Hypersequent Calculi Have Ackermann Complexity

A. R. Balasubramanian
Max Planck Institute for Software Systems (MPI-SWS), Kaiserslautern, Germany

Vitor Greati[1]
Bernoulli Institute, University of Groningen, The Netherlands

Revantha Ramanayake
Bernoulli Institute, University of Groningen, The Netherlands

Abstract

For substructural logics with contraction or weakening admitting cut-free sequent calculi, proof search was analyzed using well-quasi-orders on $\mathbb{N}^d$ (Dickson's lemma), yielding Ackermann upper bounds via controlled bad-sequence arguments. For hypersequent calculi, that argument lifted the ordering to the powerset, since a hypersequent is a (multi)set of sequents. This induces a jump from Ackermann to hyper-Ackermann complexity in the fast-growing hierarchy, suggesting that cut-free hypersequent calculi for extensions of the commutative Full Lambek calculus with contraction or weakening ($\mathbf{FL_{ec}}$/$\mathbf{FL_{ew}}$) inherently entail hyper-Ackermann upper bounds. We show that this intuition does not hold: every extension of $\mathbf{FL_{ec}}$ and $\mathbf{FL_{ew}}$ admitting a cut-free hypersequent calculus has an Ackermann upper bound on provability.

To avoid the powerset, we exploit novel dependencies between individual sequents within any hypersequent in backward proof search. The weakening case, in particular, introduces a Karp-Miller-style acceleration, and it improves the upper bound for the fundamental fuzzy logic **MTL**. Our Ackermann upper bound is optimal for the contraction case (realized by the logic $\mathbf{FL_{ec}}$).

2012 ACM Subject Classification Theory of computation → Proof theory; Theory of computation → Complexity theory and logic; Theory of computation → Linear logic

Keywords and phrases Hypersequent calculi, Substructural logics, Ackermann complexity, Well-quasi-orders

Digital Object Identifier 10.4230/LIPIcs.LICS.2026.10

Related Version *Full Version*: `https://arxiv.org/abs/2602.19229` [3]

Funding *A. R. Balasubramanian*: This research was sponsored in part by the Deutsche Forschungsgemeinschaft project 389792660 TRR 248–CPEC.
Revantha Ramanayake: Supported by Dutch Research Council (NWO) project OCENW.M.22.258.

1 Introduction

Substructural logics arise by omitting or modifying structural rules such as contraction, weakening, exchange, or associativity from classical or intuitionistic proof calculi. This yields a broad family of systems – including linear logic, relevance logics, mathematical fuzzy logics, and the full Lambek calculus [27, 30, 14, 25]. These systems are able to reason about resources in a broad sense, since assumptions can no longer be duplicated or discarded freely.

The basic substructural logics are Full Lambek calculus **FL** extended with the structural rules of exchange, weakening, or contraction, denoted $\mathbf{FL_S}$ ($\mathrm{S} \subseteq \{e, w, c\}$). These have a cut-free sequent calculus, which implies a significant restriction on the proof-search space, and this enables the proof-theoretic study of computational complexity of the logics. For provability – the topic of interest in this work – all the basic substructural logics excepting

[1] Corresponding author

$\mathbf{FL_{ec}}$ are PSPACE-complete [17]. Meanwhile, $\mathbf{FL_{ec}}$ was famously shown by Kripke [20] in 1959 to be decidable, and Urquhart [32] established its non-primitive complexity; indeed, it is complete for the class $\mathbf{F}_\omega$ in Schmitz's [31] ordinal-indexed fast-growing complexity hierarchy. Informally, a problem in $\mathbf{F}_\omega$ has complexity bounded by a single Ackermann-type function composed with primitive recursive functions.

To understand the source of the $\mathbf{F}_\omega$ upper bound, observe that backward proof search P in $\mathbf{FL_{ec}}$ can be restricted to branches that are *bad sequences* under what is essentially the product ordering on $\mathbb{N}^d$. Dickson's lemma [11] states that this is a well-quasi-ordering (wqo), i.e., every branch in P is a sequence $(s_i)_{i\in\mathbb{N}}$ of sequents with the property that there is no $i < j$ such that $s_i \leq s_j$ (read as s_j is contractible to s_i). The point of the wqo is that every bad sequence w.r.t it is finite. Hence we can do proof search in $\mathbf{FL_{ec}}$ and termination is guaranteed, and decidability follows. Successive jumps in this bad sequence are controlled in the sense that there is a primitive recursive function g (depending only on the proof calculus) and a norm $[\![\cdot]\!]$ on sequents such that $[\![s_i]\!] < g^i([\![s_0]\!])$. Every such controlled bad sequence has a maximum length that is given by a *length theorem*; in the case of $\leq$, this is an Ackermanian-type function in $[\![s_0]\!]$ (Figueira et al. [13]). The claimed upper bound follows.

Most extensions of $\mathbf{FL_{ec}}$ do not admit cut-free sequent calculi, but infinitely many do in the generalized setting of cut-free hypersequent calculi [7, 9]; specifically, axioms from every level of the substructural hierarchy [7, 18] excepting the final level $\mathcal{N}_3$, which contains axioms like distributivity that often lead to undecidability in the substructural setting. A *hypersequent* [26, 28, 1] is a multiset of sequents, written as $s_0 \mid \ldots \mid s_k$. Hypersequent rules properly generalize sequent rules by allowing multiple components in the conclusion to be utilized simultaneously. Ciabattoni et al. [7, 9] introduced a general theory to compute analytic hypersequent calculi for non-classical and substructural logics, including intermediate, many-valued, and fuzzy logics. Balasubramanian et al. [4] extended the complexity analysis of $\mathbf{FL_{ec}}$ to these hypersequent calculi, by lifting the wqo on $\mathbb{N}^d$ to the minoring ordering on $\mathcal{P}_f(\mathbb{N}^d)$. Since the length theorem on such powerset orderings is hyper-Ackermannian (i.e., in $\mathbf{F}_{\omega^\omega}$) (Balasubramanian [2]), this showed that every axiomatic extension of $\mathbf{FL_{ec}}$ that has a cut-free hypersequent calculus has hyper-Ackermann complexity. The same upper bound is obtained in [4] for extensions of $\mathbf{FL_{ew}}$; this time, the majoring wqo was coupled with forward proof search. That yielded the first upper bound for the fundamental fuzzy logic **MTL** [12] – a longstanding open question [15] – which is axiomatized over $\mathbf{FL_{ew}}$ by the prelinearity axiom $(p \to q) \vee (q \to p)$.

The above line of reasoning suggested that the jump from sequents to hypersequents (necessitated by the need for a cut-free proof calculus) also entails a jump to $\mathbf{F}_{\omega^\omega}$. In this work, we show that this intuition does *not* hold: we prove that every extension of $\mathbf{FL_{ec}}$ and $\mathbf{FL_{ew}}$ admitting a cut-free hypersequent calculus has an Ackermann upper bound on provability. The key technical insight is avoiding a lift to a powerset ordering. Our solution is to exploit novel dependencies between components in a hypersequent, so every hypersequent – now viewed as a sequence with the sequent components placed in the order of being added – in the backward proof search is itself a controlled bad sequence (in the sequent ordering!). For the contraction case, our bound is optimal because $\mathbf{FL_{ec}}$ already realizes it.

The weakening case is considerably more complicated. Vanilla backward proof search [24] will not terminate so we extend it with a Karp–Miller-style acceleration [19]. Due to the interaction of multiple sequents within a hypersequent, the precise definition here is much more sophisticated than the accelerations used for other problems in automata theory, such as Petri nets coverability or in counter automata [19, 23, 6]. We then couple this acceleration with the identification of a proof-theoretic gadget to implement a very specific form of contraction. An immediate corollary lowers the upper bound for **MTL** to Ackermann. That also applies to other (standard complete) fuzzy logics in the literature, e.g., n-contractive extensions [8, 16] and the wmn-extensions [5] of **MTL**.

Adding structure to bad sequences via proof calculi is of independent relevance to the infinite-state systems community since complexity based on wqos may be improved by a finer analysis on the bad sequences (see e.g., Lazic and Schmitz [21, 22]). Specifically, the proof calculi are variations of alternating branching vector addition systems with states (BVASS), where a configuration along a run can interact also with configurations visited before.

2 Preliminaries

2.1 Proof theory of substructural logics via hypersequents

Let P be a countably infinite set of *propositional variables*. The set L of *formulas* is defined using the following grammar.

$$\varphi := \mathrm{p} \in P \,|\, 1 \,|\, 0 \,|\, \varphi \wedge \varphi \,|\, \varphi \vee \varphi \,|\, \varphi \cdot \varphi \,|\, \varphi \to \varphi$$

A *logic* (over L) is a consequence relation $\vdash$ over L. Given a logic $\vdash$, the *provability* problem refers to deciding whether, given $\varphi \in L$, it is the case that $\varnothing \vdash \varphi$. A *sequent* is an expression of the form $\Gamma \Rightarrow \Pi$, where Γ (the *antecedent*) is a finite multiset of formulas and Π (the *succedent*) is a list that is either empty or consists of a single formula (we call it a *succedent-list* or *stoup*). We use $\Gamma, \Delta, \Theta, \Pi$ to denote finite multisets of formulas. The *multiplicity* of a formula φ in Γ is the number of copies of φ in Γ. We will denote an enumeration of the elements in Γ (including repetitions) using $[\varphi_1, \dots, \varphi_k]$. The *support* of any multiset is the set of its elements with non-zero multiplicity.

A *hypersequent* is a (possibly empty) finite multiset of sequents. It is written explicitly as follows, for $k \geq 0$,

$$\Gamma_1 \Rightarrow \Pi_1 \,|\, \dots \,|\, \Gamma_k \Rightarrow \Pi_k \tag{1}$$

Each sequent $\Gamma_i \Rightarrow \Pi_i$ is called a *component* of the hypersequent. For a hypersequent h and sequent s, we write $s \in h$ to mean that s is a component of h. Given a finite set S of formulas, an S-*(hyper)sequent* is a (hyper)sequent whose formulas are all in S. A hypersequent containing no repetitions is called *irredundant*.

A *hypersequent calculus* is a set of *hypersequent rules*. Each rule has a *conclusion* hypersequent and some number of *premise* hypersequent(s). All the calculi in this paper have the property of having only finitely many rules with the same conclusion.

Commonly, rules are presented as instances of *rule schemas* (see Example 2.1). Rule schemas and rules are usually presented in fraction notation with the premises listed above the fraction line and the conclusion below the line (see Figure 1). A rule (schema) with no premises is called an *initial hypersequent* or *axiomatic* rule. We will often see a hypersequent calculus as a collection of rule schemas, meaning that its rules are all the possibles instantiations of these schemas, also referred to as *rule instances*.

In order to properly define rule schemas we need to consider variables of different types, depending on whether they will be instantiated by formulas, multisets of formulas, sequents, or hypersequents. The conclusion and every premise of a rule schema has the form $S_1 | \dots | S_k$, where each S_i is (i) a hypersequent-variable (denoted by H), or (ii) a sequent-variable (denoted by s), or (iii) $\vec{X} \Rightarrow \Pi$, or (iv) $\vec{X} \Rightarrow\;$. Here $\vec{X}$ is a finite list of multiset-variables (we use Γ, Δ, Θ to denote multiset-variables) and formulas built from formula-variables φ, ψ, ϕ

and proposition-variables p, q, r; also, Π is a succedent-list-variable. The variables in the rule schema are collectively referred to as *schematic variables*. We refer to $S_1|\dots|S_k$ as a *hypersequent* and to each S_i as a *component*, although they are built from schematic variables, unlike in (1) where they are built from formulas; this terminology overloading is standard.

A *rule instance* is obtained from a rule schema by uniformly instantiating the schematic variables with concrete objects of the corresponding type: a hypersequent-variable with a hypersequent, a sequent-variable with a sequent, a list-variable with a list of formulas, and so on. A succedent-list-variable is instantiated by a *stoup* (i.e., a list that is either empty or consists of a single formula).

A component in the premise (resp., conclusion) of a rule schema that is not a hypersequent-variable is called an *active component* (resp., *principal component*). We use the same terminology for the corresponding component in a rule instance.

▶ **Example 2.1.** *We consider the rule schema* ($\wedge R$) *given by*

$$\frac{H \mid \boxed{\Gamma \Rightarrow \varphi} \qquad H \mid \boxed{\Gamma \Rightarrow \psi}}{H \mid \boxed{\Gamma \Rightarrow \varphi \wedge \psi}}\ (\wedge R)$$

where the active components (in the premises) and principal components (in the conclusion) have been highlighted with a surrounding box. Its premises are $H \mid \Gamma \Rightarrow \varphi$ *and* $H \mid \Gamma \Rightarrow \psi$, *and its conclusion is* $H \mid \Gamma \Rightarrow \varphi \wedge \psi$. *Below we give some rule instances of* ($\wedge R$), *where their active components have been highlighted.*

$$\frac{\boxed{\Rightarrow \mathrm{p}} \qquad \boxed{\Rightarrow \mathrm{q}}}{\boxed{\Rightarrow \mathrm{p} \wedge \mathrm{q}}} \qquad \frac{\mathrm{p} \Rightarrow \mathrm{p} \mid \boxed{\mathrm{r}, \mathrm{r} \Rightarrow \mathrm{p} \wedge \mathrm{q}} \qquad \mathrm{p} \Rightarrow \mathrm{p} \mid \boxed{\mathrm{r}, \mathrm{r} \Rightarrow \mathrm{q}}}{\mathrm{p} \Rightarrow \mathrm{p} \mid \boxed{\mathrm{r}, \mathrm{r} \Rightarrow (\mathrm{p} \wedge \mathrm{q}) \wedge \mathrm{q}}}$$

$$\frac{\Rightarrow \mathrm{p} \cdot \mathrm{p} \mid \mathrm{q} \to \mathrm{p} \Rightarrow \mathrm{p} \mid \boxed{\mathrm{r} \Rightarrow \mathrm{p} \wedge \mathrm{q}} \qquad \Rightarrow \mathrm{p} \cdot \mathrm{p} \mid \mathrm{q} \to \mathrm{p} \Rightarrow \mathrm{p} \mid \boxed{\mathrm{r} \Rightarrow \mathrm{q}}}{\Rightarrow \mathrm{p} \cdot \mathrm{p} \mid \mathrm{q} \to \mathrm{p} \Rightarrow \mathrm{p} \mid \boxed{\mathrm{r} \Rightarrow (\mathrm{p} \wedge \mathrm{q}) \wedge \mathrm{q}}}$$

The basic hypersequent calculus in this paper is $\mathbf{HFL_e}$ (see Fig 1). We omitted the (cut) rule, as cut elimination holds for this calculus. $\mathbf{HFL_{ec}}$ and $\mathbf{HFL_{ew}}$ are obtained by the addition, respectively, of the rules $\{(\mathbf{c})\}$ and $\{(\mathbf{i}), (\mathbf{o})\}$, listed below. We follow the standard convention here of writing ($\mathbf{w}$) to refer to the two rules ($\mathbf{i}$) and ($\mathbf{o}$).

$$\frac{H \mid \Gamma, \Delta \Rightarrow \Pi}{H \mid \Gamma, \varphi, \Delta \Rightarrow \Pi}\ (\mathbf{i}) \qquad \frac{H \mid \Gamma \Rightarrow}{H \mid \Gamma \Rightarrow \varphi}\ (\mathbf{o}) \qquad \frac{H \mid \Gamma, \varphi, \varphi, \Delta \Rightarrow \Pi}{H \mid \Gamma, \varphi, \Delta \Rightarrow \Pi}\ (\mathbf{c})$$

A calculus $\mathbf{H}_1$ is said to *extend* (or be an *extension of*) the calculus $\mathbf{H}_2$ if it contains all the rules of $\mathbf{H}_2$. If $\mathbf{H}$ is an extension of $\mathbf{HFL_e}$, the *extension of* $\mathbf{H}$ by a set $\mathcal{S}$ of rule schemas is the hypersequent calculus $\mathbf{H} \cup \mathcal{S}$, which we write as $\mathbf{H}(\mathcal{S})$ following standard practice.

