Infinitary Action Logic with Multiplexing

Stepan L. Kuznetsov, Stanislav O. Speranski

This is a pre-print version of the article
published in *Studia Logica*
DOI: 10.1007/s11225-022-10021-6

Abstract

Infinitary action logic can be naturally expanded by adding exponential and subexponential modalities from linear logic. In this article we shall develop infinitary action logic with a subexponential that allows multiplexing (instead of contraction). Both non-commutative and commutative versions of this logic will be considered, presented as infinitary sequent calculi. We shall prove cut admissibility for these calculi, and estimate the complexity of the corresponding derivability problems: in both cases it will turn out to be between complete first-order arithmetic and the ω^ω level of the hyperarithmetical hierarchy. Here the complexity upper bound is much lower than that for the system with a subexponential that allows contraction. The complexity lower bound in turn is much higher than that for infinitary action logic.

Keywords: Lambek calculus · infinitary action logic · multiplexing · complexity · closure ordinal

Contents

1 Introduction

The *multiplicative-additive Lambek calculus*, denoted by **MALC**, is an extension of the Lambek calculus [14] with additive connectives, and nowadays can be viewed as a non-commutative intuitionistic variant of Girard's *linear logic* [5]. From this point of view, it is natural to extend it with exponential or subexponential modalities [7]. These modalities allow structural rules of contraction, exchange, and weakening, which **MALC**, being a substructural logic, does not allow for arbitrary formulae. The exponential modality allows all these rules (for formulae under this modality), while subexponential ones may allow only some of them. We shall write $\mathbf{SMALC}_\Sigma$ for the extension of **MALC** with a family Σ of subexponential modalities.

On the other hand, **MALC** is the core of *action logic,* the algebraic logic of residuated Kleene lattices, or action lattices [19, 9]. Besides operations of **MALC**, action logic also includes a unary operation of iteration, called the *Kleene star*. In this article, we shall be interested in the algebraic logic of a subclass of action lattices called **-continuous* ones. This logic is called *infinitary action logic* and denoted by $\mathbf{ACT}_\omega$ [3, 18, 4]. In $\mathbf{ACT}_\omega$, the Kleene star is axiomatised by an ω-rule, thus making the whole system infinitary. Extending $\mathbf{ACT}_\omega$ further with a family Σ of subexponential modalities results in a system called $!_\Sigma\mathbf{ACT}_\omega$ [12].

Although both $\mathbf{SMALC}_\Sigma$ and $\mathbf{ACT}_\omega$ are undecidable, they have a rather moderate degree of undecidability: $\mathbf{ACT}_\omega$ is Π^0_1-complete [3, 18, 4], while $\mathbf{SMALC}_\Sigma$ is Σ^0_1-complete [15, 7], provided at least one subexponential in Σ allows contraction (otherwise the system is decidable).

However, combining both a subexponential allowing contraction and the Kleene star raises complexity dramatically: if Σ contains a subexponential for contraction, then $!_\Sigma\mathbf{ACT}_\omega$ is Π^1_1-complete [12], hence not even hyperarithmetical.

This huge complexity gap motivates the search for natural systems of intermediate complexity level, which somehow involve both Kleene star and some sorts of subexponential modalities. Notice that completely disallowing contraction for subexponentials does not give such a system, since in this case we return to the Π^0_1 upper bound [12].

One such fragment of $!\mathbf{ACT}_\omega$, which has an intermediate complexity level, was introduced in [10]. This fragment is defined by the following syntactic restriction: the Kleene star is not allowed to appear under the exponential. This restriction makes complexity hyperarithmetical (the fragment belongs to Δ^1_1); on the other hand, it is Π^0_2-hard, which is a lower bound higher than both Σ^0_1 and Π^0_1.

Such a restriction, however, may look artificial. In this article, we give a new example of a system with intermediate complexity, in which, instead of an external constraint on syntax, we use a modified set of rules for the subexponential modality.

Namely, we introduce an extension of infinitary action logic with a subexponential modality for *multiplexing* instead of contraction. The difference is as follows. Contraction just copies formulae under !, keeping the ! at its place and allowing its reusing upper in the proof tree:

$$\frac{\ldots !A \ldots !A \ldots \vdash C}{\ldots !A \ldots \vdash C}$$

On the other hand, multiplexing is an *introduction rule* for !, which can be used for copying the formula only once:

$$\frac{\ldots \overbrace{AA\ldots A}^{n \text{ times}} \ldots \vdash C}{\ldots !A \ldots \vdash C}$$

The finitary system of intuitionistic non-commutative linear logic with multiplexing was introduced in [8]. This system is also equipped with another subexponential, ∇, which is used for permutation. In this article, we also consider a commutative version of the system with multiplexing, where ∇ is unnecessary. The rules for introducing subexponential modalities to the right-hand side of sequents are also different from those in linear logic, and follow the ideas of "light" and "soft" linear logic [6, 13]. An accurate formulation of the calculi we consider is given in Section 2.1 below.

For infinitary action logic with multiplexing, both commutative and non-commutative, we get tighter complexity bounds than those in [10]. More precisely, the lower bound is complete first-order arithmetic, while the upper bound is $\Sigma^0_{\omega^\omega}$, i.e. the ω^ω level of the hyperarithmetical hierarchy. In order to prove the latter, we show that the closure ordinals for the derivability operators for both systems are less than or equal to ω^ω. For comparison, for $!_\Sigma\mathbf{ACT}_\omega$, when at least one subexponential allows contraction, the closure ordinal is ω_1^{CK} (the highest possible, which reflects Π^1_1-completeness) [12].

The motivation for this extension of infinitary action logic is as follows. Unlike rules used in linear logic, the multiplexing rule for ! is exactly dual to the ω-rule introducing the Kleene star.

Namely, while * stands for the "for all" quantifier, ! models the "exists" one. Complexitywise, the system obtained is quite unusual for a propositional logic: it is highly likely that its complexity lies exactly in the hyperarithmetical range.

Semantics for infinitary action logic extended with modalities for multiplexing is a subtle issue, and we leave it for further research. On one hand, $\mathbf{ACT}_\omega$ itself has natural algebraic models, which are *-continuous action algebras [19, 9, 4]. On the other hand, for soft linear logic [13], which employs multiplexing, there is also a line of research towards semantics [20]. These approaches could probably be extended to the Kleene star, at least in the commutative case. This seems to be an interesting question for future research.

The rest of this article is organised as follows. Section 2 includes the necessary preliminaries. Here we define the syntax of our calculi and prove that, thanks to the ∇ modality, the commutative system can be embedded into the non-commutative one. We also recall the basics of the theory of inductive definitions and hyperarithmetical hierarchy, which will be needed further in our arguments. In Section 3 we introduce a rank function on sequents and use it for proving an upper bound of ω^ω on the closure ordinal. Section 4 contains a sketch of cut elimination proof. In Sections 5 and 6, we prove lower and upper complexity bounds, respectively. Finally, Section 7 is a concluding one, stating some open questions for future research.

2 Preliminaries

2.1 Infinitary action logic with multiplexing

Let us introduce an extension of infinitary action logic [4] with two subexponential modalities ! and ∇ [8]. The first one allows the so-called *multiplexing* rule, and the second one is for controlled permutation. The system which we define here will be denoted by $!^{\mathsf{m}}\nabla\mathbf{ACT}_\omega$.

We start with **MALC**, multiplicative-additive Lambek calculus, as the basic logic, formulated as a sequent calculus. Formulae of **MALC** are built from variables $(p, q, r, \ldots)$ and constants $\mathbf{0}$ and $\mathbf{1}$ using five binary operations: $\cdot$ (product, or multiplicative conjunction), $\backslash$ (left division), $/$ (right division), $\wedge$ (additive conjunction), and $\vee$ (additive disjunction). Sequents are expressions of the form $\Pi \vdash A$, where A is a formula and Π is a sequence of formulae, possibly empty.

The axioms and inference rules of **MALC** are as follows:

$$\frac{}{A \vdash A}\ (\mathrm{id})$$

$$\frac{\Pi \vdash A \quad \Gamma, B, \Delta \vdash C}{\Gamma, \Pi, A \backslash B, \Delta \vdash C}\ (\backslash L) \qquad \frac{A, \Pi \vdash B}{\Pi \vdash A \backslash B}\ (\backslash R)$$

$$\frac{\Pi \vdash A \quad \Gamma, B, \Delta \vdash C}{\Gamma, B / A, \Pi, \Delta \vdash C}\ (/ L) \qquad \frac{\Pi, A \vdash B}{\Pi \vdash B / A}\ (/ R)$$

$$\frac{\Gamma, A, B, \Delta \vdash C}{\Gamma, A \cdot B, \Delta \vdash C}\ (\cdot L) \qquad \frac{\Gamma \vdash A \quad \Delta \vdash B}{\Gamma, \Delta \vdash A \cdot B}\ (\cdot R)$$

$$\frac{\Gamma, \Delta \vdash C}{\Gamma, \mathbf{1}, \Delta \vdash C}\ (\mathbf{1}L) \qquad \frac{}{\vdash \mathbf{1}}\ (\mathbf{1}R) \qquad \frac{}{\Gamma, \mathbf{0}, \Delta \vdash C}\ (\mathbf{0}L)$$

$$\frac{\Gamma, A_1, \Delta \vdash C \quad \Gamma, A_2, \Delta \vdash C}{\Gamma, A_1 \vee A_2, \Delta \vdash C}\ (\vee L) \qquad \frac{\Pi \vdash A_i}{\Pi \vdash A_1 \vee A_2}\ (\vee R_i),\ i = 1, 2$$

$$\frac{\Gamma, A_i, \Delta \vdash C}{\Gamma, A_1 \wedge A_2, \Delta \vdash C}\ (\wedge L_i),\ i = 1, 2 \qquad \frac{\Pi \vdash A_1 \quad \Pi \vdash A_2}{\Pi \vdash A_1 \wedge A_2}\ (\wedge R)$$

The calculus $!^{\mathsf{m}}\boldsymbol{\nabla}\mathbf{ACT}_\omega$ is obtained from **MALC** by adding three unary connectives, * (Kleene star), ! (multiplexing subexponential) and ∇ (permuting subexponential) with the following rules. Here and further A^n means $A, \ldots, A$ (n times); in particular, A^0 is the empty sequence.

$$\frac{(\Gamma, A^n, \Delta \vdash C)_{n \in \omega}}{\Gamma, A^*, \Delta \vdash C}\ ({*}L_\omega)$$

$$\frac{}{\vdash A^*}\ ({*}R_0) \qquad \frac{\Pi_1 \vdash A \quad \ldots \quad \Pi_n \vdash A}{\Pi_1, \ldots, \Pi_n \vdash A^*}\ ({*}R_n),\ n > 0 \text{ and each } \Pi_i \text{ is non-empty}$$

$$\frac{\Gamma, A^n, \Delta \vdash C}{\Gamma, !A, \Delta \vdash C}\ (!L_n),\ n \in \omega \qquad \frac{A \vdash B}{!A \vdash !B}\ (!R)$$

$$\frac{\Gamma, A, \Delta \vdash C}{\Gamma, \nabla A, \Delta \vdash C}\ (\nabla L) \qquad \frac{A \vdash B}{\nabla A \vdash \nabla B}\ (\nabla R)$$

$$\frac{\Gamma, B, \nabla A, \Delta \vdash C}{\Gamma, \nabla A, B, \Delta \vdash C}\ (\nabla P_1) \qquad \frac{\Gamma, \nabla A, B, \Delta \vdash C}{\Gamma, B, \nabla A, \Delta \vdash C}\ (\nabla P_2)$$

Notice that cut is not included as an official rule into the system. Thus, cut elimination will be formulated as *cut admissibility*—the set of derivable sequents is closed under cut—and proved below in Section 4.

