Infinets: The parallel syntax for non-wellfounded proof-theory

Abhishek De, Alexis Saurin

Infinets: The parallel syntax for non-wellfounded proof-theory

Abhishek De[1] and Alexis Saurin[2] ⋆

[1] IRIF, Université de Paris, F-75013 Paris, France
[2] IRIF, CNRS, Université de Paris, F-75013 Paris, France
{ade,saurin}@irif.fr

Abstract. Logics based on the μ-calculus are used to model inductive and coinductive reasoning and to verify reactive systems. A well-structured proof-theory is needed in order to apply such logics to the study of programming languages with (co)inductive data types and automated (co)inductive theorem proving. While traditional proof system suffers some defects, non-wellfounded (or infinitary) and circular proofs have been recognized as a valuable alternative, and significant progress have been made in this direction in recent years. Such proofs are non-wellfounded sequent derivations together with a global validity condition expressed in terms of *progressing threads*.
The present paper investigates a discrepancy found in such proof systems, between the sequential nature of sequent proofs and the parallel structure of threads: various proof attempts may have the exact threading structure while differing in the order of inference rules applications. The paper introduces infinets, that are proof-nets for non-wellfounded proofs in the setting of multiplicative linear logic with least and greatest fixed-points ($\mu\mathsf{MLL}^{\infty}$) and study their correctness and sequentialization.

Keywords: circular proofs · non-wellfounded proofs · fixed points · mu-calculus · linear logic · proof-nets · induction and coinduction

1 Introduction

Inductive and coinductive reasoning is pervasive in computer science to specify and reason about infinite data as well as reactive properties. Developing appropriate proof systems amenable to automated reasoning over (co)inductive statements is therefore important for designing programs as well as for analyzing computational systems. Various logical settings have been introduced to reason about such inductive and coinductive statements, both at the level of the logical languages modelling (co)induction (such as Martin Löf's inductive predicates or fixed-point logics, also known as μ-calculi) and at the level of the proof-theoretical framework considered (finite proofs with explicit (co)induction

⋆ This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 754362. Partially funded by ANR Project RAPIDO, ANR-14-CE25-0007.

rules *à la* Park [26] or infinite, non-wellfounded proofs with fixed-point unfoldings) [7–9, 4, 1, 2]. Moreover, such proof systems have been considered over classical logic [7, 9], intuitionistic logic [10], linear-time or branching-time temporal logic [23, 22, 28, 29, 14, 16, 17] or linear logic [27, 18, 4, 3, 16].

Logics based on the μ-calculus have been particularly successful in modelling inductive and coinductive reasoning and for the verification of reactive systems. While the model-theory of the μ-calculus has been well-studied, its proof-theory still deserves further investigations. Indeed, while explicit induction rules are simple to formulate (For instance, Fig. 1 shows the introduction rule *à la* Park for a coinductive property) the treatment of (co)inductive reasoning brings some highly complex proof objects.

$$\frac{\vdash S, \Delta \quad \vdash S^{\perp}, F[S/X]}{\vdash \nu X.F, \Delta}\,(\nu')$$

Fig. 1: Coinduction rule

At least two fundamental technical shortcomings prevent the application of traditional μ-calculus-based proof-systems for the study of programming languages with (co)inductive data types and automated (co)inductive theorem proving and call for alternative proposals of proof systems supporting (co)induction. Firstly, the fixed point introduction rules break the subformula property which is highly problematic for automated proof construction: at each coinduction rule, one shall guess an invariant (in the same way as one has to guess an appropriate induction hypothesis in usual mathematical reasoning). Secondly, (ν_{inv}) actually hides a cut rule that *cannot* be eliminated, which is problematic for extending the Curry-Howard correspondence to fixed-point logics.

$$\dfrac{\dfrac{\dfrac{\vdots}{\vdash \mu X.X}}{\vdash \mu X.X}\,(\mu) \qquad \dfrac{\dfrac{\vdots}{\vdash \nu X.X, \Gamma}}{\vdash \nu X.X, \Gamma}\,(\nu)}{\vdash \Gamma}\,(\mathsf{cut})$$

Fig. 2: An unsound proof

Non-wellfounded proof systems have been proposed as an alternative [7–9] to explicit (co)induction. By having the coinduction rule with simple fixed-point unfoldings and allowing for non-wellfounded branches, those proof systems address the problem of the subformula property for the cut-free systems: the set of subformula is then known as Fischer-Ladner subformulas, incorporating fixed-points unfolding but preserving finiteness of the subformula space. Moreover, the cut-elimination dynamics for inductive-coinductive rules becomes much simpler. A particularly interesting subclass of non-wellfounded proofs, is that of circular, or cyclic proofs, that have infinite but *regular* derivations trees: they have attracted a lot of attention for retaining the simplicity of the inferences of non-wellfounded proof systems but finitely representable making it possible to have an algorithmic treatment of such proof objects. However, in those proof systems when considering all possible infinite, non-wellfounded derivations (*a.k.a.* pre-proofs), it is straightforward to derive any sequent Γ (see fig. 2). Such ***pre-proofs*** are therefore unsound: one needs to impose a validity criterion to sieve the logically valid proofs from the unsound ones. This condition will actually reflect the inductive and coinductive nature of our two fixed-point connectives: a standard approach [7–9, 27, 3] is to consider a pre-proof to be valid if every infinite branch is supported by an infinitely progressing thread. As a result, the logical correctness of circular proofs becomes non-local, much in the spirit of correctness criteria for proof-nets [19, 13].

However the structure of non-wellfounded proofs has to be further investigated: the present work stems from the observation of a discrepancy between the sequential nature of sequent proofs and the parallel structure of threads. An immediate consequence is that various proof attempts may have the exact same threading structure but differ in the order of inference rule applications; moreover, cut-elimination is known to fail with more expressive thread conditions. This paper proposes a theory of proof-nets for $\mu\mathsf{MLL}^\infty$ non-wellfounded proofs.

Organization of the paper. In Section 2, we recall the necessary background from [3] on linear logic with least and greatest fixed points and its non-wellfounded proofs, we only present the unit-free multiplicative setting which is the framework in which we will define our proof-nets. In Section 3 we adapt Curien's proof-nets [11] to a very simple extension of MLL, $\mu\mathsf{MLL}^*$, in which fixed-points inferences are unfoldings and only wellfounded proofs are allowed; this allows us to set the first definitions of proof-nets and extend correctness criterion, sequentialization and cut-elimination to this setting but most importantly it sets the proof-net formalism that will be used for the extension to non-wellfounded derivations. Infinets are introduced in Section 4 as an extension of the $\mu\mathsf{MLL}^*$ proof-nets of the previous section. A correctness criterion is defined in Section 5 which is shown to be sound (every proof-nets obtained from a sequent (pre-)proof is correct). The completeness of the criterion (*i.e.* sequentialization theorem) is addressed in Section 6. We quotient proofs differing in the order of rule application in Section 7 and give a partial cut elimination result in Section 8. We conclude in Section 9 and comment on related works and future directions.

Notation. For any sequence S, let $\mathsf{Inf}(S)$ be the terms of S that appears infinitely often in S. Given a finite alphabet Σ, Σ^* and Σ^ω are the set of finite and infinite words over Σ *resp.* Let $\Sigma^\infty = \Sigma^* \cup \Sigma^\omega$. We denote the empty word by ϵ. Given two words u, u' (finite or infinite) we denote by $u \cap u'$ the greatest common prefix of u and u' and $u \sqsubseteq u'$ if u is a prefix of u'. Given a language, $\mathcal{L} \subseteq \Sigma^\infty$, $\overline{\mathcal{L}} \subseteq \Sigma^\infty$ is the set of all prefixes of the words in $\mathcal{L}$.

2 Background

We denote the multiplicative additive fragment of linear logic by MALL and the multiplicative fragment by MLL. The non-wellfounded extension of MALL with least and greatest fixed points operators, $\mu\mathsf{MALL}^\infty$, was introduced in [3, 16]. Proof-nets for additives and units are quite cumbersome [20, 6], so, in the current presentation, we will only consider the unit-free multiplicative fragment which we denote by $\mu\mathsf{MLL}^\infty$.

Definition 1. *Given an infinite set of atoms $\mathcal{A} = \{A, B, \dots\}$, and an infinite set of propositional variables, $\mathcal{V} = \{X, Y, \dots\}$, s.t. $\mathcal{A} \cap \mathcal{V} = \emptyset$,* μMLL **pre-formulas** *are given by the following grammar:*

$$\phi, \psi ::= A \mid A^\perp \mid X \mid \phi \parr \psi \mid \phi \otimes \psi \mid \sigma X.\phi$$

where $A \in \mathcal{A}$ and $X \in \mathcal{V}$, and $\sigma \in \{\mu, \nu\}$; σ binds the variable X in ϕ. When a pre-formula is closed (i.e. no free variables), we simply call it a **formula**.

