Infinitary cut-elimination via finite approximations

Matteo Acclavio
University of Southern Denmark, Denmark

Gianluca Curzi
University of Birmingham, UK

Giulio Guerrieri
Aix Marseille Univ, CNRS, LIS UMR 7020, Marseille, France

Abstract

We investigate non-wellfounded proof systems based on parsimonious logic, a weaker variant of linear logic where the exponential modality ! is interpreted as a constructor for streams over finite data. Logical consistency is maintained at a global level by adapting a standard progressing criterion. We present an infinitary version of cut-elimination based on finite approximations, and we prove that, in presence of the progressing criterion, it returns well-defined non-wellfounded proofs at its limit. Furthermore, we show that cut-elimination preserves the progressing criterion and various regularity conditions internalizing degrees of proof-theoretical uniformity. Finally, we provide a denotational semantics for our systems based on the relational model.

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

Keywords and phrases cut-elimination, non-wellfounded proofs, parsimonious logic, linear logic, proof theory, approximation, sequent calculus, non-uniform proofs

Digital Object Identifier 10.4230/LIPIcs.CSL.2024.41

Related Version Arxiv version
Full Version: `https://arxiv.org/abs/2308.07789`

Acknowledgements We would like to thank Anupam Das, Abhishek De, Farzad Jafar-Rahmani, Alexis Saurin, Tito (Lê Thành Dung Nguyên), Damiano Mazza and the anonymous reviewers for their useful comments and suggestions.

1 Introduction

Non-wellfounded proof theory studies proofs as possibly infinite (but finitely branching) trees, where logical consistency is maintained via global conditions called *progressing* (or *validity*) *criteria*. In this setting, the so-called *regular* (also called *circular*) proofs receive a special attention, as they admit a finite description in terms of (possibly cyclic) directed graphs.

This area of proof theory makes its first appearance (in its modern guise) in the modal μ-calculus [29, 14]. Since then, it has been extensively investigated from many perspectives (see, e.g., [8, 34, 13, 23]), establishing itself as an ideal setting for manipulating least and greatest fixed points, and hence for modeling induction and coinduction principles.

Non-wellfounded proof theory has been applied to constructive fixed point logics i.e., with a computational interpretation based on the *Curry-Howard correspondence* [35]. A key example can be found in the context of *linear logic* (LL) [21], a logic implementing a finer control on resources thanks to the *exponential* modalities ! and ?. In this framework, the most extensively studied fixed point logic is μMALL, defined as the exponential-free fragment of LL with least and greatest fixed point operators (respectively, μ and its dual ν) [7, 6].

In [7] Baelde and Miller have shown that the exponentials can be recovered in μMALL by exploiting the fixed points operators, i.e., by defining $!A := \nu X.(\mathbf{1} \,\&\, A \,\&\, (X \otimes X))$ and $?A := \mu X.(\bot \oplus A \oplus (X \parr X))$. As these authors notice, the fixed point-based definition of !

32nd EACSL Annual Conference on Computer Science Logic (CSL 2024).
Editors: Aniello Murano and Alexandra Silva; Article No. 41; pp. 41:1–41:19
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

and ? can be regarded as a more permissive variant of the standard exponentials, since a proof of $\nu X.(\mathbf{1} \,\&\, A \,\&\, (X \otimes X))$ could be constructed using different proofs of A, whereas in LL a proof of $!A$ is constructed uniformly using a single proof of A. This proof-theoretical notion of *non-uniformity* is indeed a central feature of the fixed-point exponentials.

However, the above encoding is not free of issues. First, as discussed in full detail in [16], the encoding of the exponentials does not verify the Seely isomorphisms, syntactically expressed by the equivalence $!(A \,\&\, B) \multimapboth (!A \otimes !B)$, an essential property for modeling exponentials in LL. Specifically, the fixed-point definition of ! relies on the multiplicative connective $\otimes$, which forces an interpretation of $!A$ based on lists rather than multisets. Secondly, as pointed out in [7], there is a neat mismatch between cut-elimination for the exponentials of LL and the one for the fixed point exponentials of μMALL. While the first problem is related to syntactic deficiencies of the encoding, and does not undermine further investigations on fixed point-based definitions of the exponential modalities, the second one is more critical. These apparent differences between the two exponentials contribute to stressing an important aspect in linear logic modalities, i.e., their *non-canonicity* [31, 12][1].

On a parallel research thread, Mazza [25, 26, 27] studied *parsimonious logic*, a variant of linear logic where the exponential modality ! satisfies Milner's law (i.e., $!A \multimapboth A \otimes !A$) and invalidates the implications $!A \multimap !!A$ (*digging*) and $!A \multimap !A \otimes !A$ (*contraction*). In parsimonious logic, a proof of $!A$ can be interpreted as a *stream* over (a finite set of) proofs of A, i.e., as a greatest fixed point, where the linear implications $A \otimes !A \multimap !A$ (*co-absorption*) and $!A \multimap A \otimes !A$ (*absorption*) can be read computationally as the *push* and *pop* operations on streams. More specifically, a formula $!A$ is introduced by an *infinitely branching rule* that takes a finite set of proofs $\mathcal{D}_1, \ldots, \mathcal{D}_n$ of A and a (possibly non-recursive) function $f : \mathbb{N} \to \{1, \ldots, n\}$ as premises, and constructs a proof of $!A$ representing a stream of proofs of the form $\mathfrak{S} = (\mathcal{D}_{f(0)}, \mathcal{D}_{f(1)}, \ldots, \mathcal{D}_{f(n)}, \ldots)$. Hence, parsimonious logic exponential modalities exploit in an essential way the above-mentioned proof-theoretical non-uniformity, which in turn deeply interfaces with notions of non-uniformity from computational complexity [27].

The analysis of parsimonious logic conducted in [26, 27] reveals that fixed point definitions of the exponentials are better behaving when digging and contraction are discarded. On the other hand, the co-absorption rule cannot be derived in LL, and so it prevents parsimonious logic becoming a genuine subsystem of the latter. This led the authors of the present paper to introduce *parsimonious linear logic*, a subsystem of linear logic (in particular, co-absorption-free) that nonetheless allows a stream-based interpretation of the exponentials.

We present two finitary proof systems for parsimonious linear logic: the system nuPLL, supporting non-uniform exponentials, and PLL, a fully uniform version. We investigate non-wellfounded counterparts of nuPLL and PLL, adapting to our setting the progressing criterion to maintain logical consistency. To recover the proof-theoretical behavior of nuPLL and PLL, we identify further global conditions on non-wellfounded proofs, that is, some forms of regularity to capture the notions of uniformity and non-uniformity. This leads us to two main non-wellfounded proof systems: *regular parsimonious linear logic* (rPLL^∞), defined via the regularity condition and corresponding to PLL, and *weakly regular parsimonious linear logic* (wrPLL^∞), defined via a *weak regularity* condition and corresponding to nuPLL.

The major contribution of this paper is the study of continuous cut-elimination in the setting of non-wellfounded parsimonious linear logic. We first introduce Scott-domains of partially defined non-wellfounded proofs, ordered by an approximation relation. Here, undefinedness in proofs is expressed by the use of an axiom introducing an arbitrary sequent;

---

[1] One can construct LL proof systems with alternative (not equivalent) exponential modalities, see [28].

this approach is analogous to the one used to define Böhm trees in the λ-calculus: intuitively, a non-wellfounded proof is kind of like a Böhm tree that may be described by its finite approximations, with the difference that—in the λ-calculus—Böhm trees, and therefore their finite approximations, are normal (that is, cut-free) by definition, whereas here proofs need not be cut-free and so the approximations too may contain cuts. Then, we define special infinitary proof rewriting strategies called *maximal and continuous infinitary cut-elimination strategies* (mc-ices) which compute (Scott-)continuous functions. Productivity in this framework is established by showing that, in presence of a good global condition (progressing, regularity or weak regularity), these continuous functions return totally defined cut-free non-wellfounded proofs and preserve the global condition: progressing (Theorem 33.1), and regularity or weak regularity (Theorem 33.2).

On a technical side, we stress that our methods and results distinguish from previous approaches to cut-elimination in a non-wellfounded setting in many respects. First, we get rid of many technical notions typically introduced to prove infinitary cut-elimination, such as the *multicut rule* or the *fairness conditions* (as in, e.g., [20, 6]), as these notions are subsumed by a *finitary approximation* approach to cut-elimination. Furthermore, we prove productivity of cut-elimination and preservation of the progressing condition in a more direct and constructive way, i.e., without going through auxiliary proof systems and avoiding arguments by contradiction (see, e.g., [6]). Finally, we prove for the first time preservation of regularity properties under continuous cut-elimination, essentially exploiting methods for compressing transfinite rewriting sequences to ω-long ones from [36, 25, 33].

Finally, we define a denotational semantics for non-wellfounded parsimonious logic based on the relational model, with a standard multiset-based interpretation of the exponentials, and we show that this semantics is preserved under continuous cut-elimination (Theorem 38). We also prove that extending non-wellfounded parsimonious linear logic with digging prevents the existence of a cut-elimination result preserving the semantics (Theorem 40). Therefore, the impossibility of a stream-based definition of ! that validates digging (and contraction).

Additional details of the proofs are provided in the extended version of this paper [2].

2 Preliminary notions

In this section we recall some basic notions from (non-wellfounded) proof theory, fixing the notation that will be adopted in this paper.

2.1 Derivations and coderivations

We assume that the reader is familiar with the syntax of sequent calculus, e.g. [37]. Here we specify some conventions adopted to simplify the content of this paper.

We consider (**sequent**) **rules** of the form $\mathsf{r}\frac{}{\Gamma}$ or $\mathsf{r}\frac{\Gamma_1}{\Gamma}$ or $\mathsf{r}\frac{\Gamma_1 \quad \Gamma_2}{\Gamma}$, and we refer to the sequents Γ_1 and Γ_2 as the **premises**, and to the sequent Γ as the **conclusion** of the rule r. To avoid technicalities of the sequents-as-lists presentation, we follow [6] and we consider **sequents** as *sets of occurrences of formulas* from a given set of formulas. In particular, when we refer to a formula in a sequent we always consider a *specific occurrence* of it.

▶ **Definition 1.** *A (binary, possibly infinite)* ***tree*** $\mathcal{T}$ *is a subset of words in* $\{1,2\}^*$ *that contains the empty word* ϵ *(the* ***root*** *of* $\mathcal{T}$*) and is* ordered-prefix-closed *(i.e., if* $n \in \{1,2\}$ *and* $vn \in \mathcal{T}$*, then* $v \in \mathcal{T}$*, and if moreover* $v2 \in \mathcal{T}$*, then* $v1 \in \mathcal{T}$*). The elements of* $\mathcal{T}$ *are called* ***nodes*** *and their* ***height*** *is the length of the word. A* ***child*** *of* $v \in \mathcal{T}$ *is any* $vn \in \mathcal{T}$ *with* $n \in \{1,2\}$*. The*

$$\mathsf{ax}\,\frac{}{A, A^{\perp}} \quad \mathsf{cut}\,\frac{\Gamma, A \quad A^{\perp}, \Delta}{\Gamma, \Delta} \quad \otimes\,\frac{\Gamma, A \quad B, \Delta}{\Gamma, \Delta, A \otimes B} \quad \parr\,\frac{\Gamma, A, B}{\Gamma, A \parr B} \quad \mathsf{1}\,\frac{}{1} \quad \perp\,\frac{\Gamma}{\Gamma, \perp} \quad \mathsf{f!p}\,\frac{\Gamma, A}{?\Gamma, !A} \quad \mathsf{?w}\,\frac{\Gamma}{\Gamma, ?A} \quad \mathsf{?b}\,\frac{\Gamma, A, ?A}{\Gamma, ?A}$$

Figure 1 Sequent calculus rules of PLL.

prefix order *is a partial order* $\leq_{\mathcal{T}}$ *on* $\mathcal{T}$ *defined by: for any* $v, v' \in \mathcal{T}$, $v \leq_{\mathcal{T}} v'$ *if* $v' = vw$ *for some* $w \in \{1, 2\}^*$. *A maximal element of* $\leq_{\mathcal{T}}$ *is a* ***leaf*** *of* $\mathcal{T}$. *A* ***branch*** *of* $\mathcal{T}$ *is a set* $\mathcal{B} \subseteq \mathcal{T}$ *such that* $\epsilon \in \mathcal{B}$ *and if* $w \in \mathcal{B}$ *is not a leaf of* $\mathcal{T}$ *then* w *has exactly one child in* $\mathcal{B}$.

