



UvA-DARE (Digital Academic Repository)

Exact bounds for acyclic higher-order recursion schemes

Afshari, B.; Wehr, D.

DOI

[10.1016/j.ic.2022.104982](https://doi.org/10.1016/j.ic.2022.104982)

Publication date

2023

Document Version

Final published version

Published in

Information and Computation

License

CC BY

[Link to publication](#)

Citation for published version (APA):

Afshari, B., & Wehr, D. (2023). Exact bounds for acyclic higher-order recursion schemes. *Information and Computation*, 290, [104982]. <https://doi.org/10.1016/j.ic.2022.104982>

General rights

It is not permitted to download or to forward/distribute the text or part of it without the consent of the author(s) and/or copyright holder(s), other than for strictly personal, individual use, unless the work is under an open content license (like Creative Commons).

Disclaimer/Complaints regulations

If you believe that digital publication of certain material infringes any of your rights or (privacy) interests, please let the Library know, stating your reasons. In case of a legitimate complaint, the Library will make the material inaccessible and/or remove it from the website. Please Ask the Library: <https://uba.uva.nl/en/contact>, or a letter to: Library of the University of Amsterdam, Secretariat, Singel 425, 1012 WP Amsterdam, The Netherlands. You will be contacted as soon as possible.



Exact bounds for acyclic higher-order recursion schemes [☆]

Bahareh Afshari ^{a,b}, Dominik Wehr ^{b,*}

^a Institute for Logic, Language and Computation, University of Amsterdam, Amsterdam, the Netherlands

^b Department of Philosophy, Linguistics and Theory of Science, University of Gothenburg, Gothenburg, Sweden



ARTICLE INFO

Article history:

Received 9 December 2021

Received in revised form 2 November 2022

Accepted 6 November 2022

Available online 21 November 2022

Keywords:

Higher-order recursion schemes

Simply typed λ -calculus

Language bounds

ABSTRACT

Beckmann [1] derives bounds on the length of reduction chains of classes of simply typed λ -calculus terms which are exact up-to a constant factor in their highest exponent. Afshari et al. [2] obtain similar bounds on acyclic higher-order recursion schemes (HORS) by embedding them in the simply typed λ -calculus and applying Beckmann's result. In this article, we apply Beckmann's proof strategy directly to acyclic HORS, proving exactness of the bounds on reduction chain length and obtaining exact bounds on the size of languages generated by acyclic HORS.

© 2022 The Author(s). Published by Elsevier Inc. This is an open access article under the CC BY license (<http://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Given a term of a rewriting system, it is natural to ask if the length of all reduction chains starting from the term can be bounded. The literature on this question is most developed with respect to simply typed λ -calculus. Investigations typically follow one of two distinct lines of inquiry. The first approach is to derive very precise bounds for any concrete term, often by semantic means. Examples can be found in the work of Dal Lago and Laurent [3], de Carvalho et al. [4] or Bernadet and Lengrand [5]. However, computing these bounds is as computationally expensive as normalizing the term itself. The alternative approach, taken by Schwichtenberg [6], Xi [7], Beckmann [1] and Clairambault [8,9], is to give collective bounds for classes of terms parameterized by various simple metrics such as their size, the maximal arity among their subterms or the maximal order among the types of their subterms. The bounds arrived at in this manner usually are towers of exponentials. Among these, Beckmann's result is especially noteworthy as it is the first exact bound in the sense that it can only be improved by a constant factor in the highest exponent.

Higher-order recursion schemes (HORS) are a rewriting system which can be considered an extension of the simply typed λ -calculus with non-determinism and unrestricted recursion. As they provide an accurate model of higher-order functional programs, HORS have been studied intensively in the field of Software Verification (see e.g. [10–12]). While the problem of bounding reduction chain lengths naturally extends to HORS, the presence of non-deterministic rewriting rules inspires another interesting question: Can one bound the language size of a given system, i.e., the number of distinct normal forms it has? Such language size bounds can be employed to estimate the number of distinct results of a non-deterministic process, as is done in [2]. However, the literature on bounds for HORS is much sparser than that of the simply typed λ -calculus. The only result we are aware of is that by Afshari et al. [2]: They restrict themselves to HORS whose call-graph is acyclic. As such acyclic schemes can be embedded into the simply typed λ -calculus, Afshari et al. derive reduction chain length bounds

[☆] Supported by the Knut and Alice Wallenberg Foundation [2015.0179 & 2020.0199] and the Swedish Research Council [2016-03502 & 2017-05111].

* Corresponding author.

E-mail address: domink.wehr@gu.se (D. Wehr).

as a corollary of Beckmann's result [1] and extend them to language bounds. However, they do not prove their reduction chain length bounds exact in the sense of Beckmann.

In this article, we derive bounds similar to those in [2] by applying Beckmann's proof strategy directly to acyclic HORS and avoiding the detour through simply typed λ -calculus. This allows us to obtain bounds both on reduction chain length and language size, which we prove exact in the sense of Beckmann. Furthermore, we derive a result which relates the reduction length bounds to the arity of HORS terms, thereby accounting for the differences in the reduction behavior of HORS and the λ -calculus. Our analysis is partly motivated by a desire to illustrate the methodology of [1] by applying it to other rewriting systems.

Outline We begin this article by sketching the overall proof strategy for obtaining the bounds and proving their exactness in Section 2. We continue in Section 3 by formally defining acyclic higher-order recursion schemes and related concepts. The article's technical work is carried out in Sections 4 and 5. In Sections 4.1 and 4.2, respectively, we derive upper and lower bound on the length of reduction chains of acyclic HORS. We use these to deduce similarly exact language size bounds in Section 5 and close with a discussion on the wider applicability of Beckmann's method in Section 6.

2. Sketching the proofs

While the details of obtaining reduction length and language size bounds are technical, the general approach, employed by Beckmann [1] and Afshari et al. [2], respectively, is strikingly elegant. We thus begin by sketching the results and proof strategies to obtain them.

2.1. Reduction chain length bounds

We derive uniform bounds on the lengths of reduction chains for certain classes of terms of acyclic HORS. These classes are characterized by the degree $\text{deg}(t)$ (the highest order among the recursive subterms of t), the recursive size $s(t)$ or the recursive height of the syntax tree $h(t)$ of these terms:

$$\begin{aligned} d(t) &:= \max\{n \mid \exists (t_i)_{i \leq n}. t_0 = t \text{ and } t_i \rightsquigarrow t_{i+1} \text{ for all } i < n\} \\ ds_n(N) &:= \max\{d(t) \mid \text{deg}(t) \leq n \text{ and } s(t) \leq N\} \\ dh_n(N) &:= \max\{d(t) \mid \text{deg}(t) \leq n \text{ and } h(t) \leq N\} \end{aligned}$$

The overall results we arrive at are that, for a fixed degree n ,

$$ds_n(N) = 2_n(\Theta(N)) \quad dh_n(N) = 2_{n+1}(\Theta(N))$$

where $2_0(N) = N$ and $2_{n+1}(N) = 2^{2_n(N)}$. More concretely, this yields

$$ds_n(N) \leq 2_n(N) \quad dh_n(N) \leq 2_{n+1}(N).$$

The proof of these results is split into two steps.

Deriving the upper bound Following Beckmann [1], the bound is derived using expanded head reduction trees, trees which "account for" certain possible future reductions along a term's reduction chains. More concretely, these trees are formalized as a derivation system $\Downarrow_\rho^\alpha t$, describing a tree of height α which accounts for all future reduction steps of non-terminals of an order at least ρ which can occur along any sequence $t \rightsquigarrow^* t'$. The upper bound is then derived by combining three results about such trees:

- (i) Any well-typed acyclic HORS t can be trivially *embedded* into derivations of $\Downarrow_{\text{deg}(t)-1}^{s(t)} t$ and $\Downarrow_{\text{deg}(t)}^{h(t)} t$.
- (ii) We may decrement the index ρ of a derivation at the cost of increasing the height α exponentially. In other words, $\Downarrow_{\rho+1}^\alpha t$ implies $\Downarrow_\rho^{2^\alpha} t$.
- (iii) If $\rho = 0$, the derivation $\Downarrow_0^\alpha t$ can be used to estimate $d(s)$ as follows: $d(s) \leq |t| \leq 2^\alpha$ where $|t|$ denotes the size of the tree $\Downarrow_0^\alpha t$.

By repeatedly applying (ii) to trees obtained via (i) until we may derive bounds based on (iii), we arrive at

$$d(t) \leq 2_{\text{deg}(t)+1}(h(t)) \quad \text{and} \quad d(t) \leq 2_{\text{deg}(t)}(s(t))$$

which readily entails

$$dh_n(N) = 2_{n+1}(O(N)) \quad \text{and} \quad ds_n(N) = 2_n(O(N))$$

Furthermore, when restricting our attention to those terms which only feature terminals and non-terminals of arity at least k , we can sharpen the estimate to $|t| \leq \frac{2^\alpha}{k+1}$ and thus subsequently

$$ds_n(N) \leq \frac{2_n(N)}{k+1} \quad dh_n(N) \leq \frac{2_{n+1}(N)}{k+1}$$

This is a result unique to higher-order recursion systems which accounts for the fact that higher arity non-terminals are unfolded in only one reduction step.

