Infinitary Cut-Elimination via Finite Approximations

Matteo Acclavio
University of Southern Denmark, Odense, Denmark
University of Sussex, Department of Informatics, Brighton, UK

Gianluca Curzi
University of Birmingham, UK
University of Gothenburg, Sweden

Giulio Guerrieri
University of Sussex, Department of Informatics, Brighton, UK

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.2024.8


Funding Matteo Acclavio: Supported by Villum Fonden, grant no. 50079

Acknowledgements We would like to thank Anupam Das, Abhishek De, Farzad Jafar-Rahmani, Alexis Saurin, Tito (Lê Thành Dung Nguyễn), 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 := νX.(1 & A & (X ⊗ X)) and ?A := µX.(⊥ ⊕ A ⊕ (X ⊗ Y)). As these authors notice, the fixed point-based definition of !
and can be regarded as a more permissive variant of the standard exponentials, since a proof of \( \nu X.(1 \& A \& (X \otimes X)) \) 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 \( (!A \otimes B) \equiv (\nu A \otimes !B) \), 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 \( \mu \text{MALL} \). 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].

On a parallel research thread, Mazza [25, 26, 27] studied parsimonious logic, a variant of linear logic where the exponential modality \( \nu \) satisfies Milner’s law (i.e., \( A \equiv A \otimes !A \)) and invalidates the implications \( !A \rightarrow !A \) (digging) and \( !A \rightarrow !A \otimes !A \) (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 \( A \otimes !A \rightarrow !A \) (co-absorption) and \( !A \rightarrow A \otimes !A \) (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 \( f : \mathbb{N} \rightarrow \{1, \ldots, n\} \) as premises, and constructs a proof of \( !A \) representing a stream of proofs of the form \( \mathcal{S} = (\mathcal{D}_{f(0)}, \mathcal{D}_{f(1)}, \ldots, \mathcal{D}_{f(n)}, \ldots) \). 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 \( \nu \text{PLL} \), supporting non-uniform exponentials, and \( \text{PLL} \), a fully uniform version. We investigate non-wellfounded counterparts of \( \nu \text{PLL} \) and \( \text{PLL} \), identifying 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 (\( \nu \text{PLL}^\infty \)), defined via the regularity condition and corresponding to \( \text{PLL} \), and weakly regular parsimonious linear logic (\( \text{wrPLL}^\infty \)), defined via a weak regularity condition and corresponding to \( \nu \text{PLL} \).

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 \( \lambda \)-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 \( \lambda \)-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 \( \omega \)-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 \( \frac{\Gamma}{\Gamma} \) or \( \frac{\Gamma_1}{\Gamma} \) or \( \frac{\Gamma_1, \Gamma_2}{\Gamma} \), and we refer to the sequents \( \Gamma_1 \) and \( \Gamma_2 \) as the premises, and to the sequent \( \Gamma \) 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.

\begin{definition}
A (binary, possibly infinite) tree \( T \) is a subset of words in \( \{1, 2\}^* \) that contains the empty word \( \epsilon \) (the root of \( T \)) and is ordered-prefix-closed (i.e., if \( n \in \{1, 2\} \) and \( vn \in T \), then \( v \in T \), and if moreover \( v2 \in T \), then \( v1 \in T \)). The elements of \( T \) are called nodes and their height is the length of the word. A child of \( v \in T \) is any \( vn \in T \) with \( n \in \{1, 2\} \). The
prefix order is a partial order \( \leq_T \) on \( T \) defined by: for any \( v, v' \in T \), \( v \leq_T v' \) if \( v' = vw \) for some \( w \in \{1, 2\}^* \). A maximal element of \( \leq_T \) is a leaf of \( T \). A branch of \( T \) is a set \( B \subseteq T \) such that \( \epsilon \in B \) and if \( w \in B \) is not a leaf of \( T \) then \( w \) has exactly one child in \( B \).

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

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

A coderivation \( D \) is \( r \)-free (for a rule \( r \in S \)) 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 \( |D| \in \mathbb{N} \)) if it labels a finite tree (with \( |D| \) nodes).

Given a set of coderivations \( X \), a sequent \( \Gamma \) is provable in \( X \) (noted \( \vdash_X \Gamma \)) if there is a coderivation in \( X \) with conclusion \( \Gamma \).

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

\[ \text{Definition 2. Let } D \text{ be a coderivation labeling a tree } T. \text{ A bar (resp. prebar) of } D \text{ is a set } V \subseteq T \text{ where:} \]

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

The height of a prebar \( V \) of \( D \) is the minimal height of the nodes of \( V \).

