T}|\}$,
- $\mathbb{C}^{\mathsf{B}}_{\to *}(t) := \{(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}} \mid t \to^* \mathsf{c}\,\vec{t}\}$, where $\to^*$ is the reflexive and transitive closure of $\to$.

Note that for any $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}$, we do not require $t_i : T_i$.

Below we define *stratifications by size functions*, and these stratifications form the interpretation $\mathbb{J}$ of sorts, which assigns a computability predicate to each sort. Notice that we use the inflationary fixed-point construction (see, e.g., [24, 1, 2]) in the case of negative arguments of constructors.

▶ **Definition 10** (Stratifications by Size Functions). *Assume that a size function is provided with each constructor. For any sort* B*, we define the stratification* $\mathcal{S}^{\mathsf{B}}$ *and the value* $\mathbb{J}(\mathsf{B})$ *of the function* $\mathbb{J}$ *from* $\mathbb{S}$ *to* $\wp(\mathbb{L})$ *by induction on* $>_{\mathbb{S}}$*. Suppose that* $\mathcal{S}^{\mathsf{C}}$ *and* $\mathbb{J}(\mathsf{C})$ *are defined for any sort* $\mathsf{C} <_{\mathbb{S}} \mathsf{B}$*. We first define* $\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}}$ *by induction on* $\mathfrak{a} \in \mathfrak{h}$*:* $\mathcal{S}^{\mathsf{B}}_0$ *is defined as the set of all terms* $t \in \mathrm{SN}$ *such that for any* $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}_{\to *}(t)$*,*

- $p^{\mathsf{c}} = 0$,
- *for any* $i \in \{1, \dots, q^{\mathsf{c}}\}$, $t_i \in \mathbb{J}^*(T_i)$, *and*
- $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c},1}}(t_1), \dots, o_{\mathcal{S}^{\mathsf{c},q^{\mathsf{c}}}}(t_{q^{\mathsf{c}}})) = 0$, *where* $\mathcal{S}^{\mathsf{c},i}$ *is a stratification* $([\mathsf{B}^{\mathsf{c}}_i : \mathcal{S}^{\mathsf{B}^{\mathsf{c}}_i}_{\mathfrak{a}}, \mathbb{J}]^*(T_i))_{\mathfrak{a}<\mathfrak{h}}$ *for any* $i \in \{1, \dots, q^{\mathsf{c}}\}$.

  We abbreviate $o_{\mathcal{S}^{\mathsf{c},1}}(t_1), \dots, o_{\mathcal{S}^{\mathsf{c},q^{\mathsf{c}}}}(t_{q^{\mathsf{c}}})$ *as* $o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})$.

When $\mathfrak{a} = \mathfrak{b} + 1$ *holds, we define* $\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}}$ *as the set of all terms* $t \in \mathrm{SN}$ *such that for any* $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}_{\to^*}(t)$,

- *for any* $k \in \{1, \dots, n^{\mathsf{c}}\}$, $t_k \in \bigcup_{\mathfrak{c}\le\mathfrak{b}} [\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{c}}, \mathbb{J}]^*(T_k)$,
- *for any* $k \in \{n^{\mathsf{c}} + 1, \dots, p^{\mathsf{c}}\}$, $t_k \in [\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{b}}, \mathbb{J}]^*(T_k)$,
- *for any* $i \in \{p^{\mathsf{c}} + 1, \dots, q^{\mathsf{c}}\}$, $t_i \in \mathbb{J}^*(T_i)$, *and*
- $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c},1}}(t_1), \dots, o_{\mathcal{S}^{\mathsf{c},q^{\mathsf{c}}}}(t_{q^{\mathsf{c}}})) \le \mathfrak{b} + 1$, *where*
  - $\mathcal{S}^{\mathsf{c},k}$ *is a stratification* $(\bigcup_{\mathfrak{d}\le\mathfrak{c}} [\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{d}}, \mathbb{J}]^*(T_k))_{\mathfrak{c}\le\mathfrak{b}}$ *for any* $k \in \{1, \dots, n^{\mathsf{c}}\}$,
  - $\mathcal{S}^{\mathsf{c},k}$ *is a stratification* $([\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{c}}, \mathbb{J}]^*(T_k))_{\mathfrak{c}\le\mathfrak{b}}$ *for any* $k \in \{n^{\mathsf{c}} + 1, \dots, p^{\mathsf{c}}\}$ *and*
  - $\mathcal{S}^{\mathsf{c},k}$ *is a stratification* $([\mathsf{B}^{\mathsf{c}}_k : \mathcal{S}^{\mathsf{B}^{\mathsf{c}}_k}_{\mathfrak{c}}, \mathbb{J}]^*(T_i))_{\mathfrak{c}<\mathfrak{h}}$ *for any* $k \in \{p^{\mathsf{c}} + 1, \dots, q^{\mathsf{c}}\}$.

  As above, we abbreviate $o_{\mathcal{S}^{\mathsf{c},1}}(t_1), \dots, o_{\mathcal{S}^{\mathsf{c},q^{\mathsf{c}}}}(t_{q^{\mathsf{c}}})$ *as* $o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})$.

When $\mathfrak{a}$ *is a limit ordinal, we define* $\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}} := \bigcup_{\mathfrak{b}<\mathfrak{a}} \mathcal{S}^{\mathsf{B}}_{\mathfrak{b}}$. *Finally, we put* $\mathcal{S}^{\mathsf{B}} := (\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}})_{\mathfrak{a}<\mathfrak{h}}$ *and* $\mathbb{J}(\mathsf{B}) := \mathcal{S}^{\mathsf{B}}_{\mathfrak{m}(\mathcal{S}^{\mathsf{B}})}$. *To justify the definition of* $\mathbb{J}(\mathsf{B})$, *we show that* $\mathcal{S}^{\mathsf{B}}$ *is monotone in Lemma 11.(2) below.*

Note that in the definition above we used induction on $>_{\mathbb{S}}$ and subinduction on $\mathfrak{a} \in \mathfrak{h}$. By the hypothesis of subinduction, we can assume in the case of $\mathfrak{a} = \mathfrak{b} + 1$ that $\mathcal{S}^{\mathsf{B}}_{\mathfrak{c}}$ is already defined for any $\mathfrak{c} \le \mathfrak{b}$. For any sort B, any set $\mathcal{X}$ of terms and any type T, we abbreviate $[\mathsf{B} : \mathcal{X}, \mathbb{J}]^*(T)$ as $[\mathsf{B} : \mathcal{X}]T$, and $t \in \mathbb{J}^*(T)$ as $t \in T$ for any term t. The lemma below shows that for any sort B, the stratification $\mathcal{S}^{\mathsf{B}}$ is monotone, and each $\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}}$ is a computability predicate.

▶ **Lemma 11.** *The following statements hold.*

1. *Let* P, Q *be two arbitrary sets of triples* $(\mathsf{c}, \vec{t}, \vec{T})$ *of a constructor, a finite sequence of terms and a finite sequence of types. If* $P \subseteq Q$ *holds then* $\{t \in \mathrm{SN} \mid \mathbb{C}^{\mathsf{B}}_{\to^*}(t) \subseteq P\} \subseteq \{t \in \mathrm{SN} \mid \mathbb{C}^{\mathsf{B}}_{\to^*}(t) \subseteq Q\}$ *holds.*
2. *For any sort* B, $\mathcal{S}^{\mathsf{B}}$ *is monotone.*
3. *For any sort* B *and any* $\mathfrak{a} < \mathfrak{h}$, $\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}}$ *is a computability predicate.*

Proof.

(1.) Straightforward.

(2.) We show by induction on $\mathfrak{b}$ that if $\mathfrak{a} \le \mathfrak{b}$ holds then $\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}} \subseteq \mathcal{S}^{\mathsf{B}}_{\mathfrak{b}}$ holds. When $\mathfrak{b}$ is 0 or a limit ordinal, the assertion is obvious. Let $\mathfrak{b} = \mathfrak{c} + 1$ be the case. First, we show $\mathcal{S}^{\mathsf{B}}_{\mathfrak{c}} \subseteq \mathcal{S}^{\mathsf{B}}_{\mathfrak{c}+1}$. Let $t \in \mathcal{S}^{\mathsf{B}}_{\mathfrak{c}}$ be the case. By the definition of $\mathcal{S}^{\mathsf{B}}$, $o_{\mathcal{S}^{\mathsf{B}}}(t)$ cannot be a limit ordinal, hence either $t \in \mathcal{S}^{\mathsf{B}}_0$ or $t \in \mathcal{S}^{\mathsf{B}}_{\mathfrak{c}_0+1}$ holds for some ordinal $\mathfrak{c}_0 < \mathfrak{c}$. Here we consider the latter case only (the former case is similar). In this case, there are sets $P_{\mathfrak{c}_0+1}$ and $P_{\mathfrak{c}+1}$ such that for any $\mathfrak{d} \in \{\mathfrak{c}_0 + 1, \mathfrak{c} + 1\}$, we have $\mathcal{S}^{\mathsf{B}}_{\mathfrak{d}} = \{u \in \mathrm{SN} \mid \mathbb{C}^{\mathsf{B}}_{\to^*}(u) \subseteq P_{\mathfrak{d}}\}$ and $P_{\mathfrak{d}}$ is the set of all $(\mathsf{c}, \vec{t}, \vec{T})$ such that

- for any $k \in \{1, \dots, n^{\mathsf{c}}\}$, $t_k \in \bigcup_{\mathfrak{c}\le\mathfrak{d}-1} [\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{c}}]T_k$,
- for any $k \in \{n^{\mathsf{c}} + 1, \dots, p^{\mathsf{c}}\}$, $t_k \in [\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{d}-1}]T_k$,
- for any $i \in \{p^{\mathsf{c}} + 1, \dots, q^{\mathsf{c}}\}$, $t_i \in T_i$, and
- $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})) \le \mathfrak{d}$.

By the assertion 1. above, it suffices to verify that $P_{\mathfrak{c}_0+1} \subseteq P_{\mathfrak{c}+1}$ holds. Assume that $(\mathsf{c}, \vec{t}, \vec{T}) \in P_{\mathfrak{c}_0+1}$ holds. First, we have $t_k \in \bigcup_{\mathfrak{b}\leq\mathfrak{c}}[\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{b}}]T_k$ for any $k \in \{1, \ldots, n^{\mathsf{c}}\}$ by $\mathfrak{c}_0 < \mathfrak{c}$. Moreover, for any $k \in \{n^{\mathsf{c}}+1, \ldots, p^{\mathsf{c}}\}$, we have $\mathcal{S}^{\mathsf{B}}_{\mathfrak{c}_0} \subseteq \mathcal{S}^{\mathsf{B}}_{\mathfrak{c}}$ by IH, hence we have $t_k \in [\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{c}}]T_k$ by $\mathrm{Pos}(\mathsf{B}, T_k) \subseteq \mathrm{Pos}^+(\mathsf{B}, T_k)$ and Proposition 3. It is obvious that $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})) \leq \mathfrak{c}+1$ holds, so we have $P_{\mathfrak{c}_0+1} \subseteq P_{\mathfrak{c}+1}$. Therefore, $\mathcal{S}^{\mathsf{B}}_{\mathfrak{c}} \subseteq \mathcal{S}^{\mathsf{B}}_{\mathfrak{c}+1}$ holds. Then, by IH, one can see that if $\mathfrak{a} \leq \mathfrak{c}+1$ holds then $\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}} \subseteq \mathcal{S}^{\mathsf{B}}_{\mathfrak{c}+1}$ holds.

(3.) By induction on $\mathfrak{a}$. Since the case of $\mathfrak{a} = 0$ is similar to the case of successors, we consider the cases of successors and limits only. As we have seen in the proof of the assertion 2. above, there is a set $P_{\mathfrak{b}+1}$ such that we have $\mathcal{S}^{\mathsf{B}}_{\mathfrak{b}+1} = \{t \in \mathrm{SN} \mid \mathbb{C}^{\mathsf{B}}_{\to^*}(t) \subseteq P_{\mathfrak{b}+1}\}$. One can show as in [9, Lemma 6] that for any set P of triples of a constructor, a finite sequence of terms and a finite sequence of types, $\{t \in \mathrm{SN} \mid \mathbb{C}^{\mathsf{B}}_{\to^*}(t) \subseteq P\}$ is a computability predicate. Therefore, $\mathcal{S}^{\mathsf{B}}_{\mathfrak{b}+1}$ is a computability predicate. If $\mathfrak{a}$ is limit, then $\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}} = \bigcup_{\mathfrak{b}<\mathfrak{a}} \mathcal{S}^{\mathsf{B}}_{\mathfrak{b}}$ holds, and each $\mathcal{S}^{\mathsf{B}}_{\mathfrak{b}}$ with $\mathfrak{b} < \mathfrak{a}$ is a computability predicate by IH. Since $((\mathcal{S}^{\mathsf{B}}_{\mathfrak{b}})_{\mathfrak{b}<\mathfrak{a}}, \subset)$ is a well ordering by the monotonicity of $\mathcal{S}^{\mathsf{B}}$, $\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}}$ is a computability predicate by Lemma 5 and the assumption that $\to$ is finitely branching. ◀

Following [1], one can say that $\mathcal{S}^{\mathsf{B}}_{\mathfrak{m}(\mathcal{S}^{\mathsf{B}})}$ is a *pre-fixed point* in the following sense: for any $(\mathsf{c}, \vec{t}, \vec{T})$ such that

- $\mathsf{c} : \vec{T} \Rightarrow \mathsf{B}$ holds, $|\vec{t}| = |\vec{T}|$ holds and $\mathsf{c}\,\vec{t}$ is normal,
- for any $k \in \{1, \ldots, n^{\mathsf{c}}\}$, $t_k \in \bigcup_{\mathfrak{c}\leq\mathfrak{m}(\mathcal{S}^{\mathsf{B}})}[\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{c}}]T_k$,
- for any $k \in \{n^{\mathsf{c}}+1, \ldots, p^{\mathsf{c}}\}$, $t_k \in [\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{m}(\mathcal{S}^{\mathsf{B}})}]T_k$,
- for any $i \in \{p^{\mathsf{c}}+1, \ldots, q^{\mathsf{c}}\}$, $t_i \in T_i$, and
- $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})) \leq \mathfrak{m}(\mathcal{S}^{\mathsf{B}}) + 1$,

we have $\mathsf{c}\,\vec{t} \in \mathcal{S}^{\mathsf{B}}_{\mathfrak{m}(\mathcal{S}^{\mathsf{B}})}$. Hereafter, we often abbreviate $t \in \mathcal{S}^{\mathsf{B}}_{\mathfrak{m}(\mathcal{S}^{\mathsf{B}})}$ as $t \in \mathcal{S}^{\mathsf{B}}$.

The statements 1–3 of the lemma below are used in the proof of its statement 4, while the statement 4 will be used in the proof of Lemma 13 below.