A ***coderivation*** *over a set of rules* $\mathcal{S}$ *is a labeling* $\mathcal{D}$ *of a tree* $\mathcal{T}$ *by sequents such that if* v *is a node of* $\mathcal{T}$ *with children* $v_1, \ldots, v_n$ *(with* $n \in \{0, 1, 2\}$*), then there is an occurrence of a rule* r *in* $\mathcal{S}$ *with conclusion the sequent* $\mathcal{D}(v)$ *and premises the sequents* $\mathcal{D}(v_1), \ldots, \mathcal{D}(v_n)$. *The* ***height*** *of* r *in* $\mathcal{D}$ *is the height of the node* $v \in \mathcal{T}$ *such that* $\mathcal{D}(v)$ *is the conclusion of* r.

The ***conclusion*** *of* $\mathcal{D}$ *is the sequent* $\mathcal{D}(\epsilon)$. *If* v *is a node of the tree, the* ***sub-coderivation*** *of* $\mathcal{D}$ *rooted at* v *is the coderivation* $\mathcal{D}_v$ *defined by* $\mathcal{D}_v(w) = \mathcal{D}(vw)$.

A coderivation $\mathcal{D}$ *is* r***-free*** *(for a rule* $\mathsf{r} \in \mathcal{S}$*) if it contains no occurrence of* r. *It is* ***regular*** *if it has finitely many distinct sub-coderivations; it is* ***non-wellfounded*** *if it labels an infinite tree, and it is a* ***derivation*** *(with* ***size*** $|\mathcal{D}| \in \mathbb{N}$*) if it labels a finite tree (with* $|\mathcal{D}|$ *nodes).*

Given a set of coderivations X, *a sequent* Γ *is* ***provable*** *in* X *(noted* $\vdash_{\mathsf{X}} \Gamma$*) if there is a coderivation in* X *with conclusion* Γ.

While derivations are usually represented as finite trees, regular coderivations can be represented as *finite* directed (possibly cyclic) graphs: a cycle is created by linking the roots of two identical subcoderivations.

▶ **Definition 2.** *Let* $\mathcal{D}$ *be a coderivation labeling a tree* $\mathcal{T}$. *A* ***bar*** *(resp.* ***prebar****) of* $\mathcal{D}$ *is a set* $\mathcal{V} \subseteq \mathcal{T}$ *where:*

- *any branch (resp. infinite branch) of the tree* $\mathcal{T}$ *underlying* $\mathcal{D}$ *contains a node in* $\mathcal{V}$;
- *any pair of nodes in* $\mathcal{V}$ *are mutually incomparable with respect to the prefix order* $\leq_{\mathcal{T}}$.

The ***height*** *of a prebar* $\mathcal{V}$ *of* $\mathcal{D}$ *is the minimal height of the nodes of* $\mathcal{V}$.

3 Parsimonious Linear Logic

In this paper we consider the set of **formulas** for propositional multiplicative-exponential linear logic with units (MELL). These are generated by a countable set of propositional variables $\mathcal{A} = \{X, Y, \ldots\}$ using the following grammar:

$$A, B ::= X \mid X^{\perp} \mid A \otimes B \mid A \parr B \mid !A \mid ?A \mid 1 \mid \perp$$

A **!-formula** (resp. **?-formula**) is a formula of the form $!A$ (resp. $?A$). **Linear negation** $(\cdot)^{\perp}$ is defined by De Morgan's laws $(A^{\perp})^{\perp} = A$, $(A \otimes B)^{\perp} = A^{\perp} \parr B^{\perp}$, $(!A)^{\perp} = ?A^{\perp}$, and $(1)^{\perp} = \perp$ while **linear implication** is defined as $A \multimap B := A^{\perp} \parr B$.

▶ **Definition 3.** ***Parsimonious linear logic****, denoted by* PLL, *is the set of rules in Figure 1, that is,* ***axiom*** *(*ax*),* ***cut*** *(*cut*),* ***tensor*** *(*$\otimes$*),* ***par*** *(*$\parr$*),* ***one*** *(*$\mathsf{1}$*),* ***bottom*** *(*$\perp$*),* ***functorial promotion*** *(*$\mathsf{f!p}$*),* ***weakening*** *(*$\mathsf{?w}$*),* ***absorption*** *(*$\mathsf{?b}$*). Rules* ax, $\otimes$, $\parr$, $\mathsf{1}$ *and* $\perp$ *are called* ***multiplicative****, while rules* $\mathsf{f!p}$, $\mathsf{?w}$ *and* $\mathsf{?b}$ *are called* ***exponential****. We also denote by* PLL *the set of derivations over the rules in* PLL.

▶ **Example 4.** Figure 2 gives some examples of derivation in PLL. The (distinct) derivations $\underline{0}$ and $\underline{1}$ prove the same formula $\mathbf{N} := !(X \multimap X) \multimap X \multimap X$. The derivation $\mathcal{D}_{\mathsf{abs}}$ proves the *absorption law* $!A \multimap A \otimes !A$; the derivation $\mathcal{D}_{\mathsf{der}}$ proves the *dereliction law* $!A \multimap A$.

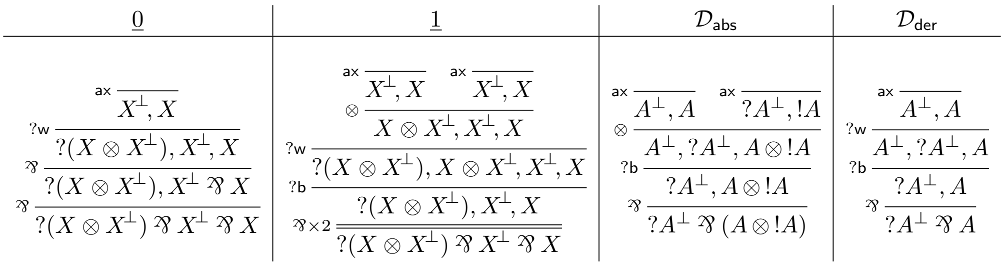


Figure 2 Examples of derivations in PLL.

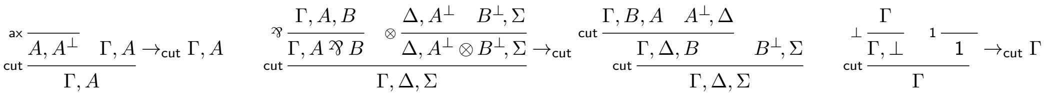


Figure 3 Multiplicative cut-elimination steps in PLL.

The **cut-elimination** relation $\to_{\mathsf{cut}}$ in PLL is the union of **principal** cut-elimination steps in Figure 3 (**multiplicative**) and Figure 4 (**exponential**) and **commutative** cut-elimination steps in Figure 5. The reflexive-transitive closure of $\to_{\mathsf{cut}}$ is noted $\to^*_{\mathsf{cut}}$.

▶ **Theorem 5.** *For every $\mathcal{D} \in \mathsf{PLL}$, there is a cut-free $\mathcal{D}' \in \mathsf{PLL}$ such that $\mathcal{D} \to^*_{\mathsf{cut}} \mathcal{D}'$.*

Sketch of proof. We associate with any derivation $\mathcal{D}$ in PLL a derivation $\mathcal{D}^{♠}$ in MELL sequent calculus. Thanks to additional commutative cut-elimination steps, we prove that cut-elimination in MELL rewrites $\mathcal{D}^{♠}$ to the translation of a derivation in PLL. The termination of cut-elimination in PLL follows from strong normalisation of (second-order) MELL [30]. ◀

Akin to light linear logic [22, 24, 32], the exponential rules of PLL are weaker than those in MELL: the usual promotion rule is replaced by f!p (*functorial promotion*), and the usual contraction and dereliction rules by ?b. As a consequence, the *digging* formula $!A \multimap !!A$ and the *contraction* formula $!A \multimap !A \otimes !A$ are not provable in PLL (unlike the dereliction formula, Example 4). This allows us to interpret computationally these weaker exponentials in terms of streams, as well as to control the complexity of cut-elimination [26, 27].

It is easy to show that MELL = PLL + digging: if we add the digging formula as an axiom (or equivalently, the *digging rule* ??d in Figure 13) to the set of rules in Figure 1, then the contraction formula becomes provable, and the obtained proof system coincides with MELL.

4 Non-wellfounded Parsimonious Linear Logic

In linear logic, a formula $!A$ is interpreted as the availability of A at will. This intuition still holds in PLL. Indeed, the Curry-Howard correspondence interprets rule f!p introducing the modality ! as an operator taking a derivation $\mathcal{D}$ of A and creating a (infinite) *stream* $(\mathcal{D}, \mathcal{D}, \dots, \mathcal{D}, \dots)$ of copies of the proof $\mathcal{D}$. Each element of the stream is accessed via the cut-elimination step f!p vs ?b in Figure 4: rule ?b is interpreted as an operator *popping* one copy of $\mathcal{D}$ out of the stream. Pushing these ideas further, Mazza [26] introduced *parsimonious logic* **PL**, a type system (comprising rules f!p and ?b) characterizing the logspace decidable problems.

Mazza and Terui then introduced in [27] another type system, $\mathbf{nuPL}_{\forall\ell}$, based on parsimonious logic and capturing the complexity class $\mathbf{P}/\mathsf{poly}$ (i.e., the problems decidable by polynomial size families of Boolean circuits [5]). Their system is endowed with a *non-uniform* version of the functorial promotion, which takes a finite set of proofs $\mathcal{D}_1, \dots, \mathcal{D}_n$ of A and a

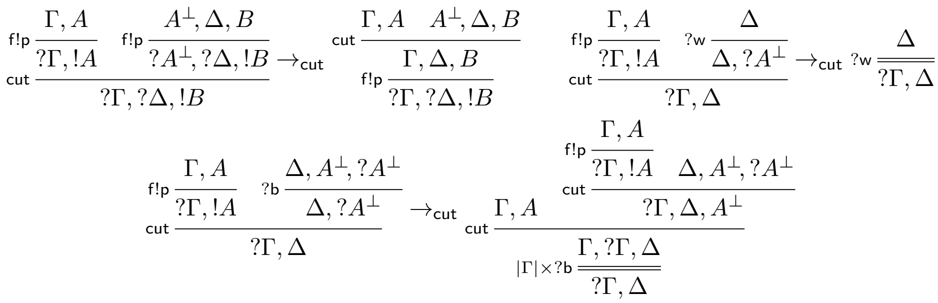


Figure 4 Exponential cut-elimination steps in PLL.

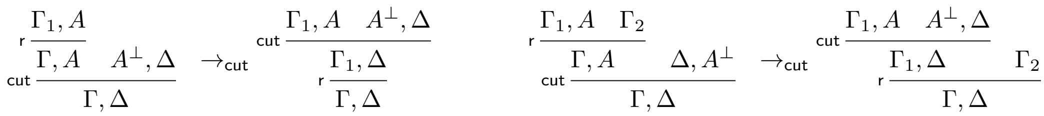


Figure 5 Commutative cut-elimination steps in PLL, where $\mathsf{r} \neq \mathsf{cut}$.

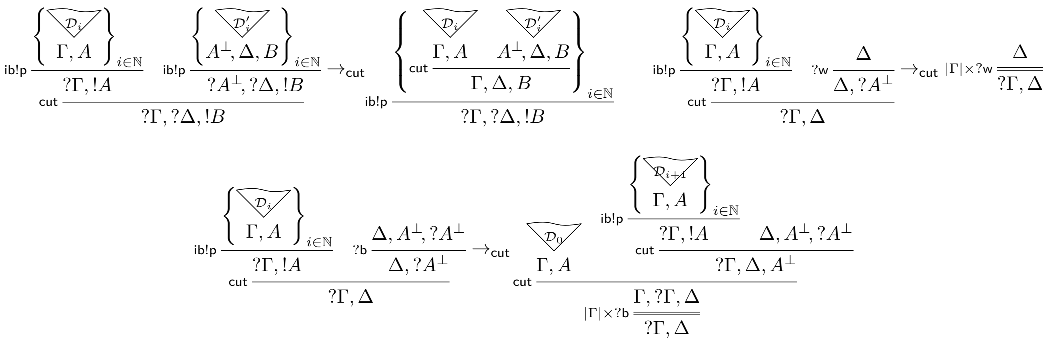


Figure 6 Exponential cut-elimination steps in nuPLL.