### 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 \( A = \{X, Y, \ldots\} \) using the following grammar:

\[
A, B ::= X \mid X^\perp \mid A \otimes B \mid A \multimap B \mid !A \mid ?A \mid 1 \mid \bot
\]

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

\[ \text{Definition 3. Parsimonious linear logic, denoted by PLL, is the set of rules in Figure 1, that is, axiom (ax), cut (cut), tensor (⊗), par (∥), one (1), bottom (⊥), functorial promotion (flip), weakening (?w), absorption (?b). Rules ax, ⊗, ∥, 1 and ⊥ are called multiplicative, while rules flip, ?w and ?b are called exponential. We also denote by PLL the set of derivations over the rules in PLL.} \]

\[ \text{Example 4. Figure 2 gives some examples of derivation in PLL. The (distinct) derivations 0 and 1 prove the same formula } N := !(X \multimap X) \multimap X \multimap X. \text{ The derivation } D_{\text{abs}} \text{ proves the absorption law } !A \multimap A \otimes !A; \text{ the derivation } D_{\text{der}} \text{ proves the dereliction law } !A \multimap A. \]
D\rightarrow X^2\otimes X
\Rightarrow !(X\otimes X^2),X^2,!,X
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2

\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otimes X^2),X^2,X\otimes X^2
\Rightarrow !(X\otils
(possibly non-recursive) function $f: \mathbb{N} \to \{1, \ldots, n\}$ as premises, and constructs a proof of $!A$ modeling the stream $(D_{f(0)}, D_{f(1)}, \ldots, D_{f(n)}, \ldots)$. This typing rule is the key tool to encode the so-called *advice* for Turing machines, an essential step to show completeness for $\text{P/poly}$.

In a similar vein, we can endow PLL with a non-uniform version of $\text{f!p}$ called **infinitely branching promotion** ($\text{ib!p}$), which constructs a stream $(D_0, D_1, \ldots, D_n, \ldots)$ with finite support, i.e., made of *finitely* many distinct derivations (of the same conclusion):\(^2\)

\[
\frac{\Gamma, A \vdash \Delta, A^+ & \vdash \Delta}{\text{ib!p}} \quad \frac{\Gamma, A \vdash \Delta, A^+ & \vdash \Delta}{\text{ib!p}} \quad \cdots \\
\frac{\vdash \Delta, A & \vdash \Delta}{\text{ib!p}} \quad \frac{\vdash \Delta, A & \vdash \Delta}{\text{ib!p}} \\
\{D_i \mid i \in \mathbb{N}\} \text{ is finite} \quad \vdash !A \quad \frac{\Gamma, A \vdash \Delta, !A}{\text{ib!p}} \\
\Gamma, \Delta, !A
\]

The side condition on $\text{ib!p}$ provides a proof theoretic counterpart to the function $f: \mathbb{N} \to \{1, \ldots, n\}$ in $\text{nuPLL}_{\text{uf}}$. Clearly, $\text{f!p}$ is subsumed by the rule $\text{ib!p}$, as it corresponds to the special (uniform) case where $D_i = D_{i+1}$ for all $i \in \mathbb{N}$.

\(^2\) Rule $\text{ib!p}$ is reminiscent of the $\omega$-rule used in (first-order) Peano arithmetic to derive formulas of the form $\forall x \phi$ that cannot be proven in a uniform way.
Definition 6. We define the set of rules nuPLL := \{\text{ax}, \otimes, \exists, 1, \bot, \text{cut}, ?b, ?w, 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 nuPLL\(_{el}\) [27]. As opposed to nuPLL, nuPLL\(_{el}\) is formulated as an intuitionistic (type) system. Furthermore, to achieve completeness for \(P/poly\), these authors introduced second-order quantifiers and the co-absorption (ib) and co-weakening (lw) rules displayed in (1).

Cut-elimination steps for nuPLL are in Figures 3, 5, and 6. In particular, the step ib!p-vs-?b in Figure 6 pops the first premise \(D_0\) of ib!p out of the stream (\(D_0, D_1, \ldots, D_n, \ldots\)).

4.1 From infinitely branching proofs to non-wellfounded proofs

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

\[
\begin{array}{c}
\Gamma, A \rightarrow A \\
\hline
\Gamma, A \\
\end{array}
\]

\[
\begin{array}{c}
?T, !A \\
\hline
?T, !A \\
\end{array}
\]

Definition 7. We define the set of rules PLL\(^\infty\) := \{\text{ax}, \otimes, \exists, 1, \bot, \text{cut}, ?b, ?w, c!p\}. We also denote by PLL\(^\infty\) the set of coderivations over the rules in PLL\(^\infty\).

In other words, PLL\(^\infty\) is the set of coderivations generated by the same rules as PLL, except that \(c!p\) is replaced by \(c!p\). From now on, we will only consider coderivations in PLL\(^\infty\).

Example 8. Figure 7 shows two non-wellfounded coderivations in PLL\(^\infty\): \(D_3\) (resp. \(D_7\)) has an infinite branch of cut (resp. ?b) rules, and is (resp. is not) regular.