Let $\mathbf{H}$ be an extension of $\mathbf{HFL_e}$. A *derivation* in $\mathbf{H}$ is a finite labelled rooted tree such that the labels of a non-leaf node and its child node(s) are respectively the conclusion and premise(s) of some rule in the calculus. A *proof* of a hypersequent h in $\mathbf{H}$ is a derivation in $\mathbf{H}$ whose root is labelled with h and all leaves are instances of axiomatic rules. Given a set of rules $\mathcal{S}$, we say that g *is obtained from* h *by applications of rules in* $\mathcal{S}$ whenever there is a derivation only built by the rules in $\mathcal{S}$ rooted at g with leaves labelled by h. We write $\vdash_{\mathbf{H}} h$ to mean that there is a proof of h in $\mathbf{H}$ (we say that h is *provable* in $\mathbf{H}$). A *branch* in a derivation is a path from the root to a leaf. The *height* of a derivation is the maximum number of nodes on a branch. We write $\vdash^{\alpha}_{\mathbf{H}} h$ (resp., $\vdash^{\leq\alpha}_{\mathbf{H}} h$) to mean that there is a proof of h in $\mathbf{H}$ of height α (resp., $\leq \alpha$). We say that $\mathbf{H}$ *is a calculus for a logic* $\vdash$ whenever $\vdash \varphi$ iff

Initial hypersequents **Structural rules**

$$\frac{}{H \mid p \Rightarrow p} \qquad \frac{}{H \mid 0 \Rightarrow} \qquad \frac{}{H \mid \ \Rightarrow 1} \qquad \frac{H \mid \Gamma \Rightarrow \Pi \mid \Gamma \Rightarrow \Pi}{H \mid \Gamma \Rightarrow \Pi}\,(\mathrm{EC}) \qquad \frac{H}{H \mid \Gamma \Rightarrow \Pi}\,(\mathrm{EW})$$

Logical rules

$$\frac{H \mid \Gamma \Rightarrow \Pi}{H \mid \Gamma, 1 \Rightarrow \Pi}\,(1\mathrm{L}) \qquad \frac{H \mid \Gamma, \varphi, \psi \Rightarrow \Pi}{H \mid \Gamma, \varphi \cdot \psi \Rightarrow \Pi}\,(\cdot\mathrm{L}) \qquad \frac{H \mid \Gamma \Rightarrow \varphi \quad H \mid \Delta \Rightarrow \psi}{H \mid \Gamma, \Delta \Rightarrow \varphi \cdot \psi}\,(\cdot\mathrm{R})$$

$$\frac{H \mid \Gamma \Rightarrow}{H \mid \Gamma \Rightarrow 0}\,(0\mathrm{R}) \qquad \frac{H \mid \Gamma, \varphi \Rightarrow \psi}{H \mid \Gamma \Rightarrow \varphi \to \psi}\,(\to\mathrm{R}) \qquad \frac{H \mid \Gamma \Rightarrow \varphi \quad H \mid \Delta, \psi \Rightarrow \Pi}{H \mid \Gamma, \Delta, \varphi \to \psi \Rightarrow \Pi}\,(\to\mathrm{L})$$

$$\frac{H \mid \Gamma, \varphi_2 \Rightarrow \Pi}{H \mid \Gamma, \varphi_1 \wedge \varphi_2 \Rightarrow \Pi}\,(\wedge\mathrm{L}_1) \qquad \frac{H \mid \Gamma, \varphi_1 \Rightarrow \Pi}{H \mid \Gamma, \varphi_1 \wedge \varphi_2 \Rightarrow \Pi}\,(\wedge\mathrm{L}_2) \qquad \frac{H \mid \Gamma \Rightarrow \varphi \quad H \mid \Gamma \Rightarrow \psi}{H \mid \Gamma \Rightarrow \varphi \wedge \psi}\,(\wedge\mathrm{R})$$

$$\frac{H \mid \Gamma \Rightarrow \varphi_2}{H \mid \Gamma \Rightarrow \varphi_1 \vee \varphi_2}\,(\vee\mathrm{R}_1) \qquad \frac{H \mid \Gamma \Rightarrow \varphi_1}{H \mid \Gamma \Rightarrow \varphi_1 \vee \varphi_2}\,(\vee\mathrm{R}_2) \qquad \frac{H \mid \Gamma, \varphi \Rightarrow \Pi \quad H \mid \Gamma, \psi \Rightarrow \Pi}{H \mid \Gamma, \varphi \vee \psi \Rightarrow \Pi}\,(\vee\mathrm{L})$$

Figure 1 The hypersequent calculus $\mathbf{HFL_e}$ for $\mathbf{FL_e}$.

$\vdash_{\mathbf{H}} (\ \Rightarrow \varphi)$. In that case we also say that $\vdash$ *is determined by* $\mathbf{H}$, and by *provability in* $\mathbf{H}$, we mean the provability problem for $\vdash$. We say that $\mathbf{H}$ is *analytic* when for any hypersequent h it proves, there is a proof of h in $\mathbf{H}$ using only subformulas of h.

Ciabattoni et al. [7] provided analytic hypersequent calculi for infinitely many axiomatic extensions of the logic $\mathbf{FL_e}$ by adding *analytic structural hypersequent rules* to $\mathbf{HFL_e}$. In terms of the substructural hierarchy [7, 9], these logics correspond to acyclic axioms in level $\mathcal{P}'_3$ for extensions of $\mathbf{FL_{ec}}$, and in level $\mathcal{P}_3$ for extensions of $\mathbf{FL_{ew}}$. We show the form of such rules in the following definition. For an index set $J := \{j_1, \dots, j_k\}$, denote the hypersequent $H \mid \Gamma_{j_1} \Rightarrow \Pi_{j_1} \mid \dots \mid \Gamma_{j_k} \Rightarrow \Pi_{j_k}$ by $H \mid \Gamma_j \Rightarrow \Pi_j (j \in J)$.

▶ **Definition 2.2.** *An* analytic structural rule schema *(henceforth,* analytic rule*) has the following form, where all symbols in antecedents are multiset-variables, I, J and L are sets of indices, K_i is a set of indices for each $i \in I$, $\alpha_i \in \mathbb{N}$ for each $i \in I$, $\beta_j \in \mathbb{N}$ for each $j \in J$, and each $\vec{\Gamma}_{ik}$ and $\vec{\Delta}_l$ is a list of multiset-variables:*

$$\frac{\{H \mid Y_i, \vec{\Gamma}_{ik} \Rightarrow \Pi_i\}_{i \in I, k \in K_i} \qquad \{H \mid \vec{\Delta}_l \Rightarrow\}_{l \in L}}{H \mid Y_i, X_{i1}, \dots, X_{i\alpha_i} \Rightarrow \Pi_i \ (i \in I) \mid Z_{j1}, \dots, Z_{j\beta_j} \Rightarrow (j \in J)}$$

Moreover, it satisfies, among others (see [29]), the following properties: (i) linear conclusion *(the multiset-variables that occur in the conclusion occur exactly once) and (ii)* strong subformula property *(each multiset-variable in the premise occurs in the conclusion).*

▶ **Example 2.3** (Communication rule).

$$\frac{H \mid \Delta_1, \Gamma_1 \Rightarrow \Pi_1 \qquad H \mid \Delta_2, \Gamma_2 \Rightarrow \Pi_2}{H \mid \Delta_2, \Gamma_1 \Rightarrow \Pi_1 \mid \Delta_1, \Gamma_2 \Rightarrow \Pi_2}\,(com)$$

The fuzzy logic **MTL** *is determined by the calculus* $\mathbf{HFL_{ew}}(\{com\})$. *For more examples of such schemas, see [7].*

We will always use $\mathcal{R}$ to denote a finite set of analytic structural rule schemas. In this way, logics determined by calculi of the form $\mathbf{HFL_{ec}}(\mathcal{R})$ and $\mathbf{HFL_{ew}}(\mathcal{R})$ will be our main focus here. Observe that in such calculi every rule schema except for (EC) and (EW) have the following form, for some $n, m \geq 1$, where $T_1, \dots, T_m, S_1, \dots, S_n$ are components, and H is a hypersequent-variable.

$$\frac{H \mid T_1 \qquad \cdots \qquad H \mid T_m}{H \mid S_1 \mid \cdots \mid S_n}$$

We will design algorithms that accept hypersequents as input. For the size of the input, we employ the following norm, which is related to the length $\langle h \rangle$ of representation on irredundant hypersequents by a polynomial factor.

▶ **Definition 2.4** ((Hyper)sequent norms)**.** *For any sequent s, we define $[\![s]\!]$ as the maximum multiplicity of a formula in the antecedent of s. For any hypersequent h, we define $[\![h]\!] := \max_{s \in h} [\![s]\!]$ (i.e., the max of the norms of its components).*

As we shall later see, for our purposes, it would be helpful to think of h as a sequence of sequents, with no repetitions. In that context, $[\![h]\!]$ tracks the largest size of any sequent that we have seen so far. Note that for hypersequents with no repetitions of sequents, this is a proper norm (see Section 2.2), i.e., for any number n, there can only be finitely many such hypersequents h such that $[\![h]\!] = n$.

2.1.1 Invertible calculi

In order to develop our proof search procedures, we modify the usual hypersequent calculi to allow components in the conclusion of a rule to persist upwards.

Fix a finite set $\mathcal{R}$ of analytic structural rule schemas and let $\mathbf{H} \in \{\mathbf{HFL_{ec}}(\mathcal{R}), \mathbf{HFL_{ew}}(\mathcal{R})\}$. Given a rule schema r that is not in $\mathcal{S} := \{(\mathrm{EC}), (\mathrm{EW})\}$, the *invertible form* $\mathsf{inv}(\mathsf{r})$ *of* r is obtained by including the principal components in every premise. The *principal* and *active* components of $\mathsf{inv}(\mathsf{r})$ are the principal and active components in r.

▶ **Example 2.5.** *The invertible form of* (com) *is*

$$\frac{h_c \mid \Delta_1, \Gamma_1 \Rightarrow \Pi_1 \qquad h_c \mid \Delta_2, \Gamma_2 \Rightarrow \Pi_2}{h_c}\ \mathsf{inv}(com)$$

where $h_c = (H \mid \Delta_2, \Gamma_1 \Rightarrow \Pi_1 \mid \Delta_1, \Gamma_2 \Rightarrow \Pi_2)$. The principal components of $\mathsf{inv}(com)$ *are $\Delta_2, \Gamma_1 \Rightarrow \Pi_1$ and $\Delta_1, \Gamma_2 \Rightarrow \Pi_2$; the active components are $\Delta_1, \Gamma_1 \Rightarrow \Pi_1$ and $\Delta_2, \Gamma_2 \Rightarrow \Pi_2$.*

▶ **Definition 2.6** (Invertible calculi)**.** *The calculus $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ec}}(\mathcal{R})$ (resp., $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})$) is obtained by replacing in $\mathbf{HFL_{ec}}(\mathcal{R})$ (resp., in $\mathbf{HFL_{ew}}(\mathcal{R})$) every rule schema r not in $\mathcal{S}$ by $\mathsf{inv}(\mathsf{r})$.*

The following is straightforward to prove by using (EC) and (EW).

▶ **Lemma 2.7.** *$\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ec}}(\mathcal{R})$ and $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})$ prove the same hypersequents as $\mathbf{HFL_{ec}}(\mathcal{R})$ and $\mathbf{HFL_{ew}}(\mathcal{R})$, respectively.*

It is also clear that $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ec}}(\mathcal{R})$ and $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})$ satisfy the subformula property, because $\mathbf{HFL_{ec}}(\mathcal{R})$ and $\mathbf{HFL_{ew}}(\mathcal{R})$ do.

2.2 Well-quasi-orders

A *well-quasi-order* (or wqo) is a structure $\mathbf{A} := \langle A, \leq_{\mathbf{A}} \rangle$ where A is a set and $\leq_{\mathbf{A}} \subseteq A \times A$ is a reflexive and transitive relation s.t. for every infinite sequence of elements $a_0, a_1, \ldots,$ of A there exists $i < j$ s.t. $a_i \leq_{\mathbf{A}} a_j$. A *proper norm* on a set A is a function $[\![\cdot]\!]_{\mathbf{A}} : A \to \mathbb{N}$ such that for every $n \in \mathbb{N}$, the set $\{a : [\![a]\!]_{\mathbf{A}} \leq n\}$ is finite. A *normed wqo* (or nwqo) is a structure $\mathbf{A} := \langle A, \leq_{\mathbf{A}}, [\![\cdot]\!]_{\mathbf{A}} \rangle$ such that $\langle A, \leq_{\mathbf{A}} \rangle$ is a wqo and $[\![\cdot]\!]_{\mathbf{A}}$ is a proper norm.

It is easy to see that the set of natural numbers $\mathbb{N}$ with the usual ordering and the norm being the identity function is a nwqo. It turns out that this observation can be generalized as follows. For any $d \geq 1$, let $\mathbb{N}^d$ denote the set of all d-tuples of natural numbers. Given

$\mathbf{x} = (x_1, \ldots, x_d)$ and $\mathbf{y} = (y_1, \ldots, y_d)$ in $\mathbb{N}^d$ we say that $\mathbf{x} \leq_{\mathbb{N}^d} \mathbf{y}$ if $x_i \leq y_i$ for each i. Finally, we set the norm of an element $\mathbf{x} = (x_1, \ldots, x_d)$ as $[\![x]\!]_{\mathbb{N}^d} = \max_{1 \leq i \leq d} x_i$. By Dickson's lemma [11], it is known that $\mathbf{N}^d \coloneqq \langle \mathbb{N}^d, \leq_{\mathbb{N}^d}, [\![\cdot]\!]_{\mathbb{N}^d} \rangle$ is an nwqo for every d. This nwqo is called the product nwqo over $\mathbb{N}^d$. In the sequel, when the dimension d is clear from context, we will drop the subscript $\mathbb{N}^d$ from $\leq_{\mathbb{N}^d}$ and $[\![\cdot]\!]_{\mathbb{N}^d}$. We will also make use of *disjoint sums* of nwqos, in particular of nwqos of the form $k \cdot \mathbf{N}^d$ $(k, d \in \mathbb{N})$, where $(i, \mathbf{x}) \leq_{k \cdot \mathbb{N}^d} (j, \mathbf{y})$ iff $i = j$ (i.e., $\mathbf{x}$ and $\mathbf{y}$ are in the same copy of $\mathbb{N}^d$) and $\mathbf{x} \leq_{\mathbb{N}^d} \mathbf{y}$; here, the norm is defined by $[\![(i, \mathbf{x})]\!]_{k \cdot \mathbb{N}^d} \coloneqq [\![\mathbf{x}]\!]_{\mathbb{N}^d}$.

Controlled bad sequences

A *control function* is any function $f : \mathbb{N} \to \mathbb{N}$ that is monotone and satisfies $f(x) \geq x$ for all $x \in \mathbb{N}$. Now, let f be any control function, let $n \in \mathbb{N}$ and let $\mathbf{A} \coloneqq \langle A, \leq_{\mathbf{A}}, [\![\cdot]\!]_{\mathbf{A}} \rangle$ be any nwqo. A sequence of elements $a_0, a_1, \ldots$ over A is called a (f, n)-*controlled bad sequence* iff

- There is no $i < j$ such that $a_i \leq_{\mathbf{A}} a_j$.
- For every i, $[\![a_i]\!]_A \leq f^i(n)$ where f^i denotes the i-fold composition of f with itself.

By definition of a wqo and by definition of the first property of a (f, n)-controlled bad sequence, it follows that no such sequence can be infinite. It turns out that, by an application of König's lemma, for every f, every n and every nwqo $\mathbf{A}$, there is a maximum length for (f, n)-controlled bad sequences of $\mathbf{A}$. Hence, we can define a *length function* $L_{\mathbf{A},f}$ for $\mathbf{A}$ and f which maps a number n to the maximum length of (f, n)-controlled bad sequences of $\mathbf{A}$.

Length function bounds for $k \cdot \mathbf{N}^d$

The complexity upper bounds that we prove in this paper will ultimately depend on length functions for nwoqs $k \cdot \mathbf{N}^d$ (for some $d, k \in \mathbb{N}$). Hence, we recall existing results on upper bounds for them. To state these upper bounds, we first need to introduce the so-called *fast-growing complexity classes*, which we shall do now. We will only state those notions and facts that are needed here.