(possibly non-recursive) function $f\colon \mathbb{N} \to \{1, \dots, n\}$ as premises, and constructs a proof of $!A$ modeling the stream $(\mathcal{D}_{f(0)}, \mathcal{D}_{f(1)}, \dots, \mathcal{D}_{f(n)}, \dots)$. This typing rule is the key tool to encode the so-called *advices* for Turing machines, an essential step to show completeness for $\mathbf{P}/\mathsf{poly}$.

In a similar vein, we can endow PLL with a non-uniform version of f!p called **infinitely branching promotion** (ib!p), which constructs a stream $(\mathcal{D}_0, \mathcal{D}_1, \dots, \mathcal{D}_n, \dots)$ with finite support, i.e., made of *finitely* many distinct derivations (of the same conclusion):[2]

$$\mathsf{ib!p}\,\frac{\overset{\mathcal{D}_0}{\Gamma, A} \quad \overset{\mathcal{D}_1}{\Gamma, A} \quad \cdots \quad \overset{\mathcal{D}_n}{\Gamma, A} \quad \cdots}{?\Gamma, !A}\ \{\mathcal{D}_i \mid i \in \mathbb{N}\}\text{ is finite} \qquad \Bigg| \qquad \mathsf{!w}\,\frac{}{!A} \qquad \mathsf{!b}\,\frac{\Gamma, A \quad \Delta, !A}{\Gamma, \Delta, !A} \tag{1}$$

The side condition on ib!p provides a proof theoretic counterpart to the function $f\colon \mathbb{N} \to \{1, \dots, n\}$ in $\mathbf{nuPL}_{\forall\ell}$. Clearly, f!p is subsumed by the rule ib!p, as it corresponds to the special (uniform) case where $\mathcal{D}_i = \mathcal{D}_{i+1}$ for all $i \in \mathbb{N}$.

[2] Rule ib!p is reminiscent of the ω-rule used in (first-order) Peano arithmetic to derive formulas of the form $\forall x \phi$ that cannot be proven in a uniform way.

$$\mathcal{D}_{\lightning} := \mathsf{cut}\,\dfrac{\mathsf{ax}\,\dfrac{}{A^\perp, A} \qquad \mathsf{cut}\,\dfrac{\mathsf{ax}\,\dfrac{}{A^\perp, A} \qquad \mathsf{cut}\,\dfrac{\vdots}{\Gamma, A}}{\Gamma, A}}{\Gamma, A} \qquad\qquad \mathcal{D}_{?} := \mathsf{?b}\,\dfrac{\mathsf{?b}\,\dfrac{\mathsf{?b}\,\dfrac{\vdots}{A, A, ?A}}{A, ?A}}{?A}$$

Figure 7 Two non-wellfounded and non-progressing coderivations in PLL^∞.

▶ **Definition 6.** *We define the set of rules* $\mathsf{nuPLL} := \{\mathsf{ax}, \otimes, \parr, 1, \perp, \mathsf{cut}, \mathsf{?b}, \mathsf{?w}, \mathsf{ib!p}\}$. *We also denote by* nuPLL *the set of derivations over the rules in* nuPLL.[3]

There are some notable differences between nuPLL and Mazza and Terui's original system $\mathbf{nuPL}_{\forall\ell}$ [27]. As opposed to nuPLL, $\mathbf{nuPL}_{\forall\ell}$ is formulated as an intuitionistic (type) system. Furthermore, to achieve completeness for $\mathbf{P}/\mathsf{poly}$, these authors introduced second-order quantifiers and the co-absorption ($\mathsf{!b}$) and co-weakening ($\mathsf{!w}$) rules displayed in (1).

Cut-elimination steps for nuPLL are in Figures 3, 5, and 6. In particular, the step $\mathsf{ib!p}$-vs-$\mathsf{?b}$ in Figure 6 *pops* the first premise $\mathcal{D}_0$ of $\mathsf{ib!p}$ out of the stream $(\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_n, \ldots)$.

4.1 From infinitely branching proofs to non-wellfounded proofs

In this paper we explore a dual approach to the one of $\mathbf{nuPL}_{\forall\ell}$ (and nuPLL): instead of considering (wellfounded) derivations with infinite branching, we consider (non-wellfounded) coderivations with finite branching. For this purpose, the infinitary rule $\mathsf{ib!p}$ of nuPLL is replaced by the binary rule below, called **conditional promotion** ($\mathsf{c!p}$):

$$\mathsf{c!p}\,\dfrac{\Gamma, A \qquad ?\Gamma, !A}{?\Gamma, !A} \tag{2}$$

▶ **Definition 7.** *We define the set of rules* $\mathsf{PLL}^\infty := \{\mathsf{ax}, \otimes, \parr, 1, \perp, \mathsf{cut}, \mathsf{?b}, \mathsf{?w}, \mathsf{c!p}\}$. *We also denote by* PLL^∞ *the set of coderivations over the rules in* PLL^∞.

In other words, PLL^∞ is the set of coderivations generated by the same rules as PLL, except that $\mathsf{f!p}$ is replaced by $\mathsf{c!p}$. From now on, we will only consider coderivations in PLL^∞.

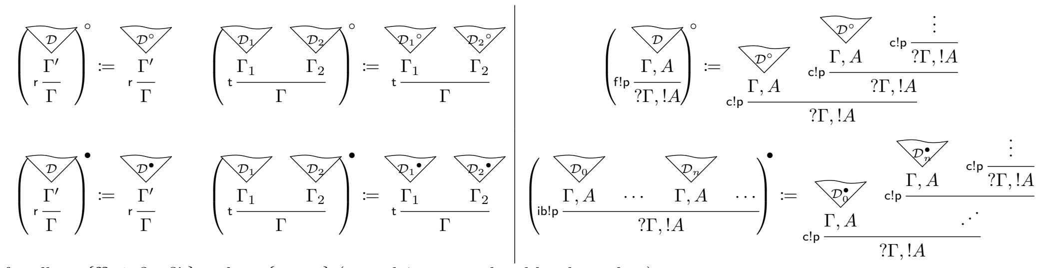
▶ **Example 8.** Figure 7 shows two non-wellfounded coderivations in PLL^∞: $\mathcal{D}_{\lightning}$ (resp. $\mathcal{D}_?$) has an infinite branch of cut (resp. $\mathsf{?b}$) rules, and is (resp. is not) regular.

We can embed PLL and nuPLL into PLL^∞ via the conclusion-preserving translations $(\cdot)^\circ \colon \mathsf{PLL} \to \mathsf{PLL}^\infty$ and $(\cdot)^\bullet \colon \mathsf{nuPLL} \to \mathsf{PLL}^\infty$ defined in Figure 8 by induction on derivations: they map all rules to themselves except $\mathsf{f!p}$ and $\mathsf{ib!p}$, which are "unpacked" into non-wellfounded coderivations that iterate infinitely many times the rule $\mathsf{c!p}$.

An infinite chain of $\mathsf{c!p}$ rules (Figure 9) is a structure of interest in itself in PLL^∞.

▶ **Definition 9.** *A* ***non-wellfounded box*** *(nwb for short) is a coderivation* $\mathcal{D} \in \mathsf{PLL}^\infty$ *with an infinite branch* $\{\epsilon, 2, 22, \dots\}$ *(the* ***main branch*** *of* $\mathcal{D}$*) all labeled by* $\mathsf{c!p}$ *rules as in Figure 9, where* $!A$ *in the conclusion is the* ***principal formula*** *of* $\mathcal{D}$*, and* $\mathcal{D}_0, \mathcal{D}_1, \ldots$ *are the* ***calls*** *of* $\mathcal{D}$*. We denote* $\mathcal{D}$ *by* $\mathsf{c!p}_{(\mathcal{D}_0, \ldots, \mathcal{D}_n, \ldots)}$.

[3] To be rigorous, this requires a slight change in Definition 1: the tree labeled by a derivation in nuPLL must be over $\mathbb{N}^\omega$ instead of $\{1,2\}^*$, in order to deal with infinitely branching derivations.



for all $\mathsf{r} \in \{⅋, \bot, ?\mathsf{w}, ?\mathsf{b}\}$ and $\mathsf{t} \in \{\mathsf{cut}, \otimes\}$ (ax and 1 are translated by themselves).

Figure 8 Translations $(\cdot)^\circ$ from PLL to PLL$^\infty$, and $(\cdot)^\bullet$ from nuPLL to PLL$^\infty$.

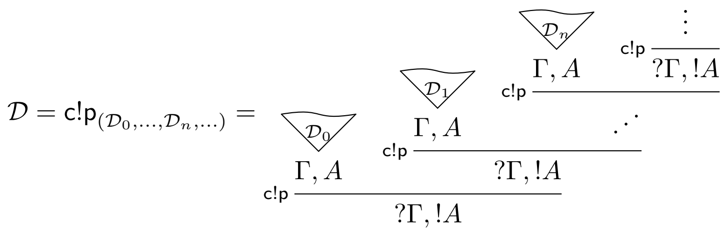
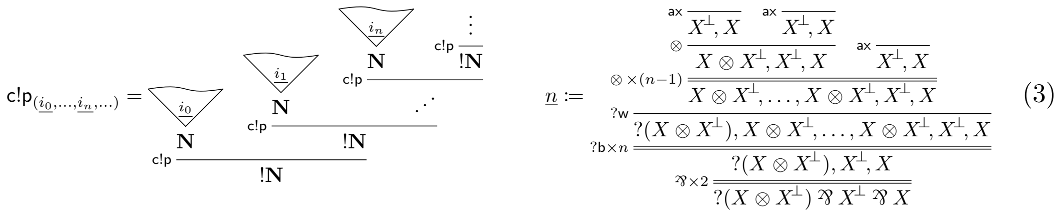


Figure 9 A non-wellfounded box in PLL$^\infty$.

Let $\mathfrak{S} = \mathsf{c!p}_{(\mathcal{D}_0,\ldots,\mathcal{D}_n,\ldots)}$ be a nwb. *We may write $\mathfrak{S}(i)$ to denote $\mathcal{D}_i$. We say that $\mathfrak{S}$ has **finite support** (resp. is **periodic** with **period** k) if $\{\mathfrak{S}(i) \mid i \in \mathbb{N}\}$ is finite (resp. if $\mathfrak{S}(i) = \mathfrak{S}(k+i)$ for any $i \in \mathbb{N}$). A coderivation $\mathcal{D}$ has **finite support** (resp. is **periodic**) if any* nwb *in $\mathcal{D}$ has finite support (resp. is periodic).*

▶ **Example 10.** The only cut-free derivations of the formula $\mathbf{N} := !(X \multimap X) \multimap X \multimap X$ are of the form $\underline{n}$ below on the right, for all $n \in \mathbb{N}$, up to permutations of the rules $?\mathsf{w}$, $?\mathsf{b}$ and $\otimes$ (the derivations $\underline{0}$ and $\underline{1}$ in Example 4 are special cases of it)



Consider the nwb $\mathsf{c!p}_{(\underline{i_0},\ldots,\underline{i_n},\ldots)}$ above on the left, proving the formula $!\mathbf{N}$, where $i_j \in \{0,1\}$ for all $j \in \mathbb{N}$. Thus $\mathsf{c!p}_{(\underline{i_0},\ldots,\underline{i_n},\ldots)}$ has finite support, as its only calls can be $\underline{0}$ or $\underline{1}$, and it is periodic if and only if so is the infinite sequence $(i_0, \ldots, i_n, \ldots) \in \{0,1\}^\omega$.

The *cut-elimination* steps $\to_{\mathsf{cut}}$ for PLL$^\infty$ are in Figures 3, 5, and 10. Computationally, they allow the c!p rule to be interpreted as a *coinductive* definition of a stream of type $!A$ from a stream of the same type to which an element of type A is prepended. In particular, the cut-elimination step c!p vs ?b accesses the head of a stream: rule ?b acts as a *pop* operator.

