

Non-Wellfounded and Cyclic Proofs for LTL: A Syntactic Correspondence with Linear Nested Sequents (Full Version)

Tim S. Lyon

TU Dresden
Nöthnitzer Straße 46, 01187 Dresden, Germany
timothy_stephen.lyon@tu-dresden.de

Lukas Zenger*

Peking University
5 Yiheyuan Rd., Haidian District, 100871 Beijing, China
lukas.zenger@pku.edu.cn

We introduce the formalism of non-wellfounded and cyclic linear nested sequent calculi, developing concrete systems for linear temporal logic (LTL). The paper addresses two central problems, which we call *cycle recognition* and *unraveling*. Cycle recognition concerns identifying cycles in non-wellfounded proofs in order to extract corresponding cyclic proofs, while unraveling studies the converse transformation, from cyclic proofs to non-wellfounded ones. Although these processes are well understood for Gentzen sequents, they have received little attention for more expressive sequent formalisms and become more challenging in the linear nested sequent setting. To address cycle recognition, we show the completeness of non-wellfounded proofs relative to a particular normal form exhibiting a property we call *saturation recurrence*, which enables the systematic extraction of cyclic proofs. To address unraveling, we introduce a specialized procedure that shifts rule applications forward along linear nested sequents, allowing non-wellfounded proofs to be reconstructed from cyclic ones. Overall, our work provides new proof-theoretic techniques for cycle recognition and unraveling in expressive multisequent formalisms.

1 Introduction

Modal fixed-point and program logics extend modal languages with operators for expressing inductive and coinductive definitions. Due to these constructs, such logics often require reasoning principles that go beyond finite, well-founded proofs [2, 14, 25]. One approach to obtain cut-free and analytic proof systems for such logics is to employ *non-wellfounded proofs*, which take the form of non-wellfounded trees and rely on global soundness conditions to ensure correctness. Such proofs can be viewed as a formalization of reasoning by infinite descent. The relationship between induction, infinite descent, and non-wellfounded proofs has been studied in particular by Brotherston [8]. He discusses the sense in which induction and infinite descent can be regarded as equivalent proof principles, while emphasizing that the precise correspondence depends on the underlying formalization. When non-wellfounded proofs have the shape of a *regular tree*—containing only finitely many distinct subtrees—they can be folded into finite proofs in which certain leaves are linked back to internal nodes, thereby creating cycles. This gives rise to *cyclic proofs*, which provide a finite representation of non-wellfounded proofs and are amenable to automated proof-search [33] and to computing interpolants [3, 34].

In the setting of *Gentzen sequents* (pairs of (multi)sets of logical formulae), the relationship between non-wellfounded and cyclic proofs is well understood. As alluded to above, if a non-wellfounded proof can be transformed into a regular tree, then it is straightforwardly convertible into a cyclic proof; conversely, one can unravel a cyclic proof into a non-wellfounded proof. By contrast, the correspondence between non-wellfounded and cyclic proofs has received far less attention in the context of *multisequents*.

*Supported by the Swiss National Science Foundation, Grant P500-2_235387.

Multisequents are generalizations of Gentzen sequents, obtained by embedding Gentzen sequents into more complex data structures. Examples include hypersequents (multisets of Gentzen sequents) [6, 29], linear nested sequents (lines of Gentzen sequents) [22, 16], and labeled sequents (graphs of Gentzen sequents) [35, 36]. For a survey of multisequent systems and their relationships, see [18]. Multisequents were introduced over the past few decades to provide cut-free and analytic sequent-style systems for modal and related logics—particularly for logics where a cut-free Gentzen sequent calculus was not known to exist. Multisequent systems often enjoy properties that can make them better suited for proof analysis and applications than traditional Gentzen systems. For instance, many multisequent calculi consist entirely of invertible rules, feature symmetric rules that facilitate cut-elimination, and use structures conducive to counter-model extraction.