First, we define a hierarchy of functions $\{F_i\}_{i \leq \omega}$ as follows. Let $F_1(n) = 2n$ and let $F_{i+1}(n) = F_i^n(n)$. We then set $F_\omega(n) = F_n(n)$, where we can think of ω as an element that is larger than any natural number. Using these functions, we can define a family of functions $\mathscr{F}_i$ for each $i \in \mathbb{N} \cup \{\omega\}$ by closing the F_i function (and a bunch of other simple functions like the successor function) under composition and limited primitive recursion. We let $\mathbf{F}_i^\star$ denote the class of functions given by $\bigcup_{j<i} \bigcup_{p \in \mathscr{F}_j} \{F_i(p(n))\}$. All that we require for our purposes regarding $\mathbf{F}_i^\star$ are the following facts.

▶ **Theorem 2.8** ([31]). *If f is a function in $\mathbf{F}_\omega^\star$ and g_1, g_2 are primitive recursive functions, then the function $g_1 \circ f \circ g_2$ is also in $\mathbf{F}_\omega^\star$.*

▶ **Theorem 2.9** (Length theorem for Dickson's lemma [13]). *For any control function f, there is $U : \mathbb{N}^3 \to \mathbb{N}$ in $\mathbf{F}_\omega^\star$ s.t., for any $k, d \geq 0$, $U(k, d, x)$ upper bounds the length function $L_{k \cdot \mathbb{N}^d, f}(x)$.*

Let $\mathbf{F}_i$ be the complexity class defined as the set of decision problems that can be solved by a deterministic Turing machine in time $f \in \mathbf{F}_i^\star$. The distinction between determinism and non-determinism and time and space becomes irrelevant for $\mathbf{F}_i$ with $i > 2$, because such classes are closed under exponential-time reductions. Of primary interest to us is the class $\mathbf{F}_\omega$, whose members are said to have *Ackermann* complexity. The hierarchy is, actually, more commonly defined for ordinals up to ϵ_0 [31]. In particular, $\mathbf{F}_{\omega^\omega}$ is the class of *hyper-Ackermann* problems.

3 Ackermann bounds for analytic hypersequent extensions of $\mathbf{FL_{ec}}$

3.1 Refined calculi and their key properties

Ramanayake [29] constructed a refined version of $\mathbf{HFL_{ec}}(\mathcal{R})$ in which (EW), (EC) and $(\mathbf{c})$ are *height-preserving admissible* (or *hp-admissible*, for short), meaning that whenever a premise of one of these rules has a proof with height $\alpha \in \mathbb{N}$, its conclusion has a proof of height at most α. We recall the construction, and a key consequence of it: *Curry's lemma.*

In order to facilitate the formulation of the refined calculus and the proof that it satisfies the desired hp-admissibility results, we define the binary relations $\rightsquigarrow^k_{\mathbf{c}}$, $\rightsquigarrow^k_{(\mathrm{EC})}$ and $\rightsquigarrow^k_l$ on hypersequents (for each $k, l \geq 0$) in the following way.

- $h_1 \rightsquigarrow^k_{\mathbf{c}} h_2$ iff h_2 can be obtained from h_1 by applying some number of instances of $\mathbf{c}$ such that every formula in a component in h_2 occurs up to k times less in that component than in the corresponding component in h_1.
- $h_1 \rightsquigarrow^l_{(\mathrm{EC})} h_2$ iff h_2 can be obtained from h_1 by applying some number of instances of (EC) such that every component of h_2 occurs up to l times less in h_2 than in h_1.
- $h_1 \rightsquigarrow^k_l h_2$ iff there exists h' s.t. $h_1 \rightsquigarrow^k_{\mathbf{c}} h'$ and $h' \rightsquigarrow^l_{(\mathrm{EC})} h_2$.

The following is a fact that comes easily from these definitions.

▶ **Proposition 3.1.** *If $h_1 \rightsquigarrow^k_l h_2$, then, for each $s_1 \in h_1$, there is $s_2 \in h_2$ such that $s_1 \rightsquigarrow^k_{\mathbf{c}} s_2$. We call s_2 a* descendant *of s_1 in h_2.*

The idea of the refined calculus is that we introduce to each rule instance a fixed amount of contractions (internal and external). This amount is determined by the syntactic form of the calculus, using the values fm and acn defined below.

▶ **Definition 3.2** ([29]). *The* formula multiplicity *of a rule schema is the maximum, over all antecedents in its principal components, of the number of variables occurring in those antecedents. The* active component number *of a rule schema is the number of components in its conclusion (including the hypersequent variable).*

The formula multiplicity fm *of a calculus* $\mathbf{HFL_{ec}}(\mathcal{R})$ *is the maximum of the formula multiplicities of its rule schemas. Similarly, the active component number* acn *is the maximum of the active component numbers of its rule schemas.*

E.g., the conclusion $H \mid \Gamma \Rightarrow \Pi$ of the (EW) rule schema has a single principal component; its antecedent consists of a single (schematic) variable Γ, so its formula multiplicity is 1. Its active component number is 2 since the conclusion has two components: this principal component as well as the hypersequent variable. The conclusion $H \mid \Delta_2, \Gamma_1 \Rightarrow \Pi_1 \mid \Delta_1, \Gamma_2 \Rightarrow \Pi_2$ of (*com*) (Example 2.3) has two principal components, with antecedents Δ_2, Γ_1 and Δ_1, Γ_2. Each of these lists contains two (schematic) variables, so its formula multiplicity is 2.

▶ **Definition 3.3** (Refined contraction-calculus). *Let* $\mathbf{H}$ *be a hypersequent calculus containing* (EC) *and* $(\mathbf{c})$*. Define* $\mathbf{H}^{\rightsquigarrow}$ *as the set of rules* $\mathsf{r}^{\rightsquigarrow}$ *of the form* $(h_1, \ldots, h_n)/g$*, where* $(h_1, \ldots, h_n)/h_0$ *is a rule instance of* $\mathsf{r} \in \mathbf{H}$ *and* $h_0 \rightsquigarrow^{\mathsf{fm}-1}_{\mathsf{acn}} g$*. The instance with conclusion* h_0 *is called the* base instance *of* $\mathsf{r}^{\rightsquigarrow}$*.*

Clearly, the conclusion g of a non-base instance is obtained from the conclusion h_0 of the base instance by performing a restricted number of applications of $(\mathbf{c})$ (via $\rightsquigarrow^{\mathsf{fm}-1}_{\mathbf{c}}$) and then a restricted number of applications of (EC) (via $\rightsquigarrow^{\mathsf{acn}}_{(\mathrm{EC})}$).

▶ **Lemma 3.4** ([29]). *Let* $\mathcal{R}$ *be a finite set of analytic structural rules.*

1. *The calculi* $\mathbf{HFL_{ec}}(\mathcal{R})$ *and* $\mathbf{HFL_{ec}}(\mathcal{R})^{\rightsquigarrow}$ *prove the same hypersequents.*
2. (EW)*,* (EC) *and* $(\mathbf{c})$ *are hp-admissible in* $\mathbf{HFL_{ec}}(\mathcal{R})^{\rightsquigarrow}$*.*

Now we obtain the so-called *Curry's lemma* for hypersequents.

▶ **Definition 3.5.** *Given hypersequents g and h, define $g \Leftarrow_{\mathsf{str}} h$ iff g is obtained from h by applications of* $\{(\mathrm{EW}), (\mathrm{EC}), (\mathbf{c})\}$.

▶ **Lemma 3.6** (Hypersequent Curry's lemma). *Let h' and h be hypersequents s.t. $h' \Leftarrow_{\mathsf{str}} h$. If $\vdash^{\alpha}_{\mathbf{HFL_{ec}}(\mathcal{R})^{\rightsquigarrow}} h$, then $\vdash^{\leq\alpha}_{\mathbf{HFL_{ec}}(\mathcal{R})^{\rightsquigarrow}} h'$.*

From [4], we know that $\Leftarrow_{\mathsf{str}}$ is a wqo when restricted to hypersequents over the same support. Then the whole point of Curry's lemma is to prove existence of $\Leftarrow_{\mathsf{str}}$-*minimal proofs* in $\mathbf{HFL_{ec}}(\mathcal{R})^{\rightsquigarrow}$, that is, every provable hypersequent is witnessed by a proof in which every branch is a bad sequence with respect to $\Leftarrow_{\mathsf{str}}$. Then a proof-search procedure for $\mathbf{HFL_{ec}}(\mathcal{R})^{\rightsquigarrow}$ looks for minimal proofs with respect to this wqo. Our new algorithm will not follow this path, because $\Leftarrow_{\mathsf{str}}$ is too complex and leads to $\mathbf{F}_{\omega^\omega}$ upper bounds, instead of $\mathbf{F}_{\omega}$. In order to design this algorithm, we need to combine the two transformations on hypersequent calculi described above, namely $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ec}}(\mathcal{R})$ and $\mathbf{HFL_{ec}}(\mathcal{R})^{\rightsquigarrow}$ into one.

3.2 Invertible refined calculi for contraction

Our focus from now on will be on calculi of the form $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ec}}(\mathcal{R})^{\rightsquigarrow}$, meaning the result of applying the $(\cdot)^{\rightsquigarrow}$ transformation to $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ec}}(\mathcal{R})$. The following result shows that provability is height-preserved via such transformation with respect to $\mathbf{HFL_{ec}}(\mathcal{R})^{\rightsquigarrow}$.

▶ **Lemma 3.7.** *Let h be a hypersequent and $\alpha \in \mathbb{N}$. If $\vdash^{\alpha}_{\mathbf{HFL_{ec}}(\mathcal{R})^{\rightsquigarrow}} h$, then $\vdash^{\leq\alpha}_{\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ec}}(\mathcal{R})^{\rightsquigarrow}} h$, and vice versa.*

Then, by Lem 3.6 and Lem 3.7:

▶ **Lemma 3.8.** *Let h' and h be hypersequents such that $h' \Leftarrow_{\mathsf{str}} h$. If $\vdash^{\alpha}_{\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ec}}(\mathcal{R})^{\rightsquigarrow}} h$, then $\vdash^{\leq\alpha}_{\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ec}}(\mathcal{R})^{\rightsquigarrow}} h'$.*

3.3 Proof-search procedure

The core idea is the following: instead of controlling the length of each branch in the proof search (wqo on hypersequents), a complete proof-search procedure is given where every hypersequent in the proof search tree is itself a bad sequence (wqo on sequents).

Throughout this section, we denote by $\mathbf{H}$ a calculus of the form $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ec}}(\mathcal{R})^{\rightsquigarrow}$ for some finite set $\mathcal{R}$ of analytic structural rules.

A *linearisation* of a hypersequent is any sequence of its components (preserving the multiplicities). For example, if $h = s_1 \mid s_2 \mid s_3$ is a hypersequent, then a linearization of h is any sequence (s_i, s_j, s_k), where $\{i, j, k\} = \{1, 2, 3\}$.

Next, we define a nwqo on S-sequents, where S is a finite set of formulas. Fix an enumeration $S = \{\varphi_1, \dots, \varphi_d\}$. A multiset Γ of such formulas can be viewed as a vector in $v(\Gamma) \in \mathbb{N}^d$, where the i^{th} component is the number of occurrences of φ_i in Γ. We say that $\Gamma_1 \leq \Gamma_2$ iff $v(\Gamma_1) \leq_{\mathbb{N}^d} v(\Gamma_2)$ according to the product wqo over $\mathbb{N}^d$. Hence, the defined order among multisets of formulas is also a wqo.

Now, for two S-sequents s_1, s_2 we say that $s_1 \leq_S s_2$ iff $s_1 := \Gamma_1 \Rightarrow \Pi_1$, $s_2 = \Gamma_2 \Rightarrow \Pi_2$, $\Pi_1 = \Pi_2$ and $\Gamma_1 \leq \Gamma_2$. It is easily seen that this order on the S-sequents is also a wqo. Hence, if we now consider the norm for sequents defined in Def 2.4, we get a nwqo for S-sequents, which we denote by $\mathbf{W}(S)$.

We ultimately want to consider only those hypersequents in our proof search whose linearisation yields a controlled bad sequence of sequents (for some suitable control function). To that end, we say that a function f is an $\mathbf{H}$-*control* if $[\![h']\!] \leq f([\![h]\!])$ for every rule instance in

$\mathbf{H}$ having h as conclusion and h' as a premise. (Here we are using the norm for hypersequents in Def 2.4.) It is easy to see that a primitive recursive $\mathbf{H}$-control exists – indeed, even a linear function – for every hypersequent calculus $\mathbf{H}$ considered here.

▶ **Lemma 3.9.** *The function* $f(x) = (\mathsf{fm}^2\mathsf{acn}^2 \cdot x) + 1$ *is an* $\mathbf{H}$*-control.*

Now, let π be a derivation of some hypersequent h_0 in $\mathbf{H}$, and let $h_0, \ldots, h_n$ be the sequence of hypersequents along any branch B in π. Let r_k $(0 < k \leq n)$ be the corresponding rule instance in π with conclusion h_{k-1} and premise h_k, and let s_k be the active component in that premise. For any linearisation $\bar{h}_0$ of h_0, the sequence $\bar{h}_0, s_1, \ldots, s_k$ is called the $\bar{h}_0$*-sequence of* B.

Let $\bar{h}$ be any linearisation of a hypersequent h and let S be the set of subformulas of h; let $\bar{h}(0)$ denote the first element in the sequence. Given a control function f, a f*-minimal proof of* $\bar{h}$ in $\mathbf{H}$ is a proof of h such that the $\bar{h}$-sequence of every branch is a $(f, [\![\bar{h}(0)]\!])$-controlled bad sequence over $\mathbf{W}(S)$. This is well-defined as proofs in $\mathbf{H}$ are cut-free and thus all elements of the $\bar{h}$-sequences are S-sequents.

▶ **Lemma 3.10** (Existence of minimal proofs)**.** *Let f be any $\mathbf{H}$-control. Let $\bar{h}$ be a linearization of a hypersequent h that is a $(f, [\![\bar{h}(0)]\!])$-controlled bad sequence over $\mathbf{W}(S)$, where S is the set of subformulas of h. Then, h is provable in $\mathbf{H}$ iff $\bar{h}$ has an f-minimal proof in $\mathbf{H}$.*

Proof. The right-to-left direction is trivial: an f-minimal proof of $\bar{h}$ is a proof of h. For the left-to-right direction, suppose that π is a proof of h in $\mathbf{H}$. Induction on the height α of π.

(*Base case*) If π has height 1, then h must be an initial hypersequent. The latter is a proof containing a singleton branch. Consequently, the $\bar{h}$-sequence of this branch is simply $\bar{h}$. The latter is a $(f, [\![\bar{h}(0)]\!])$-bad sequence by hypothesis, so the claim holds.

(*Inductive step*) Suppose π has height $\alpha > 1$. Consider the last rule $\mathsf{r}^{\rightsquigarrow}$ applied in π. Let π_i be the subproof of i-th premise of $\mathsf{r}^{\rightsquigarrow}$, with height $\alpha_i < \alpha$. Note that the root of π_i has the form $g \,|\, C \,|\, s_i$, where s_i is the active component of the rule and C is the multiset of principal components of the base instance of $\mathsf{r}^{\rightsquigarrow}$. Here $g \,|\, C \rightsquigarrow^{\mathsf{fm}-1}_{\mathsf{acn}} h$.

$$\dfrac{\begin{matrix}\pi_1 \\ g \,|\, C \,|\, s_1\end{matrix} \qquad \ldots \qquad \begin{matrix}\pi_n \\ g \,|\, C \,|\, s_n\end{matrix}}{g \,|\, C}\ \mathsf{r}$$

We then have that $h \,|\, s_i \Leftarrow_{\mathsf{str}} g \,|\, C \,|\, s_i$ for each $1 \leq i \leq n$, and thus, by Lem 3.8, there are proofs π'_i of each $h \,|\, s_i$ with height $\leq \alpha_i < \alpha$. We now reason by cases on the sequences $\bar{h}, s_i$:

1. $\bar{h}, s_i$ is a $(f, [\![\bar{h}(0)]\!])$-controlled bad sequence for every i: Applying the IH to π'_i, we obtain a f-minimal proof π''_i of $\bar{h}, s_i$ for each i. By definition, the $(\bar{h}, s_i)$-sequence of every branch is a $(f, [\![\bar{h}(0)]\!])$-controlled bad sequence. We now claim that the following is a proof in $\mathbf{H}$, with $h \,|\, C \rightsquigarrow^{\mathsf{fm}-1}_{\mathsf{acn}} h$.

$$\dfrac{\begin{matrix}\pi''_1 \\ h \,|\, C \,|\, s_1\end{matrix} \qquad \ldots \qquad \begin{matrix}\pi''_n \\ h \,|\, C \,|\, s_n\end{matrix}}{h \,|\, C}\ \mathsf{r}$$

For that, it is enough to show that $h \,|\, C \rightsquigarrow^{\mathsf{fm}-1}_{\mathsf{acn}} h$. We know that $g \,|\, C \rightsquigarrow^{\mathsf{fm}-1}_{\mathsf{acn}} h$. Therefore, by Prop 3.1, for each $s' \in C$, there is $s'' \in h$ such that $s' \rightsquigarrow^{\mathsf{fm}-1}_{\mathsf{c}} s''$. Thus, $h \,|\, C \rightsquigarrow^{\mathsf{fm}-1}_{\mathsf{c}} h \,|\, C'$, where every component in C' appears in h. But $|C| \leq \mathsf{acn}$ because C is the multiset of principal components of the base instance, and so $h \,|\, C' \rightsquigarrow^{\mathsf{acn}}_{(\mathrm{EC})} h$. Hence, $h \,|\, C \rightsquigarrow^{\mathsf{fm}-1}_{\mathsf{acn}} h$, as desired.
It only remains to argue that the above proof, which we call π_0, is an f-minimal proof of $\bar{h}$. The $\bar{h}$-sequence of any branch in π_0 has the form $\bar{h}, s_i, \ldots$ for some i. By inspection, it is the $(\bar{h}, s_i)$-sequence of some branch in π''_i. We already observed that the latter is a $(f, [\![\bar{h}(0)]\!])$-controlled bad sequence. Thus, π_0 is an f-minimal proof of $\bar{h}$.