Note that negation is not a part of the syntax, so that we do not need any positivity condition on the fixed-point expressions. We define negation, $(\bullet)^\perp$, as a meta-operation on the pre-formulas and will use it only on formulas.

Definition 2. **Negation** *of a pre-formula ϕ, $\phi^\perp$, is the involution satisfying:*

$$(\phi \otimes \psi)^\perp = \psi^\perp \parr \phi^\perp,\ X^\perp = X,\ (\mu X.\phi)^\perp = \nu X.\phi^\perp.$$

Example 1. As a running example, we will consider the formulas $\phi = A \parr A^\perp \in$ MLL and $\psi = \nu X.X \otimes \phi \in \mu\mathsf{MLL}^\infty$. Observe that $\phi^\perp = A^\perp \otimes A$ as usual in MLL and by def. 2, $\psi^\perp = \mu X.X \parr \phi^\perp$.

The reader may find it surprising to define $X^\perp = X$, but it is harmless since our proof system only deals with formulas. Note that $(F[G/X])^\perp = F^\perp[G^\perp/X]$.

Definition 3. *An* **(infinite) address** *is a finite(resp. infinite) word in $\{l, r, i\}^\infty$. Negation extends over addresses as the morphism satisfying $l^\perp = r$, $r^\perp = l$, and $i^\perp = i$. We say that α' is a* **sub-address** *of α if $\alpha' \sqsubseteq \alpha$. We say that α and β are disjoint if $\alpha \cap \beta$ is not equal to α or β.*

Definition 4. *A* **formula occurrence** *(denoted by $F, G, ...$) is given by a formula, ϕ, and a finite address, α, and written ϕ_α. Let $\mathsf{addr}(\phi_\alpha) = \alpha$. We say that occurrences are disjoint when their addresses are. Operations on formulas are extended to occurrences as follows: ${\phi_\alpha}^\perp = \phi^\perp_{\alpha^\perp}$, for any $\star \in \{\parr, \otimes\}$, $F \star G = (\phi \star \psi)_\alpha$ if $F = \phi_{\alpha l}$ and $G = \psi_{\alpha r}$, and for $\sigma \in \{\mu, \nu\}$, $\sigma X.F = (\sigma X.\phi)_\alpha$ if $F = \phi_{\alpha i}$. Substitution of occurrences forgets addresses i.e. $(\phi_\alpha)[\psi_\beta/X] = (\phi[\psi/X])_\alpha$. Finally, we use $\lceil \bullet \rceil$ to denote the address erasure operation on occurrences.*

Fixed-points logics come with a notion of subformulas (and suboccurrences) slightly different from usual:

Definition 5. *The* **Fischer-Ladner closure** *of a formula occurrence F, $\mathsf{FL}(F)$, is the least set of formula occurrences s.t. $F \in \mathsf{FL}(F)$, $G_1 \star G_2 \in \mathsf{FL}(F) \implies G_1, G_2 \in \mathsf{FL}(F)$ for $\star \in \{\parr, \otimes\}$, and $\sigma X.G \in \mathsf{FL}(F) \implies G[\sigma X.G/X] \in \mathsf{FL}(F)$ for $\sigma \in \{\mu, \nu\}$. We say that G is a* **FL-suboccurrence** *of F (denoted $G \leq F$) if $G \in \mathsf{FL}(F)$ and G is an* **immediate FL-suboccurrence** *of F (denoted $G \lessdot F$) if $G \leq F$ and for every H s.t. $G \leq H \leq F$ either $H = G$ or $H = F$. The* **FL-subformulas** *of F are elements of $\{\phi \mid \phi = \lceil G \in \mathsf{FL}(F) \rceil\}$.*

Clearly, we could have defined Fischer-Ladner closure on the level of formulas. By abuse of notation, we will sometimes use $\mathsf{FL}(\bullet), \leq, \lessdot$ on formulas.

Remark 1. Observe that for any F, the number of FL-subformulas of F is finite.

The usual notion of subformula (say in MLL) is obtained by traversing the syntax tree of a formula. In the same way, the notion of FL-subformula can be obtained by traversing the graph of the formula (*resp.* occurrence).

$$\frac{\lceil F \rceil = \lceil G \rceil^{\perp}}{\vdash F, G}\,(\mathsf{ax}) \quad \frac{\vdash F, \Delta_1 \quad \vdash F^{\perp}, \Delta_2}{\vdash \Delta_1, \Delta_2}\,(\mathsf{cut}) \quad \frac{\vdash F, G, \Delta}{\vdash F \parr G, \Delta}\,(\parr)$$

$$\frac{\vdash F, \Delta_1 \quad \vdash G, \Delta_2}{\vdash F \otimes G, \Delta_1, \Delta_2}\,(\otimes) \quad \frac{\vdash G[\mu X.G/X], \Delta}{\vdash \mu X.G, \Delta}\,(\mu) \quad \frac{\vdash G[\nu X.G/X], \Delta}{\vdash \nu X.G, \Delta}\,(\nu)$$

Fig. 3: Inference rules for $\mu\mathsf{MLL}^{\infty}$

Definition 6. *The* **FL-graph** *of a formula ϕ, denoted $\mathfrak{G}(\phi)$, is the graph obtained from $\mathsf{FL}(\phi)$ by identifying the nodes of bound variable occurrences with their binders (i.e. $\phi \to \psi$ if $\phi \lessdot \psi$).*

Example 2. The graphs of the formulas ϕ and ψ of example 1 are the following:

Observe that the graph of a MLL formula is acyclic corresponding to the usual syntax tree but the graph of a $\mu\mathsf{MLL}^{\infty}$ formula could potentially contain a cycle.

As usual with classical linear logic $\Gamma, \phi \vdash \Delta$ is provable iff the sequent $\Gamma \vdash \phi^{\perp}, \Delta$ is provable. Hence, it is enough to consider the one-sided proof system. A one-sided $\mu\mathsf{MLL}^{\infty}$ sequent is an expression $\vdash \Delta$ where Δ is a finite set of pairwise disjoint formula occurrences.

Definition 7. *A* **pre-proof** *of $\mu\mathsf{MLL}^{\infty}$ is a possibly infinite tree generated from the inference rules given in fig. 3. Given a pre-proof, π, $\mathsf{addr}(\pi) \subseteq \{l, r, i\}^{\infty}$ is a set of addresses s.t. an address $\alpha \in \mathsf{addr}(\pi)$ iff there is an occurrence, F, in an axiom in π with $\mathsf{addr}(F) = \alpha$ and an infinite address $\alpha \in \mathsf{addr}(\pi)$ iff all the strict prefixes of α are addresses of occurrences appearing in π.*

Definition 8. *A* **thread** *of a formula occurrence, F, is a sequence, $t = \{F_i\}_{i \in I}$, where $I \in \omega + 1$, $F_0 = F$, and for every $i \in I$ s.t. $i + 1 \in I$ either F_i is suboccurrence of F_{i+1} or $F_i = F_{i+1}$. We denote by $\lceil t \rceil$ the sequence $\{\lceil F_i \rceil\}_{i \in I}$ where $t = \{F_i\}_{i \in I}$. A thread, t, is said to be* **valid** *if $\min(\mathsf{Inf}(\lceil t \rceil))$ is a ν-formula where minimum is taken in the $\leq$ ordering.*

Remark 2. Observe that for any infinite thread t of a formula occurrence F, $\mathsf{Inf}(\lceil t \rceil)$ is non-empty since F has finitely many FL-subformulas.

Definition 9. *A $\mu\mathsf{MLL}^{\infty}$* **proof** *is a pre-proof in which every infinite branch contains a valid thread. A* **circular** *pre-proof is a regular $\mu\mathsf{MLL}^{\infty}$ pre-proof i.e. one which has a finite number of distinct subtrees.*

Example 3. The following non-wellfounded pre-proof of the sequent $\vdash \psi_{\alpha}$ (α is an arbitrary address) is circular and is a proof because the only infinite thread $\{\psi_{\alpha(il)^n}\}_{n=0}^{\infty}$ is valid.

$$\dfrac{\dfrac{\star}{\vdash \psi_{\alpha il}} \quad \dfrac{\dfrac{}{\vdash A_{\alpha irl}, A^{\perp}_{\alpha irr}}\,(\mathsf{ax})}{\vdash A \parr A^{\perp}_{\alpha ir}}\,(\parr)}{\dfrac{\vdash \psi \otimes (A \parr A^{\perp})_{\alpha i}}{\star \vdash \psi_{\alpha}}\,(\nu)}\,(\otimes)$$

3 A first taste of proof-nets in logics with fixed points

Proof-nets are a geometrical method of representing proofs, introduced by Girard that eliminates two forms of bureaucracy which differentiate sequent proofs: irrelevant syntactical features and the order of rules. As a stepping stone, we first consider proof nets in $\mu\mathsf{MLL}^*$ which is the proof system with the same inference rules as $\mu\mathsf{MLL}^\infty$ (fig. 3) but with finite proofs. $\mu\mathsf{MLL}^*$ is strictly weaker than $\mu\mathsf{MLL}^\infty$.