The non-emptiness restriction on the $({*}R_n)$ rule does not actually alter the set of derivable sequents. Indeed, if a Π_j is empty, we may just remove the corresponding premise, which will make the rule even stronger. If all Π_i's happen to be empty, then $({*}R_n)$ reduces to the $({*}R_0)$ axiom. This non-emptiness restriction, however, helps in analysis of derivations.

Since our calculus includes an ω-rule, $({*}L_\omega)$, we should be careful when defining derivations and derivability. Here we use the same approach as in [12], which follows Aczel [1, Definition 1.4.4] and Buchholz [2, § 1]. Let us briefly recall it. Traditionally, the derivability of a sequent is defined as the existence of a derivation for it. A derivation, in turn, is a well-founded (but possibly infinitely branching) labelled tree $\mathfrak{T}$ such that each vertex v of $\mathfrak{T}$ is labelled by a sequent, and this sequent can be obtained from the sequents labelling the children of v by applying one of the inference rules. In particular, axioms are treated as nullary inference rules; thus the leaves of $\mathfrak{T}$ are labelled by instances of axioms. Below (see Proposition 2.2 and Section 2.2) we shall also discuss alternative, but equivalent, ways of defining derivability.

We also consider a commutative version of $!^{\mathsf{m}}\boldsymbol{\nabla}\mathbf{ACT}_\omega$, denoted by $!^{\mathsf{m}}\mathbf{CommACT}_\omega$. This commutative version is obtained from $!^{\mathsf{m}}\boldsymbol{\nabla}\mathbf{ACT}_\omega$ by adding the unrestricted permutation rule:

$$\frac{\Gamma, B, A, \Delta \vdash C}{\Gamma, A, B, \Delta \vdash C}\ (P)$$

and removing ∇, which is now unnecessary. Alternatively, one can reformulate the language of sequents, making their left-hand sides multisets rather than sequences. In $!^{\mathsf{m}}\mathbf{CommACT}_\omega$, two divisions, $A \setminus B$ and $B \,/\, A$, are equivalent, and we denote them by $A \multimap B$. The notion of derivability in $!^{\mathsf{m}}\mathbf{CommACT}_\omega$ is defined in the same way as in $!^{\mathsf{m}}\boldsymbol{\nabla}\mathbf{ACT}_\omega$.

In fact, $!^{\mathsf{m}}\mathbf{CommACT}_\omega$ can be conservatively embedded into $!^{\mathsf{m}}\boldsymbol{\nabla}\mathbf{ACT}_\omega$ using the ∇ modality. More precisely, we define two translations of formulae from $!^{\mathsf{m}}\mathbf{CommACT}_\omega$, denoted by $A^{\nabla+}$ and $A^{\nabla-}$, by joint recursion:

$$\begin{array}{ll}
p^{\nabla+} = p & p^{\nabla-} = \nabla p \\
\mathbf{0}^{\nabla+} = \mathbf{0} & \mathbf{0}^{\nabla-} = \nabla\mathbf{0} \\
\mathbf{1}^{\nabla+} = \mathbf{1} & \mathbf{1}^{\nabla-} = \nabla\mathbf{1} \\
(A \multimap B)^{\nabla+} = A^{\nabla-} \setminus B^{\nabla+} & (A \multimap B)^{\nabla-} = \nabla(A^{\nabla+} \setminus B^{\nabla-}) \\
(A \cdot B)^{\nabla+} = A^{\nabla+} \cdot B^{\nabla+} & (A \cdot B)^{\nabla-} = \nabla(A^{\nabla-} \cdot B^{\nabla-}) \\
(A \wedge B)^{\nabla+} = A^{\nabla+} \wedge B^{\nabla+} & (A \wedge B)^{\nabla-} = \nabla(A^{\nabla-} \wedge B^{\nabla-}) \\
(A \vee B)^{\nabla+} = A^{\nabla+} \vee B^{\nabla+} & (A \vee B)^{\nabla-} = \nabla(A^{\nabla-} \vee B^{\nabla-}) \\
(A^*)^{\nabla+} = (A^{\nabla+})^* & (A^*)^{\nabla-} = \nabla\big((A^{\nabla-})^*\big) \\
(!A)^{\nabla+} = !A^{\nabla+} & (!A)^{\nabla-} = \nabla !A^{\nabla-}
\end{array}$$

Theorem 2.1.
A sequent $A_1, \ldots, A_n \vdash B$ is derivable in $!^{\mathsf{m}}\mathbf{CommACT}_\omega$ iff its translation $A_1^{\nabla-}, \ldots, A_n^{\nabla-} \vdash B^{\nabla+}$ is derivable in $!^{\mathsf{m}}\boldsymbol{\nabla}\mathbf{ACT}_\omega$.

Proof. The given $!^{\mathsf{m}}\mathbf{CommACT}_\omega$ derivation is translated into the corresponding $!^{\mathsf{m}}\boldsymbol{\nabla}\mathbf{ACT}_\omega$ derivation. The translation is straightforward. When a new formula A is introduced to the left-hand side of the sequent, we add an extra (∇L) application which adds the ∇ needed to form $A^{\nabla-}$. The same happens with the (id) axiom. Introducing new formulae to the right is translated as is. Finally, permutation (P) is translated into (∇P_1), since each formula $A_i^{\nabla-}$ in the left-hand side is of the form ∇F.

For the backwards translation, we just take the $!^{\mathsf{m}}\boldsymbol{\nabla}\mathbf{ACT}_\omega$ derivation of the translated sequent and remove all ∇'s. This gives a valid derivation in $!^{\mathsf{m}}\mathbf{CommACT}_\omega$ (in particular, (∇P_1) and (∇P_2) transform into (P)) of the original sequent. ∎

We finish this section with an alternative, but equivalent way to define derivability:

Proposition 2.2.
Let $\mathbf{L} \in \{!^{\mathsf{m}}\nabla\mathbf{ACT}_\omega, !^{\mathsf{m}}\mathbf{CommACT}_\omega\}$*. A sequent is derivable in* $\mathbf{L}$ *(i.e. there exists a derivation* $\mathfrak{T}$ *in the sense defined above) iff it belongs to the least (w.r.t. inclusion) set of sequents which is closed under the inference rules of* $\mathbf{L}$*.*

Proof. See [12, Proposition 2.2], where the same was proved for a closely related system $!\mathbf{ACT}_\omega$. The proof does not depend on the concrete form of inference rules. □

In Section 2.2 below, we shall reformulate this alternative definition in terms of the least fixed point of the immediate derivability operator on sets of sequents.

2.2 Inductive definitions

To simplify our discussion, let

$$\begin{aligned} \mathsf{Ord} &:= \text{ the class of all ordinals,} \\ \mathsf{LOrd} &:= \text{ the class of all limit ordinals,} \\ \mathsf{COrd} &:= \text{ the class of all constructive ordinals.}^{1} \end{aligned}$$

The least element of $\mathsf{Ord} \setminus \mathsf{COrd}$ is traditionally called the *Church–Kleene ordinal*, and denoted by ω_1^{CK}. Since COrd forms an initial segment of Ord, we have

$$\mathsf{COrd} \;=\; \left\{\alpha \in \mathsf{Ord} \mid \alpha < \omega_1^{\mathrm{CK}}\right\}.$$

Naturally, if one wishes to use transfinite recursion in an effective way, it is reasonable to focus attention on ordinals below ω_1^{CK}.

Let F be a monotone function from $\mathcal{P}(\omega)$ to $\mathcal{P}(\omega)$, i.e. for any $P, Q \subseteq \omega$,

$$P \subseteq Q \quad\Longrightarrow\quad F(P) \subseteq F(Q).$$

Then for each $S \subseteq \omega$ we can inductively define

$$F^{\alpha}(S) \;:=\; \begin{cases} S & \text{if } \alpha = 0, \\ F\left(F^{\beta}(S)\right) & \text{if } \alpha = \beta + 1, \\ \bigcup_{\beta<\alpha} F^{\beta}(S) & \text{if } \alpha \in \mathsf{LOrd} \setminus \{0\}. \end{cases}$$

Evidently, the resulting transfinite sequence is monotone as a class function from Ord to $\mathcal{P}(\omega)$, i.e. $\alpha \leqslant \beta$ implies $F^{\alpha}(S) \subseteq F^{\beta}(S)$. This observation quickly leads to:

Folklore 2.3.
Let F be a monotone function from $\mathcal{P}(\omega)$ to $\mathcal{P}(\omega)$. Then for every $S \subseteq \omega$, if $S \subseteq F(S)$, then there exists $\alpha \in \mathsf{Ord}$ such that $F^{\alpha+1}(S) = F^{\alpha}(S)$ — so $F^{\alpha}(S)$ is the least fixed point of F containing S.

[1] Recall that an ordinal α is *constructive*, or *computable*, iff there exists a computable well-ordering (on a subset of ω) that has order-type α; see [21, §§11.7–8] for more information.

By the *closure ordinal* of a monotone $F : \mathcal{P}(\omega) \to \mathcal{P}(\omega)$ we mean the least $\alpha \in \mathsf{Ord}$ such that $F^{\alpha+1}(\varnothing) = F^{\alpha}(\varnothing)$. This ordinal indicates how many steps are needed to get the least fixed point of F, which exists by Folklore 2.3.

Denote by $\mathfrak{N}$ the standard model of arithmetic. It is convenient to assume that the signature of $\mathfrak{N}$ contains a symbol for every computable function or relation. We shall be concerned with monotone operators definable in $\mathfrak{N}$. Let $\mathcal{L}_2$ be the language of monadic second-order logic based on the signature of $\mathfrak{N}$.[2] So $\mathcal{L}_2$ includes two sorts of variables:

- *individual variables* $x, y, \ldots$ (intended to range over ω);
- *set variables* $X, Y, \ldots$ (intended to range over $\mathcal{P}(\omega)$).

Accordingly one needs to distinguish between *individual* and *set quantifiers*, viz.

$$\exists x,\ \forall x,\ \exists y,\ \forall y,\ \ldots \quad \text{and} \quad \exists X,\ \forall X,\ \exists Y,\ \forall Y,\ \ldots$$

Let $n \in \omega \setminus \{0\}$. Recall that an $\mathcal{L}_2$-formula is in Σ^0_n (Π^0_n) iff it has the form

$$\underbrace{\exists \vec{x}_1\, \forall \vec{x}_2\, \ldots\, \vec{x}_n}_{n-1 \text{ alternations}}\ \Psi \quad \text{(respectively } \underbrace{\forall \vec{x}_1\, \exists \vec{x}_2\, \ldots\, \vec{x}_n}_{n-1 \text{ alternations}}\ \Psi)$$

where $\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n$ are tuples of individual variables and Ψ is quantifier-free. We say that a subset of ω belongs to Σ^0_n (Π^0_n) iff it is definable in $\mathfrak{N}$ by a Σ^0_n-formula (Π^0_n-formula). Further, an $\mathcal{L}_2$-formula is in Σ^1_1 (Π^1_1) iff it has the form

$$\exists \vec{X}\, \Psi \quad \text{(respectively } \forall \vec{X}\, \Psi)$$

where $\vec{X}$ is a tuple of set variables and Ψ contains no set quantifiers. Naturally, a subset of ω belongs to Σ^1_1 (Π^1_1) iff it is definable in $\mathfrak{N}$ by a Σ^1_1-formula (Π^1_1-formula).