▶ **Lemma 12.** *The following statements hold:*

1. *$t \in \mathcal{S}^{\mathsf{B}}_0$ holds iff $t \in \mathcal{S}^{\mathsf{B}}$ holds and for any $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}_{\to^*}(t)$, $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})) = p^{\mathsf{c}} = 0$ holds.*
2. *$t \in \mathcal{S}^{\mathsf{B}}_{\mathfrak{a}+1}$ holds iff*
   - *$t \in \mathcal{S}^{\mathsf{B}}$ holds,*
   - *for any $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}_{\to^*}(t)$, $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})) \leq \mathfrak{a}+1$ holds, and for any $k \in \{1, \ldots, p^{\mathsf{c}}\}$, $o_{\mathcal{S}^{\mathsf{c},k}}(t_k) \leq \mathfrak{a}$ holds.*
3. *If $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}$ and $\mathsf{c}\,\vec{t} \in \mathcal{S}^{\mathsf{B}}$ holds, then we have $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t}) \geq \Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t}))$ and $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t}) > o_{\mathcal{S}^{\mathsf{c},k}}(t_k)$ for any $k \in \{1, \ldots, p^{\mathsf{c}}\}$.*
4. *Let δ be the function on $\mathfrak{h}$ such that $\delta(\mathfrak{a}) = \mathfrak{a}+1$ if $\mathfrak{a}$ is a limit ordinal, and $\delta(\mathfrak{a}) = \mathfrak{a}$ otherwise. If $t \in \mathcal{S}^{\mathsf{B}}$ holds, then we have $o_{\mathcal{S}^{\mathsf{B}}}(t) = \delta(\sup(R \cup S \cup T))$ with*
   - *$R = \{o_{\mathcal{S}^{\mathsf{B}}}(t') \mid t \to t'\}$,*
   - *$S = \{o_{\mathcal{S}^{\mathsf{c},k}}(t_k)+1 \mid (\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}},\ t = \mathsf{c}\,\vec{t},\ 1 \leq k \leq p^{\mathsf{c}}\}$,*
   - *$T = \{\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t}))\}$ with $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}$ and $t = \mathsf{c}\,\vec{t}$.*

Proof.

(1.) ($\Longrightarrow$) Obvious. ($\Longleftarrow$) Since $t \in \mathcal{S}^{\mathsf{B}}$ holds, we have $t \in \mathcal{S}^{\mathsf{B}}_{\mathfrak{a}}$ with $\mathfrak{a} = o_{\mathcal{S}^{\mathsf{B}}}(t)$. Then, by definition, we have $t \in \mathrm{SN}$, and for any $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}_{\to^*}(t)$ and any $i \in \{p^{\mathsf{c}}+1, \ldots, q^{\mathsf{c}}\}$, $t_i \in T_i$ holds. Therefore, we have $t \in \mathcal{S}^{\mathsf{B}}_0$ because $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})) = 0$ holds.

(2.) ($\Longrightarrow$) Obvious. ($\Longleftarrow$) If $p^{\mathsf{c}} = 0$ holds then we immediately have $t \in \mathcal{S}^{\mathsf{B}}_{\mathfrak{a}+1}$ by assumption. Suppose that $p^{\mathsf{c}} \geq 1$ holds. We have $t \in \mathcal{S}^{\mathsf{B}}_{\mathfrak{c}+1}$ with $\mathfrak{c}+1 = o_{\mathcal{S}^{\mathsf{B}}}(t)$, so $t \in \mathrm{SN}$ holds. If $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}_{\to^*}(t)$ holds, then for any $i \in \{p^{\mathsf{c}}+1, \ldots, q^{\mathsf{c}}\}$, $t_i \in T_i$ holds. Moreover, we have $o_{\mathcal{S}^{\mathsf{c},k}}(t_k) \leq \mathfrak{a}$ for any $k \in \{1, \ldots, p^{\mathsf{c}}\}$. Therefore, $t_k \in \bigcup_{\mathfrak{b}\leq\mathfrak{a}}[\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{b}}]T_k$ holds for any

$k \in \{1, \ldots, n^{\mathsf{c}}\}$, and $t_k \in [\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{a}}]T_k$ holds for any $k \in \{n^{\mathsf{c}}+1, \ldots, p^{\mathsf{c}}\}$. Then, $t \in \mathcal{S}^{\mathsf{B}}_{\mathfrak{a}+1}$ holds because we have $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})) \leq \mathfrak{a}+1$.

(3.) One can see that $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t})$ is either 0 or a successor $\mathfrak{a}+1$. If $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t}) = 0$ holds, then $\mathsf{c}\,\vec{t} \in \mathcal{S}^{\mathsf{B}}_0$ holds and so we have $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t}) = 0 \geq \Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t}))$ by definition. If $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t}) = \mathfrak{a}+1$ holds, then $\mathsf{c}\,\vec{t} \in \mathcal{S}^{\mathsf{B}}_{\mathfrak{a}+1}$ holds and so we have $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t}) = \mathfrak{a}+1 \geq \Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t}))$ by definition. Next, we show that $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t}) > o_{\mathcal{S}^{\mathsf{c},k}}(t_k)$ holds for any $k \in \{1, \ldots, p^{\mathsf{c}}\}$. If $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t}) = 0$ holds, then $p^{\mathsf{c}} = 0$ holds and so the assertion holds vacuously. Let $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t}) = \mathfrak{a}+1$ be the case, and take a natural number $k \in \{1, \ldots, p^{\mathsf{c}}\}$. If $k \leq n^{\mathsf{c}}$ holds, then we have $t_k \in \bigcup_{\mathfrak{b}<\mathfrak{a}}[\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{b}}]T_k$ and so $o_{\mathcal{S}^{\mathsf{c},k}}(t_k) \leq \mathfrak{a}$ holds. Otherwise we have $t_k \in [\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{a}}]T_k$, hence $o_{\mathcal{S}^{\mathsf{c},k}}(t_k) \leq \mathfrak{a}$ holds as well.

(4.) We put $\mathfrak{a} := \sup(R \cup S \cup T)$ and $\mathfrak{b} := o_{\mathcal{S}^{\mathsf{B}}}(t)$. First, we show $\mathfrak{b} \geq \delta(\mathfrak{a})$. Let $t \to t'$ be the case, then we have $\mathfrak{b} \geq o_{\mathcal{S}^{\mathsf{B}}}(t')$ by Lemma 7. We have $\mathfrak{b} \geq \sup(S \cup T)$ by the statement 3. above, hence $\mathfrak{b} \geq \mathfrak{a}$. Since $\mathfrak{b}$ cannot be a limit ordinal, if $\mathfrak{a}$ is a limit ordinal then $\mathfrak{b} > \mathfrak{a}$ holds, so $\mathfrak{b} \geq \delta(\mathfrak{a})$ holds. Otherwise, we have $\mathfrak{b} \geq \mathfrak{a} = \delta(\mathfrak{a})$.
Next, we show $\delta(\mathfrak{a}) \geq \mathfrak{b}$. It suffices to show $t \in \mathcal{S}^{\mathsf{B}}_{\delta(\mathfrak{a})}$. Since $t \in \mathcal{S}^{\mathsf{B}}$ holds, we have $t \in \mathrm{SN}$, and for any $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}_{\to *}(t)$ and any $i \in \{p^{\mathsf{c}}+1, \ldots, q^{\mathsf{c}}\}$, $t_i \in T_i$ holds.
- $\delta(\mathfrak{a}) = 0$: let $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}_{\to *}(t)$ be the case. By the statement 1. above, it suffices to show $p^{\mathsf{c}} = 0$ and $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})) = 0$. First, consider the case of $t = \mathsf{c}\,\vec{t}$. Then, S must be empty and so $p^{\mathsf{c}} = 0$ holds. Moreover, we have $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})) = 0$ by $\sup(T) = 0$. Next, let $t \to t' \to_* \mathsf{c}\,\vec{t}$ be the case. Then, we have $o_{\mathcal{S}^{\mathsf{B}}}(t') = 0$ and so $t' \in \mathcal{S}^{\mathsf{B}}_0$ holds. It follows that $p^{\mathsf{c}} = 0$ and $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})) = 0$ hold, because $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}_{\to *}(t')$ holds.
- $\delta(\mathfrak{a}) = \mathfrak{c}+1$: let $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}_{\to *}(t)$ be the case. By the statement 2. above, it suffices to show that $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})) \leq \mathfrak{c}+1$ holds and $o_{\mathcal{S}^{\mathsf{c},k}}(t_k) \leq \mathfrak{c}$ holds for any $k \in \{1, \ldots, p^{\mathsf{c}}\}$. Consider the case of $t = \mathsf{c}\,\vec{t}$ first. We have $\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})) = \sup(T) \leq \mathfrak{c}+1$. Moreover, for any $k \in \{1, \ldots, p^{\mathsf{c}}\}$, we have $o_{\mathcal{S}^{\mathsf{c},k}}(t_k) + 1 \leq \sup(S) \leq \mathfrak{a} \leq \mathfrak{c}+1$. Next, let $t \to t' \to_* \mathsf{c}\,\vec{t}$ be the case. We have $t', \mathsf{c}\,\vec{t} \in \mathcal{S}^{\mathsf{B}}$ because $\mathcal{S}^{\mathsf{B}}_{\mathfrak{m}(\mathcal{S}^{\mathsf{B}})}$ is a computability predicate. By Lemma 7 and the statement 3. above, we have

  $$\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t})) \leq o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t}) \leq o_{\mathcal{S}^{\mathsf{B}}}(t') \leq \mathfrak{c}+1.$$

  For any $k \in \{1, \ldots, p^{\mathsf{c}}\}$, we have

  $$o_{\mathcal{S}^{\mathsf{c},k}}(t_k) < o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t}) \leq o_{\mathcal{S}^{\mathsf{B}}}(t') \leq \mathfrak{c}+1$$

  by Lemma 7 and the statement 3. again. Therefore, we have $t \in \mathcal{S}^{\mathsf{B}}_{\delta(\mathfrak{a})}$. ◀

By using Lemma 12.(4) as in the proof of [9, Corollaries 1 and 2], one can prove the following lemma, which is a key lemma for our termination criterion (Theorem 24 below).

▶ **Lemma 13.** *Let c be a constructor, and assume that Σ^{c} is strictly extensive with respect to recursive arguments of c, i.e., $\mathfrak{a}_i < \Sigma^{\mathsf{c}}(\vec{\mathfrak{a}})$ holds for any $\vec{\mathfrak{a}}$ and any $i \in \{1, \ldots, p^{\mathsf{c}}\}$. Moreover, we suppose that $\Sigma^{\mathsf{c}}(\vec{\mathfrak{a}})$ is not a limit ordinal for any $\vec{\mathfrak{a}}$.*

1. *For any $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}$ such that $\mathsf{c}\,\vec{t} \in \mathcal{S}^{\mathsf{B}}$ holds and $\mathsf{c}\,\vec{t}$ is normal, we have $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t}) = \Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t}))$.*
2. *If Σ^{c} is monotone, then for any $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}$ with $\mathsf{c}\,\vec{t} \in \mathcal{S}^{\mathsf{B}}$, we have $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t}) = \Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c}}}(\vec{t}))$.*

4 Computability of Well-Typed Terms

In this section, we first define the notion of annotation for constructors (Definition 17). An annotation for constructors determines the corresponding size functions (Definition 18), so one can define stratifications by size functions and the interpretation $\mathbb{J}$ of sorts along

Definition 10 above whenever an annotation for constructors is given. Next, we define the notion of annotations for function symbols (Definition 20). Finally, we give our termination criterion, namely, a sufficient condition for the termination of $\to$ which consists of several criteria concerning annotations for constructors and function symbols (Theorem 24).

First of all, we recall *accessible subterms* in [9], and define a variant of these terms called *quasi-accessible subterms*.

▶ **Definition 14** (Accessible Subterms and Quasi-Accessible Subterms). *Assume that a family* $(\Sigma^{\mathsf{c}}, \vec{\mathsf{B}^{\mathsf{c}}})_{\mathsf{c}\in\mathbb{C}}$ *of size functions is given. We say that a triple* (u, U, C) *is accessible in a tuple* (t, T, B) *and write* $(u, U, \mathsf{C}) \trianglelefteq_{\mathrm{a}} (t, T, \mathsf{B})$ *if and only if either 1. or 2. below is satisfied:*

1. $(u, U, \mathsf{C}) = (t, T, \mathsf{B})$.
2. *There are a tuple* $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}$ *and an integer* $k \in \{n^{\mathsf{c}} + 1, \dots, q^{\mathsf{c}}\}$ *such that* $t = \mathsf{c}\,\vec{t}$, $T = \mathsf{B}$ *and* $(u, U, \mathsf{C}) \trianglelefteq_{\mathrm{a}} (t_k, T_k, \mathsf{B}^{\mathsf{c}}_k)$ *hold.*

We say that (u, U, C) *is quasi-accessible in* (t, T, B) *and write* $(u, U, \mathsf{C}) \trianglelefteq_{\mathrm{qa}} (t, T, \mathsf{B})$ *if and only if either 1. above or 2'. below is satisfied:*

2'. *There are a tuple* $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}$ *and an integer* $k \in \{1, \dots, q^{\mathsf{c}}\}$ *such that* $t = \mathsf{c}\,\vec{t}$, $T = \mathsf{B}$ *and* $(u, U, \mathsf{C}) \trianglelefteq_{\mathrm{qa}} (t_k, T_k, \mathsf{B}^{\mathsf{c}}_k)$ *hold.*

Note that if $(u, U, \mathsf{C}) \trianglelefteq_{\mathrm{qa}} (t, T, \mathsf{B})$ holds by the condition 2' above, then either $(u, U, \mathsf{C}) \trianglelefteq_{\mathrm{a}} (t, T, \mathsf{B})$ holds or there are finitely many distinct tuples $(u_1, U_1, \mathsf{C}_1), \dots, (u_n, U_n, \mathsf{C}_n)$ $(n \geq 1)$ such that $(u_n, U_n, \mathsf{C}_n) \trianglelefteq_{\mathrm{a}} \cdots \trianglelefteq_{\mathrm{a}} (u_1, U_1, \mathsf{C}_1) = (t, T, \mathsf{B})$ holds with $u_n = \mathsf{c}\,\vec{t}$ and we have $u = t_k$ for some negative argument k of c. In other words, $\trianglelefteq_{\mathrm{qa}} \setminus \trianglelefteq_{\mathrm{a}}$ (i.e., the difference of $\trianglelefteq_{\mathrm{qa}}$ and $\trianglelefteq_{\mathrm{a}}$) holds at most once at the beginning. For instance, consider two constructors $\mathsf{c}_0 : (\mathsf{B} \Rightarrow \mathsf{B}) \Rightarrow \mathsf{B}, \mathsf{c}_1 : \mathsf{B} \Rightarrow \mathsf{B}$ and a variable $x : \mathsf{B} \Rightarrow \mathsf{B}$. Then, $(x, \mathsf{B} \Rightarrow \mathsf{B}, \mathsf{B}) \trianglelefteq_{\mathrm{qa}} (\mathsf{c}_1\,(\mathsf{c}_0\,x), \mathsf{B}, \mathsf{B})$ holds by the condition 2', and we have $(x, \mathsf{B} \Rightarrow \mathsf{B}, \mathsf{B}) \trianglelefteq_{\mathrm{qa}} (\mathsf{c}_0\,x, \mathsf{B}, \mathsf{B}) \trianglelefteq_{\mathrm{a}} (\mathsf{c}_1\,(\mathsf{c}_0\,x), \mathsf{B}, \mathsf{B})$ in this case. As this example shows, if one has $(u, U, \mathsf{C}) \trianglelefteq_{\mathrm{qa}} (t, T, \mathsf{B})$ or $(u, U, \mathsf{C}) \trianglelefteq_{\mathrm{a}} (t, T, \mathsf{B})$ with $(u, U, \mathsf{C}) \neq (t, T, \mathsf{B})$, then $T = \mathsf{B}$ must hold by definition. That is why $\trianglelefteq_{\mathrm{qa}} \setminus \trianglelefteq_{\mathrm{a}}$ holds at most once: if U is a type of some negative argument of a constructor for C and we have $(u, U, \mathsf{C}) \trianglelefteq_{\mathrm{qa}} (t, T, \mathsf{B})$, then $U \neq \mathsf{C}$ must hold.