2. $\bar{h}, s_i$ is not a $(f, [\![\bar{h}(0)]\!])$-controlled bad sequence for some i: Note that $\bar{h}$ is a $(f, [\![\bar{h}(0)]\!])$-controlled bad sequence by assumption. We now argue that this sequence must be $(f, [\![\bar{h}(0)]\!])$-controlled (so that it must fail to be bad). Because $[\![g \mid C \mid s_i]\!] \leq f([\![h]\!])$ by Lem 3.9, and by definition of $[\![\cdot]\!]$ we have $[\![s_i]\!] \leq [\![g \mid C \mid s_i]\!]$, we obtain $[\![s_i]\!] \leq f([\![h]\!])$. By definition of the norms for hypersequents we get that $[\![h]\!] = [\![\bar{h}(j)]\!]$ for some $0 \leq j < |h|$. By the control property, we have $[\![\bar{h}(j)]\!] \leq f^j([\![\bar{h}(0)]\!]) \leq f^{|h|-1}([\![\bar{h}(0)]\!])$. Thus $[\![s_i]\!] \leq f(f^{|h|-1}([\![\bar{h}(0)]\!])) = f^{|h|}([\![\bar{h}(0)]\!])$, and $\bar{h}, s_i$ is controlled. So the sequence must fail to be bad, hence there exists $t \in \bar{h}$ s.t. t can be obtained from s_i by (zero or more) contractions.
In either case, it follows that $h \Leftarrow_{\mathsf{str}} h|s_i$. By Lem 3.8 we obtain a proof π_0 of h of height $< \alpha$. Applying the IH to π_0, we obtain an f-minimal proof of $\bar{h}$. ◀

▶ **Definition 3.11** (f-minimal proof-search tree)**.** *The* f-minimal proof search tree *in* $\mathbf{H}$ *for the sequent s is defined as the limit tree of the following.*

$T_0(s)$ *A single node labelled with s.*

$T_{n+1}(s)$ *For each leaf ℓ (labelled by the hypersequent h, say) in $T_n(s)$, and each rule instance that has h as conclusion, add the premises as children of ℓ, provided the tree so obtained is f-minimal.*

Branching can happen for two reasons: rules with multiple premises, and the applicability of multiple rule instances. We do not have to differentiate between these children because the proof-search tree is only an over-approximation: later we extract a subtree that is a proof.

▶ **Lemma 3.12.** *Given a sequent s, there is a least $N \in \mathbb{N}$ such that $T_N(s) = T_{N+1}(s)$. Moreover, there is $U \in \mathbf{F}^\star_\omega$ such that N is upper bounded by $U([\![s]\!])$ for any sequent s.*

Proof. Let S be the set of subformulas of s. By construction, for every n, every branch in $T_n(s)$ corresponds to a $(f, [\![s]\!])$-controlled bad sequence over $\mathbf{W}(S)$. Suppose there is no $N \in \mathbb{N}$ such that $T_N(s) = T_{N+1}(s)$. Then, consider the infinite limit tree $\bigcup_{n \geq 0} T_n(s)$. This tree must be finitely branching, since every rule schema of $\mathbf{H}$ can only be instantiated in finitely many ways for every hypersequent. By König's Lemma there would exist an infinite branch, which induces an infinite bad sequence over $\mathbf{W}(S)$, which is absurd. Moreover, this N is uniformly upper bounded by a function in $\mathbf{F}^\star_\omega$ because each step in the construction adds a new element to a $(f, [\![s]\!])$-controlled bad sequence producing a longer $(f, [\![s]\!])$-controlled bad sequence, and by Thm 2.9 we know that the length of such a sequence is upper bounded by a function in $\mathbf{F}^\star_\omega$ on $[\![s]\!]$. ◀

From the above stabilization of the proof-search tree construction, we now conclude:

▶ **Theorem 3.13.** *Provability in $\mathbf{H}$ is in $\mathbf{F}_\omega$.*

Proof. Given an input sequent s, by Lem 3.12 we have that $T_N(s)$ can be constructed in space primitive recursive on $U \in \mathbf{F}^\star_\omega$, which is thus also a function in $\mathbf{F}^\star_\omega$ by Thm 2.8. Furthermore, we can search among its subtrees for a minimal proof of s, which is again a primitive-recursive procedure by Thm 2.8. By construction, any f-minimal proof of s will be a subtree of $T_N(s)$. By Lem 3.10, its existence is equivalent to provability of s. ◀

Finally, from Lem 2.7, Lem 3.7 and Thm 3.13, we obtain:

▶ **Theorem 3.14** (Main Theorem I)**.** *Provability in any extension of $\mathbf{FL_{ec}}$ by finitely many acyclic $\mathcal{P}'_3$-axioms (i.e, provability in any analytic structural extension of $\mathbf{HFL_{ec}}$) has Ackermann complexity.*

4 Ackermann bounds for hypersequent extensions of $\mathbf{FL_{ew}}$: ω-calculi

We take inspiration from the Karp-Miller algorithm for computing coverability sets in vector addition systems [19]. Let $L_\omega \coloneqq L \cup \{\varphi^\omega \mid \varphi \in L\}$ be the language of *ω-formulas*. Note that, in this language we allow, e.g., $(\varphi \wedge \psi)^\omega$, but not $\varphi^\omega \wedge \psi^\omega$. Then we call *$\omega$-components* (resp., *$\omega$-hypersequents*) those sequents (resp., *hypersequents*) in which ω-formulas may appear, and we call *ω-calculus* a calculus whose rules manipulate ω-hypersequents. An *ω-proof* is a proof in an ω-calculus.

The basic idea behind the new strategy is to perform backward ω-proof search over a suitable ω-calculus, such that if the active component of a premise is strictly larger than an earlier component, we record this increment by replacing every formula φ that has strictly larger multiplicity with φ^ω. The proof rules of the ω-calculus will guarantee that once a formula becomes an ω-formula, it remains so from that point on. This mechanism together with a careful bookkeeping of proof-theoretical information about how new components are introduced in the proof search will guarantee termination and correctness.

This section starts with a more detailed high-level description of the algorithm and why it works, and then we move to the formal definition of the ω-calculus. The remaining sections will then formalize the proof search and prove its correctness.

An overview of the ω-proof-search algorithm

Given an input hypersequent, perform the usual backward proof search, exhaustively applying the rules, until an active component is encountered (in a premise) that is larger than some component that is already present. Instead of concatenating this active component to the conclusion to get the premise, we concatenate a revised version of the active component, by replacing every ψ^k in it that had strictly larger multiplicity than its counterpart (in the already-present component) with ψ^ω. So our proof search is built on ω-hypersequents, itself built from ω-components, where the antecedent consists of an ω-set – the set of ω-formulas – and a vector in $\mathbb{N}^d$ holding the information about the formulas with finite multiplicities. A formula is an ω-formula or has finite multiplicity (but not both).

We must show that it is possible to obtain a usual proof from the ω-proof (soundness), the proof search terminates, and it finds a proof whenever there is one to be found (completeness).

For soundness, the idea is that the path from an ω-hypersequent where its active component strictly increased its ω-set down to its ω-partner below can be iterated as often as required, to replace, in the usual proof we are now building, ψ^ω with ψ^k for suitably large k. Moreover, we can work out the required value for k by a leaf-to-root analysis of the (finite) ω-proof. It is helpful to consider the path in the other direction: from the rule instance introducing the ω-partner to the revised active component with strictly larger ω-set. We need to pass the additional copies of formula ψ created at each iteration, so that the multiplicity keeps growing. For this to work, we need the ω-partner and the revised active component to be related also in proof-theoretic sense; the additional condition is that the pair should be in the transitive closure of the relation R of components that are instantiations of components in the rule schema sharing a multiset-variable (instantiations of these multiset-variables carry the additional formulas along to the next step in the iteration).

To show termination, suppose that the proof search tree were infinite. We can easily extract an infinite R-tree from this. To get a contradiction, we need to extract an infinite branch on this tree, and for that we need to ensure that R is finitely branching. The way we described R in the previous paragraph does not ensure finite branching: after all, the input hypersequent – since each premise is obtained by concatenating the conclusion with an active component, it is the initial segment of every hypersequent in the proof search – might have been used infinitely often. The solution is to be even more stringent on the R relation. Specifically, of the principal components in the conclusion of a rule instance that are proof-theoretically related to the active component in question, we relate the *rightmost* one ("*key ancestor*"). Now, if this key ancestor were subsequently related to infinitely many components under R, there must be a repeat since the number of components that can be generated from that key ancestor and those to its left (i.e., a fixed hypersequent) is finite.

For completeness, we will show that whatever is done in a usual proof can be simulated in the ω-proof. Since ψ^ω has a natural meaning as infinitely many copies of ψ, the suitable rules for ω-formulas are quite clear to deduce. In particular, ω-formulas persist upwards.

4.1 Preliminary notation

Throughout this and the remaining sections, as usual, we use $\mathcal{R}$ to mean an arbitrary finite set of analytic hypersequent structural rule schemas (recall Def 2.2). Since $\mathbf{HFL}^{\rightleftharpoons}_{\mathbf{ew}}(\mathcal{R})$ satisfies the subformula property, a proof of an input hypersequent can be restricted to the finite set of its subformulas, say $S \coloneqq \{\varphi_1, \ldots, \varphi_d\}$. For the latter, we use $\mathbb{N}_{\leq d} \coloneqq \{0, 1, \ldots, d\}$, where 0 represents the empty stoup. I.e., a formula is represented by its index in a fixed enumeration of S. We also let $\mathbb{N}^+_{\leq d} \coloneqq \{1, \ldots, d\}$. Also, as in the previous section, we read a multiset over S as a vector $\mathbf{x} \in \mathbb{N}^d$, such that $\mathbf{x}(a)$ is the multiplicity of the formula (with index) a in the multiset. This motivates the following formalization.

▶ **Definition 4.1** (ω-(hyper)sequents)**.** *Let $d \geq 1$. An element $(\Omega; \mathbf{x}) \in \mathcal{P}(\mathbb{N}^+_{\leq d}) \times \mathbb{N}^d$ such that $\Omega \cap \{i \in \mathbb{N}^+_{\leq d} \mid \mathbf{x}(i) > 0\} = \varnothing$ is called an (ω, d)-*antecedent*. Elements in Ω (resp., $\mathbf{x}$) are called ω-*coordinates*. An (ω, d)-*sequent* is an expression $(\Omega; \mathbf{x}) \Rightarrow b$, where $(\Omega; \mathbf{x})$ is an (ω, d)-antecedent and $b \in \mathbb{N}_{\leq d}$. We call Ω the ω-*set* of the ω-sequent. An (ω, d)-*hypersequent* is a multiset of (ω, d)-sequents. We may omit the d from the notation when it can be inferred from the context. An ω-sequent is ω-*free* when its ω-set is empty, and an ω-hypersequent is ω-*free* when all of its components are ω-free.*

Observe that an (ω, d)-antecedent $(\Omega; \mathbf{x})$ has the property that every formula $a \in \mathbb{N}^+_{\leq d}$ is either an ω-formula (i.e., $a \in \Omega$ and $\mathbf{x}(i) = 0$), or it is not an ω-formula ($a \notin \Omega$).

▶ **Notation 4.2.** *Given $\kappa \subseteq \mathbb{N}^+_{\leq d}$, $k \in \mathbb{N}$ and $\mathbf{x} \in \mathbb{N}^d$, let $\mathbf{x}[\kappa \mapsto k]$ denote the vector obtained by setting the coordinates of $\mathbf{x}$ in κ to k and leaving the other coordinates unchanged. For $\mathbf{x} \in \mathbb{N}^d$ and $1 \leq a \leq d$, we write $\mathbf{x}, a$ to mean $\mathbf{x} + \mathbf{e}_a$, where $\mathbf{e}_a$ is the a-th unit vector.*

▶ **Definition 4.3** (Adding formulas to $(\Omega; \mathbf{x})$)**.** *Define the ω-antecedent $((\Omega; \mathbf{x}), a)$ obtained by the addition of the formula a to $(\Omega; \mathbf{x})$ to be $(\Omega; \mathbf{x})$ if $a \in \Omega$, else it is $(\Omega; \mathbf{x}, a)$. For a multiset $I = [a_1, \ldots, a_k]$ of formulas, define $(\Omega; \mathbf{x}), I$ as $(\ldots((\Omega; \mathbf{x}), a_1), \ldots), a_k$. This does not depend on the ordering in the enumeration of I so it is well-defined.*

In words, the addition of the formula a to $(\Omega; \mathbf{x})$ leaves the latter unchanged if a is already an ω-formula (i.e., $a \in \Omega$), else its finite multiplicity is incremented (i.e., $\mathbf{x}$ becomes $\mathbf{x}, a$).

4.2 The calculus $\mathbf{HFL}^{\rightleftharpoons}_{\mathbf{ew}}(\mathcal{R})_\omega$

Our first calculus dealing with ω-formulas is denoted by $\mathbf{HFL}^{\rightleftharpoons}_{\mathbf{ew}}(\mathcal{R})_\omega$, and its rule instances are presented in Fig. 2.

In order to understand this calculus fully, we need to present the notion of *ω-structural rules*. Every $\mathsf{r} \in \mathcal{R}$ has the form in Def 2.2, so every principal component corresponds to an index in $I \cup J$, and every active component corresponds either to an index (i, k) with $i \in I$

Initial hypersequents

$$\frac{}{H \mid (\Omega; p) \Rightarrow p} \qquad \frac{}{H \mid (\Omega \cup \{p\}; \varnothing) \Rightarrow p} \qquad \frac{}{H \mid (\Omega; 0) \Rightarrow}$$

$$\frac{}{H \mid (\Omega \cup \{0\}; \varnothing) \Rightarrow} \qquad \frac{}{H \mid (\Omega; \varnothing) \Rightarrow 1}$$

Structural rules

$$\frac{h_c \mid (\Omega; \Gamma) \Rightarrow \Pi}{H \mid (\Omega; \Gamma, \varphi) \Rightarrow \Pi}\,\mathrm{w} \qquad \frac{H \mid (\Omega; \Gamma) \Rightarrow \varphi \mid (\Omega; \Gamma) \Rightarrow \Pi}{H \mid (\Omega; \Gamma) \Rightarrow \Pi}\,(\mathrm{EC}) \qquad \frac{H}{H \mid (\Omega; \Gamma) \Rightarrow \Pi}\,(\mathrm{EW})$$

All rules schemas in $\mathcal{R}_\omega$ (see Def. 4.4).