Deriving the lower bound We bound the length of reduction chains from below by giving terms of suitable degree, height and size which have sufficiently long reduction chains. For example, for the size-based bound, we construct terms $D_n^s(N)$ such that $s(D_n^s(N)) = O(N)$, $\deg(D_n^s(N)) = n$ and $d(D_n^s(N)) \geq 2_n(N)$. Alongside analogous terms $D_n^h(N)$, they witness that

$$dh_n(N) = 2_{n+1}(\Omega(N)) \quad \text{and} \quad ds_n(N) = 2_n(\Omega(N))$$

2.2. Language size bounds

We derive uniform bounds on the size of the language generated by an acyclic HORS term. For this, we first define size- and height-based measures. These depend on one additional parameter: the branching-factor $\text{bf}(t)$ of a given HORS term t , the maximal number of distinct expressions the non-terminals occurring within t may be unfolded to.

$$l(t) := |\{t' \mid t \rightsquigarrow^* t' \text{ and } t' \text{ normal}\}|$$

$$ls_n^k(N) := \max\{l(t) \mid \deg(t) \leq n \text{ and } \text{bf}(t) \leq k \text{ and } s(t) \leq N\}$$

$$lh_n^k(N) := \max\{l(t) \mid \deg(t) \leq n \text{ and } \text{bf}(t) \leq k \text{ and } h(t) \leq N\}$$

The bounds we derive are

$$ls_n^k(N) \leq k^{2n(N)} \quad lh_n^k(N) \leq k^{2_{n+1}(N)}$$

which are exact in the sense that, for any fixed k ,

$$ls_n^k(N) = 2_{n+1}(\Theta(N)) \quad lh_n^k(N) = 2_{n+2}(\Theta(N))$$

which is again obtained in two separate steps.

Deriving the upper bound The proof follows the approximation given by Afshari et al. [2]. Essentially, as any normal form of an acyclic HORS can be derived via the leftmost-reduction strategy, there can only be at most “ $\text{bf}(t)$ different meaningful reduction steps” at any point along the reduction chain. This means that $l(t) \leq \text{bf}(t)^{d(t)}$ and, therefore, using the previous results,

$$ls_n^k(N) \leq k^{2_{n+1}(N)} = 2_{n+1}(O(N)) \quad lh_n^k(N) \leq k^{2_n(N)} = 2_{n+2}(O(N))$$

Deriving the lower bound Our approach directly parallels that for reduction chain length bounds. For the size-based bound, we build AHORS terms $L_{n,k}^s(N)$ with $s(L_{n,k}^s(N)) = O(N)$, $\deg(L_{n,k}^s(N)) = n$, $\text{bf}(L_{n,k}^s(N)) = k$ and $l(L_{n,k}^s(N)) \geq k^{2_n(N)}$. For the height-based bound, analogous terms $L_{n,k}^h(N)$ are constructed. These terms then witness the lower bounds

$$lh_n(N) = 2_{n+2}(\Omega(N)) \quad \text{and} \quad ls_n(N) = 2_{n+1}(\Omega(N))$$

3. Acyclic higher-order recursion schemes

The rewriting system we examine are acyclic higher-order recursion schemes (AHORS). The syntax of (possibly cyclic) HORS expressions and their types is given below.

$$s, t \in \text{TM} ::= x \mid a \mid st$$

$$\sigma, \tau \in \text{TY} ::= 0 \mid \sigma \rightarrow \tau$$

In addition to variables and applications, HORS expressions can contain letters a, b, c, \dots which are separated into *terminals* and *non-terminals*. While terminals, similarly to variables, cannot be reduced further, non-terminals mirror defined procedures common to programming languages in that they act as shorthands which unfold into more complex HORS expressions of ground type 0. In a departure from the behavior of procedures in programming languages, this shorthand relation need not be functional, that is, a non-terminal a may be defined as a shorthand for multiple different expressions t_1, \dots, t_n . In such cases, an application of a may, non-deterministically, unfold a to any expression t_i . We write

" $a : \tau := x_1, \dots, x_n \mapsto t_1^a \mid \dots \mid t_k^a$ " to denote that a is such a shorthand of type τ for the expressions t_1^a, \dots, t_n^a which may refer to the parameters x_1, \dots, x_n . Clearly, some choices of shorthands, such as $a : 0 := a$, allow for expressions whose reduction sequences diverge, trivializing the question of reduction chain bounds. We thus restrict our attention to acyclic HORS, meaning those whose call-graph (i.e. the graph tracking which shorthands refer to which other shorthands) is acyclic.

We begin by defining the inductively defined judgment $\mathcal{N}; \mathcal{T}; \Gamma \vdash s : \tau$ which characterizes HORS which are both acyclic and well-typed. Here, \mathcal{N} is a context of typed non-terminals of the shape $a : \tau := \vec{x} \mapsto t_1^a \mid \dots \mid t_k^a$ as explained above. To simplify arguments involving variable binding and substitution, we require that the choices of parameter variables for each non-terminal are fully distinct from each other. For example, if some non-terminal b with definition $b : \sigma := \vec{y} \mapsto t^b$ appears in the body t^a of a non-terminal $a : \tau := \vec{x} \mapsto t^a$, this ensures that $\vec{x} \notin \text{FV}(t^b)$. \mathcal{T} and Γ list typed terminals and variables with the syntax $a : \tau$ and $x : \tau$, respectively. We denote the arity of a type τ by $\text{aty}(\tau)$ as defined below. We sometimes speak of the arity of an expression, in which case we mean the arity of its type.

$$\begin{array}{c} \text{VAR} \frac{x : \tau \in \Gamma}{\mathcal{N}; \mathcal{T}; \Gamma \vdash x : \tau} \quad \text{TERM} \frac{a : \tau \in \mathcal{T}}{\mathcal{N}; \mathcal{T}; \Gamma \vdash a : \tau} \quad \text{APP} \frac{\mathcal{N}; \mathcal{T}; \Gamma \vdash s : \sigma \rightarrow \tau \quad \mathcal{N}; \mathcal{T}; \Gamma \vdash t : \sigma}{\mathcal{N}; \mathcal{T}; \Gamma \vdash st : \tau} \\ \text{DEF} \frac{\mathcal{N}; \mathcal{T}; \text{args}(\vec{x}, \tau) \vdash t_1^a : 0 \quad \dots \quad \mathcal{N}; \mathcal{T}; \text{args}(\vec{x}, \tau) \vdash t_k^a : 0 \quad n = \text{aty}(\tau)}{\mathcal{N}, (a : \tau := x_1, \dots, x_n \mapsto t_1^a \mid \dots \mid t_k^a), \mathcal{N}'; \mathcal{T}; \Gamma \vdash a : \tau} \\ \text{args}(\varepsilon, 0) := \varepsilon \quad \text{args}(x_0 \vec{x}, \sigma \rightarrow \tau) := x_0 : \sigma, \text{args}(\vec{x}, \tau) \\ \text{aty}(\sigma \rightarrow \tau) := \text{aty}(\tau) + 1 \quad \text{aty}(0) := 0 \end{array}$$

There are multiple things worth noting about these definitions: First of all, observe that $\mathcal{N}; \mathcal{T}; \Gamma \vdash s : \tau$ also asserts the acyclicity of s . Whenever the DEF-rule is applied, all shorthand definitions before and including the one currently being unfolded are removed from \mathcal{N} . This ensures that the order in which non-terminals appear in \mathcal{N} is a strict linear order $<$ such that $b < a$ for any non-terminal b occurring (recursively) in any t_i^a . It is well-known that a graph, in this case the call-graph, can be linearized in such a fashion if and only if it is acyclic.

Secondly, note that whenever the DEF-rule is applied, the variable context Γ is replaced with the argument variables of the non-terminal which is being unfolded. This is because an expression associated to a non-terminal must not refer to variables outside of its arguments.

Lastly, observe that $\mathcal{N}; \mathcal{T}; \Gamma \vdash s : \tau$ only ensures the well-typedness of all non-terminals occurring recursively in s . It is possible that there is some $(a : \tau := \vec{x} \mapsto t_1^a \mid \dots \mid t_n^a) \in \mathcal{N}$ with $\mathcal{N}; \mathcal{T}; \text{args}(\vec{x}, \tau) \not\vdash t^a : 0$ which simply does not occur recursively in s and is thus never "checked" by $\mathcal{N}; \mathcal{T}; \Gamma \vdash s : \tau$. For our purposes, this does not spell any trouble as only those non-terminals which may occur along the reduction sequence of s are of relevance to the question at hand, which are precisely those occurring recursively in s .

The recursive size $s(t)$ and height $h(t)$ of a term t is defined on the derivation of $\mathcal{N}; \mathcal{T}; \Gamma \vdash t : \tau$. This is well-defined as long as each of the terminals and non-terminals are distinct and only one shorthand definition for each non-terminal is contained in \mathcal{N} , which we will assume going forward. Under these circumstances the derivation of $\mathcal{N}; \mathcal{T}; \Gamma \vdash t : \tau$ (if it exists) is unique. We thus define

$$\begin{array}{ll} s(tt') = s(t) + s(t') & h(tt') = 1 + \max\{h(t), h(t')\} \\ s(a) = 1 + \max\{s(t_i^a) \mid i \leq n\} & h(a) = 1 + \max\{h(t_i^a) \mid i \leq n\} \\ s(b) = 1 & h(b) = 0 \\ s(x) = 1 & h(x) = 0 \end{array}$$

where $(a := \vec{x} \mapsto t_1^a \mid \dots \mid t_n^a) \in \mathcal{N}$ and $b \in \mathcal{T}$.