▶ **Lemma 15.** *Assume that a family* $(\Sigma^{\mathsf{c}}, \vec{\mathsf{B}^{\mathsf{c}}})_{\mathsf{c}\in\mathbb{C}}$ *of size functions is given. If* $(u, U, \mathsf{C}) \trianglelefteq_{\mathrm{a}} (t, T, \mathsf{B})$ *and* $t \in T$ *hold then* $u \in U$ *holds.*

Proof. The assertion is trivial if $(u, U, \mathsf{C}) = (t, T, \mathsf{B})$ holds, so assume that there are a tuple $(\mathsf{c}, \vec{t}, \vec{T}) \in \mathbb{C}^{\mathsf{B}}$ and an integer $k \in \{n^{\mathsf{c}} + 1, \dots, q^{\mathsf{c}}\}$ with $t = \mathsf{c}\,\vec{t}$, $T = \mathsf{B}$ and $(u, U, \mathsf{C}) \trianglelefteq_{\mathrm{a}} (t_k, T_k, \mathsf{B}^{\mathsf{c}}_k)$. Since $\mathsf{c}\,\vec{t} \in T = \mathsf{B}$ holds, we have $\mathsf{c}\,\vec{t} \in \mathcal{S}^{\mathsf{B}}_{\mathfrak{a}}$ for some $\mathfrak{a}$ which is zero or a successor ordinal. If $k > p^{\mathsf{c}}$ holds, then we immediately have $t_k \in T_k$ by definition. If $k \leq p^{\mathsf{c}}$ holds, then $t_k \in [\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{b}}]T_k$ holds for some $\mathfrak{b}$ with $\mathfrak{a} = \mathfrak{b} + 1$. By Proposition 3, we have

$$[\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{b}}]T_k \subseteq [\mathsf{B} : \mathcal{S}^{\mathsf{B}}_{\mathfrak{m}(\mathcal{S}^{\mathsf{B}})}]T_k = T_k,$$

so we have $t_k \in T_k$ also in this case. Therefore, we have $u \in U$ by IH. ◀

Size algebras enable us to annotate constructors and function symbols in a way which estimates the sizes of their inputs and outputs.

▶ **Definition 16** (Size Algebras). *A size algebra consists of*

- *a set* $\mathtt{A} = T(\mathtt{F}, \mathtt{V})$ *of* $\mathtt{F}$*-terms built from a set* $\mathtt{V}$ *of size variables* $\alpha, \beta, \dots$ *and a set* $\mathtt{F}$ *of size function symbols* $\mathtt{f}, \mathtt{g}, \dots$ *of fixed arity with* $\mathtt{V} \cap \mathtt{F} = \emptyset$,
- *a quasi ordering* $\leqslant_{\mathtt{A}}$ *on* $\mathtt{A}$ *and a strict ordering* $<_{\mathtt{A}} \subseteq \leqslant_{\mathtt{A}}$ *such that for each* $R \in \{\leqslant_{\mathtt{A}}, <_{\mathtt{A}}\}$ *and any substitution* $\varphi : \mathtt{V} \to \mathtt{A}$, *if* aRb *holds then* $a\varphi R b\varphi$ *holds,*

- *a function* $\mathtt{f}_{\mathfrak{h}} : \mathfrak{h}^n \to \mathfrak{h}$ *for each size function symbol* $\mathtt{f} \in \mathtt{F}$ *of arity* n *such that for any valuation* $\mu : \mathtt{V} \to \mathfrak{h}$*, if* $a \leqslant_{\mathtt{A}} b$ *(resp.* $a <_{\mathtt{A}} b$*) holds then* $a\mu \leq b\mu$ *(resp.* $a\mu < b\mu$*) holds, where* $\alpha\mu := \mu(\alpha)$ *and* $(\mathtt{f}\ a_1 \dots a_n)\mu := \mathtt{f}_{\mathfrak{h}}(a_1\mu, \dots, a_n\mu)$*.*

Size algebras are denoted by $\mathtt{A}$*. A size algebra* $\mathtt{A}$ *is monotone if and only if for any* $\mathtt{f} \in \mathtt{F}$*, if* $\vec{a}(\leqslant_{\mathtt{A}})_{\mathrm{prod}}\vec{b}$ *holds then* $\mathtt{f}\ \vec{a} \leqslant_{\mathtt{A}} \mathtt{f}\ \vec{b}$ *holds.*

The *successor size algebra* is the size algebra $\mathtt{A}$ which consists of $\mathtt{F}, <_{\mathtt{A}}, \leqslant_{\mathtt{A}}$ below:

- $\mathtt{F} = \mathtt{C} \cup \{\mathtt{s}\}$, where $\mathtt{C}$ is a fixed countably infinite set of constants and $\mathtt{s}$ is the unary function symbol. The interpretation $\mathtt{s}_{\mathfrak{h}}$ of $\mathtt{s}$ is the successor function on ordinals, and each constant in $\mathtt{C}$ is interpreted in a fixed way.
- $<_{\mathtt{A}}$ is defined by induction: $a <_{\mathtt{A}} \mathtt{s}\ a$ for any $a \in \mathtt{A}$, and if $a <_{\mathtt{A}} b$ then $\mathtt{s}\ a <_{\mathtt{A}} \mathtt{s}\ b$. The ordering $\leqslant_{\mathtt{A}}$ is defined as the reflexive closure of $<_{\mathtt{A}}$.

One can easily see that the successor size algebra actually satisfies the conditions for being a size algebra. We will use the successor size algebra in applying our termination criterion below (Examples 27 and 28).

The set $\mathbb{T}_{\mathtt{A}}$ of *annotated types* is defined by induction: (1) $\mathbb{T} \subseteq \mathbb{T}_{\mathtt{A}}$, (2) if $\mathsf{B} \in \mathbb{S}$ and $a \in \mathtt{A}$ hold then $\mathsf{B}_a \in \mathbb{T}_{\mathtt{A}}$ holds, (3) if $U, T \in \mathbb{T}_{\mathtt{A}}$ holds then $U \Rightarrow T \in \mathbb{T}_{\mathtt{A}}$ holds. We adopt the following notations:

- $\mathrm{Var}(e)$ means the set of all size variables occurring in an expression e,
- $|T|$ means the type obtained by removing all annotations in T,
- $\mathrm{Annot}(T, \mathsf{B}, a)$ means the annotated type obtained by annotating any occurrence of B in T by a.

In addition, we extend the positive and negative positions in types to annotated types:

- $\mathrm{Pos}^s(\mathsf{B}_b) := \{1p \mid p \in \mathrm{Pos}^s(b)\}$ for each $s \in \{+, -\}$,
- $\mathrm{Pos}^+(\alpha) := \{\epsilon\}$, $\mathrm{Pos}^-(\alpha) := \emptyset$,
- if $\mathtt{f}$ is of arity 0 then we define $\mathrm{Pos}^+(\mathtt{f}) := \{\epsilon\}$ and $\mathrm{Pos}^-(\mathtt{f}) := \emptyset$, otherwise we define

$$\mathrm{Pos}^s(\mathtt{f}b_1 \dots b_n) := \{ip \mid i \in \mathrm{Mon}^+(\mathtt{f}),\ p \in \mathrm{Pos}^s(b_i)\} \cup \{ip \mid i \in \mathrm{Mon}^-(\mathtt{f}),\ p \in \mathrm{Pos}^{-s}(b_i)\},$$

where $\mathrm{Mon}^+(\mathtt{f})$ (resp. $\mathrm{Mon}^-(\mathtt{f})$) is the set of arguments in which $\mathtt{f}$ is monotone (resp. anti-monotone) with respect to $\leqslant_{\mathtt{A}}$.

The *top-extension* of a size algebra $\mathtt{A}$ is the set $\overline{\mathtt{A}} = \mathtt{A} \cup \{\infty\}$ with $\infty \notin \mathtt{A}$. We define as follows:

- $\mathsf{B}_\infty := \mathsf{B}$ for any $\mathsf{B} \in \mathbb{S}$.
- For any $a, b \in \overline{\mathtt{A}}$, $a \leqslant^{\infty}_{\mathtt{A}} b$ holds if and only if either $a \leqslant_{\mathtt{A}} b$ or $b = \infty$ holds. In addition, $a <^{\infty}_{\mathtt{A}} b$ holds if and only if either $a <_{\mathtt{A}} b$ or $a \neq \infty = b$ holds.
- For any substitution $\varphi : \mathtt{V} \to \overline{\mathtt{A}}$ and any $a \in \overline{\mathtt{A}}$, $a\varphi := \infty$ if there is a variable $\alpha \in \mathrm{Var}(a)$ with $\varphi(\alpha) = \infty$, otherwise $a\varphi$ is defined as the usual substitution.

Below we denote elements of $\mathtt{V} \cup \{\infty\}$ by $\alpha, \beta, \dots$ as well. Then, *annotations for constructors* can be formulated as follows:

▶ **Definition 17** (Annotations for Constructors)**.** *Let* $\mathtt{A}$ *be an arbitrary size algebra. An annotation for constructors is a family* $C = (\vec{\mathsf{B}^{\mathsf{c}}}, \overline{\Theta}(\mathsf{c}))_{\mathsf{c} \in \mathbb{C}}$ *such that for any* $\mathsf{c} \in \mathbb{C}$ *with* $\Theta(\mathsf{c}) = T_1 \Rightarrow \dots \Rightarrow T_{r^{\mathsf{c}}} \Rightarrow \mathsf{B}$*,*

1. $\vec{\mathsf{B}^{\mathsf{c}}}$ *consists of the sorts* $\mathsf{B}^{\mathsf{c}}_1, \dots, \mathsf{B}^{\mathsf{c}}_{q^{\mathsf{c}}}$*, and*
   - *for any* $i \in \{1, \dots, p^{\mathsf{c}}\}$, $\mathsf{B}^{\mathsf{c}}_i = \mathsf{B}$ *holds, and*
   - *for any* $i \in \{p^{\mathsf{c}} + 1, \dots, q^{\mathsf{c}}\}$, $\mathsf{B}^{\mathsf{c}}_i$ *occurs in* T_i *with* $\mathrm{Pos}(\mathsf{B}^{\mathsf{c}}_i, T_i) \subseteq \mathrm{Pos}^+(\mathsf{B}^{\mathsf{c}}_i, T_i)$ *and* $\mathsf{B}^{\mathsf{c}}_i <_{\mathbb{S}} \mathsf{B}$*,*
2. $\overline{\Theta}(\mathsf{c}) = \overline{T_1} \Rightarrow \dots \Rightarrow \overline{T_{r^{\mathsf{c}}}} \Rightarrow \mathsf{B}_{\sigma^{\mathsf{c}}}$ *with* $\sigma^{\mathsf{c}} \in \overline{\mathtt{A}}$*, where* $\overline{T_i} = \mathrm{Annot}(T_i, \mathsf{B}^{\mathsf{c}}_i, \alpha^{\mathsf{c}}_i)$ *if* $i \in \{1, \dots, q^{\mathsf{c}}\}$*, otherwise* $\overline{T_i} = T$*.*

3. $\{\alpha^{\mathsf{c}}_1, \ldots, \alpha^{\mathsf{c}}_{p^{\mathsf{c}}}\} \subseteq \mathtt{V}$, $\{\alpha^{\mathsf{c}}_{p^{\mathsf{c}}+1}, \ldots, \alpha^{\mathsf{c}}_{q^{\mathsf{c}}}\} \subseteq \mathtt{V} \cup \{\infty\}$, *and the members of* $\{\alpha^{\mathsf{c}}_1, \ldots, \alpha^{\mathsf{c}}_{q^{\mathsf{c}}}\} \cap \mathtt{V}$ *are either pairwise equal or pairwise distinct,*
4. *if* $p^{\mathsf{c}} \neq 0$ *then* $\mathrm{Var}(\sigma^{\mathsf{c}}) \subseteq \{\alpha_1, \ldots, \alpha_{q^{\mathsf{c}}}\}$ *holds, otherwise* $\sigma^{\mathsf{c}} \in \mathtt{V} \setminus \{\alpha_1, \ldots, \alpha_{q^{\mathsf{c}}}\}$ *holds,*
5. $\mathrm{Pos}(\alpha^{\mathsf{c}}_i, \sigma^{\mathsf{c}}) \subseteq \mathrm{Pos}^{+}(\alpha^{\mathsf{c}}_i, \sigma^{\mathsf{c}})$ *for any* $i \in \{1, \ldots, q^{\mathsf{c}}\}$,
6. $\alpha^{\mathsf{c}}_i <^{\infty}_{\mathtt{A}} \sigma^{\mathsf{c}}$ *holds for any* $i \in \{1, \ldots, p^{\mathsf{c}}\}$.

We stipulate that an annotation for constructors determines the corresponding size functions in the following way:

▶ **Definition 18** (Size Functions by Annotation). *Let an annotation for constructors be given. For any constructor* c*, we define the size function* Σ^{c} *induced by this annotation as follows: put* $\vec{\alpha} := \alpha^{\mathsf{c}}_1, \ldots, \alpha^{\mathsf{c}}_{q^{\mathsf{c}}}$*, and suppose that* $\mathfrak{a}_1, \ldots, \mathfrak{a}_{q^{\mathsf{c}}}$ *are arbitrary ordinals less than* $\mathfrak{h}$*. For any* $\alpha \in \mathrm{Var}(\sigma^{\mathsf{c}})$*, we first define the valuation* ν *as*

$$\nu(\alpha) := \begin{cases} 0, & \text{if } \alpha \text{ is distinct from any of } \vec{\alpha}, \\ \mathfrak{a}_i, & \text{if } \alpha = \alpha^{\mathsf{c}}_i \text{ and the members of } \{\vec{\alpha}\} \cap \mathtt{V} \text{ are pairwise distinct}, \\ \mathfrak{b}, & \text{if } \alpha = \alpha^{\mathsf{c}}_i \text{ and the members of } \{\vec{\alpha}\} \cap \mathtt{V} \text{ are pairwise equal}, \end{cases}$$

where $\mathfrak{b} = \sup\{\mathfrak{a}_i \mid 1 \leq i \leq q^{\mathsf{c}},\ \alpha^{\mathsf{c}}_i \in \mathtt{V}\}$*. Then, we define*

$$\Sigma^{\mathsf{c}}(\mathfrak{a}_1, \ldots, \mathfrak{a}_{q^{\mathsf{c}}}) := \begin{cases} 0, & \text{if } \sigma^{\mathsf{c}} = \infty, \\ \sigma^{\mathsf{c}}\nu, & \text{otherwise.} \end{cases}$$

When an annotation of constructors is given, we have the size function Σ^{c} for any constructor c by the definition above. Then, by Definition 10, we obtain stratifications by size functions. By using these stratifications with a given valuation $\mu : \mathtt{V} \to \mathfrak{h}$, we interpret annotated types $T \in \mathbb{T}_{\mathtt{A}}$ as follows:

- $\mathsf{B}\mu := \mathcal{S}^{\mathsf{B}}_{\mathfrak{m}(\mathcal{S}^{\mathsf{B}})}$, and $\mathsf{B}_a\mu = \mathcal{S}^{\mathsf{B}}_{a\mu}$,
- $(U \Rightarrow V)\mu = U\mu \Rightarrow^{*} V\mu$.