As a consequence, the nwb in Figure 9 constructs a stream $(\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_n, \ldots)$ similarly to ib!p but, unlike the latter, all the $\mathcal{D}_i$'s may be pairwise distinct. The reader expert in linear logic can see a nwb as a box with possibly *infinitely many* distinct contents (its calls), while usual linear logic boxes (and f!p in PLL) provide infinitely many copies of the *same* content.

Rules f!p in PLL and ib!p in nuPLL are mapped by $(\cdot)^\circ$ and $(\cdot)^\bullet$ into nwbs, which are non-wellfounded coderivations. Hence, the cut-elimination steps f!p vs f!p in PLL and ib!p vs ib!p in nuPLL can only be simulated by infinitely many cut-elimination steps in PLL$^\infty$.

$$\dfrac{\dfrac{\Gamma, A \quad ?\Gamma, !A}{?\Gamma, !A}\,\mathsf{c!p} \qquad \dfrac{A^\perp, \Delta, B \quad ?A^\perp, ?\Delta, !B}{?A^\perp, ?\Delta, !B}\,\mathsf{c!p}}{?\Gamma, ?\Delta, !B}\,\mathsf{cut} \;\to_{\mathsf{cut}}\; \dfrac{\dfrac{\Gamma, A \quad A^\perp, \Delta, B}{\Gamma, \Delta, B}\,\mathsf{cut} \qquad \dfrac{?\Gamma, !A \quad ?A^\perp, ?\Delta, !B}{?\Gamma, ?\Delta, !B}\,\mathsf{cut}}{?\Gamma, ?\Delta, !B}\,\mathsf{c!p}$$

$$\dfrac{\dfrac{\Gamma, A \quad ?\Gamma, !A}{?\Gamma, !A}\,\mathsf{c!p} \qquad \dfrac{\Delta}{\Delta, ?A^\perp}\,\mathsf{?w}}{?\Gamma, \Delta}\,\mathsf{cut} \;\to_{\mathsf{cut}}\; \dfrac{\Delta}{?\Gamma, \Delta}\,|\Gamma|\times\mathsf{?w} \qquad \dfrac{\dfrac{\Gamma, A \quad ?\Gamma, !A}{?\Gamma, !A}\,\mathsf{c!p} \qquad \dfrac{\Delta, A^\perp, ?A^\perp}{\Delta, ?A^\perp}\,\mathsf{?b}}{?\Gamma, \Delta}\,\mathsf{cut} \;\to_{\mathsf{cut}}\; \dfrac{\dfrac{\Gamma, A \qquad \dfrac{?\Gamma, !A \quad \Delta, A^\perp, ?A^\perp}{?\Gamma, \Delta, A^\perp}\,\mathsf{cut}}{\Gamma, ?\Gamma, \Delta}\,\mathsf{cut}}{?\Gamma, \Delta}\,|\Gamma|\times\mathsf{?b}$$

Figure 10 Exponential cut-elimination steps for coderivations of PLL^∞.

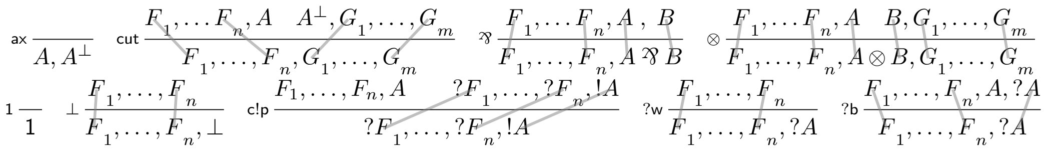


Figure 11 PLL^∞ rules: edges connect a formula in the conclusion with its parent(s) in a premise.

Note that $\mathcal{D}_\lightning \in \mathsf{PLL}^\infty$ in Figure 7 is not cut-free, and if $\mathcal{D}_\lightning \to_{\mathsf{cut}} \mathcal{D}$ then $\mathcal{D} = \mathcal{D}_\lightning$: thus $\mathcal{D}_\lightning$ cannot reduce to a cut-free coderivation, and so the cut-elimination theorem fails in PLL^∞.

4.2 Consistency via a progressing criterion

In a non-wellfounded setting such as PLL^∞, any sequent is provable. Indeed, the (non-wellfounded) coderivation $\mathcal{D}_\lightning$ in Figure 7 shows that any non-empty sequent (in particular, any formula) is provable in PLL^∞, and the empty sequent is provable in PLL^∞ by applying the cut rule on the conclusions B and $B^\perp$ (for any formula B) of two derivations $\mathcal{D}_\lightning$.

The standard way to recover logical consistency in non-wellfounded proof theory is to introduce a global soundness condition on coderivations, called *progressing criterion*. In PLL^∞, this criterion relies on tracking occurrences of !-formulas in a coderivation.

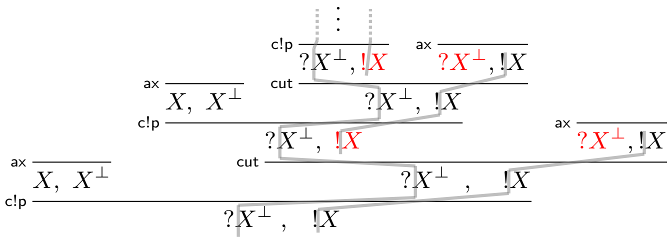
▶ **Definition 11.** *Let $\mathcal{D}$ be a coderivation in PLL^∞. It is **weakly progressing** if every infinite branch contains infinitely many right premises of $\mathsf{c!p}$-rules.*

*An occurrence of a formula in a premise of a rule r is the **parent** of an occurrence of a formula in the conclusion if they are connected according to the edges depicted in Figure 11.*

*A **!-thread** (resp. **?-thread**) in $\mathcal{D}$ is a maximal sequence $(A_i)_{i\in I}$ of !-formulas (resp. ?-formulas) for some downward-closed $I \subseteq \mathbb{N}$ such that A_{i+1} is the parent of A_i for all $i \in I$. A !-thread $(A_i)_{i\in I}$ is **progressing** if A_j is in the conclusion of a $\mathsf{c!p}$ for infinitely many $j \in I$. $\mathcal{D}$ is **progressing** if every infinite branch contains a progressing !-thread. We define pPLL^∞ (resp. wpPLL^∞) as the set of progressing (resp. weak-progressing) coderivations in PLL^∞.*

▶ Remark 12. Clearly, any progressing coderivation is weakly progressing too, but the converse fails (Example 13), therefore $\mathsf{pPLL}^\infty \subsetneq \mathsf{wpPLL}^\infty$. Moreover, the main branch of any nwb contains by definition a progressing !-thread of its principal formula.

▶ **Example 13.** Coderivations in Figure 7 are not weakly progressing (hence, not progressing): the rightmost branch of $\mathcal{D}_\lightning$, i.e., the branch $\{\epsilon, 2, 22, \dots\}$, and the unique branch of $\mathcal{D}_?$ are infinite and contain no $\mathsf{c!p}$-rules. In contrast, the nwb $\mathsf{c!p}_{(\underline{i_0},\dots,\underline{i_n},\dots)}$ in Example 10 is progressing by Remark 12, since its main branch is the only infinite branch. Below, a regular, weakly progressing but not progressing coderivation ($!X$ in the conclusion of $\mathsf{c!p}$ is a cut formula, so the branch $\{\epsilon, 2, 21, 212, 2121, \dots\}$ is infinite but has no progressing !-thread).



▶ **Lemma 14.** *Let Γ be a sequent. Then, $\vdash_{\mathsf{PLL}} \Gamma$ if and only if $\vdash_{\mathsf{wpPLL}^\infty} \Gamma$.*

Proof. Given $\mathcal{D} \in \mathsf{PLL}$, $\mathcal{D}^\circ \in \mathsf{PLL}^\infty$ preserves the conclusion and is progressing, hence weakly progressing (see Remark 12). Conversely, given a weakly progressing coderivation $\mathcal{D}$, we define a derivation $\mathcal{D}^f \in \mathsf{PLL}$ with the same conclusion by applying, bottom-up, the translation:

$$\left(\frac{\overset{\mathcal{D}}{\Gamma'}}{\Gamma}\,\mathsf{r}\right)^f := \frac{\overset{\mathcal{D}^f}{\Gamma'}}{\Gamma}\,\mathsf{r} \qquad \left(\frac{\overset{\mathcal{D}_1}{\Gamma_1} \quad \overset{\mathcal{D}_2}{\Gamma_2}}{\Gamma}\,\mathsf{r}\right)^f := \frac{\overset{\mathcal{D}_1{}^f}{\Gamma_1} \quad \overset{\mathcal{D}_2{}^f}{\Gamma_2}}{\Gamma}\,\mathsf{r} \qquad \left(\frac{\overset{\mathcal{D}}{\Gamma, A} \quad \overset{\mathcal{D}'}{?\Gamma, !A}}{?\Gamma, !A}\,\mathsf{c!p}\right)^f := \frac{\overset{\mathcal{D}^f}{\Gamma, A}}{?\Gamma, !A}\,\mathsf{f!p}$$

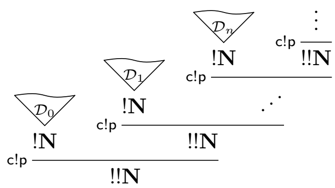
with $\mathsf{r} \neq \mathsf{c!p}$. Note that the derivation $\mathcal{D}^f$ is well-defined because $\mathcal{D}$ is weakly progressing. ◀

▶ **Corollary 15.** *The empty sequent is not provable in wpPLL^∞ (and hence in pPLL^∞).*

Proof. If the empty sequent were provable in wpPLL^∞, then there would be a cut-free derivation $\mathcal{D} \in \mathsf{PLL}$ of the empty sequent by Lemma 14 and Theorem 5, but this is impossible since cut is the only rule in PLL that could have the empty sequent in its conclusion. ◀

4.3 Recovering (weak forms of) regularity

The progressing criterion cannot capture the finiteness condition of the rule $\mathsf{ib!p}$ in the derivations in nuPLL. By means of example, consider the nwb below, which is progressing but cannot be the image of the rule $\mathsf{ib!p}$ via $(\cdot)^\bullet$ (see Figure 8) since $\{\mathcal{D}_i \mid i \in \mathbb{N}\}$ is infinite.



with $\mathcal{D}_i = \mathsf{c!p}_{(\underbrace{1,\dots,1}_{i},0,\dots)}$ for each $i \in \mathbb{N}$. (4)

To identify in pPLL^∞ the coderivations corresponding to derivations in nuPLL and in PLL via the translations $(\cdot)^\bullet$ and $(\cdot)^\circ$, respectively, we need additional conditions.

▶ **Definition 16.** *A coderivation is **weakly regular** if it has only finitely many distinct sub-coderivations whose conclusions are left premises of $\mathsf{c!p}$-rules; it is **finitely expandable** if any branch contains finitely many cut and $?\mathsf{b}$ rules. We denote by wrPLL^∞ (resp. rPLL^∞) the set of weakly regular (resp. regular) and finitely expandable coderivations in pPLL^∞.*

▶ Remark 17. Regularity implies weak regularity and the converse fails as shown in Example 18 below, so $\mathsf{rPLL}^\infty \subsetneq \mathsf{wrPLL}^\infty$. Given $\mathcal{D} \in \mathsf{PLL}^\infty$ progressing and finitely expandable, it is regular (resp. weakly regular) if and only if any nwb in $\mathcal{D}$ is periodic (resp. has finite support).

▶ **Example 18.** Coderivations $\mathcal{D}_{\not{\iota}}$ and $\mathcal{D}_{?}$ in Figure 7 are not finitely expandable, as their infinite branch has infinitely many cut or $?\mathsf{b}$, but they are weakly regular, since they have no $\mathsf{c!p}$ rules. The coderivation in (4) is not weakly regular because $\{\mathcal{D}_i \mid i \in \mathbb{N}\}$ is infinite.

An example of a weakly regular but not regular coderivation is the nwb $\mathsf{c!p}_{(\underline{i_0},\ldots,\underline{i_n},\ldots)}$ in Example 10 when the infinite sequence $(i_j)_{j\in\mathbb{N}} \in \{0,1\}^\omega$ is not periodic: $\underline{0}$ and $\underline{1}$ are the only coderivations ending in the left premise of a $\mathsf{c!p}$ rule (so the nwb is weakly regular), but there are infinitely many distinct coderivations ending in the right premise of a $\mathsf{c!p}$ rule (so the nwb is not regular). Moreover, that nwb is finitely expandable, as it contains no $?\mathsf{b}$ or cut.