Proof-nets are usually defined as vertex labelled, edge labelled directed multigraphs. In this presentation a proof structure is "almost" a forest (*i.e.* a collection of trees) with the leaves joined by axioms or cuts. We use a different presentation due to Curien [11] to separate the forest of syntax trees and the space of axiom links for reasons that will become clearer later.

Definition 10. *A* **syntax tree** *of a formula occurrence, F, is the (possibly infinite) unfolding tree of $\mathfrak{G}(F)$. The syntax tree induces a prefix closed language, $\mathcal{L}_F \subset \{l, r, i\}^\infty$ s.t. there is a natural bijection between the finite (resp. infinite) words in $\mathcal{L}_F$ and the finite (resp. infinite) paths of the tree. A* **partial syntax tree***, F^U, is a subtree of the syntax tree of the formula occurrence, F, such that the set of words, $U \subseteq \mathcal{L}_F$ represents a "frontier" of the syntax tree of F i.e. any $u, u' \in U$ are pairwise disjoint and for every $uav \in U$, there is a v' s.t. $ua^\perp v' \in U$. For a finite $u \in \overline{U}$, we denote by (F, u) the unique suboccurrence of F with the address $\mathsf{addr}(F).u$.*

Example 4. The syntax tree of ψ is the unfolding of $\mathfrak{G}(\psi)$ and induces the language $\overline{i(li)^*r(l+r)} + (il)^\omega$. Further, given an arbitrary address α, $\psi_\alpha^{\{ili,irl,irr\}}$ is a partial syntax tree whereas $\psi_\alpha^{\{ilil,irl,irr\}}$ is not. If $u = ililir$ then $(\psi_\alpha, u) = A \parr A^\perp_{\alpha ililir}$.

Definition 11. *A* **proof structure** *is given by $[\Theta']\{B_i^{U_i}\}_{i\in I}[\Theta]$, where*

- *I is a finite index set;*
- *for every $i \in I$, B_i is a formula occurrence, $B_i^{U_i}$ is a partial syntax tree with $U_i \subset \{l, r, i\}^*$;*
- *Θ' is a (possibly empty) collection of disjoint subsets of $\{B_i\}_{i\in I}$ of the form $\{C, C^\perp\}$;*
- *Θ is a partition of $\bigcup_{i\in I}\{\alpha_i u_i \mid \mathsf{addr}(B_i) = \alpha_i, u_i \in U_i\}$ s.t. the partitions are of the form $\{\alpha_i u_i, \alpha_j u_j\}$ with $\lceil(B_i, u_i)\rceil = \lceil(B_j, u_j)\rceil^\perp$.*

Each class of Θ represents an axiom, each of class of Θ' represents a cut, and $\{B_i\}_{i\in I} \setminus \bigcup_{\theta\in\Theta'} \theta$ are the conclusions of the proof structure.

$$\dfrac{\dfrac{\pi_1}{\vdash \Gamma, F} \quad \dfrac{\pi_2}{\vdash \Delta, F^{\perp}}}{\vdash \Gamma, \Delta}\,(\mathsf{cut}) \qquad \dfrac{\dfrac{\pi_1}{\vdash \Gamma, F} \quad \dfrac{\pi_2}{\vdash \Delta, G}}{\vdash \Gamma, \Delta, F \otimes G}\,(\otimes) \qquad \dfrac{\dfrac{\pi_0}{\vdash \Gamma, F, G}}{\vdash \Gamma, F \parr G}\,(\parr) \qquad \dfrac{\dfrac{\pi_0}{\vdash \Gamma, F[\mu X.F/X]}}{\vdash \Gamma, \mu X.F}\,(\mu)$$

(a) (b) (c) (d)

Fig. 4

Definition 12. *Let π be a $\mu\mathsf{MLL}^*$ proof.* **Desequentialization** *of π, denoted $\mathsf{Deseq}(\pi)$, is defined by induction on the structure of the proof:*

- *The base case is a proof with only an* ax *rule, say* $\dfrac{}{F, G^{\perp}}\,(\mathsf{ax})$. *Then*

$$\mathsf{Deseq}(\pi) = [\emptyset]\{F^{\{\epsilon\}}, (G^{\perp})^{\{\epsilon\}}\}[\{\{\mathsf{addr}(F), \mathsf{addr}(G^{\perp})\}\}]$$

- *If $\mathsf{Deseq}(\pi_1) = [\Theta'_1]\Gamma_1 \cup \{F^U\}[\Theta_1]$ and $\mathsf{Deseq}(\pi_2) = [\Theta'_2]\Gamma_1 \cup \{F^{\perp U'}\}[\Theta_2]$, then $\mathsf{Deseq}(\pi) = [\Theta'_1 \cup \Theta'_2 \cup \{F, F^{\perp}\}]\Gamma_1 \cup \Gamma_2[\Theta_1 \cup \Theta_2]$ where π is fig. 4(a).*
- *If $\mathsf{Deseq}(\pi_1) = [\Theta'_1]\Gamma_1 \cup \{F^U\}[\Theta_1]$ with $\mathsf{addr}(F) = \alpha l$ and $\mathsf{Deseq}(\pi_2) = [\Theta'_2]\Gamma_1 \cup \{G^{U'}\}[\Theta_2]$ with $\mathsf{addr}(G) = \alpha r$, then $\mathsf{Deseq}(\pi) = [\Theta'_1 \cup \Theta'_2]\Gamma_1 \cup \Gamma_2 \cup \{F \otimes G^{l \cdot U + r \cdot U'}\}[\Theta_1 \cup \Theta_2]$ with $\mathsf{addr}(F \otimes G) = \alpha$ where π is fig. 4(b).*
- *If $\mathsf{Deseq}(\pi_0) = [\Theta'_0]\Gamma_0 \cup \{F^U, G^{U'}\}[\Theta_0]$ with $\mathsf{addr}(F) = \alpha l, \mathsf{addr}(G) = \alpha r$ then $\mathsf{Deseq}(\pi) = [\Theta_0]\Gamma_0 \cup \{F \parr G^{l \cdot U + r \cdot U'}\}[\Theta_0]$ with $\mathsf{addr}(F \parr G) = \alpha$ where π is fig. 4(c).*
- *If $\mathsf{Deseq}(\pi_0) = [\Theta'_0]\Gamma_0 \cup \{F[F/X]^U\}[\Theta_0]$ with $\mathsf{addr}(F[F/X]) = \alpha i$ then $\mathsf{Deseq}(\pi) = [\Theta_0]\Gamma_0 \cup \{\mu X.F^{i \cdot U}\}[\Theta_0]$ with $\mathsf{addr}(\mu X.F) = \alpha$ where π is fig. 4(d).*
- *The case for ν follows exactly as μ.*

Example 5. Consider the following proof π of the sequent $\vdash \nu X.X \parr \mu X.X$.

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash \nu X.X_{\alpha l}, \mu Y.Y_{\beta i}}\,(\mathsf{ax})}{\vdash \nu X.X_{\alpha l}, \mu Y.Y_{\beta}}\,(\mu) \quad \dfrac{\dfrac{}{\vdash \nu Y.Y_{\beta^{\perp} i}, \mu X.X_{\alpha r}}\,(\mathsf{ax})}{\vdash \nu Y.Y_{\beta^{\perp}}, \mu X.X_{\alpha r}}\,(\nu)}{\vdash \nu X.X_{\alpha l}, \mu X.X_{\alpha r}}\,(\mathsf{cut})}{\vdash \nu X.X \parr \mu X.X_{\alpha}}\,(\parr)$$

We choose α, β *s.t.* they are disjoint. We have that $\mathsf{Deseq}(\pi) = [\Theta']\Gamma[\Theta]$ *s.t.*

$$\Theta' = \big\{\{\mu Y.Y_{\beta}, \nu Y.Y_{\beta^{\perp}}\}\big\} \qquad \Theta = \big\{\{\alpha l, \beta i\}, \{\alpha r, \beta^{\perp} i\}\big\}$$
$$\Gamma = \Big\{\nu X.X \parr \mu X.X_{\alpha}^{\{l,r\}}, \mu Y.Y_{\beta}^{\{i\}}, \nu Y.Y_{\beta^{\perp}}^{\{i\}}\Big\}$$