▶ **Lemma 19.** *Assume that an annotation* C *for constructors is given.*

1. *Let* c *be an arbitrary constructor with* $p^{\mathsf{c}} \neq 0$*, and* ν, μ *be arbitrary valuations from* $\{\alpha_1, \ldots, \alpha_{q^{\mathsf{c}}}\}$ *to* $\mathfrak{h}$*. If* $\alpha_i\nu \leq \alpha_i\mu$ *holds for any* $i \in \{1, \ldots, q^{\mathsf{c}}\}$ *with* $\alpha_i \in \mathtt{V}$*, then* $\sigma^{\mathsf{c}}\nu \leq \sigma^{\mathsf{c}}\mu$ *holds.*
2. *Let* Σ^{c} *be the size function* c *induced by* C*. If* $\sigma^{\mathsf{c}} \neq \infty$ *holds, then* $\mathfrak{a}_i < \Sigma^{\mathsf{c}}(\vec{\mathfrak{a}})$ *holds for any* $\vec{\mathfrak{a}}$ *and any* $i \in \{1, \ldots, p^{\mathsf{c}}\}$.

Proof.

(1.) Since $p^{\mathsf{c}} \neq 0$ holds, we have $\mathrm{Var}(\sigma^{\mathsf{c}}) \subseteq \{\alpha_1, \ldots, \alpha_{q^{\mathsf{c}}}\}$. Then, the assertion follows because $\mathrm{Pos}(\alpha^{\mathsf{c}}_i, \sigma^{\mathsf{c}}) \subseteq \mathrm{Pos}^{+}(\alpha^{\mathsf{c}}_i, \sigma^{\mathsf{c}})$ holds for any $i \in \{1, \ldots, q^{\mathsf{c}}\}$ by the definition of C.

(2.) By the definition of C, if $\sigma \neq \infty$ holds then we have $\alpha^{\mathsf{c}}_i <_{\mathtt{A}} \sigma^{\mathsf{c}}$ for any $i \in \{1, \ldots, p^{\mathsf{c}}\}$. It follows from the definition of size algebras that $\alpha^{\mathsf{c}}_i\nu < \sigma^{\mathsf{c}}\nu$ holds where ν is the valuation defined in Definition 18, hence we have $\mathfrak{a}_i < \Sigma^{\mathsf{c}}(\vec{\mathfrak{a}})$ for any $\vec{\mathfrak{a}}$ and any $i \in \{1, \ldots, p^{\mathsf{c}}\}$. ◀

Without loss of generality, we may assume that for any $\mathsf{f} \in \mathbb{F}$, there is a natural number $q^{\mathsf{f}} \geq 0$ such that the first q^{f} arguments of f are all sorts. We denote the i-th argument of f by $\mathsf{B}^{\mathsf{f}}_i$ for any $i \in \{1, \ldots, q^{\mathsf{f}}\}$. Notice that there may be a natural number $i \in \{q^{\mathsf{f}} + 1, \ldots, r^{\mathsf{f}}\}$ with T_i a sort. Using these notations, we define *annotations for function symbols* as follows:

▶ **Definition 20** (Annotations for Function Symbols). *Let* $\mathtt{A}$ *be an arbitrary size algebra. An annotation* F *for function symbols consists of a well founded quasi ordering* $\leq_{\mathbb{F}}$ *on* $\mathbb{F} \cup \mathbb{C} \cup \mathbb{V}$ *and a family* $((\mathbb{D}^{\mathsf{f}}_{\mathtt{A}}, \leqslant^{\mathsf{f}}_{\mathtt{A}}, <^{\mathsf{f}}_{\mathtt{A}}, \zeta^{\mathsf{f}}_{\mathtt{A}}), (\mathbb{D}^{\mathsf{f}}_{\mathfrak{h}}, \leqslant^{\mathsf{f}}_{\mathfrak{h}}, <^{\mathsf{f}}_{\mathfrak{h}}, \zeta^{\mathsf{f}}_{\mathfrak{h}}), \overline{\Theta}(\mathsf{f}))_{\mathsf{f} \in \mathbb{F}}$ *which satisfy the following: for any* $\mathsf{f} \in \mathbb{F}$*,*

1. $h <_{\mathbb{F}} \mathsf{f}$ *holds for any* $h \in \mathbb{C} \cup \mathbb{V}$,
2. $\overline{\Theta}(\mathsf{f}) = \overline{T_1} \Rightarrow \cdots \Rightarrow \overline{T_{r^\mathsf{f}}} \Rightarrow \mathsf{B}_{\sigma^\mathsf{f}}$ *with* $\sigma^\mathsf{f} \in \overline{\mathtt{A}}$,
3. $\overline{T_i} = \mathrm{Annot}(\mathsf{B}^\mathsf{f}_i, \mathsf{B}^\mathsf{f}_i, \alpha^\mathsf{f}_i)$ *for any* $i \in \{1, \ldots, q^\mathsf{f}\}$, *and* $\overline{T_i} = T_i$ *for any* $i \in \{q^\mathsf{f}+1, \ldots, r^\mathsf{f}\}$,
4. $\mathrm{Var}(\sigma^\mathsf{f}) \subseteq \{\alpha^\mathsf{f}_1, \ldots, \alpha^\mathsf{f}_{q^\mathsf{f}}\} \subseteq \mathtt{V}$, *and* $\alpha^\mathsf{f}_1, \ldots, \alpha^\mathsf{f}_{q^\mathsf{f}}$ *are pairwise distinct,*
5. *for each* $X \in \{\mathtt{A}, \mathfrak{h}\}$, $\mathbb{D}^\mathsf{f}_X$ *is a set,* $\leqslant^\mathsf{f}_X$ *is a quasi ordering on* $\mathbb{D}^\mathsf{f}_X$, $<^\mathsf{f}_X$ *is a well founded relation with* $<^\mathsf{f}_X \subseteq \leqslant^\mathsf{f}_X$, *and* ζ^f_X *is a mapping from* X^{q^f} *to* $\mathbb{D}^\mathsf{f}_X$ *such that*
   - *if* $\mathsf{f} \simeq_{\mathbb{F}} \mathsf{g}$ *holds, then* $(\mathbb{D}^\mathsf{f}_X, \leqslant^\mathsf{f}_X, <^\mathsf{f}_X) = (\mathbb{D}^\mathsf{g}_X, \leqslant^\mathsf{g}_X, <^\mathsf{g}_X)$ *holds for each* $X \in \{\mathtt{A}, \mathfrak{h}\}$,
   - *if* $\vec{a} <^{\mathsf{g},\mathsf{f}}_{\mathtt{A}} \vec{b}$ *holds, then* $\vec{a}\mu <^{\mathsf{g},\mathsf{f}}_{\mathfrak{h}} \vec{b}\mu$ *for any valuation* $\mu : \mathtt{V} \to \mathfrak{h}$,
   - $<^{\mathsf{g},\mathsf{f}}_{\mathfrak{h}} \circ \leq_{\mathrm{prod}} \subseteq <^{\mathsf{g},\mathsf{f}}_{\mathfrak{h}}$,

   where $\circ$ *denotes the composition of two relations, and for each* $X \in \{\mathtt{A}, \mathfrak{h}\}$, $(x_1, \ldots, x_{q^\mathsf{g}}) <^{\mathsf{g},\mathsf{f}}_X (y_1, \ldots, y_{q^\mathsf{f}})$ *holds iff* $\mathsf{g} \simeq_{\mathbb{F}} \mathsf{f}$ *and* $\zeta^\mathsf{g}_X(x_1, \ldots, x_{q^\mathsf{g}}) <^\mathsf{f}_X \zeta^\mathsf{f}_X(y_1, \ldots, y_{q^\mathsf{f}})$ *hold.*

As an example from [9, Example 3], we consider the subtraction $\mathsf{sub} \in \mathbb{F}$ on natural numbers: we have $\Theta(\mathsf{sub}) = \mathsf{N} \Rightarrow \mathsf{N} \Rightarrow \mathsf{N}$, and the rewrite rules are

$$\mathsf{sub}\ x\ 0 \to x \qquad \mathsf{sub}\ 0\ y \to 0 \qquad \mathsf{sub}\ (\mathsf{s}\ x)(\mathsf{s}\ y) \to \mathsf{sub}\ x\ y$$

where $0 : \mathsf{N}$ (zero) and $\mathsf{s} : \mathsf{N} \Rightarrow \mathsf{N}$ (the successor function) are the constructors for N. Using the successor size algebra, we annotate $0, \mathsf{s}$ and sub as follows:

$$\overline{\Theta}(0) = \mathsf{N}_\alpha,\ \overline{\Theta}(\mathsf{s}) = \mathsf{N}_\beta \Rightarrow \mathsf{N}_{\mathsf{s}\beta},\ \overline{\Theta}(\mathsf{sub}) = \mathsf{N}_\gamma \Rightarrow \mathsf{N} \Rightarrow \mathsf{N}_\gamma \text{ with } q^{\mathsf{sub}} = 1 \text{ and } \alpha^{\mathsf{sub}} = \sigma^{\mathsf{sub}} = \gamma.$$

The annotation $\overline{\Theta}(\mathsf{sub})$ indicates that the size of $\mathsf{sub}\ x\ y$ does not increase from the size of x. For each $X \in \{\mathtt{A}, \mathfrak{h}\}$, we take $\mathbb{D}^{\mathsf{sub}}_X$ as X, and $\zeta^{\mathsf{sub}}_X : X \to \mathbb{D}^{\mathsf{sub}}_X$ as the identity function. In addition, $\leqslant^{\mathsf{sub}}_{\mathfrak{h}}$ and $<^{\mathsf{sub}}_{\mathfrak{h}}$ are defined as $\leq$ and $<$ on ordinals, respectively, while we put $\leqslant^{\mathsf{sub}}_{\mathtt{A}} := \leqslant_{\mathtt{A}}$ and $<^{\mathsf{sub}}_{\mathtt{A}} := <_{\mathtt{A}}$. The annotation with the identity function is often useful, as we will see below.

Notice that when both an annotation for constructors and an annotation of function symbols are given, we have a mapping $\overline{\Theta} : \mathbb{C} \cup \mathbb{F} \to \mathbb{T}_{\mathtt{A}}$. We adopt similar notations for $\overline{\Theta}$ to the ones for $\Theta : \mathbb{C} \cup \mathbb{F} \to \mathbb{T}$.

Below we define two orderings $<_{\mathtt{A}}$ and $<_{\mathfrak{h}}$ on function calls, where a function call is expressed as a pair (f, ξ) of a function symbol f and a substitution ξ with the domain $\{\alpha^\mathsf{f}_1, \ldots, \alpha^\mathsf{f}_{q^\mathsf{f}}\}$. Here the substitution ξ provides an instance of function calls by instantiating the variables $\alpha^\mathsf{f}_1, \ldots, \alpha^\mathsf{f}_{q^\mathsf{f}}$.

▶ **Definition 21** (The Orderings on Function Calls). *Let* $\mathsf{g}, \mathsf{f} \in \mathbb{F}$ *be the case. Assume that an annotation for function symbols is given and that for each* $X \in \{\mathtt{A}, \mathfrak{h}\}$, *two mappings* $\xi : \{\alpha^\mathsf{g}_1, \ldots, \alpha^\mathsf{g}_{q^\mathsf{g}}\} \to X$ *and* $\eta : \{\alpha^\mathsf{f}_1, \ldots, \alpha^\mathsf{f}_{q^\mathsf{f}}\} \to X$ *are given. Then,* $(\mathsf{g}, \xi) <_X (\mathsf{f}, \eta)$ *holds iff either* $\mathsf{g} <_{\mathbb{F}} \mathsf{f}$ *or* $(\alpha^\mathsf{g}_1\xi, \ldots, \alpha^\mathsf{g}_{q^\mathsf{g}}\xi) <^{\mathsf{g},\mathsf{f}}_X (\alpha^\mathsf{f}_1\eta, \ldots, \alpha^\mathsf{f}_{q^\mathsf{f}}\eta)$ *holds.*

The following type system gives a condition of the termination criterion below (see the condition **Subject Reduction and Decreasingness**). Roughly speaking, the rule (app-decr) guarantees that the size of any function call in $h\vec{t}$ is strictly less than (f, φ), and the rule (sub) with the subtyping rules enables us to reason about the subtype relation between annotated types.

▶ **Definition 22** (Typing Rules for the Computability Closure). *Assume that annotations for* $\mathbb{C}$ *and* $\mathbb{F}$ *are given, and let* f *be a function symbol with a symbolic valuation* $\varphi : \{\vec{\alpha}^\mathsf{f}\} \to \mathtt{A}$. *The typing rules for the computability closure of* (f, φ) *are as follows:*
Rule **(app-decr)**. *If the following conditions*

1. $(h, \vec{V} \Rightarrow V) \in \Gamma \cup \overline{\Theta}$,
2. *either* $h \simeq_{\mathbb{F}} \mathsf{f}$ *holds or* $h <_{\mathbb{F}} \mathsf{f}$ *holds,*
3. *either* $h \in \mathbb{F}$ *holds and* ψ *is a symbolic valuation from* $\{\vec{\alpha}^h\} \to \mathtt{A}$ *with* $(h, \psi) <_{\mathtt{A}} (\mathsf{f}, \varphi)$, *or* $h \in \mathbb{V} \cup \mathbb{C}$ *holds and* ψ *is the empty function,*

are satisfied, then the rule

$$\frac{\Gamma \vdash^{\mathsf{f}}_{\varphi} t_1 : V_1\psi \quad \cdots \quad \Gamma \vdash^{\mathsf{f}}_{\varphi} t_{|\vec{V}|} : V_{|\vec{V}|}\psi}{\Gamma \vdash^{\mathsf{f}}_{\varphi} h\vec{t} : V\psi}\ (\text{app-decr})$$

is a typing rule for the computability closure of (f, φ).