For each $\mathcal{L}_2$-formula $\Phi(x, X)$ (where x is an individual variable and X is a set variable) we define the function $[\Phi]$ from $\mathcal{P}(\omega)$ to $\mathcal{P}(\omega)$ by

$$[\Phi](S) \;:=\; \{n \in \omega \mid \mathfrak{N} \models \Phi(n, S)\}.$$

Now call $\Phi(x, X)$ *positive* iff no free occurrence of X in Φ is in the scope of an odd number of nested negations, provided $\to$ is treated as defined using $\neg$ and $\vee$. Obviously,

$$\Phi(x, X) \text{ is positive} \quad \Longrightarrow \quad [\Phi] \text{ is monotone.}$$

We say that a function F from $\mathcal{P}(\omega)$ to $\mathcal{P}(\omega)$ is *elementary* iff $F = [\Phi]$ for some positive $\mathcal{L}_2$-formula $\Phi(x, X)$ with no set quantifiers.[3]

Folklore 2.4 (cf. Theorem 1D.3, Corollary 2B.3 and Theorem 8D.1 in [17])**.**
Let F be an elementary function from $\mathcal{P}(\omega)$ to $\mathcal{P}(\omega)$. Then:

[2]Here the restriction to monadic formulae is not essential because first-order arithmetic allows us to code tuples of natural numbers as natural numbers.

[3]For discussion and related results one may consult [17].

i. the least fixed point of F belongs to Π^1_1;

ii. the closure ordinal of F is less than or equal to ω_1^{CK}.

Consider $\mathbf{L} \in \{!^{\mathsf{m}}\nabla\mathbf{ACT}_\omega, !^{\mathsf{m}}\mathbf{CommACT}_\omega\}$. We associate with $\mathbf{L}$ its *immediate derivability operator* $\mathscr{D}_{\mathbf{L}}$ on the sets of sequents as follows: for any sequent s and set of sequents S,

$$s \in \mathscr{D}_{\mathbf{L}}(S) \quad \Longleftrightarrow \quad \begin{array}{c} s \text{ is an element of } S \text{ or } s \text{ can be obtained from} \\ \text{elements of } S \text{ by one application of some rule of } \mathbf{L}. \end{array}$$

Notice that the least fixed point of $\mathscr{D}_{\mathbf{L}}$ coincides with the collection of all sequents derivable in $\mathbf{L}$. Assuming some effective Gödel numbering for sequents, we can identify $\mathscr{D}_{\mathbf{L}}$ with a function from $\mathcal{P}(\omega)$ to $\mathcal{P}(\omega)$. Finally, it is straightforward to check that this function is elementary.[4] So Folklore 2.4 applies to $\mathscr{D}_{\mathbf{L}}$.

2.3 Hyperarithmetical hierarchy

Given $S \subseteq \omega$, take $\mathtt{U}^S$ to be one's favourite partial S-computable two-place function on ω that is universal for the class of all partial S-computable one-place functions on ω. For each $n \in \omega$ we use $\mathtt{U}^S_n$ to denote the n-th projection of $\mathtt{U}^S$, i.e.

$$\mathtt{U}^S_n := \lambda m.\left[\mathtt{U}^S(n,m)\right].$$

If $S = \varnothing$, the superscript S in $\mathtt{U}^S$ can be omitted. Further, we write $\leqslant$ for many-one reducibility and $\leqslant_T$ for Turing reducibility. So for any $S, P \subseteq \omega$:

$$\begin{array}{rcl} S \leqslant P & \Longleftrightarrow & \text{there exists } n \in \omega \text{ such that } \mathtt{U}_n \text{ is total and } S = (\mathtt{U}_n)^{-1}[P]; \\ S \leqslant_T P & \Longleftrightarrow & \text{there exists } n \in \omega \text{ such that } \chi_S = \mathtt{U}^P_n. \end{array}$$[5]

The corresponding equivalence relations are denoted by $\equiv$ and $\equiv_T$ respectively.

Take $\mathtt{J}$ to be the Turing jump operator on the powerset of ω, which can be defined by

$$\mathtt{J}(S) := \left\{n \in \omega \mid \mathtt{U}^S(n,n) \text{ converges}\right\}.$$

The importance of this operator emerges from the following.

[4] To this end, it suffices to show that for each rule R of $\mathbf{L}$ there exists a positive $\mathcal{L}_2$-formula $\Phi_{\mathrm{R}}(x, X)$ with no set quantifiers such that for any sequent s and set of sequents S,

$$\mathfrak{N} \models \Phi_{\mathrm{R}}(s, S) \quad \Longleftrightarrow \quad \begin{array}{c} s \text{ can be obtained from} \\ \text{elements of } S \text{ by one application of R} \end{array}$$

(where the rule scheme $(*R_n)$ is treated as a single rule). In effect, a perfectly analogous argument was used in the proof of Proposition 5.1 in [12].

[5] Here χ_A is the characteristic function of A.

Folklore 2.5.
Let $S \subseteq \omega$. Then for every $P \subseteq \omega$,

$$P \text{ is computably enumerable in } S \quad \Longleftrightarrow \quad P \text{ is many-one reducible to } \mathtt{J}(S).$$

Thus every subset of ω computably enumerable in S is many-one reducible to $\mathtt{J}(S)$, and $\mathtt{J}(S)$ itself is computably enumerable in S.

Among other things, it leads to an alternative characterisation of the arithmetical hierarchy:

Folklore 2.6.
Let $n \in \omega \setminus \{0\}$. Then for every $S \subseteq \omega$,

$$S \text{ belongs to } \Sigma^0_n \quad \Longleftrightarrow \quad S \text{ is many-one reducible to } \mathtt{J}^n(\varnothing)$$

where $\mathtt{J}^n(\varnothing)$ denotes the result of applying the jump operator n times to $\varnothing$. Moreover, the sets belonging to Π^0_n are the complements of those belonging to Σ^0_n.

Roughly speaking, the *hyperarithmetical hierarchy* is obtained by iterating $\mathtt{J}$ over COrd. To do this properly, one needs *Kleene's system of notation for* COrd, which consists of:

- a special partial function $|\cdot|_O$ from ω onto COrd;
- a special ordering relation $<_O$ on $\operatorname{dom}|\cdot|_O$ which mimics $\in$ on COrd.[6]

We say that $n \in \omega$ is a *notation for* $\alpha \in \mathsf{COrd}$ iff $|n|_O = \alpha$. All notations of this kind are of the form 2^m or $3 \cdot 5^m$. Moreover, $|\cdot|_O$ has the following properties:

- the ordinal 0 receives only one notation, namely 1;
- if $\alpha = |2^m|_O$ where $m \neq 0$, then $\alpha \notin \mathsf{LOrd}$ and $\alpha = |m|_O + 1$;
- if $\alpha = |3 \cdot 5^m|_O$, then $\alpha \in \mathsf{LOrd}$ and $\alpha = \sup\left\{|k|_O \mid k <_O 3 \cdot 5^m\right\}$.

Therefore every finite ordinal receives only one notation, viz.

$$0 = |1|_O, \quad 1 = \left|2^1\right|_O, \quad 2 = \left|2^{2^1}\right|_O, \quad \ldots$$

We shall write $n \in O$ instead of $n \in \operatorname{dom}|\cdot|_O$ and often omit the subscript O in $|n|_O$. Intuitively, each $n \in O$ encodes a program that computes some well-ordering having order-type $|n|$. Please see [21] or [22] for more about constructive ordinals and systems of notation.

Folklore 2.7.
There exists a computable $\gamma : \omega \longrightarrow \omega$ such that for every $n \in O$,

$$\operatorname{dom} \mathtt{U}_{\gamma(n)} = \{k \mid k <_O n\}.$$

Thus the restriction of $<_O$ to $\{k \mid k <_O n\}$ is computably enumerable uniformly in n.

[6]As usual, if f is a partial function, we write $\operatorname{dom} f$ for its domain.

Now at the heart of the *hyperarithmetical hierarchy* is an indexed family $\langle H(n) : n \in O \rangle$ of subsets of ω such that

$$H(n) = \begin{cases} \varnothing & \text{if } n = 1 \\ \mathtt{J}(H(m)) & \text{if } n = 2^m \neq 1 \\ \{\mathsf{c}(i,j) \mid i \in H(j) \text{ and } j <_O 3 \cdot 5^m\} & \text{if } n = 3 \cdot 5^m \end{cases}$$

where c denotes one's favourite computable bijection from $\omega \times \omega$ onto ω (e.g. the well-known Cantor pairing function). This family behaves nicely with respect to $|\cdot|$:

Folklore 2.8.
For any $m, n \in O$, if $|m| = |n|$, then $H(m) \equiv_T H(n)$.

Let Σ^0_0 be the collection of all computable subsets of ω. For each $n \in O \setminus \{1\}$:

- if $|n| < \omega$, then take $\Sigma^0_{|n|}$ to be $\{S \subseteq \omega \mid S \leqslant H(n)\}$;
- if $|n| \geqslant \omega$, then take $\Sigma^0_{|n|}$ to be $\{S \subseteq \omega \mid S \leqslant \mathtt{J}(H(n))\}$.

These are well-defined because of Folklore 2.8. Also, for every $\alpha \in \mathsf{COrd}$ we define

$$\Pi^0_\alpha := \left\{\omega \setminus S \mid S \in \Sigma^0_\alpha\right\} \quad \text{and} \quad \Delta^0_\alpha := \Sigma^0_\alpha \cap \Pi^0_\alpha.$$

This gives us the *hyperarithmetical hierarchy*. Evidently, its initial segment of type ω coincides with the arithmetical hierarchy (by Folklore 2.6). As one would expect, we have:

Folklore 2.9.
For any $\alpha, \beta \in \mathsf{COrd}$:

- $\Sigma^0_\alpha \setminus \Pi^0_\alpha \neq \varnothing$ *and* $\Pi^0_\alpha \setminus \Sigma^0_\alpha \neq \varnothing$;
- $\Sigma^0_\alpha \cup \Pi^0_\beta \subsetneq \Sigma^0_{\alpha+1} \cap \Pi^0_{\beta+1}$.

Call $S \subseteq \omega$ *hyperarithmetical* iff $S \in \Delta^0_\alpha$ for some $\alpha \in \mathsf{COrd}$.

Folklore 2.10.
For each $S \subseteq \omega$ the following conditions are equivalent:

i. S is hyperarithmetical;

ii. S belongs to Δ^1_1, i.e. to both Σ^1_1 and Π^1_1.

3 Ranking sequents

In this section we shall prove an upper bound on the closure ordinals for $!^{\mathsf{m}}\mathbf{\nabla ACT}_\omega$ and $!^{\mathsf{m}}\mathbf{CommACT}_\omega$. We introduce *ranking* on formulae and sequents. This ranking is not the same as the one used by Palka [18], but the idea is similar. Each sequent will receive a rank, which is an ordinal below ω^ω, and inference rules (except permutation rules) will strictly increase the rank (see Lemma 3.1 below). This enables induction on the rank for derivable sequents. The definition is basically the same as in [12], but now ! is handled in the same way as *.