Definition 13 (Graph of proof structure). *Let $\mathcal{S} = [\Theta']\{B_i^{U_i}\}_{i \in I}[\Theta]$ be a proof structure. The graph of $\mathcal{S}$ denoted $\mathsf{Gr}(\mathcal{S})$ is the graph formed by:*

- *taking the transpose (i.e. reversal of every edge) of the partial syntax tree $\{B_i^{U_i}\}_{i \in I}$;*
- *for each $\{B_i, B_j\} \in \Theta'$, adding a node labelled* cut *with two incoming edges from (B_i, ϵ) and (B_j, ϵ);*

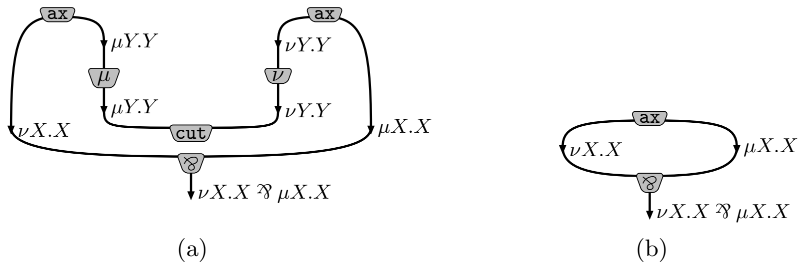


Fig. 5: Graph of $\mu\mathsf{MLL}^\infty$ proof structures

- *for each* $\{\alpha_i u_i, \alpha_j u_j\} \in \Theta$, *adding a node labelled* ax *with two outgoing edges to* (B_i, u_i) *and* (B_j, u_j) *where* $\mathsf{addr}(B_i)$ *and* $\mathsf{addr}(B_j)$ *is* α_i *and* α_j *resp.*

Example 6. The graph of the proof structure in example 5 is Fig. 5a.

$\mathsf{Gr}(\mathcal{S})$ are exactly the proof structures that we obtain from directly lifting the formalism of MLL proof nets à la Girard to $\mu\mathsf{MLL}^*$.

As usual in the theory of proof nets, we need a correctness criterion on the $\mu\mathsf{MLL}^*$ proof structures to exactly characterize the class of proof nets. The following correctness criterion lifts to $\mu\mathsf{MLL}^*$ a criterion first investigated by Danos and Regnier [13]. We present it in a slightly different syntax using the notion of orthogonal partitions [12, 13].

Definition 14. *Let* P_1 *and* P_2 *be partitions of a set* S. *The graph induced by* P_1 *and* P_2 *is defined as the undirected bipartite multigraph* (P_1, P_2, E) *s.t. for every* $p \in P_1$ *and* $p' \in P_2$, $(p, p') \in E$ *if* $p \cap p' \neq \emptyset$. *Finally,* P_1 *and* P_2 *are said to be* **orthogonal** *to each other if the graph induced by them is acyclic and connected.*

Definition 15. *Given a proof structure,* $\mathcal{S} = [\Theta']\{B_i^{U_i}\}_{i\in I}[\Theta]$, *define a set of* **switchings** *of* $\mathcal{S}$, $sw = \{sw_i\}_{i\in I}$ *s.t. for every* $i \in I$, $sw_i : P_i \to \{l, r\}$ *is a function over* P_i, *the* ⅋ *nodes of* $B_i^{U_i}$. *The* **switching graph** $\mathcal{S}^{sw}$ *associated with sw is formed by:*

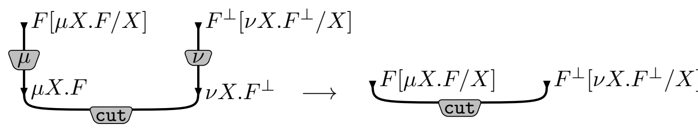
- *taking the partial syntax tree* $\{B_i^{U_i}\}_{i\in I}$ *as an undirected graph;*
- *for each* $\{B_i, B_j\} \in \Theta'$, *adding a node labelled cut with two edges to* (B_i, ϵ) *and* (B_j, ϵ);
- *for each node* $(B_i, u) \in P_i$, *removing the edge between* (B_i, u) *and* $(B_i, u \cdot sw((B_i, u)))$.

Let $\Theta_{\mathcal{S}}^{sw}$ *be the partition over* $\bigcup_{i\in I}\{\alpha_i u_i \mid \mathsf{addr}(B_i) = \alpha_i, u_i \in U_i\}$ *induced by the connected component of* $\mathcal{S}^{sw}$.

Definition 16. *A proof structure,* $\mathcal{S}$, *is said to be* **OR-correct** *if for any switching sw,* $\Theta_{\mathcal{S}}^{sw}$ *and* Θ *is orthogonal. The graph induced by* $\Theta_{\mathcal{S}}^{sw}$ *and* Θ *is called a* **correction graph** *of* $\mathcal{S}$.

Proposition 1. *Let π be a $\mu\mathsf{MLL}^*$ proof. Then $\mathsf{Deseq}(\pi)$ is an OR-correct proof structure. Conversely, given an OR-correct $\mu\mathsf{MLL}^*$ proof structure, it can be sequentialized into a $\mu\mathsf{MLL}^*$ sequent proof.*

Definition 17. $\mu\mathsf{MLL}^*$ **cut-reduction rules** *is obtained by adding the following rule to the usual cut-reduction rules for MLL proof nets:*



Proposition 2. *Cut elimination on $\mu\mathsf{MLL}^*$ proof-nets preserves correctness and is strongly normalizing and confluent.*

The proofs of propositions 1, 2 are straightforward extensions from MLL.

Example 7. The proof structure in example 5 after cut-elimination produces the proof structure in Fig. 5b.

Remark 3. Now the question is how this translates to non-wellfounded proofs. Consider the proof in example 3. Firstly observe that there is no finite proof of this sequent *i.e.* it is not provable in $\mu\mathsf{MLL}^*$. Now, if we naively translate it into a proof structure using the same recipe as def. 12 (except allowing for infinite partial syntax trees), we have

$$[\emptyset]\left\{\psi_\alpha^{\{i(li)^*r(l+r)+(il)^\omega\}}\right\}[\{\alpha i(li)^n rl, i(li)^n rr\}_{n\geq 0}].$$

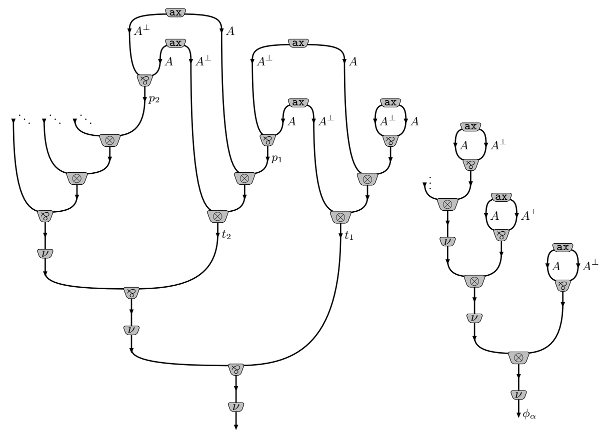
Observe that $(il)^\omega$ is not in any partition. In fact, it represents a thread in an infinite branch and must be accounted for. Hence the partition should account for the threads invariant by an infinite branch in a proof (in particular, in the example above there should be a singleton partition, $\{(il)^\omega\}$). This is also the reason we will not use the graphical presentation for non-wellfounded proof-nets since we would potentially need to join two infinite paths by a node which is unclear graph-theoretically. However we will sometimes draw the "graph" of non-wellfounded proof-nets for ease of presentation by using ellipsis points (for example Fig. 6b represents the proof-net we discussed above).

4 Infinets

We will now lift our formalism for defining proof nets for $\mu\mathsf{MLL}^*$ to $\mu\mathsf{MLL}^\infty$.

Definition 18. *A* **non-wellfounded proof structure***(NWFPS) is given by* $[\Theta']\{B_i^{U_i}\}_{i\in I}[\Theta]$*, where*

- *I is a possibly infinite index set;*
- *for every $i \in I$, B_i is a formula occurrence, $B_i^{U_i}$ is a partial syntax tree;*



(a) (b)

Fig. 6: Graph of $\mu\mathsf{MLL}^\infty$ NWFPS

- *Θ' is a (possibly empty) collection of disjoint subsets of $\{B_i\}_{i\in I}$ of the form $\{C, C^\perp\}$;*
- *Θ is a partition of $\bigcup_{i\in I}\{\alpha_i u_i \mid \mathsf{addr}(B_i) = \alpha_i, u_i \in U_i\}$ s.t. the partitions are one of the following forms:*
  - *$\{\alpha_i u_i, \alpha_j u_j\}$ s.t. u_i, u_j are finite and $\lceil (B_i, u_i)\rceil = \lceil (B_j, u_j)\rceil^\perp$.*
  - *It contains an elements of the form $\alpha_i u_i$ s.t. u is an infinite address;*
- *$\{B_i\}_{i\in I} \setminus \bigcup_{\theta\in\Theta'} \theta$ is necessarily finite.*

Intuitively, each class of Θ represents either an axiom or an infinite branch in a sequentialization. In fact, the infinite addresses in a partition correspond exactly to the infinite threads in a proof. Hence it is also straightforward to define a valid NWFPS.