Furthermore, we define the order $\text{ord}(\tau)$ recursively as

$$\text{ord}(\sigma \rightarrow \tau) := \max\{\text{ord}(\sigma) + 1, \text{ord}(\tau)\} \quad \text{ord}(0) = 0$$

If $t : \tau$ is clear from the context, we sometimes write $\text{ord}(t)$ for $\text{ord}(\tau)$. We write $\text{deg}(s)$ for the degree of s which is the maximal order $\text{ord}(\sigma)$ of the types σ occurring in the derivation of $\mathcal{N}; \mathcal{T}; \Gamma \vdash s : \tau$.

We close this section by defining the *reduction steps* $\mathcal{N} \vdash s \rightsquigarrow s'$ of the AHORS expressions. Observe that the context \mathcal{N} of non-terminals is needed to determine which terms t_i^a any given non-terminal a may be unfolded to. We usually omit \mathcal{N} , simply writing $s \rightsquigarrow s'$, if \mathcal{N} is clear from the context. Furthermore, note that a non-terminal may only be unfolded when all of its arguments are present.

$$\delta \frac{(a : \tau := x_1, \dots, x_n \mapsto t_1^a \mid \dots \mid t_n^a) \in \mathcal{N} \quad n = \text{aty}(\tau)}{\mathcal{N} \vdash a s_1 \dots s_n \rightsquigarrow t_i^a [\vec{s}/\vec{x}]} \quad \frac{i \in \{0, 1\} \quad \mathcal{N} \vdash s_i \rightsquigarrow s'_i \quad s_{1-i} = s'_{1-i}}{\mathcal{N} \vdash s_0 s_1 \rightsquigarrow s'_0 s'_1}$$

4. Reduction chain length bounds

4.1. Deriving an upper bound

The key tool for establishing the upper bounds on reduction chains are *expanded head reduction trees*. These are derivations of judgments of the form $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} t : \tau$ according to the rules given in Fig. 1. Intuitively, the judgment expresses that the term t of type τ is strongly normalizing.¹ The parameter α is an upper bound on the depth of the tree deriving $\Downarrow_{\rho}^{\alpha} t$ and ρ restricts which instances of the CUT-rule may be used. Crucially, $\rho = 0$ indicates a judgment in the CUT-free fragment of the derivation system.

$$\begin{array}{c}
 \text{VAR} \frac{x : \tau_1 \rightarrow \dots \rightarrow \tau_n \rightarrow \tau \in \Gamma \quad \mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} t_1 : \tau_1 \quad \dots \quad \mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} t_n : \tau_n}{\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha+n} x t_1 \dots t_n : \tau} \\
 \\
 \text{TERM} \frac{a : \tau_1 \rightarrow \dots \rightarrow \tau_n \rightarrow \tau \in \mathcal{T} \quad \mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} t_1 : \tau_1 \quad \dots \quad \mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} t_n : \tau_n}{\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha+n} a t_1 \dots t_n : \tau} \\
 \\
 \delta \frac{(a : \vec{\tau} \rightarrow \sigma := \vec{x}\vec{y} \mapsto t_1^a \mid \dots \mid t_k^a) \in \mathcal{N} \quad |\vec{\tau}| = |\vec{x}| = n \quad \mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} s_i : \tau_i \text{ for } 1 \leq i \leq n \quad \mathcal{N}; \mathcal{T}; \Gamma, \text{args}(\vec{y}, \sigma) \Downarrow_{\rho}^{\alpha} t_i^a[\vec{s}/\vec{x}] : 0 \text{ for } 1 \leq i \leq k}{\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha+n+1} a s_1 \dots s_n : \tau} \\
 \\
 \text{CUT} \frac{\text{ord}(\sigma \rightarrow \tau) \leq \rho \quad \mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} s : \sigma \rightarrow \tau \quad \mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} t : \sigma}{\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha+1} s t : \tau}
 \end{array}$$

Fig. 1. The rules of $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} s : \tau$.

As the judgments of $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} t : \tau$ are syntactically complex, we omit parts of them which are clear from the context wherever possible. Often, we simply write $\Downarrow_{\rho}^{\alpha} t$. Most of this section is concerned with proving various properties of these derivations, often the validity of various transformation procedures on them. The choices made in designing these derivation rules are justified by the various lemmas we prove in this section. We thus elaborate on these as we go along.

We begin by illustrating the ultimate purpose of the expanded head reduction tree in this proof method: Observe that for any well-typed term t , there is one unique derivation of $\Downarrow_0^{\alpha} t$ (up-to the choice of α , which does not influence the “shape” of the derivation). Derivations \Downarrow_0^{α} “account for” every possible future reduction step t can take. To prove this, one first defines a size measure on such derivations. Note that the notation $|t|$ is appropriate because of the unique “shape” of the derivation $\Downarrow_0^{\alpha} t$.

Definition 1. For $\Downarrow_0^{\alpha} t$ we define its size $|t|$ recursively on $\Downarrow_0^{\alpha} t$:

$$\begin{array}{ll}
 |x t_1 \dots t_n| := \sum_{i=1}^n |t_i| & \text{(VAR)} \\
 |b t_1 \dots t_n| := \sum_{i=1}^n |t_i| & \text{(TERM)} \\
 |a t_1 \dots t_n| := \begin{cases} 1 + \max\{|t_i^a[\vec{t}/\vec{x}]| \mid i \leq m\} + \sum_{i=1}^n |t_i| & \text{aty}(a) = n \\ \sum_{i=1}^n |t_i| & \text{otherwise} \end{cases} & (\delta)
 \end{array}$$

where $a := x_1 \dots x_k \mapsto t_1^a \mid \dots \mid t_m^a \in \mathcal{N}$ in the (δ) case.

The following lemma should be read as a formalizing that $\Downarrow_0^{\alpha} s$ “accounts for” all future reductions of s .

Lemma 2. If $\Downarrow_0^{\alpha} s$ and $s \rightsquigarrow^+ t$ then $\Downarrow_0^{\beta} t$ for some β and $|s| > |t|$.

¹ Our choice of notation differs from most of the literature, in which the judgment is usually denoted by $\vdash_{\rho}^{\alpha} t$. We opt for this break in convention to avoid confusion with the typing judgment $\mathcal{N}; \mathcal{T}; \Gamma \vdash t : \tau$. Our notation $\Downarrow_{\rho}^{\alpha} t$ is inspired by the notation for *big-step semantics* in the programming language design community, who denote by $s \Downarrow t$ that a term s normalizes to a term t . We believe the notation $\Downarrow_{\rho}^{\alpha} t$ is appropriate as $\Downarrow_{\rho}^{\alpha} t$ entails strong normalization of t .

Proof. Observe that the claim can be extended to the case $s \rightsquigarrow^* t$ weakening the inequality to $|s| \geq |t|$. We prove the claim per induction on $\Downarrow_{\rho}^{\alpha} s$.

Case VAR: Then $s = x\vec{u}$. We know that $x\vec{u} \rightsquigarrow^+ t$ means that $xu_1 \dots u_n \rightsquigarrow^+ xu'_1 \dots u'_n$ where $u_i \rightsquigarrow^* u'_i$. Per inductive hypothesis, we know that $\Downarrow_{\rho}^{\beta_i} u'_i$, meaning if we take $\beta := \max\{\beta_1, \dots, \beta_n\}$ we have $\Downarrow_{\rho}^{\beta} u'_i$ for $1 \leq i \leq n$. Thus $\Downarrow_{\rho}^{\beta+n} x\vec{u}'$. Per inductive hypothesis, we know $|u_i| \geq |u'_i|$. Furthermore, $x\vec{u} \rightsquigarrow^+ x\vec{u}'$ means there is $u_j \rightsquigarrow^+ u'_j$ for which we have $|u_j| > |u'_j|$. This yields the desired inequality:

$$|x\vec{u}| = \sum_{i=1}^n |u_i| > \sum_{i=1}^n |u'_i| = |x\vec{u}'|$$

Case TERM: Completely analogous to the VAR case.

Case δ : Then $s = a\vec{u}$. If $\text{aty}(a) < |\vec{u}|$ then the case is just analogous to VAR. Thus suppose $\text{aty}(a) = |\vec{u}|$. There are two possibilities how $a\vec{u} \rightsquigarrow^+ t$ can be constituted.