Let $\mathcal{N}$ be the set of all sequences of natural numbers that eventually stabilize at zero, i.e.

$$\mathcal{N} \;=\; \{(m_0, m_1, m_2, \ldots, m_n, \ldots) \mid \exists i_0\, \forall i \geq i_0\, m_i = 0\}.$$

On this set, we define the anti-lexicographical ordering, point-wise sum, lifting, and unit:

$$\begin{aligned}
&(m_0, m_1, \ldots) \prec (n_0, n_1, \ldots) \iff \exists j_0\, (m_{j_0} < n_{j_0} \text{ and } \forall j > j_0\; m_j = n_j);\\
&(m_0, m_1, \ldots) \boxplus (n_0, n_1, \ldots) = (m_0 + n_0, m_1 + n_1, \ldots);\\
&(m_0, m_1, \ldots)\!\uparrow\, = (0, m_0, m_1, \ldots);\\
&\iota = (1, 0, 0, \ldots).
\end{aligned}$$

Now we are ready to rank both formulae and sequents. The following definition is formulated for $!^{\mathsf{m}}\boldsymbol{\nabla}\mathbf{ACT}_\omega$; for $!^{\mathsf{m}}\mathbf{CommACT}_\omega$, one needs to remove the ∇ case. Given a formula A, we define $\eta(A)$, called the *rank of A*, by recursion:

$$\begin{aligned}
&\eta(p_i) = \iota \quad \text{for each variable } p_i;\\
&\eta(\mathbf{0}) = \eta(\mathbf{1}) = \iota;\\
&\eta(A \setminus B) = \eta(B \,/\, A) = \eta(A \cdot B) = \eta(A \wedge B) = \eta(A \vee B) = \eta(A) \boxplus \eta(B) \boxplus \iota;\\
&\eta(A^*) = \eta(!A) = (\eta(A)\!\uparrow) \boxplus \iota;\\
&\eta(\nabla A) = \eta(A) \boxplus \iota.
\end{aligned}$$

As for sequents, we take $\eta(A_1, \ldots, A_n \vdash B)$ to be $\eta(A_1) \boxplus \ldots \boxplus \eta(A_n) \boxplus \eta(B)$.

The ordering $\prec$ on $\mathcal{N}$ is linear and well-founded, and it is isomorphic to ω^ω by the following isomorphism $\nu \colon \mathcal{N} \to \omega^\omega$:

$$\nu((m_0, m_1, m_2, \ldots, m_n, \ldots)) = \ldots + \omega^n \cdot m_n + \ldots + \omega^2 \cdot m_2 + \omega \cdot m_1 + m_0.$$

(The sum here is always finite.)

Lemma 3.1.
All rules of our calculi, except permutation—(∇P_1) and (∇P_2) for $!^{\mathsf{m}}\boldsymbol{\nabla}\mathbf{ACT}_\omega$ and (P) for $!^{\mathsf{m}}\mathbf{CommACT}_\omega$—have the following property: the rank of each premise is strictly less, in the sense of $\prec$, than the rank of the conclusion.

For this lemma, it is important that we do not have cut as an official rule in our systems. For permutation rules, the rank does not change, so these rules are excluded from the lemma, and below we shall use special tricks to handle them. In particular, this lemma holds for $!L_n$ and $*L_\omega$, since for ! and * the rank is multiplied by ω, which is bigger than any natural number. A formal proof of Lemma 3.1 follows.

Proof of Lemma 3.1. Let us first notice that $\boxplus$ is commutative and monotone w.r.t. $\prec$: if $\alpha \prec \beta$, then $\alpha \boxplus \gamma \prec \beta \boxplus \gamma$ (here $\alpha, \beta, \gamma \in \mathcal{N}$). Moreover, we have $\alpha \prec \alpha \boxplus \beta$ if $\beta \neq (0, 0, 0, \ldots)$ (in particular, this holds if $\beta = \iota$ and if $\beta = \eta(A)$ for some A).

Therefore, for each rule we may ignore the context (whose rank could only increase) and compare only the ranks of active formulae. For each binary connective $\star$ we have $\eta(A \star B) = \eta(A) \boxplus \eta(B) \boxplus \iota \succ \eta(A) \boxplus \eta(B)$, whence $\eta(A \star B) \succ \eta(A)$ and $\eta(A \star B) \succ \eta(B)$. This gives increasing of rank for the rules introducing binary connectives. For $(\mathbf{1}L)$, the rank increases by $\eta(\mathbf{1}) = \iota$.

The interesting case is the case of unary connectives, *, $!$, and ∇ (for $!^{\mathsf{m}}\mathbf{\nabla ACT}_\omega$). For ∇, we just use the fact that $\eta(\nabla A) = \eta(A) \boxplus \iota \succ \eta(A)$ and, in the case of (∇R), the same for ∇B.

For * and $!$, we notice that $\eta(A^*)$ and $\eta(!A)$ are greater (in the sense of $\succ$) than $\eta(A) \boxplus \ldots \boxplus \eta(A)$, n times, for any natural number n. Indeed, if $\eta(A) = (m_0, m_1, \ldots, m_k, 0, \ldots)$, where m_k is the rightmost non-zero element, then $\eta(A)\!\uparrow$ has a non-zero element at the $(k+1)$-st position, which makes it greater. This consideration yields the necessary result for $(*L_\omega)$ and $(!L_n)$. For $(*R_n)$ and $(!R)$, we use its particular case: $\eta(A^*) \succ \eta(A)$, and the same for $!A$ and $!B$. $\square$

The ranking on sequents and Lemma 3.1 allow us to prove — in a way similar to [12, Theorem 5.13] — that the closure ordinals for the corresponding operators are less than or equal to ω^ω. We are going to consider both $!^{\mathsf{m}}\mathbf{\nabla ACT}_\omega$ and $!^{\mathsf{m}}\mathbf{CommACT}_\omega$ simultaneously, as there is no significant difference whether permutations may be applied to arbitrary formulae or only to ∇-formulae.

Theorem 3.2.
Let $\mathbf{L} \in \{!^{\mathsf{m}}\mathbf{\nabla ACT}_\omega, !^{\mathsf{m}}\mathbf{CommACT}_\omega\}$. *Then the closure ordinal for* $\mathscr{D}_{\mathbf{L}}$ *is at most* ω^ω.

The proof is rather straightforward, the only subtle thing is the handling of permutation rules. This issue is dealt with by reformulating the system using *generalised rules.* A generalised rule is a rule which is not permutation, followed by several applications of permutation rules below. After replacing each rule in the system by its generalised version, permutation rules themselves become unnecessary, and one can remove them. Indeed, permutation rules could appear either below other rules (and get absorbed by generalisation) or below axioms, where they are meaningless.

Proof of Theorem 3.2. If we consider the formulation of $\mathbf{L}$ with generalised rules and without permutation rules, which we denote by $\mathbf{L}_G$, then we readily have, for each derivable sequent s, that $s \in \mathscr{D}_{\mathbf{L}_G}^{\nu(\eta(s))}(\varnothing)$, which gives the ω^ω upper bound on the closure ordinal, since $\nu(\eta(s)) < \omega^\omega$ for any s.

However, we wish to prove the upper bound for the original formulation (with permutation rules), and this involves multiplying by ω. Namely, we prove by transfinite induction that $s \in \mathscr{D}_{\mathbf{L}}^{\omega \cdot \nu(\eta(s))}(\varnothing)$ for each derivable sequent s. By definition of rank, $\eta(s)$ is always an element of $\mathcal{N}$ with $m_0 \neq 0$. Thus, $\nu(\eta(s))$ is always a successor ordinal, i.e., $\nu(\eta(s)) = \beta + 1$ for some β. Consider a derivation of s in $\mathbf{L}_G$. If s is an axiom, it is also an axiom of $\mathbf{L}$, so $s \in \mathscr{D}_{\mathbf{L}}^{1}(\varnothing) \subseteq \mathscr{D}_{\mathbf{L}}^{\omega \cdot \nu(\eta(s))}(\varnothing)$, since $\nu(\eta(s)) > 0$.

Otherwise, consider the lowermost generalised rule in the derivation of s. Such a generalised rule consists of two phases: a rule which is not permutation, which derives a sequent $\tilde{s}$, and then a series of k permutations, which transform $\tilde{s}$ to s. Here k is a natural number, $k < \omega$. Since permutation does not change the rank, we have $\nu(\eta(\tilde{s})) = \nu(\eta(s)) = \beta + 1$. On the other side, for each premise s' of the rule which derived $\tilde{s}$, we have $\nu(\eta(s')) < \nu(\eta(\tilde{s}))$ by Lemma 3.1, since this rule is not permutation.

Thus, $\nu(\eta(s')) \leqslant \beta$, and by induction hypothesis we have $s' \in \mathscr{D}_{\mathbf{L}}^{\omega\cdot\beta}(\varnothing)$ for each premise s'. The goal sequent s is derived from these premises in $1 + k$ steps (the "main" rule plus k permutations), therefore, $s \in \mathscr{D}_{\mathbf{L}}^{\omega\cdot\beta+1+k}(\varnothing)$. We conclude by noticing that $\omega \cdot \beta + 1 + k < \omega \cdot (\beta + 1) = \omega \cdot \nu(\eta(s))$, therefore $s \in \mathscr{D}_{\mathbf{L}}^{\omega\cdot\nu(\eta(s))}$.

Since $\nu(\eta(s)) < \omega^\omega$, we see that the closure ordinal is bounded by $\omega \cdot \omega^\omega = \omega^\omega$. □

4 Cut admissibility

Another usage of sequent ranks and generalised rules, which were introduced in the previous section, is cut elimination. Since cut was not included in our systems, cut elimination appears as *cut admissibility:* the set of derivable sequents is closed under cut. In this section, we again consider $!^{\mathsf{m}}\nabla\mathbf{ACT}_\omega$ and $!^{\mathsf{m}}\mathbf{CommACT}_\omega$ simultaneously.

Theorem 4.1.
Let $\mathbf{L} \in \{!^{\mathsf{m}}\nabla\mathbf{ACT}_\omega, !^{\mathsf{m}}\mathbf{CommACT}_\omega\}$. *Then if* $\Pi \vdash A$ *and* $\Gamma, A, \Delta \vdash C$ *are derivable in* $\mathbf{L}$*, then so is* $\Gamma, \Pi, \Delta \vdash C$.

The proof has much in common with the proof for the finitary system without Kleene star [8], so we only sketch the proof, omitting routine proof transformations.

Proof. We proceed by nested induction. The outer induction parameter is the complexity of A, measured just as the number of connectives. The inner one is the rank of the goal sequent, $\nu(\eta(\Gamma, \Pi, \Delta \vdash C))$.

Let us consider derivations of $\Pi \vdash A$ and $\Gamma, A, \Delta \vdash C$ in $\mathbf{L}_G$ and the lowermost generalised rules used in these derivations. If at least one of the sequents is the (id) axiom, cut trivialises. The same happens if $\Pi \vdash A$ is the $(\mathbf{0}L)$ axiom. The right cut premise, $\Gamma, A, \Delta \vdash C$, cannot be the $(\mathbf{1}R)$ axiom. As for $(\mathbf{1}R)$ as the left cut premise and for $(\mathbf{0}L)$ as the right one, it will be more convenient for us to consider them below as zero-premise rules of inference rather than axioms.

As usual, we call a generalised rule application *principal,* if it introduces the formula A being cut. In particular, $!R$ and ∇R are always principal, since they introduce both formulae, on the left and on the right.

Now we consider three cases.

Case 1. The lowermost rule application in the derivation of $\Pi \vdash A$ is a non-principal one. This means that the rule operates inside Π, and it can be exchanged with cut. More precisely, since the rule is generalised, $\Pi \vdash A$ is derived by permutations from $\widetilde{\Pi} \vdash A$, and the latter is derived from some premises (maybe infinitely many). For each premise s, we have

$\eta(s) \prec \eta(\widetilde{\Pi} \vdash A) = \eta(\Pi \vdash A)$. Therefore, for a premise s of the form $\Phi \vdash A$, we have $\eta(\Phi) \prec \eta(\Pi)$. If we apply cut with $\Gamma, A, \Delta \vdash C$, we get the sequent $\Gamma, \Phi, \Delta \vdash C$, whose rank is smaller than that of $\Gamma, \Pi, \Delta \vdash C$. Thus, we may apply induction hypothesis and establish cut-free derivability of new premises of the form $\Gamma, \Phi, \Delta \vdash C$. (For $(/\, L)$ and $(/\, R)$, there is also an extra premise with another formula on the right, which is just kept as is.) Applying the original rule, which is still valid with Γ and Δ added, we get $\Gamma, \Pi, \Delta \vdash C$.