Definition 19. *Let π be a pre-proof of the $\mu\mathsf{MLL}^\infty$ sequent $\vdash \Gamma$. Desequentialization of π, denoted $\mathsf{Deseq}(\pi)$, is the NWFPS, $[\Theta']\Gamma'[\Theta]$, s.t. Θ' are the cut formulas in π, $B_i^{U_i} \in \Gamma'$ where $B_i \in \Gamma$, $U_i = \mathsf{addr}(B_i)^{-1}\mathsf{addr}(\pi)$, to any finite maximal branch of π, associate a partition in Θ containing the addresses of the occurrences that are the conclusion of the corresponding axiom rule in π and to any infinite branch β of π, associate a partition in Θ such that a finite address is in the partition if it is belongs to infinitely many sequents of β and an infinite address is in the partition if all its strict prefixes belong to β. A NWFPS that is the desequentialization of a $\mu\mathsf{MLL}^\infty$ (pre-)proof is called an (valid)* **infinet***.*

Example 8. As expected from the discussion in remark 3, desequentialization of the proof in example 3 is

$$[\emptyset]\left\{\psi_\alpha^{\{i(li)^*r(l+r)+(il)^\omega\}}\right\}[\{\alpha i(li)^n rl, i(li)^n rr\}_{n\geq 0}, \{(il)^\omega\}].$$

Remark 4. The reader might think that there is discrepancy in the way desequentialization of wellfounded and non-wellfounded proofs are defined in defs. 12, 19 *resp.* Note that def. 12 can be reformulated à la def. 19 but not vice versa. However, we choose to inductively define wellfounded desequentialization since it is closer to the standard definition in proof-net theory.

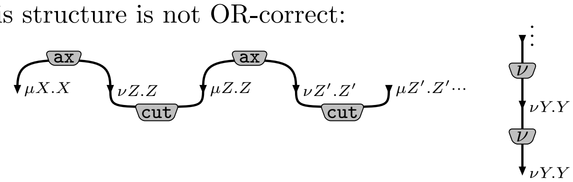
5 Correctness criteria

The OR-correctness of a NWFPS is defined as in def. 15 and def. 16 (up to the fact that the switching can be an infinite set of switching functions). However this straightforward translation is not enough to ensure soundness.

Example 9. Consider the following sequent proof with infinitely many cuts.

$$\dfrac{\dfrac{}{\mu X.X, \nu Z.Z}\,(\mathsf{ax}) \qquad \dfrac{\star \vdash \mu Z.Z, \nu Y.Y}{\vdash \mu Z.Z, \nu Y.Y}\,(\nu)}{\star \vdash \mu X.X, \nu Y.Y}\,(\mathsf{cut})$$

Observe that this structure is not OR-correct:



Consequently, we restrict ourselves to NWFPS with at most finitely many cuts. The proof structures discussed in the rest of the paper have finitely many cuts unless otherwise mentioned.

Example 10. Consider the graph of proof structure of the sequent $\vdash \nu X.X \parr (A^{\perp} \otimes (A \otimes (A^{\perp} \parr A)))$ in fig. 6a. Note that for the sake of readability, edge labels have been concealed. This proof structure is OR-correct but it is not sequentializable. Consider the $\otimes$ node labelled t_1. In any sequentialization it should be above p_1, which should be above t_2, which in turn should be above p_2 and so on. This is absurd since even in a non-wellfounded proof every rule is executed at a finite depth.

Hence we impose a "lock-free" condition (borrowing the terminology from concurrent programming) on NWFPS.

Definition 20. *Let* $[\Theta']\{B_i^{U_i}\}_{i\in I}[\Theta]$ *be a NWFPS. For any* $u_i \in U_i, u_j \in U_j$*, we say that* (u_i, u_j) *is a* **coherent pair** *if there exists* $\theta \in \Theta$ *s.t.* $\{\alpha_i u_i, \alpha_j u_j\} \subseteq \theta$*, where* $\mathsf{addr}(B_i) = \alpha_i$ *and* $\mathsf{addr}(B_j) = \alpha_j$*.*

Definition 21. *A* **switching path** *is an undirected path in a partial syntax tree s.t. it does not go consecutively through the two premises of a* $\parr$ *formula. A* **strong switching path** *is a switching path whose first edge is not the premise of a* $\parr$ *node. We denote by* $\mathsf{src}(\bullet), \mathsf{tgt}(\bullet)$ *the source and target of a switching path resp. Two switching paths* γ, γ' *are said to be* **compatible** *if* γ' *is strong and* $\mathsf{tgt}(\gamma) = \mathsf{src}(\gamma')$*.*

Proposition 3. *If γ, γ' are compatible switching paths, then their concatenation $\gamma \cdot \gamma'$ is a switching path. Furthermore, if γ is strong, then $\gamma \cdot \gamma'$ is also strong.*

The underlying undirected path of any path in a partial syntax tree is a switching path. We call such paths ***straight switching paths***. In particular, the path from any vertex, v, to the root is a straight switching path. We denote it by $\delta(v)$. By abuse of notation, we will also sometimes write $\delta((B_i, u))$ where u is infinite to mean the infinite path from the root of $B_i^{U_i}$ following u, although technically (B_i, u) is not a node per se. Observe that any straight switching path in a partial syntax tree, F^U, can be represented by a pair of words $(u, u') \in \overline{U}^2$ *s.t.* $u \sqsubset u'$. Intuitively, it means that the path is from (F, u) to (F, u').

Definition 22. *A* **switching sequence** *is a sequence $\sigma = \{\gamma_i\}_{i=1}^n$ s.t. γ_is are disjoint switching paths and for every $i \in \{1, 2, \ldots, n-1\}$, either γ_i, γ_{i+1} are compatible or they are straight and the word pairs corresponding to them, (u_i, u_i') and (u_{i+1}, u_{i+1}'), are s.t. (u_i', u_{i+1}') is a coherent pair. Two vertices, v and v', are said to be connected by the switching sequence, σ, if $\mathsf{src}(\gamma_1) = v$ and $\mathsf{tgt}(\gamma_n) = v'$. We say the switching sequence is cyclic if $\mathsf{src}(\gamma_1) = \mathsf{tgt}(\gamma_n)$.*

Proposition 4. *Let γ be a switching path in $B_j^{U_j} \in \Gamma$. Then there exists a switching sw s.t. γ is also a path in the switching graph, $\mathcal{S}^{sw}$.*

Proposition 5. *If $\mathcal{S}$ is a NWFPS containing a cyclic switching sequence, then there is switching of $\mathcal{S}$, s.t. the corresponding correction graph is contains a cycle.*

Definition 23. *Let $\mathcal{S} = [\Theta']\{B_i^{U_i}\}[\Theta]$ be a proof structure. Let $T = \{(B_i, u_i) \mid u_i \in \overline{U_i}; (B_i, u_i)$ is a $\otimes$ formula$\}$ and let $P = \{(B_i, u_i) \mid u_i \in \overline{U_i}; (B_i, u_i)$ is a $⅋$ formula$\}$. The* **dependency graph** *of $\mathcal{S}$, $D(\mathcal{S})$, is the directed graph (V, E) s.t. $V = T \uplus P$, for every $v \in V$ and $p \in P$, $(p, v) \in E$ if the premises of p are connected by a switching sequence containing v, and, for every $v, v' \in V$, $(v, v') \in E$ if $v' \in \mathsf{FL}(v)$.*

Proposition 6 (Bagnol *et al.* [5]). *If $\mathcal{S}$ is OR-correct then $D(\mathcal{S})$ is acyclic.*

From prop. 6, we can impose an order on the nodes of an OR-correct proof structure, $\mathcal{S}$, namely, $n_1 <_{D(\mathcal{S})} n_2$ if $n_1 \to n_2$ in $D(\mathcal{S})$.

Definition 24. *A NWFPS, $\mathcal{S}$, is said to be* **deeply lock-free** *if $<_{D(\mathcal{S})}$ has no infinite descending chains.*

Example 11. Consider the proof structure, $\mathcal{S} = [\emptyset]\{\nu X.X ⅋ X_\alpha^L, A \otimes B_\beta^{\{l,r\}}\}[\Theta]$ where, $L = (i(l+r))^\omega$, $\Theta = \{\{\alpha(il)^\omega, \beta l\}, \alpha \cdot (L \setminus (il)^\omega) \cup \{\beta r\}\}$.