1. Possibly, $au_1 \dots u_n \rightsquigarrow^+ au'_1 \dots u'_n$ with $u_j \rightsquigarrow^+ u'_j$ for at least one j and $u_k \rightsquigarrow^* u'_k$ for all other $j \neq k$. Then we can obtain $t_i^a[\vec{u}/\vec{x}] \rightsquigarrow^* t_i^a[\vec{u}'/\vec{x}]$ by simply replaying $u_k \rightsquigarrow^* u'_k$ and $u_j \rightsquigarrow^+ u'_j$ at each substituted instance of u_k and u_j , respectively. This then yields $\Downarrow_0^{\beta_0} t_i^a[\vec{u}'/\vec{x}]$ and $|t_i^a[\vec{u}/\vec{x}]| \geq |t_i^a[\vec{u}'/\vec{x}]|$ for each $i \leq m$ per inductive hypothesis. For each u_k , we obtain $\Downarrow_0^{\gamma_k} u'_k$ and $|u_k| \geq |u'_k|$ per inductive hypothesis, with $|u_j| > |u'_j|$. Then clearly $\Downarrow_0^{\beta+n+1} a\vec{u}'$ with $\beta := \max\{\beta_1, \dots, \beta_m, \gamma_1, \dots, \gamma_n\}$. Furthermore

$$|a\vec{u}| = 1 + \max\{|t_i^a[\vec{u}/\vec{x}]| \mid i \leq m\} + \sum_{k=1}^n |u_k| > 1 + \max\{|t_i^a[\vec{u}'/\vec{x}]| \mid i \leq m\} + \sum_{k=1}^n |u'_k| = |a\vec{u}'|$$

2. Otherwise, $au_1 \dots u_n \rightsquigarrow^* au'_1 \dots u'_n \rightsquigarrow^+ t$ with $u_j \rightsquigarrow^* u'_j$. Then we can derive $t_i^a[\vec{u}/\vec{x}] \rightsquigarrow^* t_i^a[\vec{u}'/\vec{x}]$ as in the case above. Together with $t_i^a[\vec{u}'/\vec{x}] \rightsquigarrow^* t'$ this yields $t_i^a[\vec{u}/\vec{x}] \rightsquigarrow^* t'$ which allows us to derive $\Downarrow_0^{\beta} t'$ and $|t_i^a[\vec{u}/\vec{x}]| \geq |t'|$ per inductive hypothesis on $t_i^a[\vec{u}/\vec{x}]$. Then

$$|a\vec{u}| = 1 + \max\{|t_i^a[\vec{u}/\vec{x}]| \mid i \leq m\} + \sum_{j=1}^n |u_j| > |t_i^a[\vec{u}/\vec{x}]| \geq |t'|$$

Corollary 3 (Boundedness). For any $\Downarrow_0^{\alpha} s$ we have $d(s) \leq |s|$.

Proof. For any (possibly infinite) reduction chain $s = s_0 \rightsquigarrow s_1 \rightsquigarrow s_2 \rightsquigarrow \dots$ we know that $|s_i| > |s_{i+1}|$ by Lemma 2. That means any chain can be at most of length $|s|$.

With Lemma 2 in mind, we can motivate the choice of premises for the derivation rules of $\Downarrow_{\rho}^{\alpha} s$. For illustrative purposes, we give a simplified variant of the derivation system, in which details irrelevant to this discussion have been omitted, below.

$$\text{VAR} \frac{\Downarrow t_1 \quad \dots \quad \Downarrow t_n}{\Downarrow x\vec{t}} \qquad \delta \frac{\begin{array}{c} \Downarrow u_1 \quad \dots \quad \Downarrow u_n \\ a : \tau := \vec{x} \mapsto t_1^a \mid \dots \mid t_m^a \\ \Downarrow t_1^a[\vec{u}/\vec{x}] \quad \dots \quad \Downarrow t_m^a[\vec{u}/\vec{x}] \end{array}}{\Downarrow a\vec{u}}$$

The proof of Lemma 2 relies on the fact that whenever a term $\Downarrow_0^{\alpha} s$ takes a reduction step $s \rightsquigarrow t$, this reduction step is “accounted for” somewhere within the derivation of $\Downarrow_0^{\alpha} s$. Thus each of the (non-CUT) rules has as its premises the various “places” a reduction step could lead to or take place in. Simplest of all, the VAR- and TERM-rules only have to account for their argument terms as variables and terminals cannot take reduction steps on their own, so any reduction involving them will have taken place in one of their arguments. The more interesting case is that of δ : Clearly, the reductions $a\vec{t} \rightsquigarrow t_i^a[\vec{t}/\vec{x}]$ need to be accounted for. However, this does not suffice to also account for the non-terminal’s arguments, as possibly $x_i \notin \text{FV}(t_j^a)$, meaning u_i does not occur in $t_j^a[\vec{u}/\vec{x}]$ which would in turn make it impossible to “replay” reduction steps in u_i on $t_j^a[\vec{u}/\vec{x}]$. Thus the δ -rule needs to also account for all of the arguments u_i individually as well.

We continue by proving that various transformations of derivations $\Downarrow_{\rho}^{\alpha} s$ can be carried out. Because the cases for VAR and TERM are always analogous, we only detail the case for VAR in the following proofs.

Lemma 4 (Weakening). If $\Downarrow_{\rho}^{\alpha} s$, $\alpha < \alpha'$ and $\rho < \rho'$ then $\Downarrow_{\rho'}^{\alpha'} s$.

Lemma 5 (Renaming). *If $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} s$ where $x : \tau, y : \tau \in \Gamma$ then $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} s[y/x]$.*

Proof. [Lemmas 4 and 5] Immediate by induction on $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} s$.

Lemma 6 (Appending). *If $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} s : \sigma \rightarrow \tau$ and $y : \sigma \in \Gamma$, then $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha+1} s y : \tau$.*

Proof. Begin by observing that $\Downarrow_{\rho}^{\alpha} y$ for any α via the VAR rule and Lemma 4. We proceed via induction on $\Downarrow_{\rho}^{\alpha} s$.

Case VAR: Then $s = x\vec{t}$ with $\Downarrow_{\rho}^{\alpha} t_i$ for $1 \leq i \leq n$, meaning $\Downarrow_{\rho}^{\alpha+n+1} x\vec{t}$ as $\Downarrow_{\rho}^{\alpha} y$ as well.

Case δ : Then $s = a t_1 \dots t_n$ with $\Downarrow_{\rho}^{\alpha} t_i^a[\vec{t}/\vec{x}]$, meaning $\Downarrow_{\rho}^{\alpha} t_i^a[\vec{t}/\vec{x}][y/x_{n+1}]$ via Lemma 5. Together with $\Downarrow_{\rho}^{\alpha} t_i$ and $\Downarrow_{\rho}^{\alpha} y$ this yields $\Downarrow_{\rho}^{\alpha+n+2} a t_1 \dots t_n y$.

Case CUT: Then $s = t t'$ and $\text{ord}(t) \leq \rho$. But this means that $\text{ord}(t t') \leq \text{ord}(t) \leq \rho$ as well. We can thus arrive at $\Downarrow_{\rho}^{\alpha+1} t t'$ via another CUT.

Note that the proof of Lemma 6 motivates why the δ -rule allows for partial applications of a non-terminal a (those $a\vec{t} : \tau$ with $\tau \neq 0$) as this accommodation eases the proof in the δ -case.

Lemma 7 (Substitution). *If $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha} s$ and $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\beta} t_i : \tau_i$ where $\text{ord}(\tau_i) \leq \rho$ and $x_i : \tau_i \in \Gamma$ for $1 \leq i \leq n$ then $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha+\beta} s[\vec{t}/\vec{x}]$.*

Proof. Proof per induction on $\Downarrow_{\rho}^{\alpha} s$. We write s^* as shorthand for $s[\vec{t}/\vec{x}]$

Case VAR: Then $s = y\vec{u}$. Per inductive hypothesis we immediately obtain $\Downarrow_{\rho}^{\alpha+\beta} u_i^*$. If $y \notin \vec{x}$ then $y^* = y$ and we may thus conclude that $\Downarrow_{\rho}^{\alpha+n+\beta} y\vec{u}^*$. If $y \in \vec{x}$ then $y^* = t_j$. We know that $\text{ord}(t_j) \leq \rho$ meaning we may derive $\Downarrow_{\rho}^{\alpha+n+\beta} t_j u^*$ by n applications of the CUT rule.

Case δ : Then $s = a\vec{u}$ with $(a : \tau := \vec{y}, \vec{y}' \mapsto t^a) \in \mathcal{N}$. We know by inductive hypothesis that $\Downarrow_{\rho}^{\alpha+\beta} u_i^*$ and $\Downarrow_{\rho}^{\alpha+\beta} t_i^a[\vec{u}/\vec{y}][\vec{t}/\vec{x}]$. As $\mathcal{N}; \mathcal{T}; \text{args}((\vec{y}, \vec{y}'), \tau) \vdash t_i^a : 0$ we know that $\vec{x} \notin \text{FV}(t_i^a)$ and thus that $t_i^a[\vec{t}/\vec{x}] = t_i^a$. Then $t_i^a[\vec{u}/\vec{y}][\vec{t}/\vec{x}] = t_i^a[\vec{u}^*/\vec{y}]$ which yields $\Downarrow_{\rho}^{\alpha+\beta} t_i^a[\vec{u}^*/\vec{y}]$, allowing us to conclude $\Downarrow_{\rho}^{\alpha+n+\beta+1} a\vec{u}^*$ as desired.

Case CUT: Then $s = t_0 t_1$. Per inductive hypothesis we know $\Downarrow_{\rho}^{\alpha+\beta} t_i^*$. As furthermore $\text{ord}(t_0) = \text{ord}(t_0^*)$ we conclude $\Downarrow_{\rho}^{\alpha+1+\beta} t_0^* t_1^*$.