Case 2. The lowermost rule application in the derivation of $\Gamma, A, \Delta \vdash C$ is a non-principal one. Again, we propagate the cut upwards by exchanging it with the non-principal rule.

Case 3. Both rules are principal. Here we have to consider possible cases on the structure of formula A. The cases where the main connective in A is a **MALC** connective, or if A is constant **1**, are rather standard, going back to Lambek's original paper [14]. The only difference is that now rules are generalised, i.e., we add permutations. For example, if $A = E \,/\, F$, we have

$$\dfrac{\dfrac{\dfrac{\widetilde{\Pi}, F \vdash E}{\widetilde{\Pi} \vdash E \,/\, F}\,(/\, R)}{\dfrac{\cdots}{\Pi \vdash E \,/\, F}} \qquad \dfrac{\dfrac{\widetilde{\Delta}_1 \vdash F \quad \widetilde{\Gamma}, E, \widetilde{\Delta}_2 \vdash C}{\widetilde{\Gamma}, E \,/\, F, \widetilde{\Delta}_1, \widetilde{\Delta}_2 \vdash C}\,(/\, L)}{\dfrac{\cdots}{\Gamma, E \,/\, F, \Delta \vdash C}}}{\Gamma, \Pi, \Delta \vdash C}\,(\mathrm{cut})$$

Here $\widetilde{\Pi}$ is a permutation of Π (in the non-commutative case, permutation is allowed only for ∇-formulae) and $\widetilde{\Gamma}, E \,/\, F, \widetilde{\Delta}_1, \widetilde{\Delta}_2$ is a permutation of $\Gamma, E \,/\, F, \Delta$. Since E and F are simpler than A, we may apply induction hypothesis (outer induction) and get cut-free derivability first of $\widetilde{\Gamma}, \widetilde{\Pi}, F, \widetilde{\Delta}_2 \vdash C$ (cutting E), and then of $\widetilde{\Gamma}, \widetilde{\Pi}, \widetilde{\Delta}_1, \widetilde{\Delta}_2 \vdash C$ (cutting F). Our goal sequent is now obtained by permutation.

Other **MALC** connectives are considered similarly. As for constant **0**, it has no right principal rule, so it cannot appear in Case 3.

The interesting cases are those with modalities: *, !, and, in the non-commutative case, ∇.

For $A = E^*$, if the rule introducing $\Pi \vdash A$ is the generalised version of $(*R_k)$, $k > 0$, we have the following:

$$\dfrac{\dfrac{\dfrac{\widetilde{\Pi}_1 \vdash E \quad \ldots \quad \widetilde{\Pi}_k \vdash E}{\widetilde{\Pi}_1, \ldots, \widetilde{\Pi}_k \vdash E^*}\,(*R_k)}{\dfrac{\cdots}{\Pi \vdash E^*}} \qquad \dfrac{\dfrac{\big(\widetilde{\Gamma}, E^n, \widetilde{\Delta} \vdash C\big)_{n \in \omega}}{\widetilde{\Gamma}, E^*, \widetilde{\Delta} \vdash C}\,(*L_\omega)}{\dfrac{\cdots}{\Gamma, E^*, \Delta \vdash C}}}{\Gamma, \Pi, \Delta \vdash C}\,(\mathrm{cut})$$

Here, again, $\widetilde{\Pi}_1, \ldots, \widetilde{\Pi}_n$ is obtained by permutation from Π, the same for $\widetilde{\Gamma}, E^*, \widetilde{\Delta}$ and Γ, E^*, Δ. Out of the premises of $(*L_\omega)$, we take the one with $n = k$ and apply the induction hypotheses (outer induction) k times, for the simpler formula E. This gives cut-free derivability of $\widetilde{\Gamma}, \widetilde{\Pi}, \widetilde{\Delta} \vdash C$, and our goal sequent is obtained by permutation.

The case where $\Pi \vdash E^*$ is the $(*R_0)$ axiom is simpler. Here Π should be empty, so in the derivation on the right we take the premise with $n = 0$, which is $\widetilde{\Gamma}, \widetilde{\Delta} \vdash C$, and obtain the goal sequent $\Gamma, \Delta \vdash C$ by permutation.

For $A = {!E}$, the left cut premise, $\Pi \vdash {!E}$, should be introduced by $(!R)$. This means that $\Pi = {!F}$ for some F. For $\Gamma, {!E}, \Delta \vdash C$, we have two cases, depending on which rule was used for deriving this sequent. It could be either $(!L)$ or $(!R)$, both are principal. In the case of $(!L)$, we have

$$\dfrac{\dfrac{F \vdash E}{!F \vdash !E}\ (!R) \qquad \dfrac{\dfrac{\dfrac{\widetilde{\Gamma}, E^n, \widetilde{\Delta} \vdash C}{\widetilde{\Gamma}, !E, \widetilde{\Delta} \vdash C}\ (!L_n)}{\cdots}}{\Gamma, !E, \Delta \vdash C}}{\Gamma, !F, \Delta \vdash C}\ (\mathrm{cut})$$

Here, again, $\widetilde{\Gamma}, !E, \widetilde{\Delta}$ is obtained by permutation from $\Gamma, !E, \Delta$; for $!F \vdash !E$, no permutations are possible.

Applying the induction hypothesis (outer induction) n times, for formula E, we get cut-free derivability of $\widetilde{\Gamma}, F^n, \widetilde{\Delta} \vdash C$. The generalised version of $(*L_n)$ now yields our goal sequent, $\Gamma, !F, \Delta \vdash C$.

Now let the lowermost rule in the derivation of $\Gamma, !E, \Delta \vdash C$ be $(!R)$. This means that Γ and Δ are empty and $C = !G$ for some formula G:

$$\dfrac{\dfrac{F \vdash E}{!F \vdash !E}\ (!R) \qquad \dfrac{E \vdash G}{!E \vdash !G}\ (!R)}{!F \vdash !G}\ (\mathrm{cut})$$

Again, we apply the outer induction, and get cut-free derivability of $F \vdash G$; the goal sequent $!F \vdash !G$ is obtained by $(!R)$.

The case of ∇ is similar to the case of !, since (∇R) has the same form as $(!R)$ and (∇L) has the same form as $(!L_1)$. $\square$

5 Encoding first-order arithmetic

In this section we shall prove a lower bound on the complexity of the derivability problems for $!^{\mathsf{m}}\mathbf{CommACT}_\omega$ and $!^{\mathsf{m}}\boldsymbol{\nabla}\mathbf{ACT}_\omega$. More precisely, we are going to show that the first-order theory of $\mathfrak{N}$ (i.e., complete first-order arithmetic) is many-one reducible to each of these problems. This will be first done in the commutative case, using an encoding of counter (Minsky) machines developed in [11].

A counter machine operates several counters; also it has its internal state, taken from a finite set $\{q_0, \ldots, q_m\}$. Let q_0 be the designated initial state, in which the machine starts its operation. Each of the counters keeps a natural number. The counter machine can perform two kinds of operations: (1) increase a counter by 1 and change the state; (2) conditionally decrease the counter. The latter means the following. If the counter is non-zero, it gets decreased by 1 and the machine changes the state. If the counter is zero, it is not modified, but the state is changed in a different way.

Counter machines are a Turing-complete model of computation: three counters are sufficient for computing any computable function on natural numbers [23].[7]

[7] One can further restrict to two counters, but in this case the input/output data should be specifically encoded [16, 23], which is less convenient.

We shall consider counter machines with many inputs. Such a machine has a designated counter for each input and three additional counters to be used inside the computation. Such machines, again, are capable of computing any function (with the given number of arguments), which is computable, e.g. on a Turing machine. Moreover, the translation from the Turing machine to the counter machine is itself computable. Just as with Turing machines, we may assume an appropriate coding of counter machines:

$$\mathtt{M}_0,\ \mathtt{M}_1,\ \mathtt{M}_2,\ \ldots$$

Given a machine $\mathtt{M}$ with s counters, we write $(q; n_1, \ldots, n_s)$ for the configuration of $\mathtt{M}$ where it is in state q with values $n_1, \ldots n_s$ of the counters. In effect, we shall be interested only in checking whether a given machine halts on a given configuration, and not in the value returned.

Now let us fix a natural number k and consider an arbitrary machine $\mathtt{M}$ with $2k+3$ counters. By $(*)$ let us denote the following condition:

$$\forall n_1 > 0\, \exists n_2 > 0\, \forall n_3 > 0 \ldots \exists n_{2k} > 0\, (\mathtt{M} \text{ does not halt on } (q_0; n_1, \ldots, n_{2k}, 0, 0, 0)).$$

Here the counters with numbers $2k+1$, $2k+2$, and $2k+3$ are used inside the computation, and the first $2k$ ones contain the input of $\mathtt{M}$.

For the given $k \in \omega$ consider the mass problem

$$\mathcal{S}_k \ := \ \left\{ j \in \omega \mid \mathtt{M}_j \text{ has } 2k+3 \text{ counters and obeys } (*) \right\}.$$

It is well-known that $\mathcal{S}_k$ is Π^0_{2k}-complete (see [21, Chapter 14]). In particular, $\mathcal{S}_0$ is simply the non-halting problem; it has been encoded in $\mathbf{CommACT}_\omega$ in [11]. We shall extend this encoding to $\mathcal{S}_k$ for an arbitrary k.

Let $Q = \{q_0, \ldots, q_m\}$ be the set of states of $\mathtt{M}$, and let $\mathcal{QZ} = \{q_0, \ldots, q_m, z_1, \ldots, z_{2k+3}\}$ be its extension with pseudo-states for zero checks (z_i checks for the i-th counter to be zero). Then the configuration $(q; n_1, \ldots, n_{2k+3})$ of $\mathtt{M}$, where q is a state or a pseudo-state, is encoded in our sequents by $q, a_1^{n_1}, \ldots, a_{2k+3}^{n_{2k+3}}$. Let D be the regular expression for well-formed configurations:

$$D = \left(\bigvee_{q \in Q} q \cdot a_1^* \cdot \ldots \cdot a_{2k+3}^* \right) \vee \left(\bigvee_{i=1}^{2k+3} z_i \cdot a_1^* \cdot \ldots \cdot a_{i-1}^* \cdot a_{i+1}^* \cdot \ldots \cdot a_{2k+3}^* \right).$$

Next, each instruction (operation) I of $\mathtt{M}$ is encoded by a specific formula A_I [11]. For instructions of the first kind (increment), $A_I = q \multimap (q' \cdot a_i)$, encoding "increase the i-the counter and change the state from q to q'." For instruction of the second kind (decrement), $A_I = ((q \cdot a_i) \multimap q') \wedge (q \multimap (q'' \vee z_i))$, which takes care of both zero and non-zero cases for the i-th counter: "if the i-th counter is non-zero, decrease it and change the state from q to q'; else, change the state from q to q''." Now by E we denote the big conjunction of these formulae for all instructions of $\mathtt{M}$:

$$E = \bigwedge_I A_I.$$

The main encoding statement can be formulated as follows:

Theorem 5.1 ([11][8]).
Machine $\mathtt{M}$ *does not halt on* $(q_0; n_1, \dots, n_{2k+3})$ *iff the following sequent is derivable in* $\mathbf{CommACT}_\omega$*:*

$$q_0, a_1^{n_1}, \dots, a_{2k+3}^{n_{2k+3}}, E^* \vdash D.$$

Now let us add the quantifier prefix, $\forall n_1 \, \exists n_2 \, \forall n_3 \, \dots \, \exists n_{2k}$, using a variant of the "key-and-lock" construction by Lincoln et al. [15].