Research on non-wellfounded multisequent calculi remains nascent, with only a handful of works on non-wellfounded hypersequent or labeled sequent systems. A set of non-wellfounded and cyclic hypersequent systems for modal logics with the master modality capturing different frame conditions was provided by Rooduijn [31, 32], while Das and Girlando provided a cyclic hypersequent system for transitive closure logic [12]. Docherty and Rowe provided a cut-free non-wellfounded labeled system for propositional dynamic logic as well as a cyclic labeled system that includes cut [13]. More recently, Das et al. [11] presented non-wellfounded labeled calculi for classical and intuitionistic Gödel-Löb logic, with Aguilera and Pacheco subsequently providing a cyclic version of the intuitionistic system [4]. Furthermore, Afshari et al. [1] provided non-wellfounded calculi for intuitionistic temporal logics which employ formula nesting, and their work was improved upon by Menéndez Turata [24], who obtained a cut-free labeled cyclic proof system. To the best of our knowledge, for multisequents more complex than hypersequents, the works by Aguilera and Pacheco and by Menéndez Turata serve as the only examples of cut-free cyclic multisequent systems.

Our aim in this paper is to advance this line of research by providing a deeper understanding of cyclic proofs in the context of multisequents. In particular, we focus on non-wellfoundedness and cyclicity within the *linear nested sequent (LNS)* formalism. Linear nested sequents, which consist of ordered lines of Gentzen sequents, generalize hypersequents. This formalism was introduced by Lellmann [16] and builds on the 2-sequent framework originally developed by Masini [22, 23]. As a case study, we provide cut-free non-wellfounded and cyclic linear nested sequent calculi for linear temporal logic (LTL), which is our first contribution in this paper. We then study and address two central problems, which we call *cycle recognition* and *unraveling*. The cycle recognition problem asks how to identify cycles in non-wellfounded proofs in order to construct a corresponding cyclic proof, whereas the unraveling problem concerns the reverse process: transforming a cyclic proof into a non-wellfounded one.

Detecting cycles in non-wellfounded multisequent proofs is particularly challenging because multisequents employ richer structures than traditional Gentzen sequents. Repetitions do not recur verbatim along infinite branches; instead, structures tend to grow. Consequently, cycles must be identified not as identical repeated sequents, but as recurring growth patterns. We resolve this problem by showing completeness with respect to non-wellfounded proofs satisfying the *saturation recurrence property*. This guarantees that repetitive growth patterns can be detected along infinite branches, thereby enabling the extraction of cyclic proofs. This constitutes our second contribution.

Unraveling cyclic proofs in this setting is likewise non-trivial since the standard unraveling procedure is inapplicable due to the more complex structures used in proofs. As a third contribution, we show how this problem can be addressed by introducing a specialized unraveling procedure that shifts rule applications forward along linear nested sequents. This makes it possible to reconstruct a non-wellfounded proof in which linear nested sequents exhibit repetitive growth along infinite branches. Since the non-wellfounded system for LTL is proven sound and complete, the above two transformations

yield soundness and completeness of the cyclic system as a corollary.

LTL and Related Systems. Linear temporal logic (LTL) is an important program logic introduced by Pnueli [28] for reasoning about dynamically evolving systems. The logic is equipped with the temporal operators *next* X and *until* U , which enable the specification of infinite behaviors over execution traces. This gives LTL considerable expressive power while still admitting effective decision procedures. Due to these favorable properties, LTL is widely used in model checking and formal verification, where properties such as safety (e.g., ‘no bad state will be reached’) and liveness (e.g., ‘a good state will eventually be reached’) can be specified and automatically checked.