Combining these transformations yields the CUT-admissibility lemma. It allows CUTs of order $\rho + 1$ to be performed in a derivation of $\Downarrow_{\rho}^{\alpha}$ at the cost of an increase of the height-bound of the resulting derivation.

Lemma 8 (CUT-Admissibility). *If $\Downarrow_{\rho}^{\alpha} s : \sigma \rightarrow \tau$ and $\Downarrow_{\rho}^{\beta} t : \sigma$ such that $\text{ord}(\sigma \rightarrow \tau) \leq \rho + 1$ then $\Downarrow_{\rho}^{\alpha+\beta+1} s t : \tau$.*

Proof. Pick $x \notin \Gamma$, by Lemma 6 we have $\mathcal{N}; \mathcal{T}; \Gamma, (x : \sigma) \Downarrow_{\rho}^{\alpha+1} s x : \tau$. Then apply Lemma 7 to obtain $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{\alpha+\beta+1} s t : \tau$.

By replacing each CUT-application of order $\rho + 1$ with an application of the CUT-admissibility lemma, one can lower the CUT-level to ρ at the cost of an exponential blowup of the height-bound α .

Lemma 9 (Cut Reduction). *If $\Downarrow_{\rho+1}^{\alpha} s$ then $\Downarrow_{\rho}^{2\alpha} s$.*

Proof. We show $\Downarrow_{\rho}^{2\alpha-1} s$ per induction on $\Downarrow_{\rho}^{\alpha}$. The claim follows by Lemma 4.

Case VAR: Then $s = x\vec{t}$. We directly obtain $\Downarrow_{\rho}^{2\alpha+n-1} x\vec{t}$ as $\Downarrow_{\rho}^{2\alpha-1} t_i$ per inductive hypothesis. As $2\alpha + n - 1 \leq 2\alpha + n - 1$ the claim then follows by Lemma 4.

Case δ : Then $s = a\vec{t}$. Analogously to the VAR case, we obtain $\Downarrow_{\rho}^{2\alpha+n} a\vec{t}$ and apply Lemma 4 to derive the claim.

Case CUT: Then $s = t t'$. The inductive hypothesis yields $\Downarrow_{\rho}^{2\alpha-1} t$ and $\Downarrow_{\rho}^{2\alpha-1} t'$. Then an application of Lemma 8 yields precisely $\Downarrow_{\rho}^{2\alpha+1-1} t t'$.

We next show that any well-typed acyclic HORS $\mathcal{N}; \mathcal{T}; \Gamma \vdash t : \tau$ can be embedded into $\Downarrow_{\rho}^{\alpha} t$. There are two different methods of achieving this. The “naïve” method simply replaces each \vdash -rule with its corresponding \Downarrow -rule. Notably, this means APP is replaced with CUT, resulting in a height-bound of $h(t)$ and a CUT-level of $\text{deg}(t)$. If one instead opts to

replace each APP of order $\deg(t)$ with an application of Lemma 9, one lowers the CUT-level of the resulting derivation to $\deg(t) - 1$ while increasing the height-bound to $s(t)$. As $s(t) \leq 2^{h(t)}$ the bound obtained via this embedding will never be worse because one CUT-reduction step and the resulting exponential increase of α is prevented.

Lemma 10 (Embedding). *Let $\mathcal{N}; \mathcal{T}; \Gamma \vdash t : \tau$. Then*

1. *If $\deg(t) \leq \rho$ then $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{h(t)} t : \tau$.*
2. *If $\deg(t) \leq \rho + 1$ then $\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{\rho}^{s(t)} t$.*

Proof. We prove $\Downarrow_{\rho}^{s(t)-1} t$ and $\Downarrow_{\rho}^{h(t)} t$ per induction on $\Gamma; \mathcal{N} \vdash t : \tau$. We combine the handling of both claims when possible.

Case VAR: Then $t = x$ and thus $\Downarrow_{\rho}^0 x$ by the VAR rule.

Case TERM: Then $t = a$ with $a \in \mathcal{T}$ and thus $\Downarrow_{\rho}^0 a$ by the TERM rule.

Case DEF: Then $t = a$ with $(a := \vec{x} \mapsto t_1^a \mid \dots \mid t_n^a) \in \mathcal{N}$. Per inductive hypothesis we obtain $\Downarrow_{\rho}^{s(t_i^a)-1} t_i^a$ and thus $\Downarrow_{\rho}^{s(a)-1} a$ by Lemma 4 and the δ rule because $s(a) = 1 + \max\{s(t_i^a) \mid i \leq n\}$. The claim follows analogously for $h(a)$.

Case APP: Then $t = u u'$.

1. Per inductive hypothesis we obtain $\Downarrow_{\rho}^{h(u)} u$ and $\Downarrow_{\rho}^{h(u')} u'$. As we know $\text{ord}(u) \leq \rho$ we obtain $\Downarrow_{\rho}^{h(uu')} u u'$ via CUT as $h(u u') = \max\{h(u), h(u')\} + 1$.
2. The inductive hypothesis yields $\Downarrow_{\rho}^{s(u)-1} u$ and $\Downarrow_{\rho}^{s(u')-1} u'$. Because $\text{ord}(u) \leq \rho + 1$ we can apply Lemma 8 to obtain $\Downarrow_{\rho}^{s(uu)-1} u u'$ as $s(u u') = s(u) + s(u')$.

The last lemma of this section provides a method of estimating $|s|$ in terms of α for any given $\Downarrow_0^{\alpha} s$.

Lemma 11 (Estimation). *If $\Downarrow_0^{\alpha} s$ then $|s| \leq 2^{\alpha}$.*

Proof. We prove $|s| \leq 2^{\alpha} - 1$ per induction on $\Downarrow_0^{\alpha} s$.

Case VAR: Per inductive hypothesis $|t_i| \leq 2^{\alpha} - 1$. Then

$$|x\vec{t}| = \sum_{i=1}^n |t_i| \leq \sum_{i=1}^n (2^{\alpha} - 1) = n(2^{\alpha} - 1) \leq 2^{\alpha+n} - 1$$

Case δ : Per inductive hypothesis $|t_i^a[\vec{t}/\vec{x}]|, |t_i| \leq 2^{\alpha} - 1$. Then

$$|a\vec{t}| \leq 1 + \max\{|t_i^a[\vec{t}/\vec{x}]| \mid i \leq m\} + \sum_{i=1}^n |t_i| \leq 1 + (n+1)(2^{\alpha} - 1) \leq 2^{\alpha+n+1} - 1$$

The upper bounds on the lengths of reduction chains can then be obtained by combining all previous results of this section.

Theorem 12 (Upper bounds).

$$dh_n(N) \leq 2_{n+1}(N) \quad ds_n(N) \leq 2_n(N)$$

Proof. Let $\mathcal{N}; \mathcal{T}; \Gamma \vdash t : \tau$ with $\deg(t) = n$ and $n > 0$. By Lemma 10 we know that

$$\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_n^{h(t)} t : \tau \quad \mathcal{N}; \mathcal{T}; \Gamma \Downarrow_{n-1}^{s(t)} t : \tau$$

Now we can repeatedly apply Lemma 9 to obtain

$$\mathcal{N}; \mathcal{T}; \Gamma \Downarrow_0^{2_n(h(t))} t : \tau \quad \mathcal{N}; \mathcal{T}; \Gamma \Downarrow_0^{2_{n-1}(s(t))} t : \tau$$

which allows us to estimate, using Lemma 11, that

$$|t| \leq 2_{n+1}(h(t)) \quad |t| \leq 2_n(s(t))$$

Lastly, because of Corollary 3 we know

$$d(t) \leq |t| \leq 2_{n+1}(h(t)) \quad d(t) \leq |t| \leq 2_n(s(t))$$

As this can be done for any $\mathcal{N}; \mathcal{T}; \Gamma \vdash t : \tau$ with $\deg(t) = n > 0$ this yields the bounds as desired.

For the case of $n = 0$, observe that we must have $\mathcal{N}; \mathcal{T}; \Gamma \vdash t : 0$ where t is a variable, terminal or non-terminal of type 0, such non-terminals in turn unfolding to analogous terms of degree 0. For such terms, each step $t \rightsquigarrow t'$ guarantees $s(t) > s(t')$ and $h(t) > h(t')$ as no substitutions take place. Thus we obtain

$$dh_0(N) \leq N < 2_1(N) \quad ds_0(N) \leq N = 2_0(N)$$

Corollary 13 (Strong Normalization). *All reduction chains starting at a term $\mathcal{N}; \mathcal{T}; \Gamma \vdash s : \tau$ are finite.*

Theorem 12 is completely analogous to the result for the simply typed λ -calculus obtained by Beckmann [1]. As we prove in Section 4.2, the bounds are also exact. Nonetheless, the bounds can be improved when considering another parameter: the arity of non-terminals. This result is motivated by the difference in reduction behavior between the λ -calculus and HORS. If a λ -abstraction over multiple variables is applied to multiple arguments, the arguments are substituted one-at-a-time (as pictured below). On the other hand, non-terminals in HORS (such as $a := x, y \mapsto s$ below) receive all of their arguments in one reduction step.

$$(\lambda x. \lambda y. s) u v \rightsquigarrow (\lambda y. s[x/u]) v \rightsquigarrow s[x/u, y/v] \quad a u v \rightsquigarrow s[x/u, y/v]$$

This means that high-arity HORS expressions should reduce “faster” than the equivalent λ -terms in some way. The result below makes this intuition concrete.