We can embed PLL and nuPLL into PLL\(^\infty\) via the conclusion-preserving translations (\(\cdot\)\(^5\)) : PLL \(\rightarrow\) PLL\(^\infty\) and (\(\cdot\)\(^6\)) : nuPLL \(\rightarrow\) PLL\(^\infty\) defined in Figure 8 by induction on derivations: they map all rules to themselves except \(c!p\) and ib!p, which are “unpacked” into non-wellfounded coderivations that iterate infinitely many times the rule c!p.

An infinite chain of c!p rules (Figure 9) is a structure of interest in itself in PLL\(^\infty\).

Definition 9. A non-wellfounded box (nwb for short) is a coderivation \(D \in\) PLL\(^\infty\) with an infinite branch \(\{e, 2, 22, \ldots\}\) (the main branch of \(D\)) all labeled by \(c!p\) rules as in Figure 9, where \(A\) in the conclusion is the principal formula of \(D\), and \(D_0, D_1, \ldots\) are the calls of \(D\). We denote \(D\) by \(c!p(D_0, D_1, \ldots)\).

\(^3\) To be rigorous, this requires a slight change in Definition 1: the tree labeled by a derivation in nuPLL must be over \(\mathbb{N}^\infty\) instead of \(\{1, 2\}^\infty\), in order to deal with infinitely branching derivations.
8:8 Infinitary Cut-Elimination via Finite Approximations

For all \( r \in \{ /, \bot, \omega, \nu \} \) and \( t \in \{ \bot, \odot \} \) (as 1 and 0 are translated by themselves).

**Figure 8** Translations \((\cdot)^*\) from PLL to PLL\(^\infty\), and \((\cdot)^\star\) from nuPLL to PLL\(^\infty\).

\[
D = clp(D_0, \ldots, D_n, \ldots) = \begin{array}{c} \Gamma, A \hline \Gamma, A \\
\Gamma, A \hline \Gamma, !A \\
\Gamma, !A \hline \Gamma, !A \\
\Gamma, !A \hline \Gamma, !A \\
\Gamma, !A \hline \Gamma, !A \\
\end{array}
\]

**Figure 9** A non-wellfounded box in PLL\(^\infty\).

Let \( \mathcal{G} = clp(D_0, \ldots, D_n, \ldots) \) be a nwb. We may write \( \mathcal{G}(i) \) to denote \( D_i \). We say that \( \mathcal{G} \) has finite support (resp. is periodic with period \( k \)) if \( \{ \mathcal{G}(i) \mid i \in \mathbb{N} \} \) is finite (resp. if \( \mathcal{G}(i) = \mathcal{G}(k + i) \) for any \( i \in \mathbb{N} \)). A coderivation \( D \) has finite support (resp. is periodic) if any nwb in \( D \) has finite support (resp. is periodic).

**Example 10.** The only cut-free derivations of the formula \( N := !(X \rightarrow X) ightarrow X \rightarrow X \) are of the form \( \mathcal{G}(i) \) below on the right, for all \( n \in \mathbb{N} \), up to permutations of the rules \( ?w, ?b \) and \( \odot \) (the derivations \( 0 \) and \( 1 \) in Example 4 are special cases of it)

\[
clp(D_0, \ldots, D_n, \ldots) = \begin{array}{c} N \hline N \\
N \hline !N \\
!N \hline !N \\
!N \hline !N \\
!N \hline !N \\
\end{array}
\]

Consider the nwb \( clp(i_0, i_1, \ldots) \) above on the left, proving the formula \( !N \), where \( i_j \in \{ 0, 1 \} \) for all \( j \in \mathbb{N} \). Thus \( clp(i_0, i_1, \ldots) \) has finite support, as its only calls can be \( 0 \) or \( 1 \), and it is periodic if and only if so is the infinite sequence \( (i_0, i_1, \ldots) \in \{ 0, 1 \}^\omega \).

The cut-elimination steps \( \rightarrow_{cut} \) for PLL\(^\infty\) are in Figures 3, 5, and 10. Computationally, they allow the clp 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 \( clp \) 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 \( (D_0, D_1, \ldots, D_n, \ldots) \) similarly to \( ib!p \) but, unlike the latter, all the \( D_i \)’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 \( !p \) in PLL) provide infinitely many copies of the same content.
4.2 Consistency via a progressing criterion

In a non-wellfounded setting such as PLL∞, any sequent is provable. Indeed, the (non-wellfounded) codereivation Dj 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⊥ (for any formula B) of two derivations Dj.

The standard way to recover logical consistency in non-wellfounded proof theory is to introduce a global soundness condition on codereivations, called the progressing criterion [23, 13]. In PLL∞, this criterion relies on tracking occurrences of ⊔-formulas in a codereivation.

Definition 11. Let D be a codereivation in PLL∞. It is weakly progressing if every infinite branch contains infinitely many right premises of clp-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 D is a maximal sequence (Ai)i∈I of ⊔-formulas (resp. ?-formulas) for some downward-closed I ⊆ N such that Ai+1 is the parent of Ai for all i ∈ I. A ⊔-thread (Ai)i∈I is progressing if Ai is in the conclusion of a clp for infinitely many j ∈ I.