The sets rPLL^∞ and wrPLL^∞ are the non-wellfounded counterparts of PLL and nuPLL, respectively. Indeed, we have the following correspondence via the translations $(\cdot)^\circ$ and $(\cdot)^\bullet$.

▶ **Proposition 19.** **1.** *If $\mathcal{D} \in \mathsf{PLL}$ (resp. $\mathcal{D} \in \mathsf{nuPLL}$) with conclusion Γ, then $\mathcal{D}^\circ \in \mathsf{rPLL}^\infty$ (resp. $\mathcal{D}^\bullet \in \mathsf{wrPLL}^\infty$) with conclusion Γ, and every $\mathsf{c!p}$ in $\mathcal{D}^\circ$ (resp. $\mathcal{D}^\bullet$) belongs to a nwb.*

2. *If $\mathcal{D}' \in \mathsf{rPLL}^\infty$ (resp. $\mathcal{D}' \in \mathsf{wrPLL}^\infty$) and every $\mathsf{c!p}$ in $\mathcal{D}'$ belongs to a nwb, then there is $\mathcal{D} \in \mathsf{PLL}$ (resp. $\mathcal{D} \in \mathsf{nuPLL}$) such that $\mathcal{D}^\circ = \mathcal{D}'$ (resp. $\mathcal{D}^\bullet = \mathcal{D}'$).*

Progressing and weak progressing coincide in finitely expandable coderivations.

▶ **Lemma 20.** *Let $\mathcal{D} \in \mathsf{PLL}^\infty$ be finitely expandable. If $\mathcal{D} \in \mathsf{wpPLL}^\infty$ then any infinite branch contains the main branch of a nwb. Moreover, $\mathcal{D} \in \mathsf{pPLL}^\infty$ if and only if $\mathcal{D} \in \mathsf{wpPLL}^\infty$.*

Proof. Let $\mathcal{D} \in \mathsf{wpPLL}^\infty$ be finitely expandable, and let $\mathcal{B}$ be an infinite branch in $\mathcal{D}$. By finite expandability there is $h \in \mathbb{N}$ such that $\mathcal{B}$ contains no conclusion of a cut or $?\mathsf{b}$ with height greater than h. Moreover, by weakly progressing there is an infinite sequence $h \leq h_0 < h_1 < \ldots < h_n < \ldots$ such that the sequent of $\mathcal{B}$ at height h_i has shape $?\Gamma_i, !A_i$. By inspecting the rules in Figure 1, each such $?\Gamma_i, !A_i$ can be the conclusion of either a $?\mathsf{w}$ or a $\mathsf{c!p}$ (with right premise $?\Gamma_i, !A_i$). So, there is a k large enough such that, for any $i \geq k$, only the latter case applies (and, in particular, $\Gamma_i = \Gamma$ and $A_i = A$ for some Γ, A). Therefore, h_k is the root of a nwb. This also shows $\mathcal{D} \in \mathsf{pPLL}^\infty$. By Remark 12, $\mathsf{pPLL}^\infty \subseteq \mathsf{wpPLL}^\infty$. ◀

By inspecting the steps in Figures 3, 5, and 10, we prove the following preservations.

▶ **Proposition 21.** *Cut elimination preserves weak-regularity, regularity and finite expandability. Therefore, if $\mathcal{D} \in \mathsf{X}$ with $\mathsf{X} \in \{\mathsf{rPLL}^\infty, \mathsf{wrPLL}^\infty\}$ and $\mathcal{D} \to_{\mathsf{cut}} \mathcal{D}'$, then also $\mathcal{D}' \in \mathsf{X}$.*

5 Continuous cut-elimination

Cut-elimination for (finitary) sequent calculi proceeds by introducing a proof rewriting strategy that stepwise decreases an appropriate termination ordering (see, e.g, [37]). Typically, these proof rewriting strategies consist on pushing upward the topmost cuts via the cut-elimination steps in order to eventually eliminate them.

A somewhat dual approach is investigated in the context of non-wellfounded proofs [6, 20]. It consists on *infinitary* proof rewriting strategies that gradually push upward the bottommost cuts. In this setting, the progressing condition is essential to guarantee *productivity*, i.e., that such proof rewriting strategies construct strictly increasing approximations of the cut-free proof, which can thus be obtained as a (well-defined) *limit*.

A major obstacle of this approach arises when the bottommost cut r is below another one r'. In this case, no cut-elimination step can be applied to r, so proof rewriting runs into an apparent stumbling block. To circumvent this problem, in [6, 20] a special cut-elimination step is introduced, which merges r and r' in a single, generalized cut rule called *multicut*.

In this section we study a continuous cut-elimination method that does not rely on multicut rules, following an alternative idea in which the notion of approximation plays an even more central rule, inspired by the topological approaches to infinite trees [9]. To this end, we assume the reader familiar with basic definitions on domain-theory (see, e.g., [4]).

5.1 Approximating coderivations

We introduce *open coderivations* to approximate coderivations. They form Scott-domains, on top of which we define *continuous cut elimination.* We also exploit them to *decompose* a finitely expandable and progressing coderivation into a *finite* approximation beneath nwbs.

▶ **Definition 22.** *We define the set of rules* $\mathsf{oPLL}^\infty := \mathsf{PLL}^\infty \cup \{\mathsf{hyp}\}$*, where* $\mathsf{hyp} := {\scriptstyle\mathsf{hyp}}\frac{}{\Gamma}$ *for any sequent* Γ.[4] *We will also refer to* oPLL^∞ *as the set of coderivations over* oPLL^∞*, which we call* ***open coderivations****. An open coderivation is* ***normal*** *if no cut-elimination step can be applied to it, that is, if one premise of each* cut *is a* hyp*. An* ***open derivation*** *is a derivation in* oPLL^∞*. We denote by* $\mathsf{oPLL}^\infty(\Gamma)$ *the set of open coderivations with conclusion* Γ*.*

▶ **Definition 23.** *Let* $\mathcal{D}$ *be an open coderivation,* $\mathcal{V} \subseteq \{1,2\}^*$ *be a set of mutually incomparable (w.r.t. the prefix order) nodes of* $\mathcal{D}$*, and* $\{\mathcal{D}'_\nu\}_{\nu\in\mathcal{V}}$ *be a set of open coderivations where* $\mathcal{D}'_\nu$ *has the same conclusion as the subderivation* $\mathcal{D}_\nu$ *of* $\mathcal{D}$*. We denote by* $\mathcal{D}\{\mathcal{D}'_\nu/\nu\}_{\nu\in\mathcal{V}} = \mathcal{D}(\mathcal{D}'_{\nu_1}/\nu_1, \dots, \mathcal{D}'_{\nu_n}/\nu_n)$*, the open coderivation obtained by replacing each* $\mathcal{D}_\nu$ *with* $\mathcal{D}'_\nu$*.*

The ***pruning*** *of* $\mathcal{D}$ *over* $\mathcal{V}$ *is the open coderivation* $\lfloor\mathcal{D}\rfloor_\mathcal{V} = \mathcal{D}\{\mathsf{hyp}/\nu\}_{\nu\in\mathcal{V}}$*. If* $\mathcal{D}$ *and* $\mathcal{D}'$ *are two open coderivations, then we say that* $\mathcal{D}$ *is an* ***approximation*** *of* $\mathcal{D}'$ *(noted* $\mathcal{D} \preceq \mathcal{D}'$*) iff* $\mathcal{D} = \lfloor\mathcal{D}'\rfloor_\mathcal{V}$ *for some* $\mathcal{V} \subseteq \{1,2\}^*$*. An approximation is* ***finite*** *if it is an open derivation.*

We denote by $\mathcal{K}(\mathcal{D})$ *the set of finite approximations of* $\mathcal{D}$*.*

Note that $\mathcal{D}$ and $\lfloor\mathcal{D}\rfloor_\mathcal{V}$ (and hence $\mathcal{D}'$ if $\mathcal{D} \preceq \mathcal{D}'$) have the same conclusion. Any open coderivation $\mathcal{D}$ is the supremum of its finite approximations, i.e. $\mathcal{D} = \bigsqcup_{\mathcal{D}'\in\mathcal{K}(\mathcal{D})} \mathcal{D}'$. Indeed:

▶ **Proposition 24.** *For any sequent* Γ*, the poset* $(\mathsf{oPLL}^\infty(\Gamma), \preceq)$ *is a Scott-domain with least element the open derivation* hyp *and with maximal elements the coderivations (in* PLL^∞*) with conclusion* Γ*. The compact elements are precisely the open derivations in* $\mathsf{oPLL}^\infty(\Gamma)$*.*

Cut-elimination steps essentially do not increase the size of open derivations, hence:

▶ **Lemma 25.** $\rightarrow_{\mathsf{cut}}$ *over open derivations is strongly normalizing and confluent.*

Progressing and finitely expandable coderivations can be approximated in a canonical way. Indeed, by Lemma 20 we have:

▶ **Proposition 26.** *If* $\mathcal{D} \in \mathsf{pPLL}^\infty$ *is finitely expandable, then there is a prebar* $\mathcal{V} \subseteq \{1,2\}^*$ *of* $\mathcal{D}$ *such that each* $v \in \mathcal{V}$ *is the root of a* nwb *in* $\mathcal{D}$*.*

▶ **Definition 27.** *Let* $\mathcal{D} \in \mathsf{pPLL}^\infty$ *be finitely expandable. The* ***decomposition prebar*** *of* $\mathcal{D}$ *is the minimal prebar* $\mathcal{V}$ *of* $\mathcal{D}$ *such that, for all* $\nu \in \mathcal{V}$*,* $\mathcal{D}_\nu$ *is a* nwb*. We denote with* $\mathsf{border}(\mathcal{D})$ *such a bar and we set* $\mathsf{base}(\mathcal{D}) := \lfloor\mathcal{D}\rfloor_{\mathsf{border}(\mathcal{D})}$*.*

Note that, by weak König lemma, in the above definition $\mathsf{border}(\mathcal{D})$ is finite and $\mathsf{base}(\mathcal{D})$ is a finite approximation of $\mathcal{D}$.

---

[4] Previously introduced notions and definitions on coderivations extend to open coderivations in the obvious way, e.g. the global conditions of Definitions 11 and 16 and the cut-elimination relation $\rightarrow_{\mathsf{cut}}$.

5.2 Domain-theoretic approach to continuous cut-elimination

In this subsection we define *maximal and continuous infinitary cut-elimination strategies* (mc-ices), special rewriting strategies that stepwise generate ω-chains approximating the cut-free version of an open coderivation. In other words, a mc-ices computes a (Scott-)continuous function from open coderivations to cut-free open coderivations. Then, we introduce the *height-by-height* mc-ices, a notable example of mc-ices that will be used for our results, and we show that any two mc-icess compute the same (Scott-)continuous function.

In what follows, σ denotes a countable sequence of coderivations, and $\sigma(i)$ denotes the $(i+1)$-th coderivation in σ. We denote the length of a sequence σ by $\ell(\sigma) \leq \omega$.

▶ **Definition 28.** *An* ***infinitary cut elimination strategy*** *(or* ices *for short) is a family* $\sigma = \{\sigma_\mathcal{D}\}_{\mathcal{D}\in\mathsf{oPLL}^\infty}$ *where, for all* $\mathcal{D} \in \mathsf{oPLL}^\infty$*,* $\sigma_\mathcal{D}$ *is a sequence of open coderivations such that* $\sigma_\mathcal{D}(0) = \mathcal{D}$ *and* $\sigma_\mathcal{D}(i) \to_{\mathsf{cut}} \sigma_\mathcal{D}(i+1)$ *for all* $0 \leq i < \ell(\sigma_\mathcal{D})$*. Given an* ices σ*, we define the function* $f_\sigma \colon \mathsf{oPLL}^\infty(\Gamma) \to \mathsf{oPLL}^\infty(\Gamma)$ *as* $f_\sigma(\mathcal{D}) \coloneqq \bigsqcup_{i=0}^{\ell(\sigma_\mathcal{D})} \mathsf{cf}(\sigma_\mathcal{D}(i))$ *where* $\mathsf{cf}(\mathcal{D}_i)$ *is the greatest cut-free approximation of* $\mathcal{D}_i$ *(w.r.t.* $\preceq$*).* [5]*. An* ices σ *is a* mc-ices *if it is:*

- ***maximal:*** $\sigma_\mathcal{D}(\ell(\sigma_\mathcal{D}))$ *is normal for any open derivation* $\mathcal{D}$ *(*$\ell(\sigma_\mathcal{D}) < \omega$ *by Lemma 25);*
- ***(Scott)-continuous:*** f_σ *is Scott-continuous.*

Roughly, a maximal ices is an ices that applies cut-elimination steps to open derivations (i.e., finite approximations) until a normal (possibly cut-free) open derivation is reached. The following property states that all mc-icess induce the same continuous function, an easy consequence of Lemma 25 and continuity.