Lemma 14 (Estimation II). *Let s be an AHORS expression such that all non-terminals (recursively) occurring in s are of arities at least k . Then $|s| \leq \frac{2^\alpha}{k+1}$ whenever $\Downarrow_0^\alpha s$.*

Proof. We prove that $|s| \leq \frac{2^\alpha - 1}{k+1}$ per induction on $\Downarrow_0^\alpha s$

Case VAR: Per inductive hypothesis $|t_i| \leq \frac{2^\alpha - 1}{k+1}$. Then

$$|x\vec{t}| = \sum_{i=1}^n |t_i| \leq \sum_{i=1}^n \frac{2^\alpha - 1}{k+1} = \frac{n(2^\alpha - 1)}{k+1} \leq \frac{2^{\alpha+n} - 1}{k+1}$$

because $n(2^\alpha - 1) \leq 2^{\alpha+n} - 1$ always.

Case δ : Per inductive hypothesis $|t_i^a[\vec{t}/\vec{x}]|, |t_i| \leq \frac{2^\alpha - 1}{k+1}$. The only interesting case is that of $\text{aty}(a) = |\vec{t}| = n$. Then

$$\begin{aligned} |a\vec{t}| &= 1 + \max\{|t_i^a[\vec{t}/\vec{x}]| \mid i \leq m\} + \sum_{i=1}^n |t_i| \leq 1 + \frac{n+1}{k+1}(2^\alpha - 1) \\ &= \frac{(k+1 - n - 1) + (n+1)2^\alpha}{k+1} \leq \frac{(n+1)2^\alpha}{k+1} \leq \frac{2^{\alpha+n+1} - 1}{k+1} \end{aligned}$$

because $n \geq k$ and thus $k - n \leq 0$.

Corollary 15. *When restricting our attention to AHORS expressions which contain only non-terminals of arity at least k , we obtain the following upper bounds.*

$$dh_n(N) \leq \frac{2_{n+1}(N)}{k+1} \quad ds_n(N) \leq \frac{2_n(N)}{k+1}$$

4.2. Deriving a lower bound

In this section, we show that the upper bounds from Section 4.1 are exact in the sense of Beckmann [1]. More concretely, we prove that

$$ds_n(N) = 2_n(\Omega(N)) \quad dh_n(N) = 2_{n+1}(\Omega(N))$$

from which we can conclude that the bounds from Section 4.1 can only be improved to something of the shape

$$ds_n(N) \leq 2_n(\lfloor r_s N \rfloor + c_s) \quad dh_n(N) \leq 2_{n+1}(\lfloor r_h N \rfloor + c_h)$$

for constants $0 < r_s, r_h < 1$ and $c_s, c_h \in \mathbb{R}$.

To derive this result, we follow Beckmann in defining terms $D_n^s(N)$ and $D_n^h(N)$ of suitable degrees, sizes and heights which have reduction chains of at least length $2_{n+1}(N)$ and $2_n(N)$, respectively. For this, fix a terminal $B : 0 \rightarrow 0 \rightarrow 0$. For terms $t : 0$ we define b -trees of height K with t at each leaf as follows

$$T_0(t) := t \quad T_{K+1}(t) := B T_K(t) T_K(t)$$

Such trees are of interest as they allow for exponential replication of reduction steps, as proven in the following lemma:

Lemma 16. For any $t, s : 0$ with $t \rightsquigarrow^n s$ we have $T_K(t) \rightsquigarrow^{n \cdot 2^K} T_K(s)$.

Proof. Observe that $T_K(t)$ has 2^K copies of t as its nodes. Then each copy can take n steps, leading to $n \cdot 2^K$ steps from $T_K(t)$ to $T_K(s)$ overall.

Now, with a non-terminal $b : 0 \rightarrow 0 := x \mapsto b x x$ and writing $[f]^n(v)$ for the n -fold application of f to v as defined below

$$[f]^0(v) := v \quad [f]^{n+1}(v) := f([f]^n(v))$$

we can derive the following result.

Fact 17. For any $t : 0$ we have $[b]^K t \rightsquigarrow^* T_K(t)$

Proof. Simple induction on K .

The majority of this section is concerned with “compressing” these trees into terms $s(S_n^N(t)) = O(N)$ and $h(H_n^N(t)) = O(N)$ with $S_n^N(t) \rightsquigarrow^* T_{2^{n-1}(N)}(t)$ and $H_n^N(t) \rightsquigarrow^* T_{2^n(N)}(t)$, respectively.

Beckmann [1] achieves this with the help of Church encodings of natural numbers, an approach that we parallel in HORS. Unfortunately, the reduction behavior of HORS Church encodings of different orders applied to each other is somewhat erratic, as often only reduction steps in the outer-most application may be taken. We thus first develop a theory of n -repeaters, which can be seen as a semantic characterization of Church encodings and which helps us control this behavior in our proofs. Writing τ_i for the types

$$\tau_0 := 0 \quad \tau_{i+1} := \tau_i \rightarrow \tau_i$$

and $\text{tower}(n_0, \dots, n_m)$ for the tower of exponentials

$$\text{tower}(n) := n \quad \text{tower}(n, \vec{m}) := n^{\text{tower}(\vec{m})}$$

we inductively define n -repeaters below.

Definition 18.

- 0. We call an expression $f_0^n : \tau_2$ an n -repeater of order 0 if $n > 0$ and for any $m : \tau_1$ and $v : \tau_0$ we have $f_0^n m v \rightsquigarrow^* [m]^n(v)$
- i. We call an expression $f_i^{n_i} : \tau_{i+2}$ an n_i -repeater of order i if $n_i > 0$ and for any family $(f_j^{n_j})_{j < i}$ of n_j -repeaters of order j and any $m : \tau_1$ and $v : \tau_0$ we have that $f_i^{n_i} f_{i-1}^{n_{i-1}} \dots f_0^{n_0} m v \rightsquigarrow^* [m]^k(v)$ where $k = \text{tower}(n_0, n_1, \dots, n_i)$.

This definition allows us to state and prove the following crucial lemma. It is needed for proving both the well-definedness of our HORS Church encodings and the correctness of $S_n^N(t)$ and $H_n^N(t)$.

Lemma 19. For a family $(f_j^{n_j})_{j \leq i}$ of n_j -repeaters of order j , $N > 0$ and any $m : \tau_1$ and $v : \tau_0$ we have $[f_i^{n_i}]^N (f_{i-1}^{n_{i-1}}) \dots m v \rightsquigarrow^* [m]^k(v)$ with $k = \text{tower}(n_0, \dots, n_i, N)$.

Proof. We proceed by induction on the maximum order i .

$i = 0$: We proceed by induction on N .

$$N = 1: \quad \text{Then } [f_0^{n_0}]^1(m) v = f_0^{n_0} m v \rightsquigarrow^* [m]^{n_0}(v) = [m]^{n_0}(v).$$

$N + 1$:

$$\begin{aligned} [f_0^{n_0}]^{N+1}(m) v &= f_0^{n_0} ([f_0^{n_0}]^N(m)) v \rightsquigarrow^* [[f_0^{n_0}]^N(m)]^{n_0}(v) \\ &= \underbrace{[f_0^{n_0}]^N(m) \dots ([f_0^{n_0}]^N(m) v) \dots}_{n \text{ times}} \\ &\stackrel{\text{IH}}{\rightsquigarrow^*} \underbrace{[m]^{n_0} \dots ([m]^{n_0}(v)) \dots}_{n \text{ times}} = [m]^{n_0^{N+1}}(v) \end{aligned}$$

$i + 1$: Again, we perform an induction on N .

$N = 1$: Analogous to the case for $i = 0$.

$N + 1$: Then $[f_i^{n_i}]^{N+1}(f_{i-1}^{n_{i-1}}) \dots m v = f_i^{n_i}([f_i^{n_i}]^N(f_{i-1}^{n_{i-1}})) \dots m v \rightsquigarrow^* [m]^k(v)$ with $k = \text{tower}(n_0, \dots, n_{i-2}, n_{i-1}^{n_i}, n_i)$ because, per inductive hypothesis, the expression $[f_i^{n_i}]^N(f_{i-1}^{n_{i-1}})$ is an $n_{i-1}^{n_i}$ -repeater of order $i - 1$. Now, as $(n_{i-1}^{n_i})^{n_i} = n_{i-1}^{n_i * n_i} = n_{i-1}^{n_i^{N+1}}$, we can conclude that $k = \text{tower}(n_0, \dots, n_{i-1}, n_i, N + 1)$.

We now define the HORS Church encoding $e_i^n : (\tau_i \rightarrow \tau_i) \rightarrow \tau_i \rightarrow \tau_i$ of n specialized to values of order i as follows

$$e_i^n : \tau_{i+2} := f x_i \bar{x} \mapsto [f]^n(x_i) \bar{x}$$

where $\bar{x} = x_{i-1} \dots x_0$.