Let us adopt the standard abbreviation $A^+ = A \cdot A^*$. For each $m = 1, \dots, 2k$, let us define

$$K_m \ := \ \begin{cases} p_{m-1} \multimap (a_m^+ \cdot p_m) & \text{if } m \text{ is odd;} \\ p_{m-1} \multimap \,!(a_m \cdot !p_m) & \text{if } m \text{ is even.} \end{cases}$$

(Here $p_0, p_1, \dots, p_{2k}$ are fresh variables.) Also let $B := q_0 \cdot E^*$. Now our main encoding theorem is as follows:

Theorem 5.2.
Machine $\mathtt{M}$ *satisfies* $(*)$ *iff the following sequent is derivable in* $!^{\mathsf{m}}\mathbf{CommACT}_\omega$*:*

$$p_0, K_1, \dots, K_{2k}, p_{2k} \multimap B \vdash D.$$

This theorem will be proved by induction, and for the induction step we formulate the following lemma:

Lemma 5.3.
(1) The sequent

$$a_1^{n_1}, \dots, a_{2i}^{n_{2i}}, p_{2i}, K_{2i+1}, K_{2i+2}, \dots, K_{2k}, p_{2k} \multimap B \vdash D$$

is derivable iff the following sequent is derivable **for any** *non-zero value of* n_{2i+1}*:*

$$a_1^{n_1}, \dots, a_{2i}^{n_{2i}}, a_{2i+1}^{n_{2i+1}}, p_{2i+1}, K_{2i+2}, \dots, K_{2k}, p_{2k} \multimap B \vdash D.$$

(2) The sequent

$$a_1^{n_1}, \dots, a_{2i-1}^{n_{2i-1}}, p_{2i-1}, K_{2i}, K_{2i+1}, \dots, K_{2k}, p_{2k} \multimap B \vdash D$$

is derivable iff the following sequent is derivable **for some** *non-zero value of* n_{2i}*:*

$$a_1^{n_1}, \dots, a_{2i-1}^{n_{2i-1}}, a_{2i}^{n_{2i}}, p_{2i}, K_{2i+1}, \dots, K_{2k}, p_{2k} \multimap B \vdash D.$$

Before proving Lemma 5.3, let us formulate several technical statements. All of them are based on the following balancing condition. In an axiom, we have one positive and one negative occurrence of a variable. Therefore, in the goal sequent a positive occurrence of a variable, unless it is introduced inside the derivation, should correspond to a negative one.

[8] In fact, this is a slight modification of the construction from [11], using $2k + 3$ instead of 3.

Lemma 5.4.
For any variable p, the following two statements hold. If $\Pi \vdash p$ is derivable, then Π includes a positive occurrence p. If $\Pi, p \multimap F \vdash D$ is derivable, then Π includes a positive occurrence of p. (Here D is as defined above.)

Lemma 5.5.
If $\Pi \vdash D$ is derivable, then Π includes a positive occurrence of a variable from $\mathcal{QZ}$.

Lemma 5.6.
If $\Pi \vdash p_i$ is derivable and Π includes a K_{2i+1} (which includes a positive occurrence of a^+_{2i+1}), then Π should also include a negative occurrence of a_{2i+1}.

Proof of Lemma 5.3. For the "if" direction, we explicitly construct the derivations of the sequents in question (recall the definitions of K_{2i+1} and K_{2i}):

(1)

$$\dfrac{p_{2i} \vdash p_{2i} \qquad \dfrac{\dfrac{(a_1^{n_1}, \dots, a_{2i}^{n_{2i}}, a_{2i+1}^{n}, p_{2i+1}, K_{2i+2}, \dots, K_{2k}, p_{2k} \multimap B \vdash D)_{n>0}}{a_1^{n_1}, \dots, a_{2i}^{n_{2i}}, a_{2i+1}, a^*_{2i+1}, p_{2i+1}, K_{2i+2}, \dots, K_{2k}, p_{2k} \multimap B \vdash D}\ {*L_\omega}}{a_1^{n_1}, \dots, a_{2i}^{n_{2i}}, a^+_{2i+1} \cdot p_{2i+1}, K_{2i+2}, \dots, K_{2k}, p_{2k} \multimap B \vdash D}\ {\cdot L}\text{ (2 times)}}{a_1^{n_1}, \dots, a_{2i}^{n_{2i}}, p_{2i}, K_{2i+1}, K_{2i+2}, \dots, K_{2k}, p_{2k} \multimap B \vdash D}\ {\multimap L}$$

(2) Let n be the given value of n_{2i}.

$$\dfrac{p_{2i-1} \vdash p_{2i-1} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{a_1^{n_1}, \dots, a_{2i-1}^{n_{2i-1}}, a_{2i}^{n}, p_{2i}, K_{2i+1}, \dots, K_{2k}, p_{2k} \multimap B \vdash D}{a_1^{n_1}, \dots, a_{2i-1}^{n_{2i-1}}, a_{2i}^{n}, !p_{2i}, K_{2i+1}, \dots, K_{2k}, p_{2k} \multimap B \vdash D}\ {!L_1}}{a_1^{n_1}, \dots, a_{2i-1}^{n_{2i-1}}, \underbrace{a_{2i}, !p_{2i}, \dots, a_{2i}, !p_{2i}}_{n-1 \text{ times}}, a_{2i}, !p_{2i}, K_{2i+1}, \dots, K_{2k}, p_{2k} \multimap B \vdash D}\ {!L_0}\ (n-1\text{ times})}{a_1^{n_1}, \dots, a_{2i-1}^{n_{2i-1}}, (a_{2i} \cdot !p_{2i})^n, K_{2i+1}, \dots, K_{2k}, p_{2k} \multimap B \vdash D}\ {\cdot L}\ (n-1\text{ times})}{a_1^{n_1}, \dots, a_{2i-1}^{n_{2i-1}}, !(a_{2i} \cdot !p_{2i}), K_{2i+1}, \dots, K_{2k}, p_{2k} \multimap B \vdash D}\ {!L_n}}{a_1^{n_1}, \dots, a_{2i-1}^{n_{2i-1}}, p_{2i-1}, K_{2i}, K_{2i+1}, \dots, K_{2k}, p_{2k} \multimap B \vdash D}\ {\multimap L}$$

The interesting direction, both for (1) and (2), is the "only if" one. We start with several observations, common for both situations. Let us consider the last rule applied in a cut-free derivation of the sequent in question.

First, we may suppose that this rule is $\multimap L$. Indeed, since D is constructed using only $\cdot$, $\vee$, and *, all "right" rules introducing these connectives are interchangable upwards with $\multimap L$. In the antecedent, the only top-level connectives are $\multimap$ ones.

Second, in the application of $\multimap L$ the rightmost formula $p_{2k} \multimap B$, unless it is the active one, should go to the right premise. Otherwise, the right premise includes no variables from the set $\mathcal{QZ}$ (they are only in B), and its derivability violates Lemma 5.5. Thus, $p_{2k} \multimap E^*$, whether it is active or not, could never to the antecedent of the left premise.

Let us start with (1). We claim that the active formula of $\multimap L$ should be $K_{2i+1} = p_{2i} \multimap (a^+_{2i+1} \cdot p_{2i+1})$. Suppose the contrary. Then the left premise of $\multimap L$ is of the form $\Pi \vdash p_j$, where $j > 2i$. By Lemma 5.4, Π should include a positive occurrence of p_j, that is, it includes K_j. Since K_j is of the form $p_{j-1} \multimap F$, Lemma 5.4 requires Π to include K_{j-1}, and so on, until

$K_{2i} = p_{2i} \multimap (a^+_{2i+1} \cdot p_{2i+1})$. On the other hand, as noticed above, Π does not include $p_{2k} \multimap B$. This violates Lemma 5.6: we have a positive a^+_{2i+1} inside K_{2i+1}, but no negative occurrence of a_{2i+1} (they are all in B).

Now we claim that actually the left premise of $\multimap L$ is $p_{2i} \vdash p_{2i}$. From the above we know that this premise is of the form $\Pi \vdash p_{2i}$, and by Lemma 5.4 p_{2i} should be in Π. Again, $p_{2k} \multimap B$ is not in Π, and, moreover, the argument above shows that if K_j is in Π for some $j > i$, then so is K_i, which contradicts with the fact that K_i was the active formula of $\multimap L$. Thus, Π consists of p_{2i} and maybe some a_k's, and $\Pi \vdash p_{2i}$ is derivable only for $\Pi = p_{2i}$.

Now the right premise of $\multimap L$ is

$$a_1^{n_1}, \ldots, a_{2i}^{n_{2i}}, a^+_{2i+1} \cdot p_{2i+1}, K_{2i+2}, \ldots, K_{2k}, \ldots, p_{2k} \multimap B \vdash D,$$

and by invertibility of $\cdot L$ and $*L_\omega$ we get derivability of

$$a_1^{n_1}, \ldots, a_{2i}^{n_{2i}}, a_{2i+1}^{n_{2i+1}}, p_{2i+1}, K_{2i+2}, \ldots, K_{2k}, p_{2k} \multimap B \vdash D$$

for any non-zero value of n_{2i+1}.

The (2) case is a bit trickier. The difference here is that the multiplexing rule $!L_n$ is, in general, not invertible, so we have to perform deeper proof analysis.

As in (1), the active formula of $\multimap L$ should be K_{2i}. Indeed, otherwise by descending on p_j's we show that K_{2i} should be in the antecedent Π of the left premise, while $p_{2k} \multimap B$ should be not. If K_{2i+1} is also in Π, we immediately violate Lemma 5.6. Otherwise K_{2i+1} is the active formula, and the left premise of $\multimap L$ is $\Phi, p_{2i-1}, p_{2i-1} \multimap !(a_{2i} \cdot !p_{2i}) \vdash p_{2i}$, where Φ is a sequence of a_k's. By Lemma 5.6, Φ should be empty. Now we have $p_{2i-1}, p_{2i-1} \multimap !(a_{2i} \cdot !p_{2i}) \vdash p_{2i}$, and one can explicitly show that this sequent is not derivable.

Next, the left premise is again $p_{2i-1} \vdash p_{2i-1}$ (by the same argument as in (1)), and we get the following right premise:

$$a_1^{n_1}, \ldots, a_{2i-1}^{n_{2i-1}}, K_{2i+1}, \ldots, K_{2k}, p_{2k} \multimap B \vdash D.$$

Again, right rules operating D can be shifted upwards, so the lowermost rule in the derivation is $\multimap L$ or $!L$. Now the same argument as above shows that this rule could *not* be $\multimap L$. In particular, $\multimap L$ with K_{2i+1} as the active formula would yields $!(a_{2i} \cdot !p_{2i}) \vdash p_{2i}$, which is not derivable. Therefore, the lowermost rule is the multiplexing rule $!L$, and its premise is

$$a_1^{n_1}, \ldots, a_{2i-1}^{n_{2i-1}}, (a_{2i} \cdot !p_{2i})^{n_{2i}}, K_{2i+1}, \ldots, K_{2k}, p_{2k} \multimap B \vdash D.$$

for some value n_{2i}. Since $\cdot L$ is invertible, we get

$$a_1^{n_1}, \ldots, a_{2i-1}^{n_{2i-1}}, a_{2i}^{n_{2i}}, (!p_{2i})^{n_{2i}}, K_{2i+1}, \ldots, K_{2k}, p_{2k} \multimap B \vdash D.$$

In this sequent, we have exactly one positive occurrence of p_{2i}, namely, the one in $K_{2i+1} = p_{2i} \multimap (a^+_{2i+1} \cdot p_{2i+1})$, and several negative occurrences of $!p_{2i}$. Let us go up along the derivation tree. In each sequent, there could be several negative occurrences of $!p_{2i}$ or p_{2i} itself, and one or zero positive occurrences of p_{2i}. If there are no positive occurrences of p_{2i}, by Lemma 5.4 there are also no negative occurrences of p_{2i}. As for $!p_{2i}$'s, let us remove them. For sequents with one positive p_{2i}, let us replace all negative ones with just one copy of p_{2i}.