A variety of sequent calculi have been proposed for LTL. Paech introduced a Gentzen-style system for the logic [27]; however, this system contains an induction rule that is an instance of cut, and is therefore not cut-free. To address this issue, Br nnler and Lange [10] later developed a cut-free cyclic Gentzen calculus that employs annotations—called *histories*—to detect cycles in proofs. A variant of Br nnler and Lange’s calculus was discussed by Kokkinis and Studer [15] for the unary fragment of LTL, in which weakening is syntactically admissible. More recently, Alonderis et al. [5] provided a non-annotated variant of the Br nnler-Lange calculus for full LTL. Boretti [7] introduced an infinitary labeled system for LTL, which relies on an ω -rule with infinitely many premises, rather than on non-wellfounded proofs, to establish validity. On the satisfiability side, Reynolds [30] provided a one-pass tree-shaped tableau for LTL satisfiability that employs the LOOP and PRUNE rules to detect satisfiable and unsatisfiable cycles in tableau branches, respectively. While the cycle detection underlying Reynolds’ tableau is reminiscent of the cycle recognition studied in the present work, the precise relationship between these approaches remains open.

Since our linear nested sequent systems fall within the paradigm of non-wellfounded proof theory, they are most closely related to [10, 15, 5]. However, they differ in that we employ the richer formalism of linear nested sequents rather than traditional Gentzen sequents, our calculus is *formula-driven* (i.e., it does not contain any structural rules), and all rules are invertible. We remark that our choice of formalism is motivated by two considerations. First, syntactic cut-elimination remains open for all existing non-wellfounded Gentzen calculi for LTL [10, 15, 5], and we conjecture that the additional structural expressiveness of linear nested sequents provides a pathway to resolving this problem. Second, our work contributes to the broader goal of developing a general theory of non-wellfounded and cyclic multisequent systems: the cycle recognition and unraveling techniques introduced here serve as a foundation for extending these methods to other expressive formalisms beyond the linear nested setting.

Outline of Paper. In Section 2 we introduce the preliminaries for LTL. Section 3 introduces our non-wellfounded and cyclic linear nested sequent systems for LTL. Section 4 establishes soundness and Section 5 establishes completeness of the non-wellfounded calculus. Section 6 provides our solutions to the cycle recognition and unraveling problems, establishing a syntactic correspondence between non-wellfounded and cyclic linear nested sequent proofs, which entails the soundness and completeness of the cyclic system. Finally, in Section 7, we conclude and discuss future work.

2 Logical Preliminaries: Linear Temporal Logic

Let $\text{Atm} := \{p, q, r, \dots\}$ be a denumerable set of (*propositional*) *atoms*. We define the language $\mathcal{L}$ to be the collection of all formulae A generated via the following grammar in BNF:

$$A, B ::= p \mid \perp \mid (A \rightarrow B) \mid XA \mid (AUB)$$

with $p \in \text{Atm}$. We use $A, B, C, \dots$ to denote formulae from $\mathcal{L}$. The formula $\mathbf{X}A$ expresses ‘ A is true in the *next* time-step’ while $A \mathbf{U} B$ expresses ‘ A is true *until* B is true (and B eventually becomes true)’. We have opted to use a minimal signature for LTL to simplify the presentation of our linear nested sequent systems in the subsequent section. We define $\neg A := A \rightarrow \perp$, $\top := \neg \perp$, $A \vee B := \neg A \rightarrow B$, $A \wedge B := \neg(A \rightarrow \neg B)$, and $A \leftrightarrow B := (A \rightarrow B) \wedge (B \rightarrow A)$. For a finite set of formulae Γ , $\bigwedge \Gamma$ and $\bigvee \Gamma$ are defined as usual, where

$$\bigwedge \emptyset := \top, \quad \bigvee \emptyset := \perp.$$

Other temporal operators are defined in terms of $\mathbf{U}$ as follows: $\mathbf{F}A := \top \mathbf{U} A$ expresses ‘ A is *eventually* true’, $\mathbf{G}A := \neg \mathbf{F} \neg A$ expresses ‘ A is *henceforth* true’, and $A \mathbf{R} B := \neg(\neg A \mathbf{U} \neg B)$ expresses ‘ A *releases* B ’. A *complex formula* is a formula of the form $A \rightarrow B$, $\mathbf{X}A$, or $A \mathbf{U} B$. We let $\text{sufo}(A)$ denote the set of all *subformulae* of A , defined in the usual way, and call a formula B a *subformula* of A iff $B \in \text{sufo}(A)$. For a set of formulae Γ , we define $\text{sufo}(\Gamma) := \bigcup_{A \in \Gamma} \text{sufo}(A)$.