▶ **Proposition 29.** *If* σ *and* σ' *are two* mc-icess*, then* $f_\sigma = f_{\sigma'}$*.*

Therefore, we define a specific mc-ices we use in our proofs, where cut-elimination steps are applied in a deterministic way to the minimal reducible cut-rules.

▶ **Definition 30.** *The* ***height-by-height*** ices *is defined as* $\sigma^\infty = \{\sigma^\infty_\mathcal{D}\}_{\mathcal{D}\in\mathsf{oPLL}^\infty}$ *where* $\sigma^\infty_\mathcal{D}(0) = \mathcal{D}$ *for each* $\mathcal{D} \in \mathsf{oPLL}^\infty$*, and* $\sigma^\infty_\mathcal{D}(i+1)$ *is the open coderivation obtained by applying a cut-elimination step to the leftmost reducible* cut*-rule with minimal height in* $\sigma^\infty_\mathcal{D}(i)$*.*

▶ **Proposition 31.** *The* ices σ^∞ *is a* mc-ices*.*

Proof. By Lemma 25, any open derivation $\mathcal{D}$ normalizes in $n_\mathcal{D} \in \mathbb{N}$ steps; so, if $\mathcal{D}$ is an open derivation, $\ell(\sigma^\infty_\mathcal{D}) = n_\mathcal{D}$ with $\sigma^\infty_\mathcal{D}(n_\mathcal{D})$ normal by definition of σ^∞. Hence, σ^∞ is maximal.

Since $\sigma^\infty_\mathcal{D}(i)$ is defined by applying a finite number of cut-eliminations steps to $\mathcal{D}$, then there is $\mathcal{D}' \in \mathcal{K}(\mathcal{D})$ such that $\sigma^\infty_\mathcal{D}(i) = \sigma^\infty_{\mathcal{D}'}(i)$, and therefore $\mathsf{cf}(\sigma^\infty_\mathcal{D}(i)) = \mathsf{cf}(\sigma^\infty_{\mathcal{D}'}(i)) \preceq f_{\sigma^\infty}(\mathcal{D}')$ for all $0 \leq i \leq \ell(\sigma^\infty)$. Thus $f_{\sigma^\infty}(\mathcal{D}) \preceq \bigsqcup_{\mathcal{D}'\in\mathcal{K}(\mathcal{D})} f_{\sigma^\infty}(\mathcal{D}')$. Moreover $\bigsqcup_{\mathcal{D}'\in\mathcal{K}(\mathcal{D})} f_{\sigma^\infty}(\mathcal{D}') \preceq f_{\sigma^\infty}(\mathcal{D})$ because σ^∞ is monotone by construction. We conclude by showing that f_{σ^∞} is continuous. ◀

In order to prove our results, we introduce the notion of chain of cut-rules, which allows us to keep track of the dynamic of cut-elimination steps during infinitary rewriting. Note that the definition of cut-chain is the analogue of the *multi-cut reduction sequences* from [6].

▶ **Definition 32** (Chains). *Let* $\sigma = \{\sigma_\mathcal{D}\}_{\mathcal{D}\in\mathsf{oPLL}^\infty}$ *be an* ices*. We write* $\mathsf{r}_i \mapsto_\sigma \mathsf{r}_{i+1}$ *if* r_{i+1} *is a* cut*-rule in* $\sigma_\mathcal{D}(i+1)$ *produced by applying a cut-elimination step to the* cut*-rule* r_i *in* $\sigma_\mathcal{D}(i)$*.*

A cut***-chain*** *in* $\sigma_\mathcal{D}$ *is a sequence* $(\mathsf{r}_i)_{i<\alpha}$ *of* cut *rules with* $\alpha \leq \ell(\sigma_\mathcal{D})$*, such that* r_i *a rule in* $\sigma_\mathcal{D}(i)$*, and either* $\mathsf{r}_i = \mathsf{r}_{i+1}$ *or* $\mathsf{r}_i \mapsto_\sigma \mathsf{r}_{i+1}$*. We say that a chain* ***starts*** *at* r_0 *and that each* r_{i+1} *is a* ***descendant*** *of* r_i*.*

---

[5] f_σ is well-defined, as $(\mathsf{cf}(\sigma_\mathcal{D}(i)))_{0\leq i<\ell(\sigma_\mathcal{D})}$ is an ω-chain in oPLL^∞ and so its sup exists by Proposition 24.

We conclude this section by providing the sketch of proof for the continuous cut-elimination theorem, the main contribution of this paper, establishing a productivity result and showing that continuous cut-elimination preserves all global conditions.

▶ **Theorem 33** (Continuous Cut-Elimination)**.**
1. *If* $\mathcal{D} \in \mathsf{pPLL}^\infty$*, then so is* $f_{\sigma^\infty}(\mathcal{D})$*.*
2. *If* $\mathcal{D} \in \mathsf{wrPLL}^\infty$ *(resp.* $\mathcal{D} \in \mathsf{rPLL}^\infty$*), then so is* $f_{\sigma^\infty}(\mathcal{D})$*.*

Sketch of the proof.
1. We have to prove that $f_{\sigma^\infty}(\mathcal{D})$ is hyp-free (i.e., *productivity*) and that any of its infinite branches contains a progressing !-thread. To facilitate our argument, leveraging on symmetry of the cut rules, we assume w.l.o.g. that !-formulas can only be cut in the left-hand premise of a cut-rule.
   We first show that, for any infinite cut-chain $(\mathsf{r}_i)_i$ there is a descendant r_i in $\sigma^\infty_{\mathcal{D}}(i)$ whose right premise is the conclusion of a c!p-rule. Since the cut-chain is infinite and since cut-elimination steps preserve progressing condition (Proposition 21), there is a $i_0 \geq 0$ such that all descendants of r_{i_0} in $(\mathsf{r}_i)_i$ cut formulas from the same ?-thread along an infinite branch $\mathcal{B}$. Moreover, since $\mathcal{B}$ has infinitely many c!p rules by progressing condition, every cut-rule with a premise in $\mathcal{B}$ is eventually reducible, so that there are infinitely many $i \geq i_0$ such that $\mathsf{r}_i \mapsto_\sigma \mathsf{r}_{i+1}$. Therefore, if the right-premise of r_i did not eventually become conclusion of a c!p-rule we could identify an infinite branch of $\mathcal{D}$ that has no progressing !-thread.
   Now, let $\mathcal{B}^*$ be a branch of $f_{\sigma^\infty}(\mathcal{D})$. If $\mathcal{B}^*$ has been obtained from $\mathcal{D}$ after finitely many cut-elimination steps then it is clearly hyp-free and, if infinite, it has a progressing !-thread (Proposition 21). Otherwise, $\mathcal{B}^*$ has been constructed by an infinite cut-chain $(\mathsf{r}_i)_i$ with minimal height. By repeatedly applying the above property, we have that there are infinitely many r_i whose rightmost premise is the conclusion of a c!p-rule r^*, and such that $\mathsf{r}_i \mapsto_\sigma \mathsf{r}_{i+1}$ is a step permuting r^* downward (since r^* it is on the left premise of r_i, its principal !-formula cannot be a cut-formula of r_i by assumption). This means that $\mathcal{B}^*$ contains infinitely many c!p rules, and so it is hyp-free. To prove that there is a progressing !-thread in $\mathcal{B}^*$ it suffices to show that infinitely many c!p rules of $\mathcal{B}^*$ are descendants of the same branch $\mathcal{B}$ of $\mathcal{D}$, as the existence of a progressing !-thread of $\mathcal{B}^*$ would follow directly from the existence of a (unique) progressing !-thread of $\mathcal{B}$.
2. Akin to linear logic, we define the *depth* of a coderivation as the maximal number of nested nwb*s*, and we prove that the depth of (weakly) regular coderivations is always finite. Moreover, by Proposition 26, a progressing and finitely expandable coderivation $\mathcal{D}$ can be decomposed to a nwb-free finite approximation base($\mathcal{D}$) and a series of nwb*s* whose calls have smaller depth. Using this property we define, by induction on the depth of $\mathcal{D}$, a maximal and *transfinite* ices reducing the calls of the nwb*s* one by one. The proof of preservation of (weak) regularity under cut-elimination for such an ices follows by construction since, by Remark 17, if we reduce a nwb with finite support (resp. a periodic nwb) via our transfinite ices, then we obtain in the limit a cut-free nwb with finite support (resp. a periodic nwb). We then show that this transfinite ices can be compressed to a (ω-long) mc-ices using methods studied in [36, 33], and we conclude the proof by Item 1 and by the fact that $f_{\sigma^\infty}(\mathcal{D})$ is finitely expandable and (weakly) regular for such a mc-ices. ◀

By definition (as the sup of cut-free open coderivations) $f_{\sigma^\infty}(\mathcal{D})$ is cut-free. Each item of Theorem 33 says in particular that $f_{\sigma^\infty}(\mathcal{D})$ is hyp-free, which means that $f_{\sigma^\infty}(\mathcal{D})$ is obtained by eliminating *all* the cuts in $\mathcal{D}$. This may not be the case if $\mathcal{D}$ does not fulfill any of the

$$\left[\!\left[\ \mathsf{ax}\,\frac{}{A, A^{\perp}}\ \right]\!\right]_n = \{ (x, x) \mid x \in [\![A]\!] \} \qquad \left[\!\left[\ \mathsf{cut}\,\frac{\begin{array}{c}\mathcal{D}'\\ \Gamma, A\end{array} \quad \begin{array}{c}\mathcal{D}''\\ \Delta, A^{\perp}\end{array}}{\Gamma, \Delta}\ \right]\!\right]_n = \left\{ (\vec{x}, \vec{y}) \;\middle|\; \exists z \in [\![A]\!] \text{ s.t. } \begin{array}{c} (\vec{x}, z) \in [\![\mathcal{D}']\!]_{n-1} \\ \text{and} \\ (z, \vec{y}) \in [\![\mathcal{D}'']\!]_{n-1} \end{array} \right\}$$

$$\left[\!\left[\ \bot\,\frac{\begin{array}{c}\mathcal{D}'\\ \Gamma\end{array}}{\Gamma, \bot}\ \right]\!\right]_n = \{ (\vec{x}, *) \mid \vec{x} \in [\![\mathcal{D}']\!]_{n-1} \} \qquad \left[\!\left[\ \parr\,\frac{\begin{array}{c}\mathcal{D}'\\ \Gamma, A, B\end{array}}{\Gamma, A \parr B}\ \right]\!\right]_n = \{ (\vec{x}, (y, z)) \mid (\vec{x}, y, z) \in [\![\mathcal{D}']\!]_{n-1} \}$$

$$\left[\!\left[\ 1\,\frac{}{1}\ \right]\!\right]_n = \{*\} \qquad \left[\!\left[\ \otimes\,\frac{\begin{array}{c}\mathcal{D}'\\ \Gamma, A\end{array} \quad \begin{array}{c}\mathcal{D}''\\ \Delta, B\end{array}}{\Gamma, \Delta, A \otimes B}\ \right]\!\right]_n = \left\{ (\vec{x}, \vec{y}, (x, y)) \;\middle|\; \begin{array}{c} (\vec{x}, x) \in [\![\mathcal{D}']\!]_{n-1} \\ \text{and} \\ (\vec{y}, y) \in [\![\mathcal{D}'']\!]_{n-1} \end{array} \right\} \qquad \left[\!\left[\ \mathsf{hyp}\,\frac{}{\Gamma}\ \right]\!\right]_n = \varnothing$$

$$\left[\!\left[\ ?\mathsf{w}\,\frac{\begin{array}{c}\mathcal{D}'\\ \Gamma\end{array}}{\Gamma, ?A}\ \right]\!\right]_n = \{ (\vec{x}, [\,]) \mid \vec{x} \in [\![\mathcal{D}']\!]_{n-1} \} \qquad \left[\!\left[\ ?\mathsf{b}\,\frac{\begin{array}{c}\mathcal{D}'\\ \Gamma, A, ?A\end{array}}{\Gamma, ?A}\ \right]\!\right]_n = \{ (\vec{x}, [y] + \mu) \mid (\vec{x}, y, \mu) \in [\![\mathcal{D}']\!]_{n-1} \}$$

$$\left[\!\left[\ \mathsf{c!p}\,\frac{\begin{array}{c}\mathcal{D}'\\ \Gamma, A\end{array} \quad \begin{array}{c}\mathcal{D}''\\ ?\Gamma, !A\end{array}}{?\Gamma, !A}\ \right]\!\right]_n = \left\{ (\vec{[\,]}, [\,]) \right\} \cup \left\{ ([x_1] + \mu_1, \dots, [x_k] + \mu_k, [x] + \mu) \;\middle|\; \begin{array}{c} (x_1, \dots, x_k, x) \in [\![\mathcal{D}']\!]_{n-1} \\ \text{and} \\ (\mu_1, \dots, \mu_k, \mu) \in [\![\mathcal{D}'']\!]_{n-1} \end{array} \right\}$$

Figure 12 Inductive definition of the set $[\![\mathcal{D}]\!]_n$, for $n > 0$.

global conditions in the hypotheses of Theorem 33: $f_{\sigma^\infty}(\mathcal{D})$ is still cut-free but may contain some "truncating" hyp that "prevented" eliminating some cut in $\mathcal{D}$, as in the example below.