Lemma 20. *The term $e_i^n : \tau_{i+2}$ is a n -repeater of order i .*

Proof. Proof per induction on i .

$i = 0$: $e_0^n m v = [m]^n(v)$ per definition.

$i + 1$: If $(f_j^{n_j})_{j < i}$ are n_j -repeaters of order j then

$$e_i^n f_{i-1}^{n_{i-1}} f_{i-2}^{n_{i-2}} \dots m v \rightsquigarrow [f_{i-1}^{n_{i-1}}]^n(f_{i-2}^{n_{i-2}}) \dots m v \rightsquigarrow^* [m]^k(v)$$

with $k = \text{tower}(n_0, \dots, n_{i-1}, n)$ by Lemma 19. Observe that for $i = 1$ we have $f_{i-2}^{n_{i-2}} = m$ which does not cause any issues.

Combining all previous results allows us to define the terms

$$S_n^N(t) := [e_{n-2}^2]^N(e_{n-3}^2 e_{n-4}^2 \dots e_0^2 B t)$$

We parameterize $S_n^N(t)$ over a term t because we employ it to derive both the lower bound on reduction chain length and on language size.

Lemma 21.

1. We have $S_n^N(t) \rightsquigarrow^* T_{2^{n-1}(N)}(t)$.
2. For a fixed n , we have $s(S_n^N(t)) = O(N)$.
3. We have $\text{deg}(S_n^N(t)) = n$.

Proof.

1. As the e_i^2 are all n -repeaters of order i by Lemma 20, we know by Lemma 19 that

$$S_n^N(t) = [e_{n-2}^2]^N(e_{n-3}^2) \dots B t \rightsquigarrow^* [B]^{2^{n-1}(N)}(t)$$

and furthermore by Fact 17 that $[B]^{2^{n-1}(N)}(t) \rightsquigarrow^* T_{2^{n-1}(N)}(t)$.

2. Observe that for arbitrary m and v we have $s([m]^N(v)) = s(v) + N(s(m) + 1)$ and thus that $s(e_i^n) = n + i + 1$ and $s([e_{n-2}^2]^N(e_{n-3}^2)) = n + N(n + 1) = O(N)$. Together, this yields

$$s(S_n^N(t)) = \underbrace{s(t) + s(B) + 2 + \sum_{j=0}^{j < n-3} (s(e_j^2) + 1)}_{\text{constant for fixed } n} + \underbrace{s([e_{n-2}^2]^N(e_{n-3}^2))}_{O(N)} = O(N)$$

3. Observe that $\text{deg}(S_n^N(t)) = \text{ord}(e_{n-2}^2) = n$ as $e_i^2 : \tau_{i+2}$.

Corollary 22. $ds_n(N) = 2_n(\Omega(N))$

Proof. Fix some terminal $\text{off} \in \mathcal{T}$ and non-terminal on with $(\text{on} \mapsto \text{off}) \in \mathcal{N}$. By Lemma 21, the terms $D_n^s(N) := S_n^N(\text{on})$ have $s(D_n^s(N)) = O(N)$ and $\deg(D_n^s(N)) = n$. Furthermore, $D_n^s(N) \rightsquigarrow^* T_{2n-1(N)}(\text{on})$ and by Lemma 16 $d(T_{2n-1(N)}(\text{on})) \geq 2_n(N)$, meaning $d(D_n^s(N)) \geq 2_n(N)$ overall, yielding the desired lower bound.

Unfortunately, this approach does not extend directly to the definition of $H_n^N(t)$. If we were to take $H_n^N(t) := S_n^{2N}(t)$ this would yield the desired $d(H_n^N(t)) \rightsquigarrow^* T_{2n(N)}(t)$ however, we would also have $h(H_n^N(t)) = O(2^N)$. We thus have to take an additional ‘‘compression step’’, squeezing $[e_{n-2}^2]^{2^N}(\cdot)$ into a term $h(c_i^N) = O(N)$. For this, we take advantage of the fact a binary tree of height N may have 2^N nodes.

$$c_i^1 : \tau_{i+2} := e_i^2 \quad c_i^{N+1} : \tau_{i+2} := f \vec{x} \mapsto [c_i^N]^2(f) \vec{x}$$

Lemma 23. The term $c_i^N : \tau_{i+2}$ is a 2^{2^N} -repeater of order i .

Proof. Denote by $(f_j^{n_j})_{j < i}$ a family n_j -repeaters of order j . Proof per induction on N .

$N = 1$: Follows directly from Lemma 20 as $2^{2^0} = 2$.

$N + 1$: Then

$$c_i^{N+1} f_{i-1}^{n_{i-1}} \dots m v \rightsquigarrow [c_i^N]^2 (f_{i-1}^{n_{i-1}}) \dots m v \rightsquigarrow [m]^k(v)$$

with $k = \text{tower}(n_0, \dots, n_{i-1}, 2^{2^N}, 2)$ by Lemma 19 and the inductive hypothesis. As $(2^{2^N})^2 = 2^{2^{N+1}}$ we then obtain $k = \text{tower}(n_0, \dots, n_{i-1}, 2^{2^{N+1}})$ as desired.

Now we may simply take $H_n^N(t) := c_{n-2}^N e_{n-3}^2 \dots e_0^2 B t$.

Lemma 24.

1. We have $H_n^N(t) \rightsquigarrow^* T_{2n(N)}(t)$.
2. For a fixed n , we have $h(H_n^N(t)) = O(N)$.
3. We have $\deg(H_n^N(t)) = n$.

Proof.

1. We obtain

$$H_n^N(t) = c_{n-2}^N e_{n-3}^2 \dots B t \rightsquigarrow^* [B]^{2n(N)}(t) \rightsquigarrow^* T_{2n(N)}(t)$$

by Lemma 23 and Fact 17.

2. Inductively we can see $h(c_i^N) \leq N(i+1) = O(N)$ and thus

$$h(H_n^N(t)) = \max\{\underbrace{h(c_i^N)}_{O(N)}, \underbrace{h(x e_{n-2}^2 \dots B t)}_{\text{constant for fixed } n}\} = O(N)$$

3. Simply observe that $\deg(H_n^N(t)) = \text{ord}(c_{n-2}^N) = n$.

Corollary 25. $dh_n(N) = 2_{n+1}(\Omega(N))$

5. Language size bounds

To be able to state the language size bounds as a closed expression, one needs² to separate terms into classes according to one further parameter.

Definition 26. For any sequence \mathcal{N} of non-terminal assignments, we define its *branching factor* $\text{bf}(\mathcal{N})$ as the least n such that all non-terminal mappings $(a := \vec{x} \mapsto t_1^a \mid \dots \mid t_m^a) \in \mathcal{N}$ have $m \leq n$.

² An alternative approach, as taken by Afshari et al. [2], is to fix a global upper bound on the branching factor in the HORS under consideration.

Overall, we obtain the bounds

$$l_s^k(N) \leq k^{2n(N)} \quad lh_n(N) \leq k^{2n+1(N)}$$

where k is the branching factor of the terms under consideration.

The strategy for obtaining the upper bound on language size follows that employed by Afshari et al. [2] to derive such upper bounds from their reduction chain length bounds.

Lemma 27. *Let $\mathcal{N}; \mathcal{T}; \Gamma \vdash s : \tau$ then $l(s) \leq \text{bf}(\mathcal{N})^{d(s)}$.*

Proof. Per induction on $d(s)$. If $d(s) = 0$ then s is normal and thus $L(s) = \{s\}$ meaning $l(s) = 1 = \text{bf}(\mathcal{N})^0$. Thus let $d(s) > 1$. Then let s_1, \dots, s_n be the set of expressions arrived at by reducing the left-most redex in s . It is well known that $L(s) = \bigcup \{L(s_i) \mid i \leq n\}$. By the inductive hypothesis, we know that $l(s_i) \leq \text{bf}(\mathcal{N})^{d(s_i)}$ for each i . Thus we obtain

$$l(s) \leq \sum_{i=1}^n l(s_i) = \sum_{i=1}^n \text{bf}(\mathcal{N})^{d(s_i)} \leq \text{bf}(\mathcal{N})^{d(s)}$$

because $n \leq \text{bf}(\mathcal{N})$ and $d(s) \geq d(s_i) - 1$ for each $i \leq n$.

Corollary 28. *We obtain the bounds*

$$l_s^k(N) \leq k^{2n(N)} = 2_{n+1}(O(N)) \quad \text{and} \quad lh_n(N) \leq k^{2n+1(N)} = 2_{n+2}(O(N)).$$

The strategy to obtaining the lower bounds is very similar to that in Beckmann [1] and Section 4.2: Define terms which “blow up” to a sufficient degree and many copies of a non-deterministic non-terminal, which in turn generate a large language. The proof also requires a result relating binary tree expressions with their resulting language size, similar to Lemma 16 for the reduction chain length bounds.

Lemma 29. *Let $n \leq l(s)$ then $n^{2^K} \leq l(T_K(s))$.*

Proof. Proof per induction on K . The case for $K = 0$ is trivial. For the $K + 1$ case,

$$l(T_{K+1}(s)) \geq l(T_K(s))^2 \stackrel{\text{IH}}{\geq} (n^{2^K})^2 = n^{2^{K+1}}$$

because $L(T_{K+1}(s)) \supseteq \{bt't' \mid t, t' \in L(T_K(s))\}$.