After this transformation, the derivation remains valid (multiplexing rules for $!p_{2i}$ disappear), and yields the desired sequent:

$$a_1^{n_1}, \ldots, a_{2i-1}^{n_{2i-1}}, a_{2i}^{n_{2i}}, p_{2i}, K_{2i+1}, \ldots, K_{2k}, p_{2k} \multimap B \vdash D.$$

□

Now we are ready to prove the main result.

Proof of Theorem 5.2. Applying Lemma 5.3 $2k$ times, alternating (1) and (2), we get the following. The sequent

$$p_0, K_1, \ldots, K_{2k}, p_{2k} \multimap B \vdash D$$

is derivable iff the following statement is true:

$$\forall n_1 > 0\, \exists n_2 > 0\, \forall n_3 > 0\, \ldots\, \exists n_{2k} > 0$$
$$\big(\text{the sequent } a_1^{n_1}, \ldots, a_{2k}^{n_{2k}}, p_{2k}, p_{2k} \multimap B \vdash D \text{ is derivable}\big)$$

Derivability of $a_1^{n_1}, \ldots, a_{2k}^{n_{2k}}, p_{2k}, p_{2k} \multimap B \vdash D$ is equivalent to that of $a_1^{n_1}, \ldots, a_{2k}^{n_{2k}}, B \vdash D$. Indeed, the right-to-left direction here is just application of $\multimap L$. For the left-to-right one we again postpone any right rule applications and consider the $\multimap L$ whose active formula is $p_{2k} \multimap B$ (the only complex formula in the antecedent). The left premise is $p_{2k} \vdash p_{2k}$ (otherwise it is not derivable), so the right premise is $a_1^{n_1}, \ldots, a_{2k}^{n_{2k}}, B \vdash D$, which is equiderivable with

$$q_0, a_1^{n_1}, \ldots, a_{2k}^{n_{2k}}, E^* \vdash D.$$

By Theorem 5.1, the derivability of this sequent is equivalent to the fact that $\mathtt{M}$ does not halt on $(q_0; n_1, \ldots, n_{2k}, 0, 0, 0)$. Thus the derivability of the original sequent, $p_0, K_1, \ldots, K_{2k}, p_{2k} \multimap B \vdash D$, is equivalent to $(*)$. □

This yields the desired complexity lower bound:

Theorem 5.7.
The first-order theory of $\mathfrak{N}$ is many-one reducible to the derivability problem for $!^{\mathsf{m}}\mathbf{CommACT}_\omega$.

Proof. Take $\mathcal{S}$ to be $\bigcup_{k\in\omega} \mathcal{S}_k$. It is well-known that $\mathcal{S}$ is many-one equivalent to the first-order theory of $\mathfrak{N}$ — see [21, Chapter 14]. So it remains to reduce $\mathcal{S}$ to the derivability problem for $!^{\mathsf{m}}\mathbf{CommACT}_\omega$, which can be easily done by using Theorem 5.2. □

Finally, since $!^{\mathsf{m}}\mathbf{CommACT}_\omega$ can be embedded into $!^{\mathsf{m}}\nabla\mathbf{ACT}_\omega$ by Theorem 2.1, the same holds for $!^{\mathsf{m}}\nabla\mathbf{ACT}_\omega$:

Corollary 5.8.
The first-order theory of $\mathfrak{N}$ is many-one reducible to the derivability problem for $!^{\mathsf{m}}\nabla\mathbf{ACT}_\omega$.

Proof. Immediate from Theorems 5.7 and 2.1. □

6 A hyperarithmetical upper bound

Next, using some machinery of computability theory together with Theorem 3.2, we shall show that the derivability problems for $!^{\mathsf{m}}\mathbf{CommACT}_{\omega}$ and $!^{\mathsf{m}}\mathbf{\nabla ACT}_{\omega}$ both belong to the ω^{ω} level of the hyperarithmetical hierarchy — and hence, in particular, are strictly below Π^1_1.

For any $S, P \subseteq \omega$, let

$$\mathrm{Index}\,(S; P) \;:=\; \{k \in \omega \mid \mathtt{U}_k \text{ many-one reduces } S \text{ to } P\}.$$

So $k \in \mathrm{Index}\,(S; P)$ iff $\mathtt{U}_k$ is total and $(\mathtt{U}_k)^{-1}\,[P] = S$. We call elements of $\mathrm{Index}\,(S; P)$ *indices of S with respect to P*. Intuitively, each of these encodes a program that computes a function reducing S to P, and hence provides a way of showing that the complexity of S is bounded by that of P.

Lemma 6.1.
Let $n \in \omega \setminus \{0\}$ and $\Phi\,(x, X)$ be a Σ^0_n-formula. Then there exists a computable $\xi : \omega \to \omega$ such that for any $S, P \subseteq \omega$ and $k \in \omega$,

$$k \;\in\; \mathrm{Index}\,(S; P) \quad\Longrightarrow\quad \xi\,(k) \;\in\; \mathrm{Index}\,([\Phi]\,(S); \mathtt{J}^n\,(P))$$

where $\mathtt{J}^n\,(S)$ denotes the result of applying the jump operator n times to S.

Proof. It is well-known that there exist a computable function f_n from ω to ω and a computable function g_n from the Σ^0_n-formulae with only x and X free to ω such that:

i. for any $k \in \omega$ and $S, P \subseteq \omega$,

$$k \;\in\; \mathrm{Index}\,(S, P) \quad\Longrightarrow\quad f_n\,(k) \;\in\; \mathrm{Index}\,(\mathtt{J}^n\,(S); \mathtt{J}^n\,(P));$$

ii. for any Σ^0_n-formula $\Psi\,(x, X)$ and $S \subseteq \omega$,

$$g_n\,(\Psi) \;\in\; \mathrm{Index}\,([\Psi]\,(S); \mathtt{J}^n\,(S)).^{9}$$

It should be remarked that f_n does not depend on S or P, and g_n does not depend on S. Using f_n and g_n, it is easy to construct ξ that has the desired property. □

More informally, Lemma 6.1 shows us how, given an index of S with respect to P, to effectively find an index of $[\Phi]\,(S)$ with respect to $\mathtt{J}^n\,(P)$, where n is determined by the form of Φ (cf. Folklore 2.5 and 2.6).

Define $\mathrm{succ} : \omega \times \omega \to \omega$ recursively by

$$\mathrm{succ}\,(m, n) \;:=\; \begin{cases} m & \text{if } \; n = 0; \\ 2^{\mathrm{succ}\,(m, n-1)} & \text{otherwise.} \end{cases}$$

[9] In more detail, (i) follows from an effective version of Theorem 13-I(e) in [21] — taking into account that we have $S \leqslant_T P$ whenever $S \leqslant P$. And (ii) follows from an effective version of Theorem 14-VIII(a) in [21] — taking into account that if $[\Psi]\,(S)$ is computably enumerable in $\mathtt{J}^{n-1}\,(S)$, then $[\Psi]\,(S) \leqslant \mathtt{J}^n\,(S)$ by Theorem 13-I(d). Cf. Corollary 13-I(c) and Theorem 14-VIII(c) in [21], which implicitly use our notion of index.

In this way, for each $m \in \omega$ we have

$$\operatorname{succ}(m, 0) = m, \quad \operatorname{succ}(m, 1) = 2^m, \quad \operatorname{succ}(m, 2) = 2^{2^m}, \quad \ldots$$

Obviously, succ is computable. At the same time, it is not one-one, i.e. the inverse of succ is not a function. For instance,

$$\operatorname{succ}(3, 2) = 2^{2^3} = \operatorname{succ}\left(2^3, 1\right) = \operatorname{succ}\left(2^{2^3}, 0\right).$$

However, there are computable base, step : $\omega \to \omega$ such that for every $k \in \omega$,

$$k = \operatorname{succ}(\operatorname{base}(k), \operatorname{step}(k)) \quad \text{and} \quad \log_2(\operatorname{base}(k)) \notin \omega \setminus \{0\}.$$

Let us illustrate how base and step work by examples:

- if k is not a power of 2, then $\operatorname{base}(k) = k$ and $\operatorname{step}(k) = 0$;
- if $k = 2^{2^2}$, then $\operatorname{base}(k) = 1$ and $\operatorname{step}(k) = 3$;
- if $k = 2^{2^3}$, then $\operatorname{base}(k) = 3$ and $\operatorname{step}(k) = 2$.

Hence in the case when $k \in \mathcal{O}$ we have

$$|k| = |\operatorname{base}(k)| + \operatorname{step}(k) \quad \text{and} \quad |\operatorname{base}(k)| \in \{0\} \cup \mathsf{LOrd}.$$

These functions are useful for keeping track of iterations of positive arithmetical operators over the constructive ordinals:

Theorem 6.2.
Let $n \in \omega \setminus \{0\}$ and $\Phi(x, X)$ be a positive Σ^0_n-formula. Then there is a computable $\eta : \omega \to \omega$ such that for every $k \in \mathcal{O}$:

$$\begin{aligned}
\eta(k) &\in \operatorname{Index}\left([\Phi]^{|k|}(\varnothing), \mathtt{J}^{\operatorname{step}(k)\cdot n+1}(H(\operatorname{base}(\mathrm{k})))\right)\\
&= \operatorname{Index}\left([\Phi]^{|k|}(\varnothing), H(\operatorname{succ}(\operatorname{base}(k), \operatorname{step}(k)\cdot n+1))\right)\\
&= \operatorname{Index}\left([\Phi]^{|k|}(\varnothing), \mathtt{J}(H(\operatorname{succ}(\operatorname{base}(k), \operatorname{step}(k)\cdot n)))\right).
\end{aligned}$$

In particular, if $|k| \in \mathsf{LOrd}$, then $\eta(k)$ is an index of $[\Phi]^{|k|}(\varnothing)$ with respect to $\mathtt{J}(H(k))$.

Proof. For convenience, define $\operatorname{lift}_n : \omega \to \omega$ by

$$\operatorname{lift}_n(k) := \operatorname{succ}(\operatorname{base}(k), \operatorname{step}(k)\cdot n + 1).$$

Observe that for every $k \in \omega$,

$$\begin{aligned}
\mathtt{J}^n(H(\operatorname{lift}_n(k))) &= \mathtt{J}^n(H(\operatorname{succ}(\operatorname{base}(k), \operatorname{step}(k)\cdot n+1)))\\
&= H(\operatorname{succ}(\operatorname{base}(k), \operatorname{step}(k)\cdot n+1+n))
\end{aligned}$$

$$
\begin{aligned}
&= H\left(\mathrm{succ}\left(\mathrm{base}\left(2^k\right), \mathrm{step}\left(2^k\right)\cdot n+1\right)\right)\\
&= H\left(\mathrm{lift}_n\left(2^k\right)\right).
\end{aligned}
$$

Let ξ be as in the statement of Lemma 6.1. Then for any $k \in \mathcal{O}$ and $i \in \omega$,

$$
\begin{aligned}
i \in \mathrm{Index}\left([\Phi]^{|k|}(\varnothing), H(\mathrm{lift}_n(k))\right) \quad &\Longrightarrow \\
&\xi(i) \in \mathrm{Index}\left([\Phi]^{|2^k|}(\varnothing), H\left(\mathrm{lift}_n\left(2^k\right)\right)\right).
\end{aligned}
$$

Thus we can use ξ for calculating η at successor steps.