Rules **(lam)** ***and*** **(sub)**. *We have the following rules:*

$$\frac{\Gamma, x : U \vdash^{\mathsf{f}}_{\varphi} t : V}{\Gamma \vdash^{\mathsf{f}}_{\varphi} \lambda x^U t : U \Rightarrow V}\ (\text{lam}) \qquad \frac{\Gamma \vdash^{\mathsf{f}}_{\varphi} t : U \quad U \le V}{\Gamma \vdash^{\mathsf{f}}_{\varphi} t : V}\ (\text{sub})$$

where the subtyping rules are as follows:

$$\frac{a \le^{\infty}_{\mathtt{A}} b}{\mathsf{B}_a \le \mathsf{B}_b}\ (\text{size}) \quad \frac{U' \le U \quad V \le V'}{U \Rightarrow V \le U' \Rightarrow V'}\ (\text{prod}) \quad \frac{}{U \le U}\ (\text{refl}) \quad \frac{U \le V \quad V \le T}{U \le T}\ (\text{tran})$$

▶ Remark 23. In [9], the rule (app-decr) has one more condition which states that h is applied to at least q^h arguments whenever $h \in \mathbb{F}$ and $h \simeq_{\mathbb{F}} \mathsf{f}$ holds. That is, the rule in [9] is obtained from the rule (app-decr) above by replacing the clause 2. with the clause

2'. either $h \simeq_{\mathbb{F}} \mathsf{f}$ and $|\vec{V}| \ge q^h$ holds or $h <_{\mathbb{F}} \mathsf{f}$ holds.

In fact, this condition is not needed to prove the correctness of the termination criterion in [9]. If this condition is dropped, then the criterion in [9] guarantees the termination of Hofmann's extract function for the breadth-first traversal of trees (see, e.g., [19]) as our termination criterion does. We will show in Example 27 below that our criterion guarantees the termination of Hofmann's extract function. If one keeps the condition 2', then it is, to the best of our knowledge, an open question whether the termination of Hofmann's extract function can be shown by the criterion in [9].

As an illustration of typing rules for the computability closure, we consider the annotated rewrite system of sub which we have seen above (see also [9, Example 3]). When the symbolic valuations ψ and φ are defined as $\psi := \{(\gamma, \beta)\}$ and $\varphi := \{(\gamma, \mathsf{s}\ \beta)\}$, respectively, then $(\mathsf{sub}, \psi) <_{\mathtt{A}} (\mathsf{sub}, \varphi)$ holds. Thus we have, e.g.,

$$\frac{\dfrac{x : \mathsf{N}_\beta, y : \mathsf{N} \vdash^{\mathsf{sub}}_{\varphi} x : \mathsf{N}_\beta (= \mathsf{N}_\gamma\psi) \quad x : \mathsf{N}_\beta, y : \mathsf{N} \vdash^{\mathsf{sub}}_{\varphi} y : \mathsf{N} (= \mathsf{N}\psi)}{x : \mathsf{N}_\beta, y : \mathsf{N} \vdash^{\mathsf{sub}}_{\varphi} \mathsf{sub}\ x\ y : \mathsf{N}_\beta}\ (\text{app-decr}) \quad \dfrac{\beta \le^{\infty}_{\mathtt{A}} \mathsf{s}\ \beta}{\mathsf{N}_\beta \le \mathsf{N}_{\mathsf{s}\beta}}\ (\text{size})}{x : \mathsf{N}_\beta, y : \mathsf{N} \vdash^{\mathsf{sub}}_{\varphi} \mathsf{sub}\ x\ y : \mathsf{N}_{\mathsf{s}\beta}}\ (\text{sub})$$

When an annotation C for constructors and an annotation F for function symbols are given, we denote the set of all size variables used in C by $\mathbb{V}(C)$, and the set of all size variables used in F by $\mathbb{V}(F)$.

In sum, if one provides an annotation for constructors then this annotation determines a family of size functions, and this family determines a stratification as we have seen in Section 3. If one also gives an annotation for function symbols, one obtains the typing rules above for each function symbol f and symbolic valuation φ. Then, in order to prove the termination of a given rewrite relation $\to$, it suffices to show that our termination criterion below is satisfied.

The difference between our termination criterion below and the criterion of [9] consists in the condition (3) of **Accessibility** and the condition (2b) of **Minimality**. Both of these conditions are added in order to deal with constructors of non-positive types. In particular, the condition (3) of **Accessibility** includes the quasi-accessibility, which was defined above (Definition 14).

▶ **Theorem 24** (Termination Criterion). *Let C be an annotation for constructors and $(\Sigma^{\mathsf{c}})_{\mathsf{c}\in\mathbb{C}}$ be the family of size functions induced by C such that for any c and any $\vec{\mathfrak{a}}$, $\Sigma^{\mathsf{c}}(\vec{\mathfrak{a}})$ is not a limit ordinal. Moreover, let an annotation F for function symbols be given. Then, the rewrite relation $\to$ terminates on the set of all well-typed terms if for each rule $\mathsf{f}\,l_1\cdots l_{|\vec{l}|}\to r\in\mathcal{R}$ with $\overline{\Theta}(\mathsf{f})=\overline{T_1}\Rightarrow\cdots\Rightarrow\overline{T_{r^{\mathsf{f}}}}\Rightarrow\mathsf{B}_{\sigma^{\mathsf{f}}}$,*

- *$|\vec{l}|\geq q^{\mathsf{f}}$ holds,*
- *there is a typing environment $\Gamma:\mathrm{FV}(r)\to\mathbb{T}_{\mathtt{A}}$ satisfying the following: for any $(x,U)\in\Gamma$, there are a term l_k $(1\leq k\leq|\vec{l}|)$, a sort B^x and a size variable $\alpha^x\in\mathbb{V}(C)\cup\mathbb{V}(F)$ with $\mathrm{Pos}(x,l_k)\neq\emptyset$ and $U=\mathrm{Annot}(|U|,\mathsf{B}^x,\alpha^x)$,*
- *there is a substitution $\varphi:\{\alpha_1,\dots,\alpha_{q^{\mathsf{f}}}\}\to\mathtt{A}$*

such that the following conditions are satisfied:

Monotony. *For any $i\in\{1,\dots,q^{\mathsf{f}}\}$, $\mathrm{Pos}(\alpha_i^{\mathsf{f}},\sigma^{\mathsf{f}})\subseteq\mathrm{Pos}^+(\alpha_i^{\mathsf{f}},\sigma^{\mathsf{f}})$ holds.*

Accessibility. *For any $(x,U)\in\Gamma$,*

1. *$x=l_k$ and $U=\overline{T_k}\varphi$ holds, or*
2. *$\mathrm{Pos}(\mathsf{B}^x,|U|)\subseteq\mathrm{Pos}^+(\mathsf{B}^x,|U|)$ holds, T_k is a sort and $(x,|U|,\mathsf{B}^x)\trianglelefteq_{\mathrm{a}}(l_k,T_k,T_k)$, or*
3. *$\mathrm{Pos}(\mathsf{B}^x,|U|)\not\subseteq\mathrm{Pos}^+(\mathsf{B}^x,|U|)$ holds, T_k is a sort and $(x,|U|,\mathsf{B}^x)\trianglelefteq_{\mathrm{qa}}(l_k,T_k,T_k)$.*

Minimality. *For any substitution θ such that $l_j\theta\in T_j$ holds for any $j\in\{1,\dots,|\vec{l}|\}$, there is an ordinal valuation ν satisfying*

1. *for any $i\in\{1,\dots,q^{\mathsf{f}}\}$, $\alpha_i^{\mathsf{f}}\varphi\nu=o_{\mathcal{S}^{\mathsf{B}_i}}(l_i\theta)$ holds, and*
2. *for any $(x,U)\in\Gamma$,*
   a. *if $\mathrm{Pos}(\mathsf{B}^x,|U|)\subseteq\mathrm{Pos}^+(\mathsf{B}^x,|U|)$ holds, then $o_{[\mathsf{B}^x:\mathcal{S}^{\mathsf{B}^x}]|U|}(x\theta)\leq\alpha^x\nu$ holds,*
   b. *otherwise $o_{(\bigcup_{\mathfrak{c}\leq\mathfrak{a}}[\mathsf{B}^x:\mathcal{S}^{\mathsf{B}^x}_{\mathfrak{c}}]|U|)_{\mathfrak{a}<\mathfrak{h}}}(x\theta)=\alpha^x\nu$ holds.*

Subject Reduction and Decreasingness. $\Gamma\vdash^{\mathsf{f}}_{\varphi}r:T_{|\vec{l}|+1}\Rightarrow\cdots\Rightarrow T_{r^{\mathsf{f}}}\Rightarrow\mathsf{B}_{\sigma^{\mathsf{f}}\varphi}$ *holds.*

Proof. Below we show the computability of constructors, the computability of function symbols and the computability of well-typed terms one by one. We first prove the computability of constructors: for any $\mathsf{c}\in\mathbb{C}$ with $\overline{\Theta}(\mathsf{c})=\overline{T_1}\Rightarrow\cdots\Rightarrow\overline{T_{r^{\mathsf{c}}}}\Rightarrow\mathsf{B}_{\sigma^{\mathsf{c}}}$, any $\mu:\mathtt{V}\to\mathfrak{h}$ and any $\vec{t}$ with $|\vec{t}|=r^{\mathsf{c}}$, if $t_i\in\overline{T_i}\mu$ holds for any i with $1\leq i\leq r^{\mathsf{c}}$, then $\mathsf{c}\,\vec{t}\in\mathsf{B}_{\sigma^{\mathsf{c}}}\mu$ holds. Here we need to consider constructors of non-positive types, which were not handled by [9]. Let $\mathsf{c}\in\mathbb{C}$ be the case with $\overline{\Theta}(\mathsf{c})=\overline{T_1}\Rightarrow\cdots\Rightarrow\overline{T_{r^{\mathsf{c}}}}\Rightarrow\mathsf{B}_{\sigma^{\mathsf{c}}}$. In addition, let $\mu:\mathtt{V}\to\mathfrak{h}$ be an arbitrary valuation, and assume that $t_i\in\overline{T_i}\mu$ holds for any i with $1\leq i\leq r^{\mathsf{c}}$. Putting $\vec{t}:=t_1,\dots,t_{r^{\mathsf{c}}}$, we verify $\mathsf{c}\,\vec{t}\in\mathsf{B}_{\sigma^{\mathsf{c}}}\mu$. It is obvious that $\mathsf{c}\,\vec{t}\in\mathrm{SN}$ holds. If $p^{\mathsf{c}}=0$ holds then we have $\mathsf{c}\,\vec{t}\in\mathcal{S}^{\mathsf{B}}_0$ by Definition 10, hence $\mathsf{c}\,\vec{t}\in\mathsf{B}_{\sigma^{\mathsf{c}}}\mu$ follows. Below we assume that $p^{\mathsf{c}}\neq0$ holds.

Consider an arbitrary $(\mathsf{c},\vec{u},\vec{T})\in\mathbb{C}^{\mathsf{B}}_{\to^*}(\mathsf{c}\,\vec{t})$. If $i\in\{1,\dots,n^{\mathsf{c}}\}$ holds, then we have $u_i\in\overline{T_i}\mu=[\mathsf{B}_{\alpha_i}:\mathcal{S}^{\mathsf{B}}_{\alpha_i\mu}]\overline{T_i}\mu\subseteq\bigcup_{\mathfrak{c}\leq\alpha_i\mu}[\mathsf{B}:\mathcal{S}^{\mathsf{B}}_{\mathfrak{c}}]T_i$, because $t_i\to^*u_i$ holds and $\overline{T_i}\mu$ is a computability predicate. Similarly, for each $i\in\{n^{\mathsf{c}}+1,\dots,q^{\mathsf{c}}\}$, we have $u_i\in\overline{T_i}\mu=[\mathsf{B}_i:\mathcal{S}^{\mathsf{B}_i}_{\alpha_i\mu}]T_i$. Put $\mathfrak{a}$ as $\mathfrak{a}:=\max(\{\Sigma^{\mathsf{c}}(o_{\mathcal{S}^{\mathsf{c},1}}(u_1),\dots,o_{\mathcal{S}^{\mathsf{c},q^{\mathsf{c}}}}(u_{q^{\mathsf{c}}}))\}\cup\{\alpha_i\mu\mid1\leq i\leq q^{\mathsf{c}},\ \alpha_i\in\mathtt{V}\})$. Then, $u_i\in\bigcup_{\mathfrak{c}\leq\mathfrak{a}}[\mathsf{B}:\mathcal{S}^{\mathsf{B}}_{\mathfrak{c}}]T_i$ holds for any $i\in\{1,\dots,n^{\mathsf{c}}\}$, and $u_i\in[\mathsf{B}_i:\mathcal{S}^{\mathsf{B}_i}_{\mathfrak{a}}]T_i$ holds for any $i\in\{n^{\mathsf{c}}+1,\dots,q^{\mathsf{c}}\}$ by Proposition 3. Therefore, $\mathsf{c}\,\vec{t}\in\mathcal{S}^{\mathsf{B}}_{\mathfrak{a}+1}$ holds and $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t})$ exists.

If $\sigma^{\mathsf{c}}=\infty$ holds then we immediately have $\mathsf{c}\,\vec{t}\in\mathsf{B}_{o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t})}\subseteq\mathsf{B}_{\sigma^{\mathsf{c}}\mu}$, so let $\sigma^{\mathsf{c}}\neq\infty$ be the case. By Lemma 13.(2), we have $o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{c}\,\vec{t})=\Sigma(o_{\mathcal{S}}(\vec{t}))=\sigma^{\mathsf{c}}\nu$, where ν is the valuation defined in Definition 18. If $\alpha_i\nu\leq\alpha_i\mu$ holds for any $\alpha_i\in\mathtt{V}\cap\{\vec{\alpha}\}$, then it follows from $\mathrm{Var}(\sigma^{\mathsf{c}})\subseteq\{\vec{\alpha}\}$ and Lemma 19.(1) that $\sigma^{\mathsf{c}}\nu\leq\sigma^{\mathsf{c}}\mu$ holds. We show that $\alpha_i\nu\leq\alpha_i\mu$ holds for any $\alpha_i\in\mathtt{V}\cap\{\vec{\alpha}\}$.

Since we consider the case of $\sigma^{\mathsf{c}} \neq \infty$, $\{\vec{\alpha}\} \cap \mathtt{V}$ must be non-empty. If the members of $\{\vec{\alpha}\} \cap \mathtt{V}$ are pairwise distinct, we have $\alpha_i\nu = o_{\mathcal{S}^i}(t_i) \leq \alpha_i\mu$ because $t_i \in \overline{T_i}\mu$ holds. Assume that all members of $\{\vec{\alpha}\} \cap \mathtt{V}$ are equal. We have

$$\alpha_i\nu = \sup\{o_{\mathcal{S}^j}(t_j) \mid 1 \leq j \leq q^{\mathsf{c}},\ \alpha_j \in \mathtt{V}\} = o_{\mathcal{S}^k}(t_k)$$

for some k with $1 \leq k \leq q^{\mathsf{c}}$ by definition. Therefore, $\alpha_i\nu = \alpha_k\nu \leq \alpha_k\mu = \alpha_i\mu$ holds because we have $t_k \in \overline{T_k}\mu$.