Observe that $\mathcal{S}$ is OR-correct and deeply lock-free. But $\mathcal{S}$ cannot be sequentialized into a sequent proof, because a potential sequentialization has a $\otimes$ rule at a finite depth, then either there are no subsoccurences of $\nu X.X ⅋ X_\alpha$ in the left premise in which case A cannot reside with only the left-branch in Θ, or, there are some subsoccurences of $\nu X.X ⅋ X_\alpha$ in the left premise in which case A cannot reside with any infinite branch in Θ.

Definition 25. *A NWFPS,* $\mathcal{S} = [\Theta']\{B_i^{U_i}\}_{i\in I}[\Theta]$*, is said to be* **widely lock-free** *if there is a function* $f : \mathbb{N} \to \mathbb{N}$ *s.t. for every* $(B_i, u) \in P$ *and* $(B_j, v) \in T$ *if* $((B_i, u), (B_j, v)) \in E$, $f(|v|) \geq |u|$ *where* $D(\mathcal{S}) = (T \uplus P, E)$*. We call such a function a* **wait function** *of* $\mathcal{S}$*. A proof structure is simply called* **lock-free** *if it is both deeply and widely lock-free.*

Remark 5. The wait function of a NWFPS need not be unique (if one exists).

Proposition 7. *An infinet is an OR-correct lock-free NWFPS.*

6 Sequentialization

In this section we show that any NWFPS satisfying the correctness criterion introduced in section 5 is indeed sequentializable. Since we deal with finitely many cuts, without loss of generality, we can assume that we have cut-free proof structures due to the standard trick shown in Fig. 7.

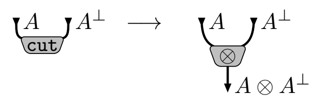


Fig. 7: Translating cuts to tensors

So, in this section, we will write NWFPS without the left component. We try to adapt the standard proof for MLL but the straightforward adaptation is not *fair* since we may never explore one branch by forever prioritizing the sequentialization of another infinite branch. We restore fairness by a time-stamping algorithm.

Definition 26. *Let* $\mathcal{S} = \Gamma[\Theta]$ *be an OR-correct NWFPS. The root,* B_i*, of a tree in* Γ *is said to be* **splitting** *if:*

- $\Gamma = \{B_i^{\epsilon}, B_j^{\epsilon}\}$,
- B_i *is a* $⅋, \mu$ *or* ν *formula, or,*
- B_i *is a* $\otimes$ *formula and there exists* Θ_1, Θ_2 *s.t.* $\Theta = \Theta_1 \uplus \Theta_2$ *and* $\mathcal{S}_1 = \Gamma_1[\Theta_1]$, $\mathcal{S}_2 = \Gamma_2[\Theta_2]$ *are OR-correct NWFPS where* $\Gamma_1 = \Gamma \setminus \{B_i^{U_i}\} \cup \{(B_i, l)^{U_l}\}$, $\Gamma_2 = \Gamma \setminus \{B_i^{U_i}\} \cup \{(B_i, r)^{U_r}\}$ *and* $U_i = lU_l + rU_r$.

Proposition 8. *Let* $\mathcal{S} = \Gamma[\Theta]$ *be an OR-correct NWFPS and* B_i *be a splitting* $\otimes$ *formula in* $\mathcal{S}$*. If* $\mathcal{S}$ *is lock-free then so is* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *as defined in def. 26.*

Dated sequentialization process. We time-stamp each node of Γ to indicate the time when it will be sequentialized. Formally, we have $(\mathcal{S}, \tau)$ where τ is a function *s.t.* $\tau : \{(B_i, u) | u \in U_i\}_{i\in I} \to \mathbb{N} \cup \{\infty\}$ where $\Gamma = \{B_i^{U_i}\}_{i\in I}$ and $\infty > n$ for all $n \in \mathbb{N}$. Define the minimal finite image, min, as

$$\min(\tau) := \min\{n \in \mathbb{N} \mid \exists i \in I, u \in U_i \textit{ s.t. } \tau((B_i, u)) = n\}.$$

We will describe the sequentialization process. Suppose we are given $\mathcal{S}(= \Gamma[\Theta], \tau)$. We maintain the following invariant:

$$\begin{array}{c}\mathcal{S} \text{ is cut-free, OR-correct and lock-free;} \\ \tau((B_i, u)) \neq \infty \text{ iff } (B_i, u) \text{ is splitting in } \mathcal{S}.\end{array} \qquad (\star)$$

Assume that Γ contains a splitting root, B_j, st. $\tau(B_j) = \min(\tau)$.

- If $\Gamma = \{B_i^\epsilon, B_j^\epsilon\}$ then we stop successfully with the proof reduced to an ax.
- If B_j is a ⅋, (co)recursively apply the sequentialization process to $\mathcal{S}_0(= \Gamma_0[\Theta], \tau_0)$ where $\Gamma_0 = \Gamma \setminus \{B_j^{U_j}\} \cup \{(B_j, l)^{U_l}, (B_j, r)^{U_r}\}$, $U_j = lU_l + rU_r$, and

$$\tau_0((B_i, u)) = \begin{cases} t & \text{if } (B_i, u) \text{ is splitting in } \mathcal{S}_0; \\ \tau((B_i, u)) & \text{otherwise.} \end{cases}$$

  where for each splitting (B_i, u), t is arbitrarily chosen to be any natural number greater than $\tau(B_j)$. We apply a ⅋ rule on the obtained proof.
- If B_j is a μ(*resp.* ν) formula, (co)recursively apply the sequentialization process to $(\mathcal{S}_0 = \Gamma_0[\Theta], \tau_0)$ where $\Gamma_0 = \Gamma \setminus \{B_j^{U_j}\} \cup \{(B_j, i)^{U_i}\}$, $U_j = iU_i$, and τ_0 is defined as above. We apply a μ (*resp.* ν) rule on the obtained proof.
- If B_j is a $\otimes$ formula we (co)recursively apply the sequentialization process to $(\mathcal{S}_1, \tau_1)$ and $(\mathcal{S}_2, \tau_2)$ where $\mathcal{S}_1, \mathcal{S}_2$ are as defined in def. 26 and τ_1, τ_2 are defined as above. We apply a $\otimes$ rule on the two obtained proofs.

Observe that the invariant $(\star)$ is maintained in this (co)recursive process. To start the sequentialization, we initialize τ by assigning arbitrary natural numbers to splitting nodes and ∞ to the other nodes.

Proposition 9. *Let T be a non-splitting conclusion in an OR-correct NWFPS. Then there exists a ⅋ formula, P, s.t. there exists disjoint switching sequences, σ, σ', from T to P which both start with a premise of T and end with a premise of P. We call (P, σ, σ') the* **witness** *for T.*

Lemma 1. *Let $\mathcal{S}$ be a cut-free OR-correct NWFPS. $\mathcal{S}$ contains a splitting root.*

Lemma 2. *The sequentialization assigns a finite natural number to every formula i.e. $\tau((B_i, u)) \neq \infty$ after some iterations of the process described above.*

Lemma. 1 crucially uses OR-correctness and lemma. 2 crucially uses lock-freeness. Lemma. 1 ensures productivity of the aforementioned sequentialization process while lemma. 2 ensures that every inference in a NWFPS is ultimately executed. From that, we conclude the following theorem.

Theorem 1. *Let $\mathcal{S} = [\Theta']\Gamma[\Theta]$ be an OR-correct lock-free NWFPS s.t. $\Theta' = \emptyset$. Then $\mathcal{S}$ is an infinet.*

Remark 6. Observe that the choice of the time-stamping function at each step of our sequentialization is non-deterministic. By considering appropriate time-stamping functions we can generate all sequentializations. The detailed study is beyond the scope of the present paper.

7 Canonicity

We started investigating proof nets for non-wellfounded proofs since we expected that the proof net formalism would quotient sequent proofs that are equivalent up to a permutation of inferences. At this point, we carry out that sanity check.