Theorem 30. *There exist terms $L_{n,k}^s(N)$ and $L_{n,k}^h(N)$ of branching-factor k with $s(L_{n,k}^s(N)) = h(L_{n,k}^h(N)) = O(N)$ and $\text{deg}(L_{n,k}^s(N)) = \text{deg}(L_{n,k}^h(N)) = n$. Furthermore,*

$$l(L_{n,k}^s(N)) \geq k^{2n(N)} \quad \text{and} \quad l(L_{n,k}^h(N)) \geq k^{2n+1(N)}.$$

Proof. Fix distinct terminals $1, \dots, k : 0$ and a non-terminal $a : 0 := 1 \mid \dots \mid k$. Then take $L_{n,k}^s(N) := S_n^N(a)$ and observe the fact that $L_{n,k}^s(N) \rightsquigarrow^* T_{2n-1}(a)$ and thus $l(L_{n,k}^s(N)) \geq k^{2n(N)}$ by Lemma 29. Similarly, take $L_{n,k}^h(N) := H_n^N(a)$.

Corollary 31.

$$l_s^k(N) = 2_{n+2}(\Omega(N)) \quad lh_n^k(N) = 2_{n+1}(\Omega(N))$$

and thus

$$l_s^k(N) = 2_{n+2}(\Theta(N)) \quad lh_n^k(N) = 2_{n+1}(\Theta(N)).$$

6. Discussion

We have obtained bounds on the lengths of reduction chains and the sizes of the languages generated by classes of acyclic HORS and have proven their exactness in the sense of Beckmann [1]. Key to these results was adapting Beckmann’s method of expanded head reduction trees to acyclic HORS. A similar undertaking can be found in the work of Clairambault [8,9] who adapts Beckmann’s method to obtain bounds on the length of plays in game semantics for higher-order programming languages.

A drawback of Beckmann's method, as he presents it in [1] and our adaptation, is that it does not easily apply to basic extensions of the systems presented. In cut-free derivations, the next applicable rule is always determined by the left-most syntactic element of the term. In contrast, consider the following rewriting rule for the projections on pairs, added to either the simply typed λ -calculus or HORS:

$$\pi_i (s_1, s_2) \rightsquigarrow s_i$$

With this definition, an expression $\pi_i s$, with s not being of the form (s_1, s_2) , cannot take the π -reduction step. This means there is no expanded head reduction tree derivation rule which applies to all expressions of the form $\pi_i s t$, at least not in the style employed by Beckmann in [1]. This problem was addressed by Beckmann and Weiermann in [13] where they give bounds for terms of Gödel's T, including the recursion combinator, which raises the same problems as the projections π_i . Instead of applying rules based on the left-most syntactic element, the applicability of these rules is determined by the term's "head redex", which may occur deep inside the term under consideration.

In [2], Afshari et al. translate derivations in the classical first-order sequent calculus with CUT into AHORS whose generated language are the derivation's Herbrand expansions. To estimate the number of Herbrand expansions which may be generated in this manner, they derive reduction chain length and language size bounds for AHORS by embedding AHORS into the λ -calculus and applying Beckmann's result [1]. To express their bound, they employ a notion of expression size which does not recurse on non-terminals. That is, for an expression $\mathcal{N}; \mathcal{T}; \Gamma \vdash t : \tau$ their size function $s'(t)$ is defined as below:

$$\begin{aligned} s'(b) &= 0 & b \text{ a terminal} \\ s'(x) &= 1 & x \text{ a variable} \\ s'(a) &= 1 & a \text{ a non-terminal} \\ s'(t t') &= s'(t) + s'(t') \end{aligned}$$

For a given expression³ $\mathcal{N}; \mathcal{T}; \Gamma \vdash t : \tau$, they then derive a reduction chain length bound of $2^{\frac{(|\mathcal{N}|+1)(k+1)}{\deg(t)+1}}$ where

$$k = \max\{\max\{s'(t_i) \mid 0 < i \leq k\} \mid a : \tau := \vec{x} \mapsto t_1^a \mid \dots \mid t_k^a \in \mathcal{N}\} \cup \{s'(t)\}$$

which, at face value, is exponentially larger than our bound of $2^{\frac{s(t)}{\deg(t)}}$. However, this difference can be attributed to the fact that our notion of size can be exponentially larger than theirs. To illustrate this, fix terminals $b : 0 \rightarrow 0 \rightarrow 0$ and $l : 0$ as well as the non-terminals below and consider $t := a_n l$.

$$a_0 x : 0 \rightarrow 0 \mapsto b x x \quad a_{n+1} x : 0 \rightarrow 0 \mapsto (a_n x) (a_n x)$$

For an \mathcal{N} which contains the definitions for a_0, \dots, a_n we have $|\mathcal{N}| = n + 1$ and $k = s'(b (a_n x) (a_n x)) = 4$, meaning $(|\mathcal{N}| + 1)(k + 1) = 5(n + 2)$. On the other hand, $s(t) = 2^{n+3} - 2$. Thus, in the worst-case scenario of "exponentially growing" AHORS, both bounds can be seen to agree as the exponential "saved" in our bound is reintroduced by our notion of size. It should be noted that for the application of the bounds in [2], our bounds are tighter than those derived by Afshari et al. because the size of the AHORS under consideration is linear in the size of the sequent calculus derivation for both notions of AHORS size, meaning the exponential worst-case illustrated above does not occur.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

No data was used for the research described in the article.

References

- [1] A. Beckmann, Exact bounds for lengths of reductions in typed λ -calculus, J. Symb. Log. 66 (3) (2001) 1277–1285, <https://doi.org/10.2307/2695106>, publisher: Cambridge University Press.
- [2] B. Afshari, S. Hetzl, G.E. Leigh, Herbrand's theorem as higher order recursion, Ann. Pure Appl. Log. 171 (6) (2020) 102792, <https://doi.org/10.1016/j.apal.2020.102792>.

³ We have adjusted the bound to be in terms of a starting expression t , as we present it, rather than in terms of a complete AHORS as it is presented in [2].

- [3] U. Dal Lago, O. Laurent, Quantitative game semantics for linear logic, in: M. Kaminski, S. Martini (Eds.), *Computer Science Logic*, in: *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2008, pp. 230–245.
- [4] D. de Carvalho, M. Pagani, L. Tortora de Falco, A semantic measure of the execution time in linear logic, *Theor. Comput. Sci.* 412 (20) (2011) 1884–1902, <https://doi.org/10.1016/j.tcs.2010.12.017>.
- [5] S.J. Lengrand, A. Bernadet, Non-idempotent intersection types and strong normalisation, *Log. Methods Comput. Sci.* 9 (4) (Oct. 2013), [https://doi.org/10.2168/LMCS-9\(4:3\)2013](https://doi.org/10.2168/LMCS-9(4:3)2013), publisher: Episciences.org.
- [6] H. Schwichtenberg, An upper bound for reduction sequences in the typed λ -calculus, *Arch. Math. Log.* 30 (5) (1991) 405–408, <https://doi.org/10.1007/BF01621476>.
- [7] H. Xi, Upper bounds for standardizations and an application, *J. Symb. Log.* 64 (1) (1999) 291–303, <https://doi.org/10.2307/2586765>, publisher: Cambridge University Press.
- [8] P. Clairambault, Estimation of the length of interactions in arena game semantics, in: M. Hofmann (Ed.), *Foundations of Software Science and Computational Structures*, in: *Lecture Notes in Computer Science*, Springer, Berlin, Heidelberg, 2011, pp. 335–349.
- [9] P. Clairambault, Bounding linear head reduction and visible interaction through skeletons, *Log. Methods Comput. Sci.* 11 (2) (Jun. 2015), [https://doi.org/10.2168/LMCS-11\(2:6\)2015](https://doi.org/10.2168/LMCS-11(2:6)2015).
- [10] N. Kobayashi, Model checking higher-order programs, *J. ACM* 60 (3) (2013) 20, <https://doi.org/10.1145/2487241.2487246>.
- [11] C.-H. Ong, On model-checking trees generated by higher-order recursion schemes, in: 21st Annual IEEE Symposium on Logic in Computer Science, LICS'06, 2006, pp. 81–90, ISSN 1043-6871, <https://doi.org/10.1109/LICS.2006.38>.
- [12] R. Suzuki, K. Fujima, N. Kobayashi, T. Tsukada, Streect automata model checking of higher-order recursion schemes, in: D. Miller (Ed.), 2nd International Conference on Formal Structures for Computation and Deduction, FSCD 2017, in: *Leibniz International Proceedings in Informatics (LIPIcs)*, ISSN 1868-8969, vol. 84, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, Dagstuhl, Germany, 2017, 32.
- [13] A. Beckmann, A. Weiermann, Analyzing Gödel's T via expanded head reduction trees, *Math. Log. Q.* 46 (4) (2000) 517–536, [https://doi.org/10.1002/1521-3870\(200010\)46:4<517::AID-MALQ517>3.0.CO;2-A](https://doi.org/10.1002/1521-3870(200010)46:4<517::AID-MALQ517>3.0.CO;2-A).