Definition 2.1. A state sequence is an infinite sequence of states $\bar{\sigma} = (\sigma_0, \sigma_1, \dots)$, where each state $\sigma_i \subseteq \text{Atm}$. The satisfaction of a formula A on $\bar{\sigma}$ at the time point $i \geq 0$, written $\bar{\sigma}, i \models A$, is defined by:

- $\bar{\sigma}, i \models p$ iff $p \in \sigma_i$;
- $\bar{\sigma}, i \not\models \perp$;
- $\bar{\sigma}, i \models A \rightarrow B$ iff $\bar{\sigma}, i \not\models A$ or $\bar{\sigma}, i \models B$;
- $\bar{\sigma}, i \models \mathbf{X}A$ iff $\bar{\sigma}, i+1 \models A$;
- $\bar{\sigma}, i \models A \mathbf{U} B$ iff there is a $j \geq i$ such that $\bar{\sigma}, j \models B$ and for every $i \leq n < j$, $\bar{\sigma}, n \not\models A$;
- $\bar{\sigma} \models A$ iff $\bar{\sigma}, 0 \models A$.

We say that A is *valid*, written $\models A$, iff for each state sequence $\bar{\sigma}$ we have $\bar{\sigma} \models A$; otherwise, a formula A is *invalid*, written $\not\models A$. We define the logic LTL to be the set of all valid formulae from $\mathcal{L}$.

Example 2.2. It is straightforward to verify, using the semantic clauses defined above, that for any formulae $A, B \in \mathcal{L}$, $A \mathbf{U} B \leftrightarrow B \vee (A \wedge \mathbf{X}(A \mathbf{U} B))$ is valid. The formula $B \vee (A \wedge \mathbf{X}(A \mathbf{U} B))$ is called the *unfolding* of $A \mathbf{U} B$.

3 Linear Nested Sequents

This section introduces linear nested sequents as well as the non-wellfounded and cyclic calculi for LTL. To ensure soundness of the cyclic calculus, we make use of focusing to keep track of good formula traces.

An *annotated formula* is a tuple (A, a) where A is a formula and $a \in \{u, f\}$ is an *annotation*. The annotation f designates that the formula is *in focus* while u designates that the formula is *unfocused*. We will write annotated formulae as A^a and define $A^u := A$ for simplicity. That is, we take unfocused formulae to just be regular formulae; this will simplify our presentation. We do not use parentheses when displaying annotated formulae, e.g., $A \mathbf{U} B^f$ should be read as $(A \mathbf{U} B)^f$.

For a set of annotated formulae Γ , we define $\Gamma^- := \{A \mid A^a \in \Gamma\}$, i.e., Γ^- is the set of formulae obtained by removing annotations. Moreover, we define $\Gamma^u := \{A^u \mid A^a \in \Gamma\}$ to be the set of annotated formulae obtained from Γ by changing any annotation to u . A *Gentzen sequent* is an expression of the form $\Gamma \vdash \Delta$ such that (1) Γ and Δ are finite sets of annotated formulae, (2) at most one formula in Γ is in focus, (3) if a formula in Γ is in focus, then it is of the form $A \mathbf{U} B$, and (4) every formula in Δ is unfocused. We call Γ the *antecedent* and Δ the *consequent* of $\Gamma \vdash \Delta$. If $\Gamma \vdash \Delta$ contains a formula in focus, then we call $\Gamma \vdash \Delta$ *focused*, and otherwise we call $\Gamma \vdash \Delta$ *unfocused*.