Consider the following proofs π_1 and π_1'.

$$\dfrac{\dfrac{\dfrac{\pi_2}{\vdash \Gamma, F[\mu X.F/X], A}}{\vdash \Gamma, \mu X.F, A}(\mu) \quad \dfrac{\pi_3}{\vdash B, \Delta}}{\pi_1 \vdash \Gamma, \mu X.F, A \otimes B, \Delta}(\otimes) \qquad \dfrac{\dfrac{\dfrac{\pi_2}{\vdash \Gamma, F[\mu X.F/X], A} \quad \dfrac{\pi_3}{\vdash B, \Delta}}{\vdash \Gamma, F[\mu X.F/X], A \otimes B, \Delta}(\otimes)}{\pi_1' \vdash \Gamma, \mu X.F, A \otimes B, \Delta}(\mu)$$

We say that $\pi \rightsquigarrow_{(\mu,\otimes_L)} \pi'$ if π is a proof with π_1 as a subproof at a finite depth and π' is π where π_1' has been replaced by π_1'. Observe that we can define $\rightsquigarrow_{\square}$ for every $\square \in P \times P$ where $P = \{\mu, \nu, ⅋, \otimes_\star, \mathsf{cut}_\star \mid \star \in \{L, R\}\}$. Let $\sim_{\blacksquare} = \bigcup_{\square \in S} \rightsquigarrow_{\square}$.

Observe that the usual notion of equivalence by permutation, *viz.* $\sim = (\sim_{\blacksquare})^*$ does not characterize equivalence by infinets. Consider the following two proofs, π_1 and π_2, *s.t.* $\pi_1 \not\sim \pi_2$ which have the same infinet,

$$[\emptyset]\{\mu X.X_\alpha^{\{i^\omega\}}, \nu X.X_\beta^{\{i^\omega\}}\}[\{\{\alpha i^\omega, \beta i^\omega\}\}].$$

$$\dfrac{\dfrac{\pi_1 \vdash \mu X.X, \nu X.X}{\vdash \mu X.X, \nu X.X}(\mu)}{\pi_1 \vdash \mu X.X, \nu X.X}(\nu) \qquad \dfrac{\dfrac{\pi_2 \vdash \mu X.X, \nu X.X}{\vdash \mu X.X, \nu X.X}(\nu)}{\pi_2 \vdash \mu X.X, \nu X.X}(\mu)$$

Suppose we allow infinite permutations. We say that $\pi(\sim_{\blacksquare})^\omega\pi'$ if there exists a proof π'' (not necessarily different from π, π') and two sequence of proofs, $\{\pi_i\}_{i=0}^{\infty}$ and $\{\pi_i'\}_{i=0}^{\infty}$, *s.t.* $\pi_0 = \pi$, $\pi_0' = \pi'$, for every i, $\pi_i \sim_{\blacksquare} \pi_{i+1}$ and $\pi_i' \sim_{\blacksquare} \pi_{i+1}'$, and $d(\pi_i, \pi'') \to 0$, $d(\pi_i', \pi'') \to 0$ as $i \to \infty$. Consider the following proofs.

$$\dfrac{\dfrac{\pi}{\vdash A} \quad \dfrac{\vdots}{\vdash B, \nu Y.Y}(\nu)}{\vdash A \otimes B, \nu Y.Y}(\otimes) \quad (\sim_{\blacksquare})^\omega \quad \dfrac{\vdots}{\vdash A \otimes B, \nu Y.Y}(\nu) \quad (\sim_{\blacksquare})^\omega \quad \dfrac{\dfrac{\pi'}{\vdash A} \quad \dfrac{\vdots}{\vdash B, \nu Y.Y}(\nu)}{\vdash A \otimes B, \nu Y.Y}(\otimes)$$

Note that equating these proofs is absurd since π and π' can have different computation behaviour (for example, $A = (X^\perp ⅋ X^\perp) ⅋ (X \otimes X)$ and π corresponds to true while π' corresponds to false). To exactly capture equivalence by infinets we need to refine this equivalence. To do that we introduce the notion of an active occurrence. We say that for a permutation step $\rightsquigarrow_{(r_i, r_i')}$, the formula occurrence F_i introduced by the rule r_i' is the ***active occurrence*** in that step.

Given two node-labelled trees T_1 and T_2, we define $d(T_1, T_2) = \frac{1}{2^\delta}$ where δ is the minimal depth of the nodes at which they differ. We say that $\pi(\sim_{\blacksquare})^\omega_{\mathsf{fair}}\pi'$ if there exists a sequence of proofs $\{\pi_i\}_{i=0}^{\infty}$ *s.t.* $\pi_0 = \pi$, for every i, $\pi_i \rightsquigarrow_{(r_i, r_i')} \pi_{i+1}$, the sequence of addresses of the active occurrences occurring infinitely often is empty, *i.e.* $\mathsf{Inf}(\{\mathsf{addr}(F_i)\}_{i=0}^{\infty}) = \emptyset$, and $d(\pi_i, \pi') \to 0$ as $i \to \infty$. Let $\sim^\infty = (\sim_{\blacksquare})^* \cup (\sim_{\blacksquare})^\omega_{\mathsf{fair}}$.

Proposition 10. $\pi_1 \sim^\infty \pi_2$ *iff* $\mathsf{Deseq}(\pi_1) = \mathsf{Deseq}(\pi_2)$.

8 Cut Elimination

In this section we provide cut elimination results albeit with two crucial restrictions: firstly, we consider only finitely many cuts as in the rest of the paper and secondly, we consider proofs with no axioms and no atoms. An infinet

$S = [\Theta']\Gamma[\Theta]$ is said to be η^∞***-expanded*** if it does not contain any axioms or atoms *i.e.* every $\theta \in \Theta$ contains only infinite addresses. Any infinet can be made η^∞-expanded in a way akin to η-expansion of axioms in MLL. There are two issues to be resolved to obtain the result: first, to specify the notion of a normal form and second, formulate how to reach that.

Proposition 11. *Let $\mathcal{S} = [\Theta']\Gamma[\Theta]$ be an η^∞-expanded infinet. Let $\{C, C^\perp\} \in \Theta$ and $B_i^{U_i}, B_j^{U_j} \in \Gamma$ s.t. $B_i = C = {B_j}^\perp$. Then, $U_i = {U_j}^\perp$ i.e. $u \in U_i$ iff $u^\perp \in U_j$.*

Proof (Sketch). Since $B_i = {B_j}^\perp$, their syntax trees are orthogonal. Since $\mathcal{S}$ is η^∞-expanded, U_i(resp. U_j) is actually the full syntax tree. Hence $U_i = {U_j}^\perp$.

Definition 27. *Let $\mathcal{S}_0 = [\Theta_0']\Gamma_0[\Theta_0]$ be a η^∞-expanded infinet. Let $\{C, C^\perp\} \in \Theta_0'$ and $B_i^{U_i}, B_j^{U_j} \in \Gamma$ s.t. $B_i = C = {B_j}^\perp$. A* **big-step** $\{C, C^\perp\}$ **elimination** *on $\mathcal{S}_0$ produces non-wellfounded proof-structure $\mathcal{S}_1 = [\Theta_1']\Gamma_1[\Theta_1]$ where,*

- $\Theta_1' = \Theta_0' \setminus \{\{C, C^\perp\}\}$
- $\Gamma_1 = \Gamma_0 \setminus \{B_i^{U_i}, B_j^{U_j}\}$
- *If $\theta \in \Theta_0$ s.t. $\theta \cap U_i = \emptyset$ and $\theta \cap U_j = \emptyset$, then $\theta \in \Theta_1$. If $u \in \theta \cap U_i$ then $\theta \cup \theta' \setminus \{u, u^\perp\} \in \Theta_1$ where $\theta' \in \Theta_0$ and $u^\perp \in \theta' \cap U_j$.*

Remark 7. Def. 27 is well-defined because of prop. 11.

Proposition 12. *A big-step operation on a valid infinet produces a valid infinet.*

Given $\mathcal{S} = [\Theta']\Gamma[\Theta]$, an η^∞-expanded infinet, we can extend the definition of a big-step $\{C, C^\perp\}$ elimination on $\mathcal{S}$, for any $\{C, C^\perp\} \in \Theta'$, to a big-step $\mathcal{C}$ elimination on $\mathcal{S}$, for $\mathcal{C} \subseteq \Theta'$. We call the big-step Θ' elimination on $\mathcal{S}$ the ***normal form*** of $\mathcal{S}$ and denote it by $[\![\mathcal{S}]\!]$.

The idea now is to show that local cut-elimination indeed produces in the limit the normal form defined above. For this we need to define a metric, d, over infinets with the same normal form so that we can formalize the limit of infinite reduction sequences. See [15] for details.

Lemma 3. *The set of all valid infinets with the same normal form together with the distance, d, forms a metric space.*

We can now define the limit of an infinite sequence of valid infinets with the same normal form in the standard way: we say that $\{\mathcal{S}_i\}_{i=0}^\infty$ converges to $\mathcal{S}$ if $d(\mathcal{S}_i, \mathcal{S}) \to 0$ as $i \to \infty$.