▶ **Example 34.** For any finite approximation $\mathcal{D}$ of the (non-weakly progressing, non-finitely expandable) open coderivation $\mathcal{D}_{\lightning}$, we have $f_{\sigma^\infty}(\mathcal{D}) = \mathsf{hyp}$, so $f_{\sigma^\infty}(\mathcal{D}_{\lightning}) = \mathsf{hyp}$ by continuity.

6 Relational semantics for non-wellfounded proofs

Here we define a denotational model for oPLL^∞ based on *relational semantics*, which interprets an open coderivation as the union of the interpretations of its finite approximations, as in [17]. We show that relational semantics is sound for oPLL^∞, but not for its extension with digging.

Relational semantics interprets exponential by finite multisets, denoted by brackets, e.g., $[x_1, \dots, x_n]$; $+$ denotes the *multiset union*, and $\mathcal{M}_f(X)$ denotes the set of finite multisets over a set X. To correctly define the semantics of a coderivation, we need to see sequents as *finite sequences* of formulas (taking their order into account), which means that we have to add an *exchange* rule to oPLL^∞ to swap the order of two consecutive formulas in a sequent.

▶ **Definition 35.** *We associate with each formula A a* **set** *$[\![A]\!]$ defined as follows:*

$$[\![X]\!] \coloneqq D_X \quad [\![1]\!] \coloneqq \{*\} \quad [\![A \otimes B]\!] \coloneqq [\![A]\!] \times [\![B]\!] \quad [\![!A]\!] \coloneqq \mathcal{M}_f([\![A]\!]) \quad [\![A^\perp]\!] \coloneqq [\![A]\!]$$

where D_X is an arbitrary set. For a sequent $\Gamma = A_1, \dots, A_n$, we set $[\![\Gamma]\!] \coloneqq [\![A_1 \parr \cdots \parr A_n]\!]$.

Given $\mathcal{D} \in \mathsf{PLL} \cup \mathsf{oPLL}^\infty$ with conclusion Γ, we set $[\![\mathcal{D}]\!] \coloneqq \bigcup_{n \geq 0} [\![\mathcal{D}]\!]_n \subseteq [\![\Gamma]\!]$, where $[\![\mathcal{D}]\!]_0 = \varnothing$ and, for all $i \in \mathbb{N} \setminus \{0\}$, $[\![\mathcal{D}]\!]_i$ is defined inductively according to Figure 12.

▶ **Example 36.** For the coderivations $\mathcal{D}_{\lightning}$ and $\mathcal{D}_?$ in Figure 7, $[\![\mathcal{D}_{\lightning}]\!] = [\![\mathcal{D}_?]\!] = \varnothing$. For the derivations $\underline{0}$ and $\underline{1}$ in Figure 2, $[\![\underline{0}]\!] = \{([\,], (x, x)) \mid x \in D_X\}$ and $[\![\underline{1}]\!] = \{([(x, y)], (x, y)) \mid x, y \in D_X\}$. For the coderivation $\mathsf{c!p}_{(\underline{i_0}, \dots, \underline{i_n}, \dots)}$ in Example 10 (with $i_j \in \{0, 1\}$ for all $j \in \mathbb{N}$), $[\![\mathsf{c!p}_{(\underline{i_0}, \dots, \underline{i_n}, \dots)}]\!] = \{[\,]\} \cup \left\{[x_{i_0}, \dots, x_{i_n}] \in \mathcal{M}_f([\![\mathbf{N}]\!]) \mid n \in \mathbb{N},\ x_{i_j} \in [\![\underline{i_j}]\!]\ \forall 0 \leq j \leq n\right\}$. For the derivation $\underline{n}$ in Example 10 (for any $n \in \mathbb{N}$), $[\![\underline{n}]\!] = \{([(x_1, x_2), \dots, (x_n, x_{n+1})], (x_1, x_{n+1})) \mid x_1, \dots, x_{n+1} \in D_X\}$. Note that $[\![\underline{n}]\!] \cap [\![\underline{m}]\!] = \varnothing$ for all $n, m \in \mathbb{N}$ such that $n \neq m$, and that

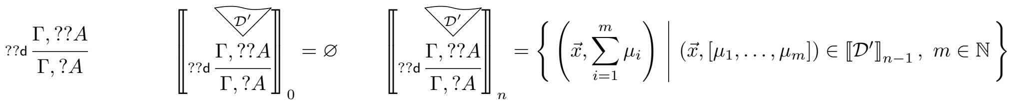


Figure 13 The rule ??d and its interpretation in the relational semantics ($n > 0$).

$[\![n]\!]$ is stable under permutations of the rules ?w, ?b and $\otimes$ in $\underline{n}$ (that is, if $\mathcal{D}$ is obtained from $\underline{n}$ by permuting the rules ?w, ?b or $\otimes$, then $[\![\mathcal{D}]\!] = [\![\underline{n}]\!]$).

By inspecting the cut-elimination steps and by continuity, we can prove the soundness of relational semantics with respect to cut-elimination (Theorem 38), thanks to the fact the interpretation of a coderivation is the union the interpretations of its finite approximation.

▶ **Lemma 37.** *Let $\mathcal{D} \in \mathsf{oPLL}^\infty$. Then, $[\![\mathcal{D}]\!] = [\![\bigsqcup_{\mathcal{D}' \in \mathcal{K}(\mathcal{D})} \mathcal{D}']\!] = \bigcup_{\mathcal{D}' \in \mathcal{K}(\mathcal{D})} [\![\mathcal{D}']\!]$.*

▶ **Theorem 38** (Soundness)**.** **1.** *Let $\mathcal{D} \in \mathsf{oPLL}^\infty$. If $\mathcal{D} \to_{\mathsf{cut}} \mathcal{D}'$, then $[\![\mathcal{D}]\!] = [\![\mathcal{D}']\!]$.*
2. *Let $\mathcal{D} \in \mathsf{oPLL}^\infty$. If σ is a* mc-ices, *then $[\![\mathcal{D}]\!] = [\![f_\sigma(\mathcal{D})]\!]$.*

By Theorem 38 and since cut-free coderivations have non-empty semantics, we have:

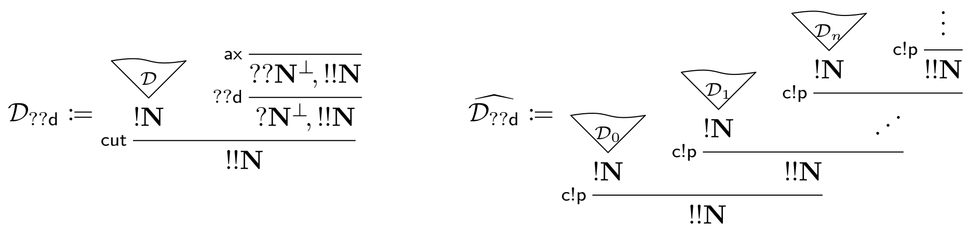
▶ **Corollary 39.** *Let $\mathcal{D} \in \mathsf{wpPLL}^\infty$. Then $[\![\mathcal{D}]\!] \neq \varnothing$.*

We define the set of rules $\mathsf{MELL}^\infty := \mathsf{PLL}^\infty \cup \{??\mathsf{d}\}$ where the rule ??d (**digging**) is defined in Figure 13. We also denote by MELL^∞ the set of coderivations over the rules in MELL^∞. Relational semantics is naturally extended to MELL^∞ as shown in Figure 13.

The proof system MELL^∞ can be seen as a non-wellfounded version of MELL. We show that, as opposed to several fragments of PLL^∞, in any good fragment of MELL^∞ with digging, cut-elimination cannot reduce to cut-free coderivations *and* preserve both the progressing condition and relational semantics.

▶ **Theorem 40.** *Let $\mathsf{X} \subseteq \mathsf{MELL}^\infty$ contain non-wellfounded coderivations with ??d. Let $\to_{\mathsf{cut}+}$ be a cut-elimination relation on X preserving the progressing condition, containing $\to_{\mathsf{cut}}$ in Figures 3, 5, and 10 and reducing every coderivation in X to a cut-free one. Then, $\to_{\mathsf{cut}+}$ does not preserve relational semantics.*

Proof. Consider the coderivations $\mathcal{D}_{??\mathsf{d}}$ and $\widehat{\mathcal{D}_{??\mathsf{d}}}$ below, where $\mathcal{D} = \mathsf{c!p}_{(\underline{0},\underline{1},\underline{0},\underline{1},\dots)}$ and, for all $i \in \mathbb{N}$, $\mathcal{D}_i \in \{\mathsf{c!p}_{(\underline{k_0^i},\dots,\underline{k_n^i},\dots)} \mid k_j^i \in \mathbb{N} \text{ for all } j \in \mathbb{N}\}$ ($\underline{n}$ is defined in Example 10 for all $n \in \mathbb{N}$).



Coderivations $\widehat{\mathcal{D}_{??\mathsf{d}}}$ are the only cut-free and progressing ones with conclusion $!!\mathbf{N}$. Indeed, any cut-free coderivation of $!!\mathbf{N}$ or $!\mathbf{N}$ must end with a c!p, and the only cut-free and progressing coderivations of $\mathbf{N}$ are the derivations of the form $\underline{n}$ for any $n \in \mathbb{N}$, up to permutations of the rules ?w, ?b and $\otimes$ (other cut-free coderivations of $\mathbf{N}$ exist, but they have an infinite branch containing infinitely many ?b rules and no c!p rules, hence they are not progressing). Therefore, for whatever definition of the cut-elimination steps concerning ??d that preserves the progressing condition, necessarily $\mathcal{D}_{??\mathsf{d}}$ will reduce to $\widehat{\mathcal{D}_{??\mathsf{d}}}$, since $\mathcal{D}_{??\mathsf{d}}$ is progressing.