Logical rules operating on ω-coordinates

$$\frac{h_c \mid (\Omega; \Gamma) \Rightarrow \Pi}{H \mid (\Omega \cup \{1\}; \Gamma) \Rightarrow \Pi}\,(1\mathrm{L}) \qquad \frac{h_c \mid (\Omega \cup \{\varphi \cdot \psi\}; \Gamma), \varphi, \psi \Rightarrow \Pi}{H \mid (\Omega \cup \{\varphi \cdot \psi\}; \Gamma) \Rightarrow \Pi}\,(\cdot\mathrm{L})$$

$$\frac{h_c \mid (\Omega \cup \{\varphi_1 \wedge \varphi_2\}; \Gamma), \varphi_i \Rightarrow \Pi}{H \mid (\Omega \cup \{\varphi_1 \wedge \varphi_2\}; \Gamma) \Rightarrow \Pi}\,(\wedge\mathrm{L})$$

$$\frac{h_c \mid (\Omega \cup \{\varphi \vee \psi\}; \Gamma), \varphi \Rightarrow \Pi \qquad h_c \mid (\Omega \cup \{\varphi \vee \psi\}; \Gamma), \psi \Rightarrow \Pi}{H \mid (\Omega \cup \{\varphi \vee \psi\}; \Gamma) \Rightarrow \Pi}\,(\vee\mathrm{L})$$

$$\frac{h_c \mid (\Omega \cup \{\varphi \to \psi\}; \Gamma) \Rightarrow \varphi \qquad h_c \mid (\Omega \cup \{\varphi \to \psi\}; \Delta), \psi \Rightarrow \Pi}{H \mid (\Omega \cup \{\varphi \to \psi\}; \Delta, \Gamma) \Rightarrow \Pi}\,(\to\mathrm{L})$$

Logical rules operating on finite multiplicities

$$\frac{h_c \mid (\Omega; \Gamma) \Rightarrow \Pi}{H \mid (\Omega; \Gamma, 1) \Rightarrow \Pi}\,(1\mathrm{L})_2 \qquad \frac{h_c \mid (\Omega; \Gamma), \varphi, \psi \Rightarrow \Pi}{H \mid (\Omega; \Gamma, \varphi \cdot \psi) \Rightarrow \Pi}\,(\cdot\mathrm{L})_2 \qquad \frac{h_c \mid (\Omega; \Gamma), \varphi_i \Rightarrow \Pi}{H \mid (\Omega; \Gamma, \varphi_1 \wedge \varphi_2) \Rightarrow \Pi}\,(\wedge\mathrm{L})_2$$

$$\frac{h_c \mid (\Omega; \Gamma), \varphi \Rightarrow \Pi \qquad h_c \mid (\Omega; \Gamma), \psi \Rightarrow \Pi}{H \mid (\Omega; \Gamma, \varphi \vee \psi) \Rightarrow \Pi}\,(\vee\mathrm{L})_2 \qquad \frac{h_c \mid (\Omega; \Gamma) \Rightarrow \varphi \qquad h_c \mid (\Omega; \Delta), \psi \Rightarrow \Pi}{H \mid (\Omega; \Gamma, \Delta, \varphi \to \psi) \Rightarrow \Pi}\,(\to\mathrm{L})_2$$

Logical rules (succedent)

$$\frac{h_c \mid (\Omega, \Gamma) \Rightarrow}{H \mid (\Omega, \Gamma) \Rightarrow 0}\,(0\mathrm{R}) \qquad \frac{h_c \mid (\Omega; \Gamma) \Rightarrow \varphi \qquad h_c \mid (\Omega; \Delta) \Rightarrow \psi}{H \mid (\Omega; \Gamma, \Delta) \Rightarrow \varphi \cdot \psi}\,(\cdot\mathrm{R}) \qquad \frac{h_c \mid (\Omega; \Gamma) \Rightarrow \varphi_i}{H \mid (\Omega; \Gamma) \Rightarrow \varphi_1 \vee \varphi_2}\,(\vee\mathrm{R})$$

$$\frac{h_c \mid (\Omega; \Gamma) \Rightarrow \varphi \qquad h_c \mid (\Omega; \Gamma) \Rightarrow \psi}{H \mid (\Omega; \Gamma) \Rightarrow \varphi \wedge \psi}\,(\wedge\mathrm{R}) \qquad \frac{h_c \mid (\Omega; \Gamma), \varphi \Rightarrow \psi}{H \mid (\Omega; \Gamma) \Rightarrow \varphi \to \psi}\,(\to\mathrm{R})$$

Figure 2 The calculus $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})_\omega$. Here, h_c denotes the meta-hypersequent in the conclusion.

and $k \in K_i$, or to an index $l \in L$. For $i \in I$ and $k \in K_i$, let $C_{i,k}$ be the set of those principal component indices $v \in I \cup J$ that have a multiset metavariable in common with the active component (i, k). Define C_l for each active component $l \in L$ analogously. Finally, define $\Omega[i, k]$ as $\bigcup_{v \in C_{i,k}} \Omega_v$ for each (i, k), and $\Omega[l]$ as $\bigcup_{v \in C_l} \Omega_v$ for each $l \in L$.

▶ **Definition 4.4** (ω-structural rules, $\mathcal{R}_\omega$). *If* r *is an analytic structural hypersequent rule schema, then* r_ω *is the* ω*-version below, where* h_c *is the conclusion meta-hypersequent.*

$$\frac{\{h_c \mid (\Omega[i, k]; \varnothing), Y_i, \vec{\Gamma}_{ik} \Rightarrow \Pi_i\}_{i \in I, k \in K_i} \qquad \{h_c \mid (\Omega[l]; \varnothing), \vec{\Delta}_l \Rightarrow\}_{l \in L}}{H \mid (\Omega_i; Y_i, X_{i1}, \ldots, X_{i\alpha_i}) \Rightarrow \Pi_i\ (i \in I) \mid (\Omega_j; Z_{j1}, \ldots, Z_{j\beta_j}) \Rightarrow\ (j \in J)}\,\mathsf{r}_\omega$$

We remind the reader that an ω-hypersequent is a multiset of ω-sequents of the form $(\Omega; \mathbf{x}) \Rightarrow b$. The proof calculus $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})_\omega$ in Fig 2 operates on (ω, d)-hypersequents. For example, in addition to the usual logical rules – in the present setting, these operate on those formulas with finite multiplicities (described by the vector $\mathbf{x}$) – there are also logical rules that operate on the ω-coordinates Ω. As a concrete example, consider the rule schema

$$\frac{h_c \,|\, (\Omega \cup \{\varphi \to \psi\}; \Gamma) \Rightarrow \varphi \qquad h_c \,|\, (\Omega \cup \{\varphi \to \psi\}; \Delta), \psi \Rightarrow \Pi}{H \,|\, (\Omega \cup \{\varphi \to \psi\}; \Delta, \Gamma) \Rightarrow \Pi}\ (\to\mathrm{L})$$

Observe that this rule schema operates on a component that has $\varphi \to \psi$ among its ω-coordinates; we also refer to this by saying that $\varphi \to \psi$ is an ω-formula in that component. In this rule schema, the ω-coordinates of the active components (in the premise) are the same as the ω-coordinates in the principal component (in the conclusion). Consider the right premise of the above. Since we do not know whether $\psi \in \Omega$, we use Def 4.3 to ensure that the formula is classified correctly; this also ensures that we maintain the property that, in any component, the ω-coordinates are disjoint from the formulas with positive finite multiplicity. In general, ω-coordinates increase from principal to active components, e.g., as seen by the $(\Omega[i,k]; \varnothing), Y_i, \vec{\Gamma}_{ik}$ term in Def 4.4.

There is an obvious bijective correspondence between a sequent $\mathbf{x} \Rightarrow b$ and the ω-sequent $(\varnothing; \mathbf{x}) \Rightarrow b$. This bijection extends from rule instances in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})$ to rule instances in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})_\omega$ that operate only on the finite coordinates of the ω-hypersequent. Since the latter calculus does not contain any rule to strictly increase the ω-coordinates from the empty set (we will subsequently introduce such a mechanism), the following is immediate.

▶ **Lemma 4.5.** *Let h be an ω-free ω-hypersequent. Then, h is provable in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})$ iff it is provable in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})_\omega$.*

4.3 Refined ω-calculus

Next, we modify $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})_\omega$ to equip it with height-preserving admissibility of (EC) and (EW), as in Sec 3.1, based on [29].

▶ **Definition 4.6** (Refined ω-calculus). *$\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})_\omega^{\rightsquigarrow}$ consists of every rule $\mathsf{r}^{\rightsquigarrow}$ below left s.t. r (below right) is a rule in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})_\omega$, and $g \rightsquigarrow^{\mathsf{acn}}_{(\mathrm{EC})} h$. Call r the* base instance *of $\mathsf{r}^{\rightsquigarrow}$.*

$$\frac{h \,|\, s_1 \qquad \cdots \qquad h \,|\, s_n}{h}\ \mathsf{r}^{\rightsquigarrow} \qquad\qquad \frac{g \,|\, s_1 \qquad \cdots \qquad g \,|\, s_n}{g}\ \mathsf{r}$$

Every component in h that is related, via applications of (EC)*, to the principal components in g is called a principal component of $\mathsf{r}^{\rightsquigarrow}$. The* schema *for an active/principal component in $\mathsf{r}^{\rightsquigarrow}$ is the corresponding schematic component in r.*

Informally, a rule instance in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})_\omega^{\rightsquigarrow}$ (seen from conclusion to premises) is obtained by applying a limited amount – up to acn, whose value depends solely on the proof rules – of (EC) backwards, and then a backward application of a rule instance in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})_\omega$, replacing g with h in the premises of the base instance.

▶ **Lemma 4.7.** *The calculi $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})_\omega$ and $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})_\omega^{\rightsquigarrow}$ prove the same ω-hypersequents.*

Similar to [29], we have that the external structural rules are hp-admissible in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})_\omega^{\rightsquigarrow}$.

▶ **Lemma 4.8.** (EW) *and* (EC) *are hp-admissible in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})_\omega^{\rightsquigarrow}$.*

▶ **Lemma 4.9.** *If h is provable in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftharpoons}(\mathcal{R})_\omega^{\rightsquigarrow}$, then it is provable without* (EW) *and* (EC)*.*

Henceforth, we may assume that proofs do not use the external structural rules.

The rules operating on ω-coordinates behave just like their familiar counterparts operating on the finite multiplicities (except ω-coordinates do not diminish with use). Hence, treating a finite multiplicity formula as belonging to the ω-set is proof-preserving:

▶ **Definition 4.10** (ω-extensions). *Given an (ω, d)-sequent $s = (\Omega, \mathbf{x}) \Rightarrow b$, we say that an (ω, d)-sequent $s' = (\Omega', \mathbf{x}') \Rightarrow b$ is an* ω-extension *of s if $\Omega \subseteq \Omega'$ and $\mathbf{x}' = \mathbf{x}[\Omega' \backslash \Omega \mapsto 0]$. An* ω-extension of an (ω, d)-hypersequent *h is obtained by replacing some components in h by some of their ω-extensions.*

▶ **Lemma 4.11.** *Let h be an ω-hypersequent provable in $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})^{\rightsquigarrow}_{\omega}$ with height α. Then, any of its ω-extensions is provable in $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})^{\rightsquigarrow}_{\omega}$ with height $\leq \alpha$.*

Proof. Let h be an ω-hypersequent having a proof π of height α in $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})^{\rightsquigarrow}_{\omega}$. It is enough to prove for the case when h' is an ω-extension of h differing from h at a single component $(\Omega', \mathbf{x}') \Rightarrow b$, and, within that component, at a single formula a. That is, $h = g \mid (\Omega, \mathbf{x}) \Rightarrow b$ with $a \notin \Omega$ and $h' = g \mid (\Omega \cup \{a\}, \mathbf{x}[a \mapsto 0]) \Rightarrow b$. The general result follows by iterating this case. We will now show by structural induction on π that h' has a proof π' of height $\leq \alpha$.

(Base case) Here, h is an initial ω-hypersequent. If the component that changed in h' is part of the instantiation of the hypersequent variable H, simply change the instantiation of H accordingly. Otherwise, it must be of one of the forms $(\Omega; p) \Rightarrow p$ or $(\Omega; 0) \Rightarrow$. In that case, use instead the initial hypersequents with principal components $(\Omega \cup \{p\}; \mathbf{0}) \Rightarrow p$ or $(\Omega \cup \{0\}; \mathbf{0}) \Rightarrow$, respectively.

(Inductive step) Assume the last rule instance applied in π has the form $(h \mid t_i)_i / h$, where

$(g|s_1|\cdots|s_m|t_1), \ldots, (g|s_1|\cdots|s_m|t_k)/(g|s_1|\cdots|s_m)$

is its base rule instace in $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})_{\omega}$ and $(g|s_1|\cdots|s_m) \rightsquigarrow^{\mathsf{acn}}_{(\mathrm{EC})} h$. Here, g is the instantiation of the hypersequent variable H.

By IH, we have $h'|t_i$ provable with height $\leq \alpha_i < \alpha$ for each i. We show that the instance $(h'|t_i)_i/h'$ is in the calculus. By cases:

1. The component that changed from h to h' is in g: use the IH and instantiate the same rule schema of $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})_{\omega}$ but changing g to g' in the instantiation of the hypersequent variable H to obtain h'. That is,

   $$(g'|s_1|\cdots|s_m|t_1), \ldots, (g'|s_1|\cdots|s_m|t_k)/(g'|s_1|\cdots|s_m)$$

   is a rule instance of $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})_{\omega}$. So, $g'|s_1|\cdots|s_m \rightsquigarrow^{\mathsf{acn}}_{(\mathrm{EC})} h'$.
2. Assume that the component that changed is s_i for some $1 \leq i \leq m$, and s_i is not present in g (otherwise the previous item would have applied). If the last rule applied operates on the first component or on succedents, note that $i = 1$ necessarily (there is only one principal component), so simply consider the component s'_1 that now has the formula a in the first component, and adapt the rule instance accordingly. In the case of rules operating on the first component, we might get in the new instantiation premises that are ω-extensions of the previous premises, so just apply the IH to obtain proofs of height $\leq \alpha_i < \alpha$ for them.

   If it is a logical rule operating on the second component or $(\mathbf{w})$, again we have $i = 1$, and now we need to consider two cases. First, the formula that changed in the component is not principal. In this case, simply use the IH and adapt the instantiation. If it is the principal formula of the rule, use the IH and then appropriately instantiate the corresponding logical rule that operates on the first component.

   If it is a structural rule from $\mathcal{R}_{\omega}$, the formula a must be in an instantiation of a multiset meta-variable. By the property of linear conclusion of these rules, this variable only occurs in the schema of s_i. By definition of r_{ω}, all premises sharing same variable get affected, and their corresponding ω-sets are changed accordingly. Thus, it is enough to change the instantiation of that variable to not contain a, and place a in Ω_i. Thus use the IH and this new instantiation.

The lemma then follows by iterating the above argument. ◀

5 The ω-proof search algorithm for extensions of $\mathbf{FL_{ew}}$

Now we build on the refined ω-calculus developed in the previous section to describe a ω-proof search procedure.

▶ **Definition 5.1.** *Define $\prec$ on (ω, d)-sequents as $s \prec t$ iff $\Omega \subseteq \Omega'$, $\mathbf{x} < \mathbf{y}$ and $b_1 = b_2$, for any $s = (\Omega, \mathbf{x}) \Rightarrow b_1$ and $t = (\Omega', \mathbf{y}) \Rightarrow b_2$.*

The following function identifies strict increments in the finite multiplicities, and includes those formulas as ω-coordinates.

▶ **Definition 5.2** (ω-function)**.** *Let $s = (\Omega, \mathbf{x}) \Rightarrow b$ and $t = (\Omega', \mathbf{y}) \Rightarrow b$ be ω-sequents such that $s \prec t$. By $\omega(s,t)$ we denote the ω-sequent $(\Omega'', \mathbf{y}') \Rightarrow b$, where $\Omega'' = \Omega' \cup \kappa$ and $\mathbf{y}' = \mathbf{y}[\kappa \mapsto 0]$, with $\kappa = \{l \in \mathbb{N}^+_{\leq d} \backslash \Omega' \mid \mathbf{x}[l] < \mathbf{y}[l]\}$.*

▶ **Example 5.3.** *Let $s_1 = (1, 3; (0, 2, 0, 4)) \Rightarrow b$ and $s_2 = (1, 3, 4; (0, 3, 0, 0)) \Rightarrow b$. Then $\omega(s_1, s_2) = (1, 2, 3, 4; (0, 0, 0, 0)) \Rightarrow b$.*

Proof search will need to track more than just the hypersequents that appear. So we first enrich each component s of the hypersequent as (i, s) where $i \in \mathbb{N}$ serves as a unique index recording the order in which the components were created.