Definition 28. *A sequence of infinets, $\{\mathcal{S}_i\}_{i=0}^\infty$, is called a* **reduction sequence** *if for every $i > 0$, $\mathcal{S}_i \to \mathcal{S}_{i+1}$ by the cut reduction rules in def. 17. A reduction sequence is said to be* **fair** *if for every i, for every cut $\{C, C^\perp\}$ in $\mathcal{S}_i$, there is a $j > i$ such that C' is a suboccurrence of C where $\{C', C'^\perp\}$ is the cut being reduced in the step $\mathcal{S}_j \to \mathcal{S}_{j+1}$.*

Theorem 2. *Let* $\{\mathcal{S}_i\}_{i=0}^{\infty}$ *be a fair reduction sequence s.t.* $\mathcal{S}_0$ *is valid. Then, it converges to* $[\![\mathcal{S}_0]\!]$.

Corollary 1. *If two reduction sequences starting from a valid* η^∞*-expanded infinet,* $\mathcal{S}$*, converges to* $\mathcal{S}_1$ *and* $\mathcal{S}_2$*, then all fair reduction sequences starting from* $\mathcal{S}_1$ *and* $\mathcal{S}_2$ *resp. converge to* $[\![\mathcal{S}]\!]$.

9 Conclusion

In this paper, we introduced infinitary proof-nets for $\mu\mathsf{MLL}^\infty$. We defined a correctness criterion and showed its soundness and completeness in characterizing those proof structures which come from non-wellfounded sequent (pre)proofs. We also gave a partial cut elimination result. For the non-wellfounded correctness criterion, we extended the Danos-Regnier criterion from the finitary case. Other more efficient criteria (like the parsing criterion [21]) are impossible to adapt since any reasonable operation over non-wellfounded structures should necessarily be of a bottom-up nature (unlike the parsing criterion).

Related and future works. The closest works we know of are Montelatici's polarized proof nets with cycles [25] and Mellies' work on higher-order parity automata [24] which considers a $\lambda\mathsf{Y}$-calculus and an infinitary λ-calculus endowed with parity conditions, therefore quotienting some of the non-determinism of sequent-calculus albeit in the case of intuitionistic logic. Our work is a first step in developing a general theory of non-wellfounded and circular proof-nets:

– We plan to strengthen the correctness criterion to capture proofs with infinitely many cuts and possibly extend our formalism to $\mu\mathsf{MALL}^\infty$.
– We plan to carry an investigation of the notion of circularity in infinets: while one can capture circular proofs as finitely representable infinets, there are non-wellfounded proofs which are not circular but which have finitely representable desequentialization. The simplest example is the proof of $\vdash \nu X.X \parr X$ which contain sequents of unbounded size and are therefore not circular. Not only is the study circular infinets interesting from a programming perspective but also it would make it possible to do a complexity analyses of our methods of checking correctness, sequentialization and cut-elimination.

Acknowledgement. We are indebted to anonymous reviewers for providing insightful comments which has immensely enhanced the presentation of the paper.

References

1. David Baelde. On the proof theory of regular fixed points. In Martin Giese and Arild Waaler, editors, *Automated Reasoning with Analytic Tableaux and Related Methods, 18th International Conference, TABLEAUX 2009, Oslo, Norway, July 6-10, 2009. Proceedings*, volume 5607 of *Lecture Notes in Computer Science*, pages 93–107. Springer, 2009.

2. David Baelde. Least and greatest fixed points in linear logic. *ACM Transactions on Computational Logic (TOCL)*, 13(1):2, 2012.
3. David Baelde, Amina Doumane, and Alexis Saurin. Infinitary proof theory: the multiplicative additive case. In *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 fuer Informatik, 2016.
4. 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.
5. Marc Bagnol, Amina Doumane, and Alexis Saurin. On the dependencies of logical rules. In Andrew M. Pitts, editor, *Foundations of Software Science and Computation Structures - 18th International Conference, FoSSaCS 2015, Held as Part of the European Joint Conferences on Theory and Practice of Software, ETAPS 2015, London, UK, April 11-18, 2015. Proceedings*, volume 9034 of *Lecture Notes in Computer Science*, pages 436–450. Springer, 2015.
6. R.F. Blute, J.R.B. Cockett, R.A.G. Seely, and T.H. Trimble. Natural deduction and coherence for weakly distributive categories. *Journal of Pure and Applied Algebra*, 113(3):229 – 296, 1996.
7. James Brotherston. *Sequent Calculus Proof Systems for Inductive Definitions*. PhD thesis, University of Edinburgh, November 2006.
8. James Brotherston and Alex Simpson. Complete sequent calculi for induction and infinite descent. In *22nd IEEE Symposium on Logic in Computer Science (LICS 2007), 10-12 July 2007, Wroclaw, Poland, Proceedings*, pages 51–62. IEEE Computer Society, 2007.
9. James Brotherston and Alex Simpson. Sequent calculi for induction and infinite descent. *J. Log. Comput.*, 21(6):1177–1216, 2011.
10. Pierre Clairambault. Least and greatest fixpoints in game semantics. In *FOSSACS*, volume 5504 of *Lecture Notes in Computer Science*, pages 16–31. Springer, 2009.
11. Pierre-Louis Curien. Introduction to linear logic and ludics, part ii, 2006.
12. Vincent Danos. *Une application de la logique linéaire à l'étude des processus de normalisation (principalement du λ-calcul)*. Thèse de doctorat, Université Denis Diderot, Paris 7, 1990.
13. Vincent Danos and Laurent Regnier. The structure of multiplicatives. *Archive for Mathematical Logic*, 28:181–203, 1989.
14. Christian Dax, Martin Hofmann, and Martin Lange. A proof system for the linear time μ-calculus. In S. Arun-Kumar and Naveen Garg, editors, *FSTTCS 2006: Foundations of Software Technology and Theoretical Computer Science, 26th International Conference, Kolkata, India, December 13-15, 2006, Proceedings*, volume 4337 of *Lecture Notes in Computer Science*, pages 273–284. Springer, 2006.
15. Abhishek De and Alexis Saurin. Infinets: The parallel syntax for non-wellfounded proof-theory. working paper or preprint, June 2019.
16. Amina Doumane. *On the infinitary proof theory of logics with fixed points. (Théorie de la démonstration infinitaire pour les logiques à points fixes)*. PhD thesis, Paris Diderot University, France, 2017.
17. Amina Doumane, David Baelde, Lucca Hirschi, and Alexis Saurin. Towards Completeness via Proof Search in the Linear Time mu-Calculus. Accepted for publication at LICS, January 2016.

18. Jérôme Fortier and Luigi Santocanale. Cuts for circular proofs: semantics and cut-elimination. In Simona Ronchi Della Rocca, editor, *Computer Science Logic 2013 (CSL 2013), CSL 2013, September 2-5, 2013, Torino, Italy*, volume 23 of *LIPIcs*, pages 248–262. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2013.
19. Jean-Yves Girard. Linear logic. *Theor. Comput. Sci.*, 50:1–102, 1987.
20. Jean-Yves Girard. Proof-nets: the parallel syntax for proof theory. 1995.
21. Stefano Guerrini. A linear algorithm for mll proof net correctness and sequentialization. *Theor. Comput. Sci.*, 412(20):1958–1978, April 2011.
22. Roope Kaivola. A simple decision method for the linear time mu-calculus. In Jörg Desel, editor, *Structures in Concurrency Theory*, Workshops in Computing, pages 190–204. Springer London, 1995.
23. Dexter Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.
24. Paul-André Melliès. Higher-order parity automata. In *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2017, Reykjavik, Iceland, June 20-23, 2017*, pages 1–12. IEEE Computer Society, 2017.
25. Raphaël Montelatici. Polarized proof nets with cycles and fixpoints semantics. In Martin Hofmann, editor, *Typed Lambda Calculi and Applications, 6th International Conference, TLCA 2003, Valencia, Spain, June 10-12, 2003, Proceedings.*, volume 2701 of *Lecture Notes in Computer Science*, pages 256–270. Springer, 2003.
26. David Park. Fixpoint induction and proofs of program properties. *Machine intelligence*, 5(59-78):5–3, 1969.
27. Luigi Santocanale. A calculus of circular proofs and its categorical semantics. In Mogens Nielsen and Uffe Engberg, editors, *Foundations of Software Science and Computation Structures*, volume 2303 of *Lecture Notes in Computer Science*, pages 357–371. Springer, 2002.
28. Igor Walukiewicz. On completeness of the mu-calculus. In *LICS*, pages 136–146. IEEE Computer Society, 1993.
29. Igor Walukiewicz. Completeness of Kozen's axiomatisation of the propositional mu-calculus. In *Proceedings, 10th Annual IEEE Symposium on Logic in Computer Science, San Diego, California, USA, June 26-29, 1995*, pages 14–24. IEEE Computer Society, 1995.