D is progressing if every infinite branch contains a progressing ⊔-thread. We define pPLL∞ (resp. wpPLL∞) as the set of progressing (resp. weak-progressing) codereivations in PLL∞.

Remark 12. Clearly, any progressing codereivation is weakly progressing too, but the converse fails (Example 13), therefore pPLL∞ ⊆ wpPLL∞. Moreover, the main branch of any nwb contains by definition a progressing ⊔-thread of its principal formula.

Example 13. Codereivations in Figure 7 are not weakly progressing (hence, not progressing): the rightmost branch of Dj, i.e., the branch {ε, 2, 22, . . . }, and the unique branch of Dj are infinite and contain no clp-rules. In contrast, the nwb clp(ε, 2, 22, . . . ) in Example 10 is
Infinitary Cut-Elimination via Finite Approximations

progressing by Remark 12, since its main branch is the only infinite branch. Below, a regular, weakly progressing but not progressing coderivation (\(\{\epsilon, 2, 21, 212, 2121, \ldots\}\)) is infinite but has no progressing \(t\)-thread).

\[
\frac{\text{clp}}{\text{cut}} \quad \frac{\text{as \ X, X}^+}{\text{as \ X, X}^+} \quad \frac{\text{as \ X, X}^+}{\text{as \ X, X}^+}
\]

\(\epsilon \vdash_{\text{PLL}} \Gamma \) if and only if \(\vdash_{\text{wpPLL}^\infty} \Gamma \).

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

\[
\left( \begin{array}{c}
D^{f1} \\
\cdot
\end{array} \right) := \left( \begin{array}{c}
D^f \\
\cdot
\end{array} \right) \quad \left( \begin{array}{c}
D^{f2} \\
\cdot
\end{array} \right) := \left( \begin{array}{c}
D^f \\
\cdot
\end{array} \right) \quad \left( \begin{array}{c}
D^{f3} \\
\cdot
\end{array} \right) := \left( \begin{array}{c}
D^f \\
\cdot
\end{array} \right)
\]

with \(r \neq \text{clp}\). Note that the derivation \(D^f\) is well-defined because \(D\) is weakly progressing.

**Corollary 15.** The empty sequent is not provable in \(\text{wpPLL}^\infty\) (and hence in \(\text{PLL}^\infty\)).

**Proof.** If the empty sequent were provable in \(\text{wpPLL}^\infty\), then there would be a cut-free derivation \(D \in \text{PLL}\) 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 \(\text{iblp}\) in the derivations in \(\text{nuPLL}\). By means of example, consider the \(\text{nwb}\) below, which is progressing but cannot be the image of the rule \(\text{iblp}\) via \((\cdot)^*\) (see Figure 8) since \(\{D_i \mid i \in \mathbb{N}\}\) is infinite.

\[
\frac{\text{clp}}{\text{cut}} \quad \frac{\text{as \ X, X}^+}{\text{as \ X, X}^+}
\]

with \(D_i = \text{clp}_{\text{PLL}^\infty(\ldots)}\) for each \(i \in \mathbb{N}\).

To identify in \(\text{PLL}^\infty\) the coderivations corresponding to derivations in \(\text{nuPLL}\) and in PLL via the translations \((\cdot)^\circ\) 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 \(\text{clp}\)-rules; it is finitely expandable if any branch contains finitely many \(\text{cut}\) and \(\text{?b}\) rules. We denote by \(\text{wrPLL}^\infty\) (resp. \(\text{rPLL}^\infty\)) the set of weakly regular (resp. regular) and finitely expandable coderivations in \(\text{PLL}^\infty\).

**Remark 17.** Regularity implies weak regularity and the converse fails as shown in Example 18 below, so \(\text{rPLL}^\infty \subseteq \text{wrPLL}^\infty\). Given \(D \in \text{PLL}^\infty\) progressing and finitely expandable, it is regular (resp. weakly regular) if and only if any \(\text{nwb}\) in \(D\) is periodic (resp. has finite support).
Example 18. Codereivations $D_1$ and $D_7$ in Figure 7 are not finitely expandable, as their infinite branch has infinitely many cut or $\text{?b}$, but they are weakly regular, since they have no clp rules. The codereivation in (4) is not weakly regular because $\{D_i \mid i \in \mathbb{N}\}$ is infinite.

An example of a weakly regular but not regular codereivation is the nwb clp$(i_{n_{i+1}},...i_{n_{i-1}})$ in Example 10 when the infinite sequence $(i_j)_{j \in \mathbb{N}} \in \{0,1\}^\omega$ is not periodic: $\emptyset$ and $\mathbb{1}$ are the only codereivations ending in the left premise of a clp rule (so the nwb is weakly regular), but there are infinitely many distinct codereivations ending in the right premise of a clp rule (so the nwb is not regular). Moreover, that nwb is finitely expandable, as it contains no $\text{?b}$ or cut.