Let $d \geq 1$. An *indexed d-component* is a pair $\underline{s} = (i, s)$, where $i \in \mathbb{N}$ and s is an (ω, d)-sequent. We denote by $\mathsf{c}(\underline{s})$ the (ω, d)-sequent of $\underline{s}$. An *indexed d-hypersequent* is a multiset of indexed d-components, such that any two distinct indexed components have distinct indices (we could use a set here, but multisets are familiar for hypersequents). We omit d from the notation when clear from context. The *support* of an indexed hypersequent is the hypersequent obtained by deleting the index of every component. If the support has no repeated components, $\underline{h}$ is said to be *irredundant*, otherwise it is said to be *redundant*.

▶ **Definition 5.4** (Relational indexed hypersequent)**.** *For $d \geq 1$, a d-*rin-hypersequent *is a tuple $\mathbf{h} = \langle \underline{h}, R_{\mathbf{h}} \rangle$, where $\underline{h}$ is an indexed d-hypersequent, and $R_{\mathbf{h}}$ is a binary relation on the indexed components in $\underline{h}$ such that $R^*_{\mathbf{h}}$ is a rooted tree. Let $\max \underline{h}$ denote the largest index in $\underline{h}$. We say that $\mathbf{h}$ is* irredundant *when $\underline{h}$ is so, otherwise we say it is* redundant.

Let $\mathsf{r} = (h \,|\, s_i)_i / h$ be a rule of $\mathbf{HFL}^{\rightleftharpoons}_{\mathbf{ew}}(\mathcal{R})^{\leadsto}_{\omega}$ and $\mathbf{h} = \langle \underline{h}, R_{\mathbf{h}} \rangle$ be an irredundant rin-hypersequent with support h. We refer to the structure $\mathfrak{I} = \langle \mathsf{r}, \mathbf{h} \rangle$ as an *instantiation context*. Let C be the multiset of principal components of h in r. We say that an indexed component in $\underline{h}$ is principal in r if its ω-sequent is in C. The *schema* of such component in r is defined as the schema of the corresponding principal component in C. Define the *key ancestor of s_i in $\mathfrak{I}$* as the principal component of $\underline{h}$ in r of largest index among those whose schema shares a variable with the schema of s_i.

If $\underline{t}$ is the key ancestor of s_i in $\mathfrak{I}$, define $\underline{s_i} = (\max \underline{h} + 1, s_i)$ and $R_i = R_{\mathbf{h}} \cup \{(\underline{t}, \underline{s_i})\}$. The *$\omega$-partner of $\underline{s_i}$ in $\mathfrak{I}$* (if it exists) is the indexed component $\underline{t}'$ in $\underline{h}$ of largest index such that $(\underline{t}', \underline{s_i}) \in R^*_i$ and $\mathsf{c}(\underline{t}') \prec s_i$. Then, for $s'_i = \omega(t', s_i)$, the indexed ω-sequent $\underline{s}'_i = (\max \underline{h} + 1, s'_i)$ is called the *ω-refinement of $\underline{s_i}$ in $\mathfrak{I}$*, and $\underline{s_i}$ is called the *pre-component of $\underline{s}'_i$ in $\mathfrak{I}$*.

▶ **Definition 5.5.** *A* legal rin-hypersequent *(or lrin-hypersequent, for short) is an irredundant rin-hypersequent $\mathbf{h} = \langle \underline{h}, R_{\mathbf{h}} \rangle$ such that if $\underline{s} R_{\mathbf{h}} \underline{t}$, then there is a rule r such that, for $\mathfrak{I} = \langle \mathsf{r}, \mathbf{h} \rangle$, (a): $\mathsf{c}(\underline{t})$ is one of the active components of r and $\underline{s}$ is the key ancestor of $\mathsf{c}(\underline{t})$ in $\mathfrak{I}$, and (b): $\underline{t}$ either has no ω-partner in $\mathfrak{I}$, or is an ω-refinement in $\mathfrak{I}$.*

▶ **Definition 5.6** (ω-proof-search tree). *For an lrin-hypersequent $\mathbf{h}_0$, the ω-proof-search tree $T_\omega(\mathbf{h}_0)$ is defined as the limit of the trees recursively constructed in the following way.*

1. $T_0(\mathbf{h}_0)$*: A single node labelled with* $\mathbf{h}_0$.
2. $T_{n+1}(\mathbf{h}_0)$*: Pick a leaf ℓ of $T_n(\mathbf{h}_0)$ and assume it is labeled by $\mathbf{h} = \langle \underline{h}, R_\mathbf{h} \rangle$. For each rule $\mathsf{r} = (h|s_i)_{i \in I}/h$ of $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftarrows}(\mathcal{R})_\omega^{\rightsquigarrow}$ resulting from an instantiation with subformulas of the support of $\mathbf{h}_0$, where h is the support of $\underline{h}$, we do the following.*
   a. *(pre-refinement redundancy) If there is $i \in I$ and $\underline{t}$ in $\underline{h}$ s.t. $s_i = \mathsf{c}(\underline{t})$, then go to the next iteration (i.e., process the next rule), else proceed:*

   Let $\mathfrak{I} := \langle \mathsf{r}, \mathbf{h} \rangle$, and, for each $i \in I$, set $m := \max \underline{h} + 1$, $\underline{s}_i := (m, s_i)$ and $R_i := R_\mathbf{h} \cup \{(\underline{t}_i, \underline{s}_i)\}$, where $\underline{t}_i$ is the key ancestor of s_i in $\mathfrak{I}$. Then perform the following.

   b. *(leaf expansion) For each $i \in I$, add $\langle \underline{h}|\underline{s}_i, R_i \rangle$ as a child of ℓ.*
   c. *(ω-refinement) For each new leaf $\langle \underline{h}|(m, s_i), R_i \rangle$, let $\underline{s}'_i$ be the ω-*refinement of $\underline{s}_i$ in $\mathfrak{I}$ *(if it exists), and replace the label of this leaf by $\langle \underline{h}|\underline{s}'_i, R_\mathbf{h} \cup \{(\underline{t}_i, \underline{s}'_i)\} \rangle$.*
   *If this refinement took place for at least one premise, we call the result an ω-*introduction rule*, with $\underline{s}'_i$ called a* productive premise *and r its* pre-rule.
   d. *(post-refinement redundancy) If the rin-hypersequent of any new leaf is redundant, delete all the new leaves introduced in this iteration.*

Observe that the tree resulting from the above construction is guaranteed to have legal rin-hypersequents as labels. Consider an lrin-hypersequent $\langle \underline{h}, R \rangle$ in $T_\omega(\mathbf{h}_0)$. Viewing the indexed hypersequent $\underline{h}$ as a sequence of indexed components, ordered by increasing index, we have access to the order in which the components were obtained. Also, $((i, t), (j, s)) \in R$ tells us t was a principal component of the rule instance that had s as an active component, and the instantiation that led to s is independent of components with index $> i$, in view of the key ancestor condition.

▶ **Definition 5.7.** *Let $\mathbf{h} := \langle \underline{h}, R_\mathbf{h} \rangle$ be an lrin-hypersequent. We define the* tree of $\mathbf{h}$*, denoted by $T_R(\mathbf{h})$, by first making the minimal elements of $R_\mathbf{h}$ the children of a special node labelled with $\bullet$, and then extending it such that $\underline{s}'$ is a child of $\underline{s}$ iff $(\underline{s}, \underline{s}') \in R_\mathbf{h}$.*

▶ **Remark 5.8.** *Let $\mathbf{h} := \langle \underline{h}, R_\mathbf{h} \rangle$ be an lrin-hypersequent occurring in $T_\omega(\mathbf{h}_0)$. First, note that $T_\omega(\mathbf{h}_0)$ is a tree labeled by lrin-hypersequents, while $T_R(\mathbf{h})$ is a tree labeled by components, plus a special symbol $\bullet$, which labels the root. Also, the children of $\bullet$ are elements of $\underline{h}_0$, so the tree is finitely branching at the root (since $\underline{h}_0$ is finite).*

We now define a more restrictive wqo than $\prec$ (Def 5.1) where indexed (ω, d)-sequents with distinct ω-sets are incomparable. Recall that $\prec$ checks if one sequent is larger than another (and this means comparing also sequents with different ω-sets); this ensures that we do not add unnecessary premises to the proof-search tree, a condition that is needed for termination. On the other hand, the more restrictive wqo is required to measure the length of proof-search branches by controlling sequences with fixed ω-sets, using the length theorem for $\mathbb{N}^d$, where d is fixed.

▶ **Definition 5.9.** *Given $\underline{s} = (i, (\Omega; \mathbf{x}) \Rightarrow b_1)$ and $\underline{s'} = (j, (\Omega'; \mathbf{y}) \Rightarrow b_2)$, let $\underline{s} \preceq_{\mathbf{W}(\omega,d)} \underline{s'}$ iff $\Omega = \Omega'$, $b_1 = b_2$ and $\mathbf{x} \leq_{\mathbb{N}^d} \mathbf{y}$. Define the structure $\mathbf{W}(\omega, d)$ on indexed (ω, d)-sequents having $\preceq_{\mathbf{W}(\omega,d)}$ as relation and $[\![(l, \underline{s})]\!] := [\![\mathsf{c}(\underline{s})]\!]$ as norm, where the norm of an ω-sequent $(\Omega; \mathbf{x}) \Rightarrow b$ is defined as $[\![\mathbf{x}]\!]_{\mathbb{N}^d}$.*

▶ **Lemma 5.10.** *$\mathbf{W}(\omega, d)$ is an nwqo and, for any primitive recursive control function f, its length function $L_{\mathbf{W}(\omega,d),f}$ is upper bounded in $\mathbf{F}^\star_\omega$, uniformly on d.*

Proof. In order to see that $\mathbf{W}(\omega, d)$ is a wqo, let $\mathbf{Q} = (2^d \cdot (d+1)) \cdot \mathbf{N}^d$. Now, uniquely assign a number $I(\Omega, b)$ in $\mathbb{N}_{\leq 2^d \cdot (d+1)}$ to the pairs (Ω, b), where Ω is a subset of $\mathbb{N}^+_{\leq d}$ and $b \in \mathbb{N}_{\leq d}$. We then code the indexed (ω, d)-sequent $\underline{s} = (i, (\Omega, \mathbf{x}) \Rightarrow b)$ as the element $\underline{s}^\# := (I(\Omega, b), \mathbf{x})$ of $\mathbf{Q}$ (note that the index is ignored). Then clearly $\underline{t} \preceq_{\mathbf{W}(\omega,d)} \underline{s}$ iff $\underline{t}^\# \leq_{\mathbf{Q}} \underline{s}^\#$. Moreover, $[\![\underline{t}^\#]\!] = [\![\underline{t}]\!]$. It then follows that any (f, n)-controlled bad sequence $\underline{t}_0, \underline{t}_1, \ldots,$ over sequents can be translated to a (f, n)-controlled bad sequence $\underline{t}_0^\#, \underline{t}_1^\#, \ldots$ Hence, the length functions of $\mathbf{W}(\omega, d)$ are upper bounded by the length functions of $\mathbf{Q}$. We conclude using Thm 2.9. ◀

▶ **Lemma 5.11.** *Let $\mathbf{h}_0 := \langle \underline{h}_0, R_{\mathbf{h}_0} \rangle$ and $\mathbf{h} := \langle \underline{h}, R_{\mathbf{h}} \rangle$ be lrin-hypersequents such that $\mathbf{h}_0$ is finite and $\mathbf{h}$ occurs in $T_\omega(\mathbf{h}_0)$. Then:*

1. *Along a branch of $T_R(\mathbf{h})$, indices of components strictly increase and ω-sets are increasing.*
2. *Every branch of $T_R(\mathbf{h})$ is an $(f, [\![h_0]\!])$-controlled bad sequence over $\mathbf{W}(\omega, d)$, where f is the function from Lem 3.9.*
3. *$T_R(\mathbf{h})$ is finitely branching.*

Therefore, $T_R(\mathbf{h})$ (and hence $\mathbf{h}$) is finite. Moreover, there is a function in $\mathbf{F}^\star_\omega$ that uniformly upper bounds the size of $T_R(\mathbf{h})$ as a function of $[\![\mathsf{c}(\underline{h}_0)]\!]$.

Proof. Recall that a branch of $T_R(\mathbf{h})$ is – after $\bullet$ – a chain of indexed components $\underline{t}_1 \, R_{\mathbf{h}} \, \underline{t}_2 \, R_{\mathbf{h}} \ldots$ such that $\underline{t}_{i+1}$ is the active component of a premise of a rule of the calculus having $\underline{t}_i$ as a principal component (in fact, as its key ancestor).

Item 1. Indices of components strictly increase because, by construction, $\underline{t}_i R_{\mathbf{h}} \underline{t}_{i+1}$ implies that the index of the component $\underline{t}_{i+1}$ is strictly larger than the index of any of its ancestors. For the next part, by the way the rules are defined, an active component has a larger ω-set than its key ancestor for any non ω-introduction rule. If an active component was created by an ω-introduction rule, then it has an ω-set that is strictly larger than its parent (and in fact, all its ancestors, by an implicit induction).

Item 2. Let $\mathsf{B} := \underline{t}_1 \, R_{\mathbf{h}} \, \underline{t}_2 \, R_{\mathbf{h}} \ldots$ be a branch of of $T_R(\mathbf{h})$. If it is not a bad sequence, there is a chain $(\Omega_i, \mathbf{x}_i) \Rightarrow b_i = \underline{t}_i \, R_{\mathbf{h}} \, \ldots \, R_{\mathbf{h}} \, \underline{t}_j = (\Omega_j, \mathbf{x}_j) \Rightarrow b_j$ in B for some $i < j$ such that $\underline{t}_i \preceq_{\mathbf{W}(\omega,d)} \underline{t}_j$ i.e., $\Omega_i = \Omega_j$, $b_i = b_j$ and $\mathbf{x}_i < \mathbf{x}_j$ (strict since $\mathbf{h}$ is irredundant). Notice that $\underline{t}_j$ must be the result of ω-refinement since otherwise it would mean that the algorithm missed a suitable ω-partner (i.e., $\underline{t}_i$). Denote the pre-component of $\underline{t}_j$ by $\underline{t}'_j = (\Omega'_j, \mathbf{y}'_j) \Rightarrow b$. Then $\Omega'_j \subset \Omega_j$ since, as observed above, the pre-component has a strictly smaller ω-set than the productive premise that replaces it. Since $j > i$, and since ω-sets are increasing (by ω-increasing property of related components from conclusion to premise), it follows that $\Omega_i \subseteq \Omega'_j$. Since $\Omega_i = \Omega_j$, we get the contradiction $\Omega_i \subseteq \Omega'_j \subset \Omega_i$.

Moreover, since $(\underline{t}_i, \underline{t}_{i+1}) \in R_{\mathbf{h}}$ means that $\underline{t}_{i+1}$ is either a premise or an ω-extension of a premise of a rule with $\underline{t}_i$ among its principal components, we have that $[\![\underline{t}_{i+1}]\!] \leq f([\![\underline{t}_i]\!])$ by Lem 3.9, and thus $[\![\underline{t}_j]\!] \leq f^j([\![\underline{t}_1]\!])$, meaning that the sequence is $(f, [\![h_0]\!])$-controlled, since $[\![\underline{t}_1]\!] \leq [\![h_0]\!]$ because $\underline{t}_1$ is in $\underline{h}_0$.

Item 3. Towards a contradiction, pick a node labelled by $\underline{t}$ having infinitely-many children, i.e., $\underline{t} \, R_{\mathbf{h}} \, \underline{s}_i$ for all $i \geq 0$. So $\underline{t}$ acted as key ancestor for each $\underline{s}_i$, meaning that it was the most recent principal component in the indexed rin-hypersequent $\mathbf{h}_i := \langle \underline{h}_i, R_i \rangle$ being expanded sharing a multiset meta-variable with the schema of $\underline{s}_i$.