We show that $[\![\widehat{\mathcal{D}_{??\mathsf{d}}}]\!] \not\subseteq [\![\mathcal{D}_{??\mathsf{d}}]\!]$. First, it can be easily shown that if, in one of the $\mathcal{D}_i = \mathsf{c!p}_{(\underline{k_0^i},\dots,\underline{k_n^i},\dots)}$ in $\widehat{\mathcal{D}_{??\mathsf{d}}}$, one of the k_j^i is different from 0 or 1, then there is $x \in [\![\widehat{\mathcal{D}_{??\mathsf{d}}}]\!] \smallsetminus [\![\mathcal{D}_{??\mathsf{d}}]\!]$ (this basically follows from the fact that $[\![\underline{n}]\!] \cap [\![\underline{m}]\!] = \varnothing$ for all $n, m \in \mathbb{N}$ such that $n \neq m$, see Example 36). Let us now suppose that in $\widehat{\mathcal{D}_{??\mathsf{d}}}$, for all $i \in \mathbb{N}$, $\mathcal{D}_i = \mathsf{c!p}_{(\underline{k_0^i},\dots,\underline{k_n^i},\dots)}$ with $k_j^i \in \{0,1\}$ for all $j \in \mathbb{N}$. Let $\hat{0}$ and $\hat{1}$ be any element of $[\![\underline{0}]\!]$ and $[\![\underline{1}]\!]$, respectively (see Example 36). Note that $\hat{0} \neq \hat{1}$. It is easy to verify that $[[\hat{0}],[\hat{0}]],[[\hat{1}],[\hat{1}]] \notin [\![\mathcal{D}_{??\mathsf{d}}]\!]$, since $[\hat{0},\hat{0}],[\hat{1},\hat{1}] \notin [\![\mathcal{D}]\!]$ (see Example 36). Concerning $[\![\widehat{\mathcal{D}_{??\mathsf{d}}}]\!]$, notice that, since $k_0^0, k_0^1, k_0^2 \in \{0,1\}$, either $k_0^0 = k_0^1$ or $k_0^1 = k_0^2$ or $k_0^2 = k_0^0$. In the first case, we have $[[k_0^0],[k_0^1]] \in [\![\widehat{\mathcal{D}_{??\mathsf{d}}}]\!]$, in the second case we have $[[k_0^1],[k_0^2]] \in [\![\widehat{\mathcal{D}_{??\mathsf{d}}}]\!]$, and in the last case we have $[[k_0^2],[k_0^0]] \in [\![\widehat{\mathcal{D}_{??\mathsf{d}}}]\!]$. ◀

7 Conclusion and future work

For future research, we envisage extending our contributions in many directions. First, our notion of finite approximation seems intimately related with that of Taylor expansion from *differential linear logic* (DiLL) [18, 19, 15], where the rule hyp (quite like the rule 0 from DiLL, [3]) serves to model approximations of *boxes*. This connection with Taylor expansions becomes even more apparent in Mazza's original systems for parsimonious logic [26, 27], which comprise co-absorption and co-weakening rules typical of DiLL. These considerations deserve further investigations. Secondly, building on a series of recent works in *Cyclic Implicit Complexity*, i.e., implicit computational complexity in the setting of circular and non-wellfounded proof theory [11, 10], we are currently working on second-order extensions of wrPLL^∞ and rPLL^∞ to characterize the complexity classes $\mathbf{P}/\mathsf{poly}$ and $\mathbf{P}$ (see [1]). These results would reformulate in a non-wellfounded setting the characterization of $\mathbf{P}/\mathsf{poly}$ presented in [27].

References

1. Matteo Acclavio, Gianluca Curzi, and Giulio Guerrieri. Non-uniform polynomial time via non-wellfounded parsimonious proofs. Unpublished. URL: `http://gianlucacurzi.com/Non-uniform-polynomial-time-via-non-wellfounded-parsimonious-proofs.pdf`.
2. Matteo Acclavio, Gianluca Curzi, and Giulio Guerrieri. Infinitary cut-elimination via finite approximations. *CoRR*, abs/2308.07789, 2023. URL: `https://doi.org/10.48550/arXiv.2308.07789`, `arXiv:2308.07789`, `doi:10.48550/ARXIV.2308.07789`.
3. Matteo Acclavio and Giulio Guerrieri. A deep inference system for differential linear logic. In *Proceedings Second Joint International Workshop on Linearity & Trends in Linear Logic and Applications, Linearity&TLLA@IJCAR-FSCD 2020*, volume 353 of *EPTCS*, pages 26–49, 2020. `doi:10.4204/EPTCS.353.2`.
4. Roberto M. Amadio and Pierre-Louis Curien. *Domains and lambda-calculi*, volume 46 of *Cambridge tracts in theoretical computer science*. Cambridge University Press, 1998.
5. Sanjeev Arora and Boaz Barak. *Computational Complexity: A Modern Approach*. Cambridge University Press, 2009. `doi:10.1017/CBO9780511804090`.
6. David Baelde, Amina Doumane, and Alexis Saurin. Infinitary proof theory: the multiplicative additive case. In Jean-Marc Talbot and Laurent Regnier, editors, *25th EACSL Annual Conference on Computer Science Logic, CSL 2016, August 29 - September 1, 2016, Marseille, France*, volume 62 of *LIPIcs*, pages 42:1–42:17. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2016. `doi:10.4230/LIPIcs.CSL.2016.42`.
7. David Baelde and Dale Miller. Least and greatest fixed points in linear logic. In Nachum Dershowitz and Andrei Voronkov, editors, *Logic for Programming, Artificial Intelligence, and Reasoning, 14th International Conference, LPAR 2007, Yerevan, Armenia, October 15-19, 2007, Proceedings*, volume 4790 of *Lecture Notes in Computer Science*, pages 92–106. Springer, 2007. `doi:10.1007/978-3-540-75560-9\_9`.

8 James Brotherston and Alex Simpson. Sequent calculi for induction and infinite descent. *Journal of Logic and Computation*, 21(6):1177–1216, 2011.

9 Bruno Courcelle. Fundamental properties of infinite trees. *Theoretical Computer Science*, 25(2):95–169, 1983. URL: `https://www.sciencedirect.com/science/article/pii/0304397583900592`, `doi:https://doi.org/10.1016/0304-3975(83)90059-2`.

10 Gianluca Curzi and Anupam Das. Cyclic implicit complexity. In *Proceedings of the 37th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '22, New York, NY, USA, 2022. Association for Computing Machinery. `doi:10.1145/3531130.3533340`.

11 Gianluca Curzi and Anupam Das. Non-uniform complexity via non-wellfounded proofs. In *31st EACSL Annual Conference on Computer Science Logic, CSL 2023*, volume 252 of *LIPIcs*, pages 16:1–16:18. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2023. `doi:10.4230/LIPIcs.CSL.2023.16`.

12 Vincent Danos and Jean-Baptiste Joinet. Linear logic and elementary time. *Inf. Comput.*, 183(1):123–137, 2003. `doi:10.1016/S0890-5401(03)00010-5`.

13 Anupam Das. On the logical strength of confluence and normalisation for cyclic proofs. In *6th International Conference on Formal Structures for Computation and Deduction, FSCD 2021*, volume 195 of *LIPIcs*, pages 29:1–29:23. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021. `doi:10.4230/LIPIcs.FSCD.2021.29`.

14 Christian Dax, Martin Hofmann, and Martin Lange. A proof system for the linear time μ-calculus. In *International Conference on Foundations of Software Technology and Theoretical Computer Science*, pages 273–284. Springer, 2006.

15 Thomas Ehrhard. An introduction to differential linear logic: proof-nets, models and antiderivatives. *Math. Struct. Comput. Sci.*, 28(7):995–1060, 2018. `doi:10.1017/S0960129516000372`.

16 Thomas Ehrhard and Farzad Jafar-Rahmani. On the denotational semantics of linear logic with least and greatest fixed points of formulas. *CoRR*, abs/1906.05593, 2019. URL: `http://arxiv.org/abs/1906.05593`, `arXiv:1906.05593`.

17 Thomas Ehrhard, Farzad Jafar-Rahmani, and Alexis Saurin. On relation between totality semantic and syntactic validity. In *5th International Workshop on Trends in Linear Logic and Applications (TLLA 2021)*, June 2021. URL: `https://hal-lirmm.ccsd.cnrs.fr/lirmm-03271408`.

18 Thomas Ehrhard and Laurent Regnier. Differential interaction nets. *Theor. Comput. Sci.*, 364(2):166–195, 2006. `doi:10.1016/J.TCS.2006.08.003`.

19 Thomas Ehrhard and Laurent Regnier. Uniformity and the Taylor expansion of ordinary lambda-terms. *Theor. Comput. Sci.*, 403(2-3):347–372, 2008. `doi:10.1016/J.TCS.2008.06.001`.

20 Jérôme Fortier and Luigi Santocanale. Cuts for circular proofs: semantics and cut-elimination. In *Computer Science Logic 2013, CSL 2013*, volume 23 of *LIPIcs*, pages 248–262. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013. `doi:10.4230/LIPIcs.CSL.2013.248`.

21 Jean-Yves Girard. Linear logic. *Theoretical Computer Science*, 50(1):1–101, 1987. `doi:10.1016/0304-3975(87)90045-4`.

22 Jean-Yves Girard. Light linear logic. *Information and Computation*, 143(2):175–204, 1998. `doi:10.1006/inco.1998.2700`.

23 Denis Kuperberg, Laureline Pinault, and Damien Pous. Cyclic proofs, system T, and the power of contraction. *Proc. ACM Program. Lang.*, 5(POPL):1–28, 2021. `doi:10.1145/3434282`.

24 Yves Lafont. Soft linear logic and polynomial time. *Theoretical Computer Science*, 318(1):163–180, 2004. Implicit Computational Complexity. `doi:10.1016/j.tcs.2003.10.018`.

25 Damiano Mazza. Non-uniform polytime computation in the infinitary affine lambda-calculus. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, volume 8573 of *Lecture Notes in Computer Science*, pages 305–317. Springer, 2014. `doi:10.1007/978-3-662-43951-7\_26`.

26 Damiano Mazza. Simple parsimonious types and logarithmic space. In *24th EACSL Annual Conference on Computer Science Logic, CSL 2015*, volume 41 of *LIPIcs*, pages 24–40. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2015. doi:10.4230/LIPIcs.CSL.2015.24.

27 Damiano Mazza and Kazushige Terui. Parsimonious types and non-uniform computation. In Magnús M. Halldórsson, Kazuo Iwama, Naoki Kobayashi, and Bettina Speckmann, editors, *Automata, Languages, and Programming - 42nd International Colloquium, ICALP 2015, Kyoto, Japan, July 6-10, 2015, Proceedings, Part II*, volume 9135 of *Lecture Notes in Computer Science*, pages 350–361. Springer, 2015. doi:10.1007/978-3-662-47666-6_28.

28 Vivek Nigam and Dale Miller. Algorithmic specifications in linear logic with subexponentials. In António Porto and Francisco Javier López-Fraguas, editors, *Proceedings of the 11th International ACM SIGPLAN Conference on Principles and Practice of Declarative Programming*, pages 129–140. ACM, 2009. doi:10.1145/1599410.1599427.

29 Damian Niwiński and Igor Walukiewicz. Games for the μ-calculus. *Theoretical Computer Science*, 163(1-2):99–116, 1996. doi:10.1016/0304-3975(95)00136-0.

30 Michele Pagani and Lorenzo Tortora de Falco. Strong normalization property for second order linear logic. *Theor. Comput. Sci.*, 411(2):410–444, jan 2010. doi:10.1016/j.tcs.2009.07.053.

31 Myriam Quatrini. *Sémantique cohérente des exponentielles: de la logique linéaire à la logique classique*. PhD thesis, Aix-Marseille 2, 1995.

32 Luca Roversi and Luca Vercelli. Safe recursion on notation into a light logic by levels. In Patrick Baillot, editor, *Proceedings International Workshop on Developments in Implicit Computational complExity, DICE 2010, Paphos, Cyprus, 27-28th March 2010*, volume 23 of *EPTCS*, pages 63–77, 2010. doi:10.4204/EPTCS.23.5.

33 Alexis Saurin. A linear perspective on cut-elimination for non-wellfounded sequent calculi with least and greatest fixed points (extended version). working paper or preprint, 2023. URL: https://hal.science/hal-04169137.

34 Alex Simpson. Cyclic arithmetic is equivalent to peano arithmetic. In Javier Esparza and Andrzej S. Murawski, editors, *Foundations of Software Science and Computation Structures - 20th International Conference, FOSSACS 2017, Proceedings*, volume 10203 of *Lecture Notes in Computer Science*, pages 283–300, 2017. doi:10.1007/978-3-662-54458-7_17.

35 Morten Heine Sørensen and Pawel Urzyczyn. *Lectures on the Curry-Howard Isomorphism, Volume 149 (Studies in Logic and the Foundations of Mathematics)*. Elsevier Science Inc., USA, 2006.

36 Terese. *Term rewriting systems*, volume 55 of *Cambridge tracts in theoretical computer science*. Cambridge University Press, 2003.

37 A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2 edition, 2000. doi:10.1017/CBO9781139168717.