Next, we show the computability of function symbols: we verify that for any $\mathsf{f} \in \mathbb{F}$ with $\overline{\Theta}(\mathsf{f}) = \overline{T_1} \Rightarrow \cdots \overline{T_{r^{\mathsf{f}}}} \Rightarrow \mathsf{B}_{\sigma^{\mathsf{f}}}$, any $\mu : \{\alpha_1^{\mathsf{f}}, \ldots, \alpha_{q^{\mathsf{f}}}^{\mathsf{f}}\} \to \mathfrak{h}$ and any $\vec{t}$ with $|\vec{t}| = r^{\mathsf{f}}$, if $t_i \in \overline{T_i}\mu$ holds for any i, then $\mathsf{f}\,\vec{t} \in \mathsf{B}_{\sigma^{\mathsf{f}}}\mu$ holds. We show this claim by induction on $((\mathsf{f}, \mu), \vec{t})$ with $(<_{\mathfrak{h}}, \leftarrow_{\mathrm{prod}})_{\mathrm{lex}}$. Let $t_i \in \overline{T_i}\mu$ be the case for any i with $1 \leq i \leq r^{\mathsf{f}}$. Since $\mathsf{f}\,\vec{t}$ is neutral, it suffices to show that $u \in \mathsf{B}_{\sigma^{\mathsf{f}}}\mu$ holds for any u with $\mathsf{f}\,\vec{t} \to u$. The case of $u = \mathsf{f}\,\vec{u}$ and $\vec{t} \to_{\mathrm{prod}} \vec{u}$ is straightforward. Otherwise, we have

(∗) $\mathsf{f}\,\vec{l} \to r \in \mathcal{R}$, $\vec{t} = \vec{l}\theta\vec{u}$ and $u = r\theta\vec{u}$,

where $\vec{u} = t_{|\vec{l}|+1}, \ldots, t_{r^{\mathsf{f}}}$ holds. Recall that if $i \leq q^{\mathsf{f}}$ holds then T_i is a sort and that if $i > q^{\mathsf{f}}$ holds then $\overline{T_i} = T_i$ holds. Therefore, for any $i \in \{1, \ldots, |\vec{l}|\}$, we have $l_i\theta \in T_i$ by $l_i\theta \in \overline{T_i}\mu$. We obtain the following valuation ν by Minimality: for any $i \in \{1, \ldots, q^{\mathsf{f}}\}$, $\alpha_i^{\mathsf{f}}\varphi\nu = o_{\mathcal{S}^{\mathsf{B}_i}}(l_i\theta)$ holds, and for any $(x, U) \in \Gamma$,

1. if $\mathrm{Pos}(\mathsf{B}^x, |U|) \subseteq \mathrm{Pos}^+(\mathsf{B}^x, |U|)$ holds, then $o_{[\mathsf{B}^x:\mathcal{S}^{\mathsf{B}^x}]|U|}(x\theta) \leq \alpha^x\nu$ holds,
2. otherwise $o_{(\bigcup_{\mathfrak{c} \leq \mathfrak{a}} [\mathsf{B}^x:\mathcal{S}^{\mathsf{B}^x}_{\mathfrak{c}}]|U|)_{\mathfrak{a}<\mathfrak{h}}}(x\theta) = \alpha^x\nu$ holds.

We prove the following claims 1–3. The claim 3 will be proved in the same way as [9, Theorem 1]. On the other hand, the claim 1 is proved without a condition on (app-decr) which was imposed in [9] (see Remark 23 above), and the claim 2 forces us to consider the new case where B^x occurs in $|U|$ negatively for some $(x, U) \in \Gamma$.

1. **Correctness of the computability closure**: let ν be the valuation given above by Minimality, and φ be the substitution given by the assumptions of this theorem. Using subinduction on $\vdash^{\mathsf{f}}_{\varphi}$, we show that for any typing environment Δ, any term t, any type T and any substitution θ', if (i) $\Delta \vdash^{\mathsf{f}}_{\varphi} t : T$ holds and (ii) $x\theta' \in U\nu$ holds for any $(x, U) \in \Delta$, then $t\theta' \in T\nu$ holds.

   We consider the principal case (app-decr) only: let $\Delta \vdash^{\mathsf{f}}_{\varphi} h\vec{w} : V\psi$ be the case, then we have $w_i\theta' \in V_i\psi\nu$ for any i with $1 \leq i \leq |\vec{V}|$ by the hypothesis of subinduction. We put $k := |\vec{V}|$. If $h \in \mathbb{V}$ holds, then $\psi = \emptyset$ holds and we have $h\theta' \in (\vec{V} \Rightarrow V)\nu$ by assumption, hence we have $h\theta'\vec{w}\theta' \in V\psi\nu$. Let $h \in \mathbb{C}$ be the case, and put $V = (U_1 \Rightarrow \cdots \Rightarrow U_m \Rightarrow \mathsf{C}_{\sigma^{\mathsf{c}}})$. To show $h\vec{w}\theta' \in V\psi\nu$, consider arbitrary $u_1 \in U_1\psi\nu, \ldots, u_m \in U_m\psi\nu$. By the computability of constructors, we have $h\vec{w}\theta'\vec{u} \in \mathsf{C}_{\sigma^{\mathsf{c}}}\psi\nu$, hence $h\vec{w}\theta' \in V\psi\nu$ holds by definition. Let $h \in \mathbb{F}$ be the case, and assume that

   $$V = (U_1 \Rightarrow \cdots \Rightarrow U_m \Rightarrow \mathsf{C}_{\sigma^h}), \quad \vec{\alpha}^h = \alpha_1^h, \ldots, \alpha_{q^h}^h, \quad \vec{\alpha}^{\mathsf{f}} = \alpha_1^{\mathsf{f}}, \ldots, \alpha_{q^{\mathsf{f}}}^{\mathsf{f}}$$

   holds. If $h <_{\mathbb{F}} \mathsf{f}$ holds, then $(h, \psi\nu) <_{\mathfrak{h}} (\mathsf{f}, \mu)$ immediately follows from Definition 21. Otherwise, we have $\vec{\alpha}^h\psi <^{h,\mathsf{f}}_{\mathtt{A}} \vec{\alpha}^{\mathsf{f}}\varphi$ by $(h, \psi) <_{\mathtt{A}} (\mathsf{f}, \varphi)$. Then, by Definition 20, we have $\vec{\alpha}^h\psi\nu <^{h,\mathsf{f}}_{\mathfrak{h}} \vec{\alpha}^{\mathsf{f}}\varphi\nu$. By the minimality of ν, we have $\vec{\alpha}^{\mathsf{f}}\varphi\nu \leq_{\mathrm{prod}} \vec{\alpha}^{\mathsf{f}}\mu$, so we have $\vec{\alpha}^h\psi\nu <^{h,\mathsf{f}}_{\mathfrak{h}} \vec{\alpha}^{\mathsf{f}}\mu$ by Definition 20 again. Therefore, $(h, \psi\nu) <_{\mathfrak{h}} (\mathsf{f}, \mu)$ holds also in this case. By the hypothesis of the main induction, for any $\vec{t}$ with $\vec{t} = t_1, \ldots, t_k, t_{k+1}, \ldots, t_{k+m}$, if $t_i \in V_i\psi\nu$ holds for any i with $1 \leq i \leq k$ and $t_{k+j} \in U_j\psi\nu$ holds for any j with $1 \leq j \leq m$, then $h\vec{t} \in \mathsf{C}_{\sigma^h}\psi\nu$ holds. Therefore, we have $h\vec{w}\theta'\vec{u} \in \mathsf{C}_{\sigma^h}\psi\nu$ for any $u_1 \in U_1\psi\nu, \ldots, u_m \in U_m\psi\nu$, hence $h\vec{w}\theta' \in V\psi\nu$ holds by definition.

2. **Computability of the matching substitution**: we show that $x\theta \in U\nu$ holds for any $(x, U) \in \Gamma$, where θ is the substitution given in $(*)$ and $\Gamma : \mathrm{FV}(r) \to \mathbb{T}_{\mathtt{A}}$ is the typing environment given by the assumptions of the theorem. Consider an arbitrary $(x, U) \in \Gamma$, then there is an integer k, a term l_k, a sort B^x and a size variable α^x such that all of them satisfy the assumptions of the theorem. If $\mathrm{Pos}(\mathsf{B}^x, |U|) \subseteq \mathrm{Pos}^+(\mathsf{B}^x, |U|)$ holds then we can prove the assertion as in [9, Theorem 1] by using Lemma 15 and Accessibility, so assume that B^x occurs in $|U|$ negatively. If $x = l_k$ holds, then k must be greater than q^{f} because $|U|$ is not a sort, hence we have

$$|U| = U = T_k = \overline{T_k}$$

and so $x\theta = l_k\theta \in \overline{T_k}\mu = U\nu$ holds. Let $x \neq l_k$ be the case, and suppose that the position of $x\theta$ in $l_k\theta$ is $p_1 \cdots p_i$ $(i \geq 1)$. By Accessibility, we have $(x, |U|, \mathsf{B}^x) \trianglelefteq_{\mathrm{qa}} (l_k, T_k, T_k)$. Therefore, for any subterm t of $l_k\theta$ whose position in $l_k\theta$ is ϵ or $p_1 \cdots p_j$ for some j with $0 \leq j < i$, t has the form $\mathsf{c}\,\vec{w}$ for some $\mathsf{c} \in \mathbb{C}$ and some $\vec{w}$. Let $\mathsf{c}_0\,\vec{w}_0, \mathsf{c}_1\,\vec{w}_1, \ldots, \mathsf{c}_{i-1}\,\vec{w}_{i-1}$ be the subterms of positions $\epsilon, p_1, \ldots, p_1 \cdots p_{i-1}$ in $l_k\theta$, respectively (hence $\mathsf{c}_0\,\vec{w}_0 = l_k\theta$). We put $\mathsf{c} := \mathsf{c}_{i-1}$ if $i - 1 \neq 0$, otherwise we put $\mathsf{c} := \mathsf{c}_0$. Then, since $l_k\theta \in \overline{T_k}\mu$ holds, we have $x\theta \in \bigcup_{\mathsf{c} \leq \mathfrak{a}} [\mathsf{B}^{\mathsf{c}} : \mathcal{S}^{\mathsf{B}^{\mathsf{c}}}_{\mathsf{c}}]|U|$ for some $\mathfrak{a}$ by applying Definition 10 repeatedly. By the minimality of ν, we have $x\theta \in U\nu$.
3. We show that $u \in \mathsf{B}_{\sigma^{\mathsf{f}}}\mu$ holds. The claims 1. and 2. above show $r\theta \in \overline{T_{|l|+1}}\varphi\nu \Rightarrow \cdots \Rightarrow \overline{T_{r^{\mathsf{f}}}}\varphi\nu \Rightarrow \mathsf{B}_{\sigma^{\mathsf{f}}}\varphi\nu$. By definition, we have $\overline{T_i} = T_i$ for any i with $|l| + 1 \leq i \leq r^{\mathsf{f}}$, so we have $r\theta\vec{u} \in \mathsf{B}_{\sigma^{\mathsf{f}}}\varphi\nu$. If $\sigma^{\mathsf{f}} = \infty$ holds then we immediately have $\mathsf{B}_{\sigma^{\mathsf{f}}}\varphi\nu = \mathsf{B}_{\sigma^{\mathsf{f}}}\mu$, so let $\sigma^{\mathsf{f}} \neq \infty$ be the case. Since φ does not assign ∞ to any of $\alpha_1, \ldots, \alpha_{q^{\mathsf{f}}}$, we have $\sigma^{\mathsf{f}}\varphi \neq \infty$, so $\nu(\sigma^{\mathsf{f}}\varphi)$ is defined. By Monotony, it suffices to verify $\alpha_i\varphi\nu \leq \alpha_i\mu$ for any i with $1 \leq i \leq q^{\mathsf{f}}$, but this follows from Minimality.

Finally, following [9, Theorem 1], we show the computability of well-typed terms by induction on $\vdash$: we show that for any Γ, t, T with $\Gamma \vdash t : T$ and any substitution θ, if $x\theta \in U$ holds for any $(x, U) \in \Gamma$, then $t\theta \in T$ holds. The termination of $\to$ follows from the computability of well-typed terms: if $\Gamma \vdash t : T$ holds, then we consider the empty substitution θ. Since any computability predicate subsumes the set $\mathbb{V}$ of variables, we have $x\theta = x \in U$ for any $(x, U) \in \Gamma$, hence $t \in T \subseteq \mathrm{SN}$ holds.

The case of function symbols: by assumption, we have $\vdash \mathsf{f} : \Theta(\mathsf{f})$ with $\Theta(\mathsf{f}) = T_1 \Rightarrow \cdots \Rightarrow T_{r^{\mathsf{f}}} \Rightarrow \mathsf{B}$ and $\overline{\Theta}(\mathsf{f}) = \overline{T_1} \Rightarrow \cdots \Rightarrow \overline{T_{r^{\mathsf{f}}}} \Rightarrow \mathsf{B}_{\sigma^{\mathsf{f}}}$. It suffices to verify that $\mathsf{f}\,\vec{t} \in \mathsf{B}$ for any $\vec{t} \in \vec{T}$ with $|\vec{t}| = r^{\mathsf{f}}$. Define the valuation $\mu : \{\alpha_1, \ldots, \alpha_{q^{\mathsf{f}}}\} \to \mathfrak{h}$ as $\mu(\alpha_i) := \mathfrak{m}(\mathcal{S}^{\mathsf{B}_i})$. Since $T_i = \mathsf{B}_i$ holds for any $i \in \{1, \ldots, q^{\mathsf{f}}\}$ and $\overline{T_j} = T_j$ holds for any $j \in \{q^{\mathsf{f}} + 1, \ldots, r^{\mathsf{f}}\}$, we have $t_k \in \overline{T_k}\mu$ for any $k \in \{1, \ldots, r^{\mathsf{f}}\}$. By the computability of function symbols, we have $\mathsf{f}\,\vec{t} \in \mathsf{B}_{\sigma^{\mathsf{f}}}\mu \subseteq \mathsf{B}$.

The case of constructors: by assumption, we have $\vdash \mathsf{c} : \Theta(\mathsf{c})$ with $\Theta(\mathsf{c}) = T_1 \Rightarrow \cdots \Rightarrow T_{r^{\mathsf{c}}} \Rightarrow \mathsf{B}$ and $\overline{\Theta}(\mathsf{c}) = \overline{T_1} \Rightarrow \cdots \Rightarrow \overline{T_{r^{\mathsf{c}}}} \Rightarrow \mathsf{B}_{\sigma^{\mathsf{c}}}$. It suffices to verify that $\mathsf{c}\,\vec{t} \in \mathsf{B}$ for any $\vec{t} \in \vec{T}$ with $|\vec{t}| = r^{\mathsf{c}}$. Define the valuation $\mu : \{\alpha_1, \ldots, \alpha_{q^{\mathsf{c}}}\} \to \mathfrak{h}$ as in the previous paragraph. Let $i \in \{1, \ldots, q^{\mathsf{c}}\}$ be the case. We have

$$t_i \in T_i = [\mathsf{B}_i : \mathsf{B}_i]T_i = [\mathsf{B}_i : \mathcal{S}^{\mathsf{B}_i}_{\mu(\alpha_i)}]T_i,$$

hence $t_i \in \overline{T_i}\mu$ holds. Therefore, we have $t_k \in \overline{T_k}\mu$ for any $k \in \{1, \ldots, r^{\mathsf{c}}\}$. By the computability of constructors, we have $\mathsf{c}\,\vec{t} \in \mathsf{B}_{\sigma}\mu \subseteq \mathsf{B}$. The proofs of the other cases are standard. ◀

Below we discuss several examples of rewrite systems. The first and the second (Examples 25 and 26) are examples of non-terminating rewrite systems with a rule for non-positive types. The third and the fourth (Examples 27 and 28) contain a non-strictly positive inductive type and a non-positive type, respectively. We show that the first two examples cannot satisfy our termination criterion, and that the last two examples satisfy the criterion.