Observe that the set A of components with smaller index than $\underline{t}$ in the whole $\underline{h}$ is the same for all such $\mathbf{h}_i$. This means that every $\underline{s}_i$ is determined by A. It follows that $\underline{s}_i = \underline{s}_j$ for some i, j, which contradicts the redundancy check. Observe also that there are only finitely-many children of the first special node $\bullet$, since $\underline{h}_0$ is finite. This establishes Item (3).

So $T_R(\mathbf{h})$ is finitely branching and all of its branches are finite; thus, by König's Lemma, $T_R(\mathbf{h})$ is finite. Now let m be the length of the longest branch in this tree. By the argument above, at any node labelled by $\underline{t}$, there can only be $p([\![\underline{t}]\!]) \leq p(f^m([\![h_0]\!]))$ many children of it for some primitive recursive function p that depends only on the rules of the underlying calculus. Hence, the size of the total tree (which is at most the maximum number of children a node can have exponentiated to the length of the longest branch) is at most $(p(f^m([\![h_0]\!]))^m$. This is a primitive recursive function in m ([10, Definition 9],[31, Section 5.3.1]). Since m is upper bounded by a function in $\mathbf{F}^\star_\omega$ by Item (2) and Lem 5.10, then by Thm 2.8 it follows that the size of the whole tree is bounded by a function in $\mathbf{F}^\star_\omega$. ◀

▶ **Theorem 5.12.** *Let $\mathbf{h}_0$ be a finite lrin-hypersequent. Then $T_\omega(\mathbf{h}_0)$ is finite, and thus the construction of $T_\omega(\mathbf{h}_0)$ terminates. Moreover, $T_\omega(\mathbf{h}_0)$ is built in time $\mathbf{F}^\star_\omega$.*

Proof. Assume that $T_\omega(\mathbf{h}_0)$ is not finite. As branching factor of proof search tree is finite (the calculus is locally finitary), $T_\omega(\mathbf{h}_0)$ must have an infinite branch by König's Lemma, which induces an infinite lrin-hypersequent $\mathbf{h} \coloneqq \langle \underline{h}, R_\mathbf{h} \rangle$, contradicting Lem 5.11, which says that there cannot be infinite lrin-hypersequent in $T_\omega(\mathbf{h}_0)$.

A uniform $\mathbf{F}^\star_\omega$ bound for the length of any branch in $T_\omega(\mathbf{h}_0)$ is given by Lem 5.11. The branching factor is also $\mathbf{F}^\star_\omega$ upper bounded. So the size of this tree is upper bounded in $\mathbf{F}^\star_\omega$. Instantiations, expanding nodes, and the checks in the construction of $T_\omega(\mathbf{h}_0)$ are primitive recursive operations over an object of size $\mathbf{F}^\star_\omega$, so the whole tree is built in this time. ◀

▶ **Definition 5.13** (ω-eager proofs)**.** *A subtree of $T_\omega(\mathbf{h}_0)$ that is a derivation in $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})^{\leadsto}_\omega$, is rooted at $\mathbf{h}_0$, and whose leaves were obtained from instances of axiomatic rules is called an* ω-eager proof of h_0*, where h_0 is the support of $\underline{h}_0$.*

To any irredundant ω-hypersequent h_0 we canonically associate the lrin-hypersequent $\mathbf{h}^\circ_0 = \langle \underline{h}_0, \varnothing \rangle$, where $\underline{h}_0$ is obtained by choosing an order for the elements of h_0 and assigning indices $\{0, \ldots, |h_0| - 1\}$ respecting this order.

▶ **Definition 5.14** (ω-proof search algorithm)**.** *Given a finite irredundant ω-hypersequent h_0 as input, the algorithm* `Wkn`(h_0) *builds the tree $T_\omega(\mathbf{h}^\circ_0)$, then looks for a subtree that is an ω-eager proof of h_0. If found, it outputs* `provable`*, otherwise it outputs* `unprovable`.

By Lem 5.11 and Lem 5.12 we obtain termination and complexity.

▶ **Theorem 5.15** (Complexity)**.** `Wkn` *terminates in $\mathbf{F}^\star_\omega$ time.*

Given a finite irredundant hypersequent h_0, we say that h_0 *has an ω-eager proof* if `Wkn`(h_0) returns `provable`. Next, we prove that this algorithm decides the extensions of $\mathbf{FL}_{\mathbf{ew}}$.

6 Correctness of the ω-proof search algorithm for extensions of $\mathbf{FL}_{\mathbf{ew}}$

▶ **Lemma 6.1** (Completeness)**.** *Let h be an irredundant ω-hypersequent. If h has a proof in $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})^{\leadsto}_\omega$, then it has an ω-eager proof.*

Proof. We prove the stronger statement: if h is provable in $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})^{\leadsto}_\omega$, then $T_\omega(\mathbf{h})$ has an ω-eager proof of h for any lrin-hypersequent $\mathbf{h} \coloneqq \langle \underline{h}, R_\mathbf{h} \rangle$, where $\underline{h}$ has support h.

By induction on the height α of a proof π of h in $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})^{\leadsto}_\omega$. Let $\mathbf{h} \coloneqq \langle \underline{h}, R_\mathbf{h} \rangle$ be an arbitrary lrin-hypersequent.

(Base case) This is obvious, as the root of $T_\omega(\mathbf{h})$ itself is an ω-eager proof of h, since h is an instance of an axiom of $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})^{\leadsto}_\omega$.

(Induction step) Assume that the last rule applied in π was $\mathsf{r} = (h|t_i)_i/h$. Thus, each $h_i = (h|t_i)$ has a proof π_i in $\mathbf{HFL}^{\rightleftarrows}_{\mathbf{ew}}(\mathcal{R})^{\leadsto}_\omega$ of height $\alpha_i < \alpha$. By cases:

$$
\dfrac{\dfrac{h_0 \mid (\varnothing; \mathbf{x}_0) \Rightarrow b \mid h_1 \mid (\varnothing; \mathbf{y}[\kappa \mapsto K_1]) \Rightarrow b}{h_0 \mid (\varnothing; \mathbf{x}_0) \Rightarrow b \mid \dots \mid h_1 \mid (\varnothing; \mathbf{x}_{K_1}) \Rightarrow b}\,(\mathrm{EW}),(\mathbf{w}) \qquad \dfrac{\{h_0 \mid (\varnothing; \mathbf{x}_0) \Rightarrow b \mid h_1 \mid t_i\}}{\{h_0 \mid (\varnothing; \mathbf{x}_0) \Rightarrow b \mid \dots \mid h_1 \mid t_i'\}}\,(\mathrm{EW}),(\mathbf{w})}{h_0 \mid (\varnothing; \mathbf{x}_0) \Rightarrow b \mid \dots \mid h_1 \mid (\varnothing; \mathbf{x}_{K_1}) \Rightarrow b \mid h_1}
$$

$$
\dots
$$

$$
\dfrac{h_0 \mid (\varnothing; \mathbf{x}_0) \Rightarrow b \mid h_1 \mid (\varnothing; \mathbf{x}_1) \Rightarrow b \mid h_1 \mid (\varnothing; \mathbf{x}_2) \Rightarrow b \qquad \dfrac{\{h_0 \mid (\varnothing; \mathbf{x}_0) \Rightarrow b \mid h_1 \mid t_i\}}{\{h_0 \mid (\varnothing; \mathbf{x}_0) \Rightarrow b \mid h_1 \mid (\varnothing; \mathbf{x}_1) \Rightarrow b \mid h_1 \mid t_i'\}}\,(\mathrm{EW}),\ (\mathbf{w})}{h_0 \mid (\varnothing; \mathbf{x}_0) \Rightarrow b \mid h_1 \mid (\varnothing; \mathbf{x}_1) \Rightarrow b \mid h_1}
$$

$$
\Pi_1
$$

$$
\dfrac{h_0 \mid (\varnothing; \mathbf{x}_0) \Rightarrow b \mid h_1 \mid (\varnothing; \mathbf{x}_1) \Rightarrow b \qquad \{h_0 \mid (\varnothing; \mathbf{x}_0) \Rightarrow b \mid h_1 \mid t_i\}}{h_0 \mid (\varnothing; \mathbf{x}_0) \Rightarrow b \mid h_1}
$$

$$
\Pi_0
$$

$$
h_0 \mid (\varnothing; \mathbf{x}_0) \Rightarrow b
$$

Figure 3 K_1 iterations of derivation in (2). Set of open leaves (non-axiomatic leaves) is unchanged.

1. t_i occurs in h for some $1 \leq i \leq k$: Say $h = h'|t_i$, thus $h_i = h'|t_i|t_i$. Since h_i has a proof of height $< \alpha$, by hp-admissibility of (EC) we know that h itself has a proof of height $< \alpha$. Because h is irredundant by assumption, by IH, $T_\omega(\mathbf{h})$ has an ω-eager proof of h.

We know that each $(h \mid t_i)$ is irredundant. Thus, by IH, any lrin-hypersequent $\mathbf{h}_i$ based on $\underline{h}_i$ has an ω-eager proof of h_i. Define the instantiation context $\mathfrak{I} \coloneqq \langle \mathsf{r}, \mathbf{h} \rangle$. Other cases:

2. For each $1 \leq i \leq k$, it is not the case that $\underline{t}_i$ has an ω-partner in $\mathfrak{I}$: Then, since t_i does not occur in h, in the first step of the construction of $T_\omega(\underline{h})$, an lrin-hypersequent $\mathbf{h}_i$ will be added as a child of the root of $T_\omega(\mathbf{h})$, and no ω-refinement takes place. Thus, by construction, $T_\omega(\mathbf{h}_i)$ is a subtree of $T_\omega(h)$ for each $1 \leq i \leq k$. Hence, picking the ω-eager proofs π_i' of h_i obtained by IH, obtain an ω-eager proof of h as a rooted subtree of $T_\omega(\underline{h})$.
3. Now, assume wlog that $\underline{t}_1, \dots, \underline{t}_m$ have ω-partners in $\mathfrak{I}$. This means that they have gone through the ω-refinement process when building $T_\omega(\mathbf{h})$, and lrin-hypersequents $\mathbf{h}_i'$ with support $h_i' = (h|t_i')$ were added as children of the root of $T_\omega(\mathbf{h})$ for all $1 \leq i \leq m$, where h_i' is clearly an ω-extension of h_i. Since h_i is provable with height $\alpha_i < \alpha$, due to Lemma 4.11 we have that $h|t_i'$ has a proof with height $\leq \alpha_i < \alpha$. We now have two cases. If some t_i' appears in h, then $h = h'|t_i'$ and $h|t_i' = h'|t_i'|t_i'$. By hp-admissibility of (EC), h has a proof of height $< \alpha$ and thus by IH it has an ω-eager proof in $T_\omega(\mathbf{h})$.
   If no t_i' appears in h, we know that h_i' has an ω-eager proof in $T_\omega(\mathbf{h}_i')$ by IH. Since $T_\omega(\mathbf{h}_i')$ is an immediate subtree of $T_\omega(\mathbf{h})$, and since the IH applies to the other premises $(h|t_j)$, $j \geq m$, and $T_\omega(\mathbf{h}_j)$ are also immediate subtrees of $T_\omega(\mathbf{h})$ since they have not gone through ω-refinement, we are done. ◀

Given an ω-hypersequent $h = (\Omega_0, \mathbf{x}_0) \Rightarrow b_0 \mid \dots \mid (\Omega_l, \mathbf{x}_l) \Rightarrow b_l$ and $K \in \mathbb{N}$, let $h[K]$ denote the ω-free ω-hypersequent $(\varnothing; \mathbf{x}_0[\Omega_0 \mapsto K]) \Rightarrow b_0 \mid \dots \mid (\varnothing; \mathbf{x}_l[\Omega_l \mapsto K]) \Rightarrow b_l$. Informally speaking, every formula in the ω-set has been assigned the multiplicity K.

▶ **Lemma 6.2** (Soundness). *If an ω-free irredundant ω-hypersequent has an ω-eager proof, then h is provable in $\mathbf{HFL}^{\rightleftharpoons}_{\mathbf{ew}}(\mathcal{R})^{\rightsquigarrow}_{\omega}$.*

Proof. Let $h = (\varnothing; \mathbf{x}) \Rightarrow b$ and let $\underline{\pi}$ be an ω-eager proof of h of height α, i.e., a subtree of $T_\omega(\mathbf{h}^\circ)$ rooted at $\mathbf{h}^\circ$. Obtain the tree π by deleting all the indices of all components appearing in $\underline{\pi}$ (i.e., π is a tree of ω-hypersequents). We must transform π into a proof in $\mathbf{HFL}^{\rightleftharpoons}_{\mathbf{ew}}(\mathcal{R})^{\rightsquigarrow}_{\omega}$ (free of ω-coordinates). For that, it suffices to give a proof of h containing no ω-introduction rules; recall that the latter are those introduced as the productive output of ω-refinement in the construction of $T_\omega(\mathbf{h}^\circ)$. Our transformation has two stages.

(Stage one). We transform π by replacing, in every component (in every ω-hypersequent occurring in the proof), each element of its ω-set by a specified finite multiplicity. This multiplicity is obtained by a leaf-to-root induction on π (equivalently, by an induction on the distance of a node in the tree from the leaves).

(Base case). The node must be one of the following, where we denote the logical constants by $\mathbf{f}$ and $\mathbf{t}$ (since 0 and 1 are overloaded):

$$g \mid (\Omega; \mathbf{e}_b) \Rightarrow b \quad g \mid (\Omega \cup \{b\}; \mathbf{0}) \Rightarrow b \quad g \mid (\Omega; \mathbf{e_f}) \Rightarrow$$
$$g \mid (\Omega \cup \{\mathbf{f}\}; \mathbf{0}) \Rightarrow \quad g \mid (\Omega; \mathbf{0}) \Rightarrow \mathbf{t}$$

Map every ω-set element to multiplicity 1 and prove using $(\mathbf{w})$:

$$g[1] \mid (\varnothing; \mathbf{e}_b[\Omega \mapsto 1]) \Rightarrow b \quad g[1] \mid (\varnothing; \mathbf{0}[\Omega \cup \{b\} \mapsto 1]) \Rightarrow b$$
$$g[1] \mid (\varnothing; \mathbf{e_f}[\Omega \mapsto 1]) \Rightarrow \quad g[1] \mid (\varnothing; \mathbf{0}[\Omega \cup \{\mathbf{f}\} \mapsto 1]) \Rightarrow$$
$$g[1] \mid (\varnothing; \mathbf{0}[\Omega \mapsto 1]) \Rightarrow \mathbf{t}$$

(Inductive step). Consider an internal node of the tree with label g, and assume that r was applied backwards in the construction of $T_\omega(\underline{h})$. Let π_i denote the subtree at the i-th premise, say with root labelled by h_i. By inductive construction, each π_i is transformed to π'_i, where the latter has root $h'_i = h_i[K_i]$ for some $K_i \in \mathbb{N}$ (i.e., every ω-set in h_i is replaced by a finite multiplicity K_i). By cases:

1. r is an instance of $\mathbf{HFL}^{\rightleftharpoons}_{\mathbf{ew}}(\mathcal{R})^{\rightsquigarrow}_{\omega}$ with conclusion g: we identify a derivation of $\mathbf{HFL}^{\rightleftharpoons}_{\mathbf{ew}}(\mathcal{R})^{\rightsquigarrow}_{\omega}$ whose every non-axiom leaf is a root of some π'_i, and whose conclusion is $g[K_0]$, for a suitable $K_0 \geq \max_i K_i$. We show an instance r' of the same rule schema with conclusion $g[K_0]$ and premises that can be weakened upward to reach the roots of π'_i. I.e.,
$$\dfrac{\dfrac{h_1[K_1]}{g_1[K_0]}\,(\mathbf{w})^* \quad \cdots \quad \dfrac{h_m[K_m]}{g_m[K_0]}\,(\mathbf{w})^*}{g[K_0]}\,r'$$
At the root of π'_i, for each component, we need K_i copies of every formula that was formerly in its ω-set. For components instantiating the hypersequent-variable in the rule schema of r, this is easy to fulfill; use the same instantiation. Meanwhile, every active component c_a in the rule schema has multiset metavariables $M_1, \ldots, M_{l(a)}$, and the latter occur in some subset $\{p_1, \ldots, p_{n(a)}\}$ of principal components. The ω-set of c_a was defined (Def. 4.4) as the union of the ω-sets of $\{p_1, \ldots, p_{n(a)}\}$. Hence, in the instantiation of each M_i, if we include K (for some $K \in \mathbb{N}$) copies of each formula from the ω-set of the principal component in which it occurs, then every formula that was formerly in the ω-set of c_a now has multiplicity $\geq K$ (excess copies may occur when an active component contains multiple multiset metavariables; these can be removed by weakening upwards to reach K_i). If r is a logical rule, then we need an additional copy in the principal component, to serve (potentially) as principal formula. Recalling fm (Def. 3.2), replace every element in the ω-set of the conclusion of r by $K_0 = (\mathsf{fm} \cdot \max_i K_i) + 1$.
2. r is an ω-introduction rule $\omega(r)$ that has a single productive premise (the argument extends by induction to the general case of multiple productive premises). For spacing reasons, here and elsewhere we may present the multiple premises stacked vertically.
$$\dfrac{\begin{array}{c} h_0 \mid (\Omega; \mathbf{x}) \Rightarrow b \mid h_1 \mid (\Omega \cup \kappa; \mathbf{y}[\kappa \mapsto 0]) \Rightarrow b \\ \{h_0 \mid (\Omega; \mathbf{x}) \Rightarrow b \mid h_1 \mid t_i\}_i \end{array}}{h_0 \mid (\Omega; \mathbf{x}) \Rightarrow b \mid h_1}\,\omega(r)$$
Setting $K_0 = \max_i K_i$, and replacing each π_i with π'_i obtained from the inductive construction, write the following.
$$\dfrac{\begin{array}{c} h_0[K_1] \mid (\varnothing; \mathbf{x}[\Omega \mapsto K_1]) \Rightarrow b \mid h_1[K_1] \mid (\varnothing; \mathbf{y}[\Omega \cup \kappa \mapsto K_1]) \Rightarrow b \\ \{(h_0 \mid (\Omega; \mathbf{x}) \Rightarrow b \mid h_1 \mid t_i)[K_i]\}_i \end{array}}{h_0[K_0] \mid (\varnothing; \mathbf{x}[\Omega \mapsto K_0]) \Rightarrow b \mid h_1[K_0]}\,p\omega(r)$$

Unlike before, we do not obtain a rule instance of $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftarrows}(\mathcal{R})_{\omega}^{\leadsto}$: the coordinates κ have some finite multiplicities in $(\varnothing; \mathbf{x}[\Omega \mapsto K])$ (conclusion), but they have multiplicity K_1 in $\mathbf{y}[\Omega \cup \kappa \mapsto K_1]$ (premise). Call the above a *pseudo-ω rule instance* $p\omega(r)$, and call $\{K_i\}_{i\in I}$ its *associated values*. We will replace these with a derivation in the subsequent stage. For now, proceeding in this way, we ultimately reach a proof π' of $(\varnothing, \mathbf{x}) \Rightarrow b$ using rules of $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftarrows}(\mathcal{R})_{\omega}^{\leadsto}$ + pseudo-ω-rule instances. In particular, π' does not contain ω-introduction rules.

(Stage two). Now, let us take π' obtained above and eliminate pseudo-ω-rule instances, obtaining a proof of h in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftarrows}(\mathcal{R})_{\omega}^{\leadsto}$.

We proceed by induction considering a slightly more general family of proofs. Let P be a proof of an ω-free hypersequent in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftarrows}(\mathcal{R})_{\omega}^{\leadsto}$ + pseudo-ω-rule instances that satisfy the ω-refinement condition, where each ω-hypersequent in P implicitly holds a (key ancestor, active component) relation on its components defined as in the ω-proof search tree in Def 5.6.

Associate a branch in P with the number of pseudo-ω-rule instance conclusions on it. Define *maximum nesting* $\rho(P)$ as the multiset of such numbers from all branches (i.e., every branch of P contributes a single number). Induction on usual multiset ordering on ρ (replacing a number by finitely many strictly smaller numbers yields a strictly smaller multiset).

(Base case). If ρ is empty, then P is already a proof in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftarrows}(\mathcal{R})_{\omega}^{\leadsto}$.

(Inductive step). Suppose that $\rho(P)$ is non-empty and consider a lowermost instance of a pseudo-ω-rule instance in P. Such a rule instance appears as in the picture below. Here, K_i is the i-th associated value of the pseudo-ω-rule instance. Note that the ω-set of every component in every ω-hypersequent along the path from the root of P to the conclusion of the chosen $p\omega(r)$ rule is empty, by choice of lowermost $p\omega(r)$ and the ω-freeness assumption.

$$\begin{array}{c} h_0 \mid (\varnothing; \mathbf{x}) \Rightarrow b \mid h_1 \mid (\varnothing; \mathbf{y}[\kappa \mapsto K_1]) \Rightarrow b \\ \dfrac{\{h_0 \mid (\varnothing; \mathbf{x}) \Rightarrow b \mid h_1 \mid t_i[\kappa \mapsto K_i]\}_i}{h_0 \mid (\varnothing; \mathbf{x}) \Rightarrow b \mid h_1}\ p\omega(r) \\ \cdots \\ h_0 \mid (\varnothing; \mathbf{x}) \Rightarrow b \end{array} \tag{2}$$

Reading the above as a derivation where the only open leaves are those shown in the picture, replace its occurrence in P by the derivation in Fig 3. Here, $\mathbf{x}_0, \ldots, \mathbf{x}_{K_1}$ is the sequence defined by $\mathbf{x}_0 := \mathbf{x}$ and $\mathbf{x}_{i+1} = \mathbf{x} + (i+1)(\mathbf{y} - \mathbf{x})$ (recall that $\mathbf{x} < \mathbf{y}$ by the condition in ω-refinement). Call this new tree P^*.

First, let us explain why this derivation has the same set of open leaves as before. Consider the first premise of $p\omega(r)$. In the (key ancestor, active component) dependency graph on components, we have that the featured $(\varnothing; \mathbf{x}_0) \Rightarrow b$ is an ancestor of the active component $(\varnothing; \mathbf{x}_1) \Rightarrow b$, by the ω-refinement condition. Moreover, looking at the rule schemas underlying the rule instances from the introduction of $(\varnothing; \mathbf{x}_0) \Rightarrow b$ to $(\varnothing; \mathbf{x}_1) \Rightarrow b$, there is a common multiset metavariable between each (key ancestor, active component) pair – ensuring this property was a major motivation in the definition of key ancestor. Now, Π_i $(i > 1)$ is like Π_0 except that the excess $\mathbf{x}_i - \mathbf{x}_0$ is included as part of the instantiation of these multiset metavariables. This excess appears in every active component of a rule instance within Π that contains such a multiset metavariable – i.e., the unproductive premises of the ω-introduction rule, or the premise of a branching rule inside a Π_i that leads towards an axiom – but all unwanted copies can be deleted by weakening upwards. Hence Π_i is indeed a derivation.

Next, we claim that the nesting depth ρ^* of P^* is strictly less than ρ in the multiset ordering. Any branch B in P either passes our chosen $p\omega(r)$, or it does not. Let ρ_1 be the multiset of nesting depths from branches of the former type, and ρ_2 from the latter. Thus,

$\rho = \rho_1 \uplus \rho_2$, where $\uplus$ is multiset union. Also, any branch in P^* either has the form σB where σ is the path from the root to an open leaf wrt the deduction (B is the remainder of the path), or it does not pass any open leaf wrt the deduction (this means that it branches inside some Π_i and heads towards an axiom). Then, $\rho^* = \rho_1^* \uplus \rho_2$, where ρ_1^* is the multiset of nesting depths from branches of the former type, and ρ_2 is from the latter. Since σ does not contain any pseudo-ω-rule instance, $\rho_1 = 1 + \rho_1^*$ (where $+$ is the obvious pointwise addition operation). Hence, the claim is established.

Inductive case: it suffices to apply IH to P^*. Applying this to π' yields desired proof. ◀

▶ **Theorem 6.3** (Correctness of `Wkn`). *For any irredundant hypersequent h_0, `Wkn`(h_0) outputs* `provable` *iff h_0 is provable in $\mathbf{HFL}_{\mathbf{ew}}^{\rightleftarrows}(\mathcal{R})_{\omega}^{\leadsto}$.*

From Lem 2.7, Lem 4.5, Lem 4.7, Thm 6.3 and Thm 5.15:

▶ **Theorem 6.4** (Main Theorem II). *Provability in any extension of $\mathbf{FL_{ew}}$ by finitely many $\mathcal{P}_3$-axioms (i.e, provability in any analytic structural extension of $\mathbf{HFL_{ew}}$) is Ackermannian.*

▶ **Corollary 6.5.** **MTL** *has Ackermann complexity.*

7 Final remarks

Our results improved the upper bound for the complexity of **MTL** from hyper-Ackermann to Ackermann. Nevertheless, a large gap remains, since the best lower bound is currently coNP-hardness; see [15]. In our proof search for extensions of $\mathbf{FL_{ew}}$, branch extension depends on earlier nodes on that branch. From an automata-theoretic perspective, this is a history-sensitive extension of VAS-type models quite different from those in [21, 22].

References

1 A. Avron. A constructive analysis of RM. *J. of Symbolic Logic*, 52(4):939–951, 1987. `doi:10.2307/2273828`.

2 A. R. Balasubramanian. Complexity of controlled bad sequences over finite sets of $\mathbb{N}^d$. In *ACM International Conference Proceeding Series*, pages 130–140. Association for Computing Machinery, July 2020. `doi:10.1145/3373718.3394753`.

3 A. R. Balasubramanian, Vitor Greati, and Revantha Ramanayake. Hypersequent calculi have Ackermannian complexity, 2026. `doi:10.48550/arXiv.2602.19229`.

4 A. R. Balasubramanian, Timo Lang, and Revantha Ramanayake. Decidability and complexity in weakening and contraction hypersequent substructural logics. In *36th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2021, Rome, Italy, June 29 - July 2, 2021*, pages 1–13. IEEE, 2021. `doi:10.1109/LICS52264.2021.9470733`.

5 P. Baldi, A. Ciabattoni, and L. Spendier. Standard completeness for extensions of MTL: an automated approach. In *WOLLIC 2012*, volume 7456 of *LNCS*, pages 154–167. Springer, 2012. `doi:10.1007/978-3-642-32621-9_12`.

6 Sébastien Bardin, Alain Finkel, Jérôme Leroux, and Laure Petrucci. FAST: acceleration from theory to practice. *Int. J. Softw. Tools Technol. Transf.*, 10(5):401–424, 2008. `doi:10.1007/S10009-008-0064-3`.

7 A. Ciabattoni, N. Galatos, and K. Terui. From axioms to analytic rules in nonclassical logics. In *LICS 2008*, pages 229–240, 2008. `doi:10.1109/LICS.2008.39`.

8 Agata Ciabattoni, Francesc Esteva, and Llus Godo. T-norm based logics with n-contraction. *Neural Network World*, 12:441–452, January 2002.

9 Agata Ciabattoni, Nikolaos Galatos, and Kazushige Terui. Algebraic proof theory: Hypersequents and hypercompletions. *Ann. Pure Appl. Logic*, 168(3):693–737, 2017. `doi:10.1016/j.apal.2016.10.012`.

10 Peter Clote. Computational models and function algebras. In Daniel Leivant, editor, *Logical and Computational Complexity. Selected Papers. Logic and Computational Complexity, International Workshop LCC '94, Indianapolis, Indiana, USA, 13-16 October 1994*, volume 960 of *Lecture Notes in Computer Science*, pages 98–130. Springer, 1994. `doi:10.1007/3-540-60178-3_81`.

11 Leonard Eugene Dickson. Finiteness of the odd perfect and primitive abundant numbers with n distinct prime factors. *American Journal of Mathematics*, 35(4):413–422, 1913.

12 Francesc Esteva and Lluıs Godo. Monoidal t-norm based logic: towards a logic for left-continuous t-norms. *Fuzzy Sets and Systems*, 124(3):271–288, 2001. Fuzzy Logic. `doi:10.1016/S0165-0114(01)00098-7`.

13 Diego Figueira, Santiago Figueira, Sylvain Schmitz, and Philippe Schnoebelen. Ackermannian and primitive-recursive bounds with Dickson's lemma. In *Proceedings - Symposium on Logic in Computer Science*, pages 269–278, 2011. `doi:10.1109/LICS.2011.39`.

14 N. Galatos, P. Jipsen, T. Kowalski, and H. Ono. *Residuated lattices: an algebraic glimpse at substructural logics*, volume 151 of *Studies in Logic and the Foundations of Mathematics*. Elsevier, 2007.

15 Zuzana Haniková. Complexity of some language fragments of fuzzy logics. *Soft Computing*, 21(1):69–77, 2017. `doi:10.1007/s00500-016-2346-0`.

16 Rostislav Horcík, Carles Noguera, and Milan Petrık. On n-contractive fuzzy logics. *Mathematical Logic Quarterly*, 53:268–288, June 2007. `doi:10.1002/malq.200610044`.

17 Rostislav Horčík and Kazushige Terui. Disjunction property and complexity of substructural logics. *Theoretical Computer Science*, 412(31):3992–4006, 2011. `doi:10.1016/j.tcs.2011.04.004`.

18 Emil Jerábek. A note on the substructural hierarchy. *Math. Log. Q.*, 62(1-2):102–110, 2016. `doi:10.1002/MALQ.201500066`.

19 Richard M. Karp and Raymond E. Miller. Parallel program schemata. *J. Comput. Syst. Sci.*, 3(2):147–195, 1969. `doi:10.1016/S0022-0000(69)80011-5`.

20 S. Kripke. The problem of entailment (abstract). *J. of Symbolic Logic*, 24:324, 1959.

21 Ranko Lazic and Sylvain Schmitz. The complexity of coverability in ν-petri nets. In Martin Grohe, Eric Koskinen, and Natarajan Shankar, editors, *Proceedings of the 31st Annual ACM/IEEE Symposium on Logic in Computer Science, LICS '16, New York, NY, USA, July 5-8, 2016*, pages 467–476. ACM, 2016. `doi:10.1145/2933575.2933593`.

22 Ranko Lazic and Sylvain Schmitz. The ideal view on Rackoff's coverability technique. *Inf. Comput.*, 277:104582, 2021. `doi:10.1016/J.IC.2020.104582`.

23 Jérôme Leroux and Grégoire Sutre. Flat counter automata almost everywhere! In Doron A. Peled and Yih-Kuen Tsay, editors, *Automated Technology for Verification and Analysis, Third International Symposium, ATVA 2005, Taipei, Taiwan, October 4-7, 2005, Proceedings*, volume 3707 of *Lecture Notes in Computer Science*, pages 489–503. Springer, 2005. `doi:10.1007/11562948_36`.

24 George Metcalfe, Nicola Olivetti, and Dov Gabbay. *Proof Theory for Fuzzy Logics*. Springer Netherlands, 2009. `doi:10.1007/978-1-4020-9409-5`.

25 George Metcalfe, Francesco Paoli, and Constantine Tsinakis. *Residuated structures in algebra and logic*, volume 277 of *Mathematical Surveys and Monographs*. American Mathematical Society, Providence, RI, 2023.

26 G. E. Minc. Some calculi of modal logic. *Trudy Mat. Inst. Steklov*, 98:88–111, 1968.

27 Francesco Paoli. *Substructural logics: a primer*, volume 13 of *Trends in Logic—Studia Logica Library*. Kluwer Academic Publishers, Dordrecht, 2002. `doi:10.1007/978-94-017-3179-9`.

28 G. Pottinger. Uniform, cut-free formulations of T, S4 and S5 (abstract). *J. of Symbolic Logic*, 48(3):900, 1983.

29 Revantha Ramanayake. Extended Kripke lemma and decidability for hypersequent substructural logics. In Holger Hermanns, Lijun Zhang, Naoki Kobayashi, and Dale Miller, editors, *LICS '20: 35th Annual ACM/IEEE Symposium on Logic in Computer Science, Saarbrücken, Germany, July 8-11, 2020*, pages 795–806. ACM, 2020. `doi:10.1145/3373718.3394802`.

30 Greg Restall. *An Introduction to Substructural Logics*. Routledge, 2000.

31 Sylvain Schmitz. Complexity hierarchies beyond elementary. *ACM Trans. Comput. Theory*, 8(1), February 2016. doi:10.1145/2858784.

32 Alasdair Urquhart. The complexity of decision procedures in relevance logic II. *The Journal of Symbolic Logic*, 64(4):1774–1802, 1999. doi:10.2307/2586811.