The sets $\text{PLL}^\infty$ and $\text{wrPLL}^\infty$ are the non-wellfounded counterparts of PLL and nuPLL, respectively. Indeed, we have the following correspondence via the translations $(\cdot)^{\infty}$ and $(\cdot)^*$.}

Proposition 19.
1. If $D \in \text{PLL}$ (resp. $D \in \text{nuPLL}$) with conclusion $\Gamma$, then $D^\infty \in \text{PLL}^\infty$ (resp. $D^* \in \text{wrPLL}^\infty$) with conclusion $\Gamma$, and every clp in $D^\infty$ (resp. $D^*$) belongs to a nwb.
2. If $D' \in \text{PLL}^\infty$ (resp. $D' \in \text{wrPLL}^\infty$) and every clp in $D'$ belongs to a nwb, then there is $D \in \text{PLL}$ (resp. $D \in \text{nuPLL}$) such that $D^\infty = D'$ (resp. $D^* = D'$).

Progressing and weak progressing coincide in finitely expandable codereivations.

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

Proof. Let $D \in \text{wpPLL}^\infty$ be finitely expandable, and let $B$ be an infinite branch in $D$. By finite expandability there is $h \in \mathbb{N}$ such that $B$ contains no conclusion of a cut or $\text{?b}$ 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 $B$ at height $h_i$ has shape $\text{?w} \cdot A_i$. By inspecting the rules in Figure 1, each such $\text{?w} \cdot A_i$ can be the conclusion of either a $\text{?w}$ or a clp (with right premise $\text{?r} \cdot A_i$). 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 $\Gamma, A$). Therefore, $h_k$ is the root of a nwb. This also shows $D \in \text{pPLL}^\infty$. By Remark 12, $\text{pPLL}^\infty \subseteq \text{wpPLL}^\infty$.

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 $D \in X$ with $X \in \{\text{PLL}^\infty, \text{wrPLL}^\infty\}$ and $D \rightarrow_{\text{cut}} D'$, then also $D' \in X$.

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 role, 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 oPLL := PLL ∪ {hyp}, where hyp := hyp → for any sequent Γ. 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 oPLL(Γ) the set of open coderivations with conclusion Γ.

#### Definition 23.

Let D be an open derivations, V ⊆ {1, 2}* be a set of mutually incomparable (w.r.t. the prefix order) nodes of D, and {D′ν}ν∈V be a set of open coderivations where D′ν has the same conclusion as the subderivation Dν of D. We denote by D{D′ν/ν}ν∈V = D(D′₁/ν₁, . . . , D′νₙ/νₙ), the open coderivation obtained by replacing each Dν with D′ν.

The pruning of D over V is the open coderication [D]V = D{hyp/ν}ν∈V. If D and D′ are two open coderivations, then we say that D is an approximation of D′ (noted D ≤ D′) iif D = [D′]V for some V ⊆ {1, 2}*.

We denote by K(D) the set of finite approximations of D.

Note that D and [D]V (and hence D′ if D ≤ D′) have the same conclusion. Any open coderivation D is the supremum of its finite approximations, i.e. D = ∞D′∈K(D) D′. Indeed:

#### Proposition 24.

For any sequent Γ, the poset (oPLL(Γ), ≤) is a Scott-domain with least element the open derivation hyp and with maximal elements the coderivations (in PLL(Γ)) with conclusion Γ. The compact elements are precisely the open derivations in oPLL(Γ).

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

#### Lemma 25.

→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 D ∈ pPLL is finitely expandable, then there is a prebar V ⊆ {1, 2}* of D such that each v ∈ V is the root of a nwb in D.

#### Definition 27.

Let D ∈ pPLL be finitely expandable. The decomposition prebar of D is the minimal prebar V of D such that, for all v ∈ V, Dν is a nwb. We denote with border(D) such a bar and we set base(D) := [D]border(D).

Note that, by weak König lemma, in the above definition border(D) is finite and base(D) is a finite approximation of 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 →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 \(\omega\)-chains approximating the cut-free version of an open codervation. In other words, a mc-ices computes a (Scott-)continuous function from open codervations to cut-free open codervations. 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-ices compute the same (Scott-)continuous function.

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

\[\textbf{Definition 28.} \text{An infinitary cut elimination strategy (or ices for short) is a family } \sigma = \{\sigma_D\}_{D \in \mathit{oPLL}^\infty} \text{ where, for all } D \in \mathit{oPLL}^\infty, \sigma_D \text{ is a sequence of open codervations such that } \sigma_D(0) = D \text{ and } \sigma_D(i) \rightarrow_{\mathrm{cut}} \sigma_D(i + 1) \text{ for all } 0 \leq i < \ell(\sigma_D). \text{ Given an ices } \sigma, \text{ we define the function } \mathit{f}_\sigma : \text{\mathit{oPLL}^\infty} \rightarrow \text{\mathit{oPLL}^\infty} \text{ as } \mathit{f}_\sigma(D) := \bigcup_{i=0}^{\ell(\sigma_D)} \text{cf}(\sigma_D(i)) \text{ where } \text{cf}(D) \text{ is the greatest cut-free approximation of } D \text{ (w.r.t. } \preceq \text{).} 5 \text{ An ices } \sigma \text{ is a mc-ices if it is:}
\]

\begin{itemize}
  \item \textbf{maximal:} \(\sigma_D(\ell(\sigma_D))\) is normal for any open derivation \(D \ (\ell(\sigma_D) < \omega \text{ by Lemma 25});\)
  \item \textbf{(Scott)-continuous:} \(\mathit{f}_\sigma\) is Scott-continuous.
\end{itemize}