▶ **Example 25.** We consider the following non-terminating rewrite system with a non-positive type B, which was discussed in [7]: put $\mathbb{S} := \{\mathsf{B}, \mathsf{A}\}$, $\mathbb{C} := \{\mathsf{c}\}$ and $\mathbb{F} := \{\mathsf{p}\}$ with $\Theta(\mathsf{c}) = (\mathsf{B} \Rightarrow \mathsf{A}) \Rightarrow \mathsf{B}$ and $\Theta(\mathsf{p}) = \mathsf{B} \Rightarrow \mathsf{B} \Rightarrow \mathsf{A}$. Define the rewrite system $\mathcal{R}$ as $\mathcal{R} := \{\mathsf{p}\,(\mathsf{c}\,x) \to x\}$ with $x \in \mathbb{V}$. If we put $\omega := \lambda y^{\mathsf{B}}\mathsf{p}\,y\,y$, then we have

$$\omega(\mathsf{c}\,\omega) \to_\beta \mathsf{p}\,(\mathsf{c}\,\omega)(\mathsf{c}\,\omega) \to_{\mathcal{R}} \omega(\mathsf{c}\,\omega) \to_\beta \cdots$$

so this rewrite system $\mathcal{R}$ is non-terminating. The system $\mathcal{R}$ cannot satisfy our termination criterion: suppose that $\mathcal{R}$ satisfies it. Then, $q^{\mathsf{p}} \leq 1$ holds and so we have $\Gamma \vdash^{\mathsf{p}}_{\varphi} x : \mathsf{B} \Rightarrow \mathsf{A}_\sigma\varphi$ by Subject Reduction and Decreasingness. That is, $\mathsf{B}^x \neq \mathsf{B}$ holds and B is not annotated in U with $(x, U) \in \Gamma$ and $U = \mathsf{B} \Rightarrow \mathsf{A}_\sigma\varphi$. Since $x \neq l_1$ holds, either $(x, |U|, \mathsf{B}^x) \trianglelefteq_{\mathrm{a}} (l_1, T_1, T_1)$ or $(x, |U|, \mathsf{B}^x) \trianglelefteq_{\mathrm{qa}} (l_1, T_1, T_1)$ must hold by Accessibility. Note that $T_1 = \mathsf{B}$ and $l_1 = \mathsf{c}\,x$ holds. If $(x, |U|, \mathsf{B}^x) \trianglelefteq_{\mathrm{a}} (l_1, T_1, T_1)$ holds, this contradicts the fact that x is the first argument of c and $n^{\mathsf{c}} = 1$ holds. If $(x, |U|, \mathsf{B}^x) \trianglelefteq_{\mathrm{qa}} (l_1, T_1, T_1)$ holds, then we must choose B in U as B^x, but this contradicts the fact that B in $U = \mathsf{B} \Rightarrow \mathsf{A}_\sigma\varphi$ is not annotated.

▶ **Example 26** (Higher-Order Abstract Syntax for Untyped λ-Calculus)**.** The following is a non-terminating rewrite system which is obtained from Example 25 above with a minor change: put $\mathbb{S} := \{\mathsf{B}\}$, $\mathbb{C} := \{\mathsf{abs}\}$ and $\mathbb{F} := \{\mathsf{app}\}$ with $\Theta(\mathsf{abs}) = (\mathsf{B} \Rightarrow \mathsf{B}) \Rightarrow \mathsf{B}$ and $\Theta(\mathsf{app}) = \mathsf{B} \Rightarrow \mathsf{B} \Rightarrow \mathsf{B}$. Then, define the rewrite system $\mathcal{R}$ as $\mathcal{R} := \{\mathsf{app}\,(\mathsf{abs}\,g)\,x \to g\,x\}$ with $\{g, x\} \subseteq \mathbb{V}$. If we put $\omega' := \mathsf{abs}\,(\lambda y^{\mathsf{B}}\mathsf{app}\,y\,y)$, then

$$\mathsf{app}\,\omega'\,\omega' = \mathsf{app}\,(\mathsf{abs}\,(\lambda y^{\mathsf{B}}\mathsf{app}\,y\,y))\,\omega' \to_{\mathcal{R}} (\lambda y^{\mathsf{B}}\mathsf{app}\,y\,y)\,\omega' \to_\beta \mathsf{app}\,\omega'\,\omega' \to_{\mathcal{R}} \cdots$$

holds, hence this rewrite system $\mathcal{R}$ is non-terminating.

We show that the system $\mathcal{R}$ cannot satisfy our termination criterion. Assume that $\mathcal{R}$ satisfies the criterion. By Accessibility, we have $\mathsf{B}^g = \mathsf{B}$ as in Example 25 above, so B must be annotated in U with $(g, U) \in \Gamma$, that is, $(g, \mathsf{B}_{\alpha^g} \Rightarrow \mathsf{B}_{\alpha^g}) \in \Gamma$ holds. Then, $(x, \mathsf{B}_{\alpha^x}) \in \Gamma$ and $\alpha^x = \alpha^g$ must hold, because we have $\vdash^{\mathsf{app}}_{\varphi} g\,x : \mathsf{B}_\sigma\varphi$ by Subject Reduction and Decreasingness.

By Lemma 11.(3), $\mathsf{B}_0 \Rightarrow \mathsf{B}_0$ is a computability predicate, hence there exists a variable $z \in \mathsf{B}_0 \Rightarrow \mathsf{B}_0$ because any variable belongs to $\mathsf{B}_0 \Rightarrow \mathsf{B}_0$. This implies that $\mathsf{abs}\,z \in \mathsf{B}$ holds. We define a substitution θ as $\theta := \{(g, z), (x, \mathsf{abs}\,z)\}$, then $l_1\theta = \mathsf{abs}\,g\theta \in \mathsf{B}$ and $l_2\theta = x\theta \in \mathsf{B}$ holds. By Minimality, we have a valuation ν such that

$$o_{(\bigcup_{\mathsf{c} \leq \mathsf{a}} \mathsf{B}_\mathsf{c} \Rightarrow \mathsf{B}_\mathsf{c})_{\mathsf{a} < \mathsf{b}}}(z) = o_{(\bigcup_{\mathsf{c} \leq \mathsf{a}} \mathsf{B}_\mathsf{c} \Rightarrow \mathsf{B}_\mathsf{c})_{\mathsf{a} < \mathsf{b}}}(g\theta) = \alpha^g\nu = \alpha^x\nu \geq o_{\mathsf{B}}(x\theta) = o_{\mathsf{B}}(\mathsf{abs}\,z)$$

holds with $o_{(\bigcup_{\mathsf{c} \leq \mathsf{a}} \mathsf{B}_\mathsf{c} \Rightarrow \mathsf{B}_\mathsf{c})_{\mathsf{a} < \mathsf{b}}}(z) = 0$, but this contradicts the definition of the stratification $\mathcal{S}^{\mathsf{B}}$.

In the two examples below, we use the successor size algebra, which was defined above.

▶ **Example 27** (Hofmann's Extract Function for the Breadth-First Traversal of Trees (see, e.g., [19]))**.** Let $\mathbb{S} := \{\mathsf{C}, \mathsf{L}\}$ be the set of sorts, $\mathbb{F} := \{\mathsf{e}\}$ be the set of function symbols with $\Theta(\mathsf{e}) = \mathsf{C} \Rightarrow \mathsf{L}$, and $\mathbb{C} := \{\mathsf{c}\}$ be the set of constructors with $\Theta(\mathsf{c}) = ((\mathsf{C} \Rightarrow \mathsf{L}) \Rightarrow \mathsf{L}) \Rightarrow \mathsf{C}$. The following rewrite rule satisfies our termination criterion: $\mathsf{e}\,(\mathsf{c}\,x) \to x\,\mathsf{e}$ with $x \in \mathbb{V}$.

We annotate c as $\overline{\Theta}(\mathsf{c}) = ((\mathsf{C}_\alpha \Rightarrow \mathsf{L}) \Rightarrow \mathsf{L}) \Rightarrow \mathsf{C}_{\mathtt{s}\alpha}$, and e as $\overline{\Theta}(\mathsf{e}) = \mathsf{C}_\beta \Rightarrow \mathsf{L}_\infty$. By Definition 18, we have the size function Σ^{c} induced by this annotation of c. We define $\Gamma := \{x : (\mathsf{C}_\alpha \Rightarrow \mathsf{L}) \Rightarrow \mathsf{L}\}$, $\mathsf{B}^x := \mathsf{C}$, $\alpha^x := \alpha$ and $\varphi := \{(\beta, \mathtt{s}\,\alpha)\}$. Then, take ζ^{e}_X as the identity function for each $X \in \{\mathtt{A}, \mathfrak{h}\}$.

Monotony. Obvious.

Accessibility. We have $(x, (\mathsf{C} \Rightarrow \mathsf{L}) \Rightarrow \mathsf{L}, \mathsf{C}) \trianglelefteq_{\mathrm{a}} (\mathsf{c}\,x, \mathsf{C}, \mathsf{C})$.

Minimality. Let θ be a substitution with $\mathsf{c}\,x\theta \in \mathsf{C}$. We define ν as $\nu := \{(\alpha, \mathfrak{a})\}$ with $\mathfrak{a} := o_{[\mathsf{C}:\mathcal{S}^{\mathsf{c}}](\mathsf{C}\Rightarrow\mathsf{L})\Rightarrow\mathsf{L}}(x\theta)$. Then, by Lemma 13, we have $\beta\varphi\nu = (\mathtt{s}\,\alpha)\nu = \mathfrak{a} + 1 = \Sigma^{\mathsf{c}}(\mathfrak{a}) = o_{\mathcal{S}^{\mathsf{c}}}(\mathsf{c}\,x\theta)$.

Subject Reduction and Decreasingness. Put $\psi := \{(\beta, \alpha)\}$, then we have the following derivation:

$$\dfrac{\dfrac{x : (\mathsf{C}_\alpha \Rightarrow \mathsf{L}) \Rightarrow \mathsf{L} \vdash^{\mathsf{e}}_{\varphi} x : (\mathsf{C}_\alpha \Rightarrow \mathsf{L}) \Rightarrow \mathsf{L} \quad \overline{\vdash^{\mathsf{e}}_{\varphi} \mathsf{e} : \mathsf{C}_\alpha \Rightarrow \mathsf{L}}}{x : (\mathsf{C}_\alpha \Rightarrow \mathsf{L}) \Rightarrow \mathsf{L} \vdash^{\mathsf{e}}_{\varphi} x\,\mathsf{e} : \mathsf{L}}\ (\text{app-decr})}{}\ (\text{app-decr})$$

where $\mathsf{C}_\alpha = \mathsf{C}_\beta\psi$ and $(\mathsf{e}, \psi) <_{\mathtt{A}} (\mathsf{e}, \varphi)$ hold. Note that we have $\vdash^{\mathsf{e}}_{\varphi} \mathsf{e} : \mathsf{C}_\alpha \Rightarrow \mathsf{L}$ because we dropped the condition $|\vec{V}| \geq q^h$ in the rule (app-decr) of [9] (see Remark 23 above).

▶ **Example 28** (β-Reduction and $\beta\eta$-Reduction of Untyped λ-Calculus)**.** Let $\mathbb{S} := \{\mathsf{B}\}$ be the set of sorts, where B denotes the type of untyped λ-terms. The following rewrite rules say that abs_β is the λ-abstraction for β-reduction and that $\mathsf{abs}_{\beta\eta}$ is the λ-abstraction for $\beta\eta$-reduction:

$$\mathsf{app}\,(\mathsf{abs}_\beta\,g)\,x \to g\,x \qquad \mathsf{app}\,(\mathsf{abs}_{\beta\eta}\,g)\,x \to g\,x \qquad \mathsf{abs}_{\beta\eta}\,(\lambda y^{\mathsf{B}}\mathsf{app}\,x\,y) \to x$$

where abs_β is a constructor of B, while $\mathsf{abs}_{\beta\eta}$ and app are function symbols. In this way, one can deal with both of β-reduction and $\beta\eta$-reduction in one rewrite system. Of course, this rewrite system is not terminating as seen above. On the other hand, consider the set $\mathcal{R}$ of the following rewrite rules:

$$\mathsf{f}\,(\mathsf{abs}_\beta\,g) \to \mathsf{abs}_{\beta\eta}\,g \qquad \mathsf{f}\,(\mathsf{abs}_{\beta\eta}\,g) \to \mathsf{abs}_\beta\,g \qquad \mathsf{f}\,(\mathsf{app}\,x\,y) \to \mathsf{app}\,(\mathsf{f}\,x)\,(\mathsf{f}\,y)$$

with $\{g, x, y\} \subseteq \mathbb{V}$. These rules enable us to replace the outermost λ-abstraction for β-reduction with the one for $\beta\eta$-reduction, and vice versa. We show that the system $\mathcal{R}$ satisfies the termination criterion. Note that abs_β, $\mathsf{abs}_{\beta\eta}$ and app are all constructors of B when we consider the system $\mathcal{R}$. We thus assume that $\mathbb{C}$ is given as $\{\mathsf{abs}_\beta, \mathsf{abs}_{\beta\eta}, \mathsf{app}\}$ with $\overline{\Theta}(\mathsf{abs}_\beta) = (\mathsf{B}_\gamma \Rightarrow \mathsf{B}_\gamma) \Rightarrow \mathsf{B}_{\mathtt{s}\gamma}$, $\overline{\Theta}(\mathsf{abs}_{\beta\eta}) = (\mathsf{B}_{\gamma'} \Rightarrow \mathsf{B}_{\gamma'}) \Rightarrow \mathsf{B}_{\mathtt{s}\gamma'}$ and $\overline{\Theta}(\mathsf{app}) = \mathsf{B}_{\gamma''} \Rightarrow \mathsf{B}_{\gamma''} \Rightarrow \mathsf{B}_{\mathtt{s}\gamma''}$. By Definition 18, we have the size functions $\Sigma^{\mathsf{abs}_\beta}, \Sigma^{\mathsf{abs}_{\beta\eta}}$ and Σ^{app}. Moreover, put $\mathbb{F} := \{\mathsf{f}\}$ with $\overline{\Theta}(\mathsf{f}) = \mathsf{B}_{\alpha^{\mathsf{f}}} \Rightarrow \mathsf{B}_{\alpha^{\mathsf{f}}}$, and take ζ^{f}_X as the identity function for each $X \in \{\mathtt{A}, \mathfrak{h}\}$.