Limit steps are a bit more complicated. Let γ be as in the statement of Folklore 2.7, and let g_1 be as in the proof of Lemma 6.1 (dealing with Σ^0_1-formulae). Given $e, k \in \omega$, take

$$
\Psi_{e,k}(x, X) := \exists y\left(y \in \mathrm{dom}\,\mathtt{U}_{\gamma(k)} \wedge \mathsf{c}\left(\mathtt{U}\left(\mathtt{U}_e(y), x\right), \mathrm{lift}_n(y)\right) \in X\right)
$$

— which can be treated as a Σ^0_1-formula, of course. Now suppose that $k \in \mathcal{O}$ and $e \in \omega$ are such that $|k| \in \mathsf{LOrd}$, and for every $m \in \mathcal{O}$,

$$
m <_{\mathcal{O}} k \quad \Longrightarrow \quad \mathtt{U}_e(m) \in \mathrm{Index}\left([\Phi]^{|m|}(\varnothing), H(\mathrm{lift}_n(m))\right).
$$

Then $\left[\Psi_{e,k}\right](H(k))$ coincides with

$$
\begin{aligned}
\bigcup_{m<_{\mathcal{O}}k} \{i \mid \mathsf{c}\left(\mathtt{U}\left(\mathtt{U}_e(m), i\right), \mathrm{lift}_n(m)\right) \in H(k)\} &= \bigcup_{m<_{\mathcal{O}}k} \{i \mid \mathtt{U}\left(\mathtt{U}_e(m), i\right) \in H(\mathrm{lift}_n(m))\}\\
&= \bigcup_{m<_{\mathcal{O}}k} \left\{i \mid i \in [\Phi]^{|m|}(\varnothing)\right\}\\
&= \bigcup_{m<_{\mathcal{O}}k} [\Phi]^{|m|}(\varnothing)\\
&= [\Phi]^{|k|}(\varnothing).
\end{aligned}
$$

(Note that $m <_{\mathcal{O}} k$ implies $\mathrm{lift}_n(m) <_{\mathcal{O}} k$ because $|k| \in \mathsf{LOrd}$.) Hence $g_1\left(\Psi_{e,k}\right)$ is an index of $[\Phi]^{|k|}(\varnothing)$ with respect to $\mathtt{J}(H(k))$, which can also be expressed as $H(\mathrm{lift}_n(k))$.

Finally, we fix an index i_0 of $\varnothing$ with respect to $\mathtt{J}(\varnothing)$, and consider a computable $\mu : \omega \to \omega$ such that for all $e, k \in \omega$,

$$
\mathtt{U}_{\mu(e)}(k) = \begin{cases} \xi\left(\mathtt{U}_e(m)\right) & \text{if } k = 2^m \neq 1,\\ g_1\left(\Psi_{e,m}\right) & \text{if } k = 3\cdot 5^m,\\ k_0 & \text{otherwise.}\end{cases}
$$

By the recursion theorem, $\mathtt{U}_{\mu(c)} = \mathtt{U}_c$ for some $c \in \omega$. As can easily be verified, the function $\mathtt{U}_c$ does the job.[10] □

[10] In particular, $\mathtt{U}_c$ turns out to be total. For otherwise let k be the least element of $\omega \setminus \mathrm{dom}(\mathtt{U}_c)$; then $k = 2^m \neq 1$ and therefore $\mathtt{U}_c(m)$ is undefined, which contradicts the choice of k.

Intuitively, Theorem 6.2 shows us how, given any $k \in O$ (which respesents the constructive ordinal $|k|$), to effectively locate $[\Phi]^{|k|}(\varnothing)$ in the hyperarithmetical hierarchy. In particular, if $[\Phi]$ is $\mathscr{D}_{\mathbf{L}}$ where $\mathbf{L} \in \{!^{\mathsf{m}}\nabla\mathbf{ACT}_\omega, !^{\mathsf{m}}\mathbf{CommACT}_\omega\}$, then for each $k \in O$ we obtain an upper bound for $\mathscr{D}_{\mathbf{L}}^{|k|}(\varnothing)$, i.e. for the collection of all sequents derivable in $\mathbf{L}$ in at most $|k|$ steps.

Finally, we are ready to prove:

Theorem 6.3.
Let $\mathbf{L} \in \{!^{\mathsf{m}}\nabla\mathbf{ACT}_\omega, !^{\mathsf{m}}\mathbf{CommACT}_\omega\}$. *Then the least fixed point of* $\mathscr{D}_{\mathbf{L}}$ *belongs to* $\Sigma^0_{\omega^\omega}$, *i.e. to the* ω^ω *level of the hyperarithmetical hierarchy.*

Proof. As we observed earlier, $\mathscr{D}_{\mathbf{L}}$ is elementary. Hence the desired result follows immediately from Theorems 6.2 and 3.2. □

The semi-formal reasoning behind the last theorem is the following. Among the rules of $\mathbf{L}$ there are $(*L_\omega)$ and $(!L_n)$, which are definable by a Π^0_1-formula and a Σ^0_1-formula respectively. Hence each application of $\mathscr{D}_{\mathbf{L}}$ may yield yet another quantifier alternation (there may be more quantifier alternations, but they can, in fact, be merged at limit steps).[11] Now by Theorem 3.2, we need at most ω^ω steps to settle the derivability problem, which suggests that $\Sigma^0_{\omega^\omega}$ is an upper bound for it. The purpose of Lemma 6.1 and Theorem 6.2 is to fill in the technical gaps in the foregoing reasoning.

7 Conclusion

In this article, we have considered two subexponential extensions of infinitary action logic. The first one is non-commutative and involves two subexponentials: one allowing multiplexing, the other allowing permutation. The second extension is commutative and uses one subexponential which allows multiplexing. For both systems we have established upper and lower complexity bounds. The upper bound is hyperarithmetical, namely $\Sigma^0_{\omega^\omega}$. The lower one is complete first-order arithmetic. Thus the complexity of each of these systems is strictly between that of infinitary action logic (which is Π^0_1-complete) and that of infinitary action logic with subexponentials which allow contraction (which is Π^1_1-complete). We have also shown that ω^ω is an upper bound for the corresponding closure ordinals.

However, there is still a gap between our lower and upper bounds. So finding the exact complexity level and the exact value of the closure ordinal for infinitary action logic with multiplexing (both in the commutative and in the non-commutative case) is left as an open problem for future research.

Acknowledgements. This work was supported by the Russian Science Foundation under grant no. 20-41-05002; see https://rscf.ru/en/project/20-41-05002/.

[11] This case is very different from that of ACT_ω or that of ACT_ω expanded with a family of subexponentials for exchange and weakening (cf. [18, Section 5] and [12, Section 5.4]).

References

[1] Aczel, P., An introduction to inductive definitions, in J. Barwise (ed.), *Handbook of Mathematical Logic*, Elsevier, 1977, pp. 739–782.

[2] Buchholz, W., Explaining Gentzen's consistency proof within infinitary proof theory, in G. Gottlob, A. Leitsch and D. Mundici (eds.), *Computational Logic and Proof Theory, 5th Kurt Gödel Colloquium, KGC 1997*, LNCS 1289, Springer, 1997, pp. 4–17.

[3] Buszkowski, W., On action logic: equational theories of action algebras, *Journal of Logic and Computation* 17(1):199–217, 2007.

[4] Buszkowski, W., and E. Palka, Infinitary action logic: complexity, models and grammars, *Studia Logica* 89(1):1–18, 2008.

[5] Girard, J.-Y., Linear logic, *Theoretical Computer Science* 50(1):1–102, 1987.

[6] Girard, J.-Y., Light linear logic, *Information and Computation* 143(2):175–204, 1998.

[7] Kanovich, M., S. Kuznetsov, V. Nigam and A. Scedrov, Subexponentials in non-commutative linear logic, *Mathematical Structures in Computer Science* 29(8):1217–1249, 2019.

[8] Kanovich, M., S. Kuznetsov, V. Nigam and A. Scedrov, Soft subexponentials and multiplexing, in N. Peltier and V. Sofronie-Stokkermans (eds.), *Automated Reasoning, 10th International Joint Conference, IJCAR 2020*, LNAI 12166, Springer, 2020, pp. 500–517.

[9] Kozen, D., On action algebras, in J. van Eijck and A. Visser (eds.), *Logic and Information Flow*, MIT Press, 1994, pp. 78–88.

[10] Kuznetsov, S. L., Complexity of a fragment of infinitary action logic with exponential via non-well-founded proofs, in A. Das and S. Negri (eds.), *Automated Reasoning with Analytic Tableaux and Related Methods, 30th International Conference, TABLEAUX 2021*, LNCS 12842, Springer, 2021, pp. 317–334.

[11] Kuznetsov, S. L., Commutative action logic, *Journal of Logic and Computation*, exac008, 2022.

[12] Kuznetsov, S. L., and S. O. Speranski, Infinitary action logic with exponentiation, *Annals of Pure and Applied Logic* 173(2):103057, 2022.

[13] Lafont, Y., Soft linear logic and polynomial time, *Theoretical Computer Science* 318(1–2):163–180, 2004.

[14] Lambek, J., The mathematics of sentence structure, *American Mathematical Monthly* 65(3):154–170, 1958.

[15] Lincoln, P., J. Mitchell, A. Scedrov, and N. Shankar, Decision problems for propositional linear logic, *Annals of Pure and Applied Logic* 56(1–3):239–311, 1992.

[16] Minsky, M. L., Recursive unsolvability of Post's problem of "Tag" and other topics in theory of Turing machines, *Annals of Mathematics* 74(3):437–455, 1961.

[17] Moschovakis, Y. N., *Elementary Induction on Abstract Structures*, North-Holland Publishing Company, 1974.

[18] Palka, E., An infinitary sequent system for the equational theory of *-continuous action lattices, *Fundamenta Informaticae* 78(2):295–309, 2007.

[19] Pratt, V., Action logic and pure induction, in J. Eijck (ed.), *Logics in AI, European Workshop JELIA 1990*, LNAI 478, Springer, 1991, pp. 97–120.

[20] Redmond, B. F., Multiplexor categories and models of soft linear logic, in S. N. Artemov and A. Nerode (eds.), *Logical Foundations of Computer Science, International Symposium, LFCS 2007*, LNCS 4514, Springer, 2007, pp. 472–485.

[21] Rogers, H., Jr., *Theory of Recursive Functions and Effective Computability*, McGraw-Hill Book Company, 1967.

[22] Sacks, G. E., *Higher Recursion Theory*, Springer, 1990.

[23] Schroeppel, R., A two counter machine cannot calculate 2^N, Massachusetts Institute of Technology, A. I. Laboratory, Artificial Intelligence Memo #257, 1972.

Stepan L. Kuznetsov
Steklov Mathematical Institute of Russian Academy of Sciences
8 Gubkina St.,
119991 Moscow, Russia
sk@mi-ras.ru

Stanislav O. Speranski
Steklov Mathematical Institute of Russian Academy of Sciences
8 Gubkina St.,
119991 Moscow, Russia
katze.tail@gmail.com