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-ices induce the same continuous function, an easy consequence of Lemma 25 and Continuity.

\[\textbf{Proposition 29.} \text{If } \sigma \text{ and } \sigma' \text{ are two mc-ices, then } \mathit{f}_\sigma = \mathit{f}_{\sigma'}.\]

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.

\[\textbf{Definition 30.} \text{The height-by-height ices is defined as } \sigma^\infty = \{\sigma_D^\infty\}_{D \in \mathit{oPLL}^\infty} \text{ where } \sigma_D^\infty(0) = D \text{ for each } D \in \mathit{oPLL}^\infty, \text{ and } \sigma_D^\infty(i + 1) \text{ is the open codervation obtained by applying a cut-elimination step to the rightmost reducible cut-rule with minimal height in } \sigma_D^\infty(i).\]

\[\textbf{Proposition 31.} \text{The ices } \sigma^\infty \text{ is a mc-ices.}\]

\[\textbf{Proof.} \text{By Lemma 25, any open derivation } D \text{ normalizes in } n_D \in \mathbb{N} \text{ steps; so, if } D \text{ is an open derivation, } \ell(\sigma_D^\infty) = n_D \text{ with } \sigma_D^\infty(n_D) \text{ normal by definition of } \sigma^\infty. \text{ Hence, } \sigma^\infty \text{ is maximal.}

Since } \sigma_D^\infty(i) \text{ is defined by applying a finite number of cut-eliminations steps to } D, \text{ then there is } D' \in K(D) \text{ such that } \sigma_D^\infty(i) = \sigma_D^\infty(i) \text{, and therefore } \text{cf}(\sigma_D^\infty(i)) = \text{cf}(\sigma_D^\infty(i)) \leq \mathit{f}_{\sigma^\infty}(D') \text{ for all } 0 \leq i \leq \ell(\sigma^\infty). \text{ Thus } \mathit{f}_{\sigma^\infty}(D) \leq \bigcup_{D' \in K(D)} \mathit{f}_{\sigma^\infty}(D'). \text{ Moreover } \bigcup_{D' \in K(D)} \mathit{f}_{\sigma^\infty}(D') \leq \mathit{f}_{\sigma^\infty}(D) \text{ because } \sigma^\infty \text{ is monotone by construction. Therefore, } \mathit{f}_{\sigma^\infty} \text{ 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].

\[\textbf{Definition 32} \text{(Chains). Let } \sigma = \{\sigma_D\}_{D \in \mathit{oPLL}^\infty} \text{ be an ices. We write } r_i \mapsto r_{i+1} \text{ if } r_{i+1} \text{ is a cut-rule in } \sigma_D(i + 1) \text{ produced by applying a cut-elimination step to the cut-rule } r_i \text{ in } \sigma_D(i).\]

A \textbf{cut-chain} in \(\sigma_D\) is a sequence \((r_i)_{i \in \alpha}\) of cut-rules with \(\alpha \leq \ell(\sigma_D)\), such that \(r_i\) a rule in \(\sigma_D(i)\), and either \(r_i = r_{i+1}\) or \(r_i \mapsto r_{i+1}\). We say that a chain \(\text{starts at } r_0\) and that each \(r_{i+1}\) is a \textbf{descendant} of \(r_i\).

\[\text{5 } \mathit{f}_\sigma \text{ is well-defined, as } (\text{cf}(\sigma_D(i)))_{0 \leq i < \ell(\sigma_D)} \text{ is an } \omega \text{-chain in } \mathit{oPLL}^\infty \text{ and so its sup exists by Proposition 24.} \]
8:14 Infinitary Cut-Elimination via Finite Approximations

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 \( D \in \text{pPLL}^\infty \), then so is \( f_{\sigma^\infty}(D) \).
2. If \( D \in \text{wrPLL}^\infty \) (resp. \( D \in \text{rPLL}^\infty \)), then so is \( f_{\sigma^\infty}(D) \).

Sketch of the proof.

1. We have to prove that \( f_{\sigma^\infty}(D) \) 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 \((r_i)\) there is a descendant \( r_i \) in \( \sigma^\infty\)(i) whose right premise is the conclusion of a \( \text{clp} \)-rule. Since \( B \) has infinitely many \( \text{clp} \) rules by progressing condition, every cut-rule with a premise in \( B \) is eventually reducible, so that there are infinitely many \( i \geq i_0 \) such that \( r_i \mapsto_{\sigma} r_{i+1} \). Therefore, if the right-premise of \( r_i \) did not eventually become conclusion of a \( \text{clp} \)-rule we could identify an infinite branch of \( D \) that has no progressing \( ! \)-thread.