Consider the first rule (the second rule can be handled in the same way). We define $\Gamma := \{g : \mathsf{B}_{\gamma'} \Rightarrow \mathsf{B}_{\gamma'}\}$, $\mathsf{B}^g := \mathsf{B}$, $\alpha^g := \gamma'$ and $\varphi := \{(\alpha^{\mathsf{f}}, \mathtt{s}\,\gamma')\}$.

Monotony. Obvious.

Accessibility. We have $(g, \mathsf{B} \Rightarrow \mathsf{B}, \mathsf{B}) \trianglelefteq_{\mathrm{qa}} (\mathsf{abs}_\beta\,g, \mathsf{B}, \mathsf{B})$.

Minimality. Let θ be a substitution with $\mathsf{abs}_\beta\,g\theta \in \mathsf{B}$. We define ν as $\nu := \{(\gamma', \mathfrak{a})\}$ with $\mathfrak{a} := o_{(\bigcup_{\mathfrak{c}\leq\mathfrak{b}}[\mathsf{B}:\mathcal{S}^{\mathsf{B}}_{\mathfrak{c}}]\mathsf{B}\Rightarrow\mathsf{B})_{\mathfrak{b}<\mathfrak{h}}}(g\theta)$. Then, by Lemma 13, we have $\alpha^{\mathsf{f}}\varphi\nu = (\mathtt{s}\,\gamma')\nu = \mathfrak{a} + 1 = \Sigma^{\mathsf{abs}_\beta}(\mathfrak{a}) = o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{abs}_\beta\,g\theta)$.

Subject Reduction and Decreasingness. We have

$$\dfrac{\dfrac{g : \mathsf{B}_{\gamma'} \Rightarrow \mathsf{B}_{\gamma'} \vdash^{\mathsf{f}}_{\varphi} g : \mathsf{B}_{\gamma'} \Rightarrow \mathsf{B}_{\gamma'}}{g : \mathsf{B}_{\gamma'} \Rightarrow \mathsf{B}_{\gamma'} \vdash^{\mathsf{f}}_{\varphi} \mathsf{abs}_{\beta\eta}\,g : \mathsf{B}_{\mathtt{s}\gamma'}}\ (\text{app-decr}) \qquad \begin{array}{c}\vdots\\ \mathsf{B}_{\mathtt{s}\gamma'} \leq \mathsf{B}\end{array}}{g : \mathsf{B}_{\gamma'} \Rightarrow \mathsf{B}_{\gamma'} \vdash^{\mathsf{f}}_{\varphi} \mathsf{abs}_{\beta\eta}\,g : \mathsf{B}}\ (\text{sub})$$

Since $\mathsf{B}\varphi = \mathsf{B}$ holds, we are done.

Next, consider the third rule $\mathsf{f}\,(\mathsf{app}\,x\,y) \to \mathsf{app}\,(\mathsf{f}\,x)\,(\mathsf{f}\,y)$. We define $\Gamma := \{x : \mathsf{B}_{\gamma''}, y : \mathsf{B}_{\gamma''}\}$, $\mathsf{B}^x := \mathsf{B}^y := \mathsf{B}$, $\alpha^x := \alpha^y := \gamma''$ and $\varphi := \{(\alpha^{\mathsf{f}}, \mathsf{s}\,\gamma'')\}$.

Monotony. Obvious.

Accessibility. We have $(x, \mathsf{B}, \mathsf{B}) \trianglelefteq_{\mathrm{a}} (\mathsf{app}\,x\,y, \mathsf{B}, \mathsf{B})$ and $(y, \mathsf{B}, \mathsf{B}) \trianglelefteq_{\mathrm{a}} (\mathsf{app}\,x\,y, \mathsf{B}, \mathsf{B})$.

Minimality. Let θ be a substitution with $\mathsf{app}\,(x\theta)\,(y\theta) \in \mathsf{B}$. We define ν as $\nu := \{(\gamma'', \mathfrak{b})\}$ with $\mathfrak{b} := \max\{o_{\mathcal{S}^{\mathsf{B}}}(x\theta), o_{\mathcal{S}^{\mathsf{B}}}(y\theta)\}$. Then, we have $o_{\mathcal{S}^{\mathsf{B}}}(x\theta) \leq \alpha^x\nu$ and $o_{\mathcal{S}^{\mathsf{B}}}(y\theta) \leq \alpha^y\nu$. Moreover, by Lemma 13, we have $\alpha^{\mathsf{f}}\varphi\nu = (\mathsf{s}\,\gamma'')\nu = \mathfrak{b} + 1 = \Sigma^{\mathsf{app}}(o_{\mathcal{S}^{\mathsf{B}}}(x\theta), o_{\mathcal{S}^{\mathsf{B}}}(y\theta)) = o_{\mathcal{S}^{\mathsf{B}}}(\mathsf{app}\,(x\theta)\,(y\theta))$.

Subject Reduction and Decreasingness. Define $\psi := \{(\alpha^{\mathsf{f}}, \gamma'')\}$. We have the following:

$$\dfrac{\dfrac{\Gamma \vdash^{\mathsf{f}}_{\varphi} x : \mathsf{B}_{\gamma''}}{\Gamma \vdash^{\mathsf{f}}_{\varphi} \mathsf{f}\,x : \mathsf{B}_{\gamma''}}\,(\text{app-decr}) \quad \dfrac{\Gamma \vdash^{\mathsf{f}}_{\varphi} y : \mathsf{B}_{\gamma''}}{\Gamma \vdash^{\mathsf{f}}_{\varphi} \mathsf{f}\,y : \mathsf{B}_{\gamma''}}\,(\text{app-decr})}{\Gamma \vdash^{\mathsf{f}}_{\varphi} \mathsf{app}\,(\mathsf{f}\,x)\,(\mathsf{f}\,y) : \mathsf{B}_{\mathsf{s}\gamma''}}\,(\text{app-decr})$$

where $\mathsf{B}_{\gamma''} = \mathsf{B}_{\alpha^{\mathsf{f}}\psi}$ and $(\mathsf{f}, \psi) <_{\mathtt{A}} (\mathsf{f}, \varphi)$ holds.

5 Concluding Remarks and Future Work

We, in this paper, have extended the termination criterion in [9] so that in some case the termination of the rewrite relation induced by rewrite rules on non-positive types can be shown. For this purpose, the inflationary fixed-point construction has been used: the inflationary fixed-point construction is crucial to the definition of stratifications by size functions for non-positive types. In addition, we have also improved the criterion in [9] with regard to non-strictly positive inductive types. We have noted that a condition on a typing rule for the computability closure can be dropped, and then we have shown the termination of Hofmann's extract function for the breadth-first traversal of trees. This example is a typical case of rewrite systems on non-strictly positive inductive types.

However, a thorough study of rewrite rules on non-positive types is far from being achieved, since it is dependent type systems that are able to include more impressive examples of non-positive types. A larger goal is thus to extend our termination criterion to dependent type systems. Setzer's Mahlo universe ([23]), which is a universe type with a strong reflection property in Martin-Löf type theory, is an example of non-positive types in dependent type systems. Exploring rewrite rules for Setzer's Mahlo universe would be a crucial step for further research on combined systems of typed λ-calculus and rewrite rules. For this purpose, we will examine whether our criterion can be extended to $\lambda\Pi/\mathcal{R}$-calculus, and whether Mahlo universe can be formulated in $\lambda\Pi/\mathcal{R}$-calculus. This calculus is a combined system of the dependent type system $\lambda\Pi$-calculus and rewrite rules. Some termination criteria for $\lambda\Pi/\mathcal{R}$-calculus were already formulated by [10, 16], but, to the best of our knowledge, rewrite rules on non-positive types in this calculus remain unexplored.

Recently, dependently typed programming languages such as Coq and Agda were combined with rewrite rules ([13, 14]). Providing these combined languages with termination criteria would be another crucial step for further research on rewrite rules in dependent type systems, since rewrite rules and several features from Coq or Agda coexist there. Termination criteria on strictly or non-strictly positive inductive types in these languages are not sufficiently examined yet, so we are planning to begin by exploring positive inductive types. In particular, it should be investigated whether the size-based termination method is applicable to formulate termination criteria on positive inductive types in these combined programming languages.

Yet another larger goal is to integrate our work into an automated termination prover for higher-order rewriting such as Blanqui's HOT. Since HOT is based on sized types and computability closure, HOT is most relevant to our work among automated termination provers.

References

1 Andreas Abel. Type-based termination, inflationary fixed-points, and mixed inductive-coinductive types. In *Proceedings 8th Workshop on Fixed Points in Computer Science, FICS 2012, Tallinn, Estonia, 24th March 2012*, pages 1–11, 2012. doi:10.4204/EPTCS.77.1.

2 Andreas Abel and Brigitte Pientka. Well-founded recursion with copatterns and sized types. *J. Funct. Program.*, 26:e2, 2016. doi:10.1017/S0956796816000022.

3 Franz Baader and Tobias Nipkow. *Term Rewriting and All That*. Cambridge University Press, 1998. doi:10.1017/CBO9781139172752.

4 Franco Barbanera, Maribel Fernández, and Herman Geuvers. Modularity of strong normalization in the algebraic-lambda-cube. *J. Funct. Program.*, 7(6):613–660, 1997. doi:10.1017/s095679689700289x.

5 Gilles Barthe and Herman Geuvers. Modular properties of algebraic type systems. In *Higher-Order Algebra, Logic, and Term Rewriting, Second International Workshop, HOA '95, Paderborn, Germany, September 21-22, 1995, Selected Papers*, pages 37–56, 1995. doi:10.1007/3-540-61254-8_18.

6 Gilles Barthe and Femke van Raamsdonk. Termination of algebraic type systems: The syntactic approach. In *Algebraic and Logic Programming, 6th International Joint Conference, ALP '97 - HOA '97, Southampton, UK, Spetember 3-5, 1997, Proceedings*, pages 174–193, 1997. doi:10.1007/BFb0027010.

7 Frédéric Blanqui. Definitions by rewriting in the calculus of constructions. *Mathematical Structures in Computer Science*, 15(1):37–92, 2005. doi:10.1017/S0960129504004426.

8 Frédéric Blanqui. Inductive types in the calculus of algebraic constructions. *Fundamenta Informaticae*, 65(1-2):61–86, 2005.

9 Frédéric Blanqui. Size-based termination of higher-order rewriting. *J. Funct. Program.*, 28:e11, 2018. doi:10.1017/S0956796818000072.

10 Frédéric Blanqui, Guillaume Genestier, and Olivier Hermant. Dependency pairs termination in dependent type theory modulo rewriting. In *4th International Conference on Formal Structures for Computation and Deduction, FSCD 2019, June 24-30, 2019, Dortmund, Germany*, pages 9:1–9:21, 2019. doi:10.4230/LIPIcs.FSCD.2019.9.

11 Frédéric Blanqui, Jean-Pierre Jouannaud, and Mitsuhiro Okada. Inductive-data-type systems. *Theoretical Computer Science*, 272(1):41–68, 2002. doi:10.1016/S0304-3975(00)00347-9.

12 Val Breazu-Tannen and Jean Gallier. Polymorphic rewriting conserves algebraic strong normalization and confluence. In Giorgio Ausiello, Mariangiola Dezani-Ciancaglini, and Simonetta Ronchi Della Rocca, editors, *ICALP 1989: Automata, Languages and Programming*, pages 137–150, Berlin, Heidelberg, 1989. Springer Berlin Heidelberg. doi:10.1007/BFb0035757.

13 Jesper Cockx. Type theory unchained: Extending agda with user-defined rewrite rules. In *25th International Conference on Types for Proofs and Programs, TYPES 2019, June 11-14, 2019, Oslo, Norway*, pages 2:1–2:27, 2019. doi:10.4230/LIPIcs.TYPES.2019.2.

14 Jesper Cockx, Nicolas Tabareau, and Théo Winterhalter. The taming of the rew: a type theory with computational assumptions. *Proc. ACM Program. Lang.*, 5(POPL):1–29, 2021. doi:10.1145/3434341.

15 Carsten Fuhs and Cynthia Kop. Polynomial interpretations for higher-order rewriting. In *23rd International Conference on Rewriting Techniques and Applications (RTA'12) , RTA 2012, May 28 - June 2, 2012, Nagoya, Japan*, pages 176–192, 2012. doi:10.4230/LIPIcs.RTA.2012.176.

16 Guillaume Genestier. *Dependently-Typed Termination and Embedding of Extensional Universe-Polymorphic Type Theory using Rewriting*. PhD thesis, University of Paris-Saclay, France, 2020. URL: `https://tel.archives-ouvertes.fr/tel-03167579`.

17 Jean-Pierre Jouannaud and Mitsuhiro Okada. Abstract data type systems. *Theoretical Computer Science*, 173(2):349–391, 1997. `doi:10.1016/S0304-3975(96)00161-2`.

18 Cynthia Kop and Femke van Raamsdonk. Dynamic dependency pairs for algebraic functional systems. *Log. Methods Comput. Sci.*, 8(2), 2012. `doi:10.2168/LMCS-8(2:10)2012`.

19 Ralph Matthes. Lambda Calculus: A Case for Inductive Definitions. Unpublished lecture notes, 2000.

20 Nax Paul Mendler. Inductive types and type constraints in the second-order lambda calculus. *Annals of Pure and Applied Logic*, 51(1):159–172, 1991. `doi:10.1016/0168-0072(91)90069-X`.

21 Mitsuhiro Okada. Strong normalizability for the combined system of the typed lambda calculus and an arbitrary convergent term rewrite system. In *Proceedings of the ACM-SIGSAM 1989 International Symposium on Symbolic and Algebraic Computation, ISSAC '89, Portland, Oregon, USA, July 17-19, 1989*, pages 357–363, 1989. `doi:10.1145/74540.74582`.

22 Erik Palmgren. On universes in type theory. In Giovanni Sambin and Jan M. Smith, editors, *Twenty Five Years of Constructive Type Theory*, Oxford Logic Guides, pages 191–204. Oxford University Press, 1998.

23 Anton Setzer. Extending Martin-Löf type theory by one Mahlo-universe. *Arch. Math. Log.*, 39(3):155–181, 2000. `doi:10.1007/s001530050140`.

24 Christoph Sprenger and Mads Dam. On the structure of inductive reasoning: Circular and tree-shaped proofs in the μ-calculus. In *Foundations of Software Science and Computational Structures, 6th International Conference, FOSSACS 2003 Held as Part of the Joint European Conference on Theory and Practice of Software, ETAPS 2003, Warsaw, Poland, April 7-11, 2003, Proceedings*, pages 425–440, 2003. `doi:10.1007/3-540-36576-1_27`.

25 Daria Walukiewicz-Chrząszcz. Termination of rewriting in the calculus of constructions. *J. Funct. Program.*, 13(2):339–414, 2003. `doi:10.1017/S0956796802004641`.