$$\begin{array}{c}
\frac{}{\mathcal{G} \parallel \Gamma, p \vdash p, \Delta \parallel \mathcal{H}} \text{id} \quad \frac{}{\mathcal{G} \parallel \Gamma, \perp \vdash \Delta \parallel \mathcal{H}} \perp\text{L} \\
\\
\frac{\mathcal{G} \parallel \Gamma, A \vdash B, \Delta \parallel \mathcal{H}}{\mathcal{G} \parallel \Gamma \vdash A \rightarrow B, \Delta \parallel \mathcal{H}} \rightarrow\text{R} \quad \frac{\mathcal{G} \parallel \Gamma, B \vdash \Delta \parallel \mathcal{H} \quad \mathcal{G} \parallel \Gamma \vdash A, \Delta \parallel \mathcal{H}}{\mathcal{G} \parallel \Gamma, A \rightarrow B \vdash \Delta \parallel \mathcal{H}} \rightarrow\text{L} \\
\\
\frac{\mathcal{G} \parallel \Gamma \vdash \Delta \parallel \Sigma, A \vdash \Pi \parallel \mathcal{H}}{\mathcal{G} \parallel \Gamma, \chi A \vdash \Delta \parallel \Sigma \vdash \Pi \parallel \mathcal{H}} \chi\text{L}_1 \quad \frac{\mathcal{G} \parallel \Gamma \vdash \Delta \parallel A \vdash}{\mathcal{G} \parallel \Gamma, \chi A \vdash \Delta} \chi\text{L}_2 \quad \frac{\mathcal{G} \parallel \Gamma^u, AUB^u \vdash \Delta \parallel \mathcal{H}}{\mathcal{G} \parallel \Gamma^u, AUB^f \vdash \Delta \parallel \mathcal{H}} \text{u} \\
\\
\frac{\mathcal{G} \parallel \Gamma \vdash \Delta \parallel \Sigma \vdash A, \Pi \parallel \mathcal{H}}{\mathcal{G} \parallel \Gamma \vdash \chi A, \Delta \parallel \Sigma \vdash \Pi \parallel \mathcal{H}} \chi\text{R}_1 \quad \frac{\mathcal{G} \parallel \Gamma \vdash \Delta \parallel \vdash A}{\mathcal{G} \parallel \Gamma \vdash \chi A, \Delta} \chi\text{R}_2 \quad \frac{\mathcal{G} \parallel \Gamma^u, AUB^f \vdash \Delta \parallel \mathcal{H}}{\mathcal{G} \parallel \Gamma^u, AUB^u \vdash \Delta \parallel \mathcal{H}} \text{f} \\
\\
\frac{\mathcal{G} \parallel \Gamma, B \vdash \Delta \parallel \Sigma \vdash \Pi \parallel \mathcal{H} \quad \mathcal{G} \parallel \Gamma, A \vdash \Delta \parallel \Sigma, AUB^a \vdash \Pi \parallel \mathcal{H}}{\mathcal{G} \parallel \Gamma, AUB^a \vdash \Delta \parallel \Sigma \vdash \Pi \parallel \mathcal{H}} \text{uL}_1 \\
\\
\frac{\mathcal{G} \parallel \Gamma \vdash A, B, \Delta \parallel \Sigma \vdash \Pi \parallel \mathcal{H} \quad \mathcal{G} \parallel \Gamma \vdash B, \Delta \parallel \Sigma \vdash AUB, \Pi \parallel \mathcal{H}}{\mathcal{G} \parallel \Gamma \vdash AUB, \Delta \parallel \Sigma \vdash \Pi \parallel \mathcal{H}} \text{uR}_1 \\
\\
\frac{\mathcal{G} \parallel \Gamma, B \vdash \Delta \quad \mathcal{G} \parallel \Gamma, A \vdash \Delta \parallel AUB^a \vdash}{\mathcal{G} \parallel \Gamma, AUB^a \vdash \Delta} \text{uL}_2 \quad \frac{\mathcal{G} \