   Now, let \( B^* \) be a branch of \( f_{\sigma^\infty}(D) \). If \( B^* \) has been obtained from \( D \) after finitely many cut-elimination steps then it is clearly hyp-free and, if infinite, it has a progressing \( ! \)-thread (Proposition 21). Otherwise, \( B^* \) has been constructed by an infinite cut-chain \((r_i)\), 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 \( \text{clp} \)-rule, and such that \( r_i \mapsto_{\sigma} r_{i+1} \) is a step permuting \( r^* \) downward (since \( r^* \) is on the left premise of \( r_i \), its principal \( ! \)-formula cannot be a cut-formula of \( r_i \) by assumption). This means that \( B^* \) contains infinitely many \( \text{clp} \) rules, and so it is hyp-free. To prove that there is a progressing \( ! \)-thread in \( B^* \) it suffices to show that infinitely many \( \text{clp} \) rules of \( B^* \) are descendants of the same branch \( B \) of \( D \), as the existence of a progressing \( ! \)-thread of \( B^* \) would follow directly from the existence of a (unique) progressing \( ! \)-thread of \( B \).

2. Akin to linear logic, we define the depth of a coderivation as the maximal number of nested nwbs, and we prove that the depth of (weakly) regular coderivations is always finite. Moreover, by Proposition 26, a progressing and finitely expandable coderivation \( D \) can be decomposed to a \( \text{nwb} \)-free finite approximation base\((D)\) and a series of \( \text{nwb} \)-bs whose calls have smaller depth. Using this property we define, by induction on the depth of \( D \), a maximal and transfinite ices reducing the calls of the \( \text{nwb} \)-bs 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 \( \text{nwb} \) with finite support (resp. a periodic \( \text{nwb} \)) via our transfinite ices, then we obtain in the limit a cut-free \( \text{nwb} \) with finite support (resp. a periodic \( \text{nwb} \)). We then show that this transfinite ices can be compressed to a (\( \omega \)-long) mc-ices using methods studied in [36, 33], and we conclude the proof by Item 1 and by the fact that \( f_{\sigma^\infty}(D) \) is finitely expandable and (weakly) regular for such a mc-ices.

By definition (as the sup of cut-free open coderivations) \( f_{\sigma^\infty}(D) \) is cut-free. Each item of Theorem 33 says in particular that \( f_{\sigma^\infty}(D) \) is hyp-free, which means that \( f_{\sigma^\infty}(D) \) is obtained by eliminating all the cuts in \( D \). This may not be the case if \( D \) does not fulfill any of the global conditions in the hypotheses of Theorem 33: \( f_{\sigma^\infty}(D) \) is still cut-free but may contain some “truncating” hyp that “prevented” eliminating some cut in \( D \), as in the example below.
where $D$ is the (non-weakly progressing, non-finitely expandable) open coderivation $D_1$, we have $f_n(D) = \text{hyp}$, so $f_n(D_1) = \text{hyp}$ by continuity.

6 Relational semantics for non-wellfounded proofs

Here we define a denotational model for oPLL$\infty$ 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$\infty$, but not for its extension with digging.

Relational semantics interprets exponential by finite multisets, denoted by brackets, e.g., $[x_1, \ldots, x_n]$; $+$ denotes the multiset union, and $M_f(X)$ denotes the set of finite multisets over a set $X$. To correctly define the semantics of a coderivation, we need to see sequents as sequences of formulas (taking their order into account), which means that we have to add an exchange rule to oPLL$\infty$ 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:

$$[X] := D_X \quad [1] := \{\ast\} \quad [A \otimes B] := [A] \times [B] \quad [[A]] := M_f([A]) \quad [A^\perp] := [[A]]$$

where $D_X$ is an arbitrary set. For a sequent $\Gamma = A_1, \ldots, A_n$, we set $[\Gamma] := [[A_1] \otimes \cdots \otimes [A_n]]$.

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

Example 36. For the codervations $D_1$ and $D_2$ in Figure 7, $[[D_1]] = [[D_2]] = \emptyset$. For the derivations $\emptyset$ and $\{1\}$ in Figure 2, $[[\emptyset]] = \{[[\{\ast\}]], (x, x) \mid x \in D_X\}$ and $[[\{1\}]] = \{[\{1\}, (y, x)] \mid x, y \in D_X\}$. For the codervation $\cdot \cdot p_{\{x_0, \ldots, x_n\}}$ in Example 10 (with $i_j \in \{0, 1\}$ for all $j \in \mathbb{N}$), $[[\cdot \cdot p_{\{x_0, \ldots, x_n\}}]] = \{[[\{\ast\}]]\} \cup \{[\{x_0, \ldots, x_n\} \in M_f([N]) \mid n \in \mathbb{N}, x_j \in [i_j] \forall 0 \leq j \leq n\}$. For the derivation $\circ$ in Example 10 (for any $n \in \mathbb{N}$), $[[\circ]] = \{[[\{x_0, \ldots, x_n\}], \{x_1, x_{n+1}\}] \mid x_0, \ldots, x_{n+1} \in D_X\}$. Note that $[[\circ]] \cap [[\ast]] = \emptyset$ for all $m, n \in \mathbb{N}$ such that $n \neq m$, and that $[[\circ]]$ is stable under permutations of the rules $\ast w, \ast b$ and $\otimes$ in $\circ$ (that is, if $D$ is obtained from $\circ$ by permuting the rules $\ast w, \ast b$ or $\otimes$, then $[[D]] = [[\circ]]$).
\[ \frac{\Gamma, \text{??} A}{\Gamma, \text{??} A} \]

\[ \frac{\Gamma, \text{??} A}{\Gamma, \text{??} A} \]

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

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 \( \mathcal{D} \in \text{oPLL}^\infty \). Then, \( [\mathcal{D}] = [\bigcup_{D' \in K(\mathcal{D})} D'] = [\bigcup_{D' \in K(\mathcal{D})} [D']] \).

**Theorem 38** (Soundness).

1. Let \( \mathcal{D} \in \text{oPLL}^\infty \). If \( \mathcal{D} \rightarrow_{\text{cut}} D' \), then \( [\mathcal{D}] = [D'] \).
2. Let \( \mathcal{D} \in \text{oPLL}^\infty \). If \( \sigma \) is a mc-ices, then \( [\mathcal{D}] = [\mathcal{f}_\sigma(\mathcal{D})] \).

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

**Corollary 39.** Let \( \mathcal{D} \in \text{wpPLL}^\infty \). Then \( [\mathcal{D}] \neq \emptyset \).

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

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

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

**Proof.** Consider the coderivations \( D_{??d} \) and \( \widehat{D}_{??d} \) below, where \( D = \text{clp}(\{0,1,0,1,\ldots\}) \) and, for all \( i \in \mathbb{N} \), \( D_i = \{ \text{clp}(\{k_i,\ldots,k_i\}) \mid k_j \in \mathbb{N} \text{ for all } j \in \mathbb{N} \} \) (\( n \) is defined in Example 10 for all \( n \in \mathbb{N} \)).

\[
D_{??d} := \begin{array}{c}
\vdash \hline
!N \\
\text{cut} \hline
??\text{??} \frac{N}{\text{??} \frac{N}{\text{??} \frac{N}{\text{??} \frac{N}{\text{??} \frac{N}{}}}}}
\end{array}
\]

\[
\widehat{D}_{??d} := \begin{array}{c}
\vdash \hline
!N \\
\text{clp} \hline
??\text{??} \frac{N}{\text{??} \frac{N}{\text{??} \frac{N}{\text{??} \frac{N}{}}}}
\end{array}
\]

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

We show that \( [\hat{D}_{??d}] \not\subseteq [D_{??d}] \). First, it can be easily shown that if, in one of the \( D_i = \text{clp}(\{k_i,\ldots,k_i\}) \) in \( \hat{D}_{??d} \), one of the \( k_j \) is different from 0 or 1, then there is \( x \in [D_{??d}] \setminus [\hat{D}_{??d}] \) (this basically follows from the fact that \( [n] \cap [m] = \emptyset \) for all \( n, m \in \mathbb{N} \) such that \( n \neq m \),...
see Example 36). Let us now suppose that in $D_{\tau d}$, for all $i \in \mathbb{N}$, $D_i = c[p_{(k_0, \ldots, k_n)}]$ with $k_j \in \{0,1\}$ for all $j \in \mathbb{N}$. Let $\tilde{0}$ and $\tilde{1}$ be any element of $[\tilde{0}]$ and $[\tilde{1}]$, respectively (see Example 36). Note that $\tilde{0} \neq \tilde{1}$. It is easy to verify that $[[\tilde{0}], [\tilde{0}]], [[\tilde{1}], [\tilde{1}]] \notin [D_{\tau d}]$, since $[\tilde{0}, \hat{0}], [\tilde{1}, \hat{1}] \notin [D]$ (see Example 36). Concerning $[D_{\tau d}]$, notice that, since $k_0, k_1, k_2 \in \{0,1\}$, either $k_0 = k_1$ or $k_0 = k_2$ or $k_1 = k_2$. In the first case, we have $[[k_0], [k_0]] \in [D_{\tau d}]$, in the second case we have $[[k_0], [k_2]] \in [D_{\tau d}]$, and in the last case we have $[[k_1], [k_2]] \in [D_{\tau d}]$. □

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 $P/poly$ and $P$ (see [1]). These results would reformulate in a non-wellfounded setting the characterization of $P/poly$ presented in [27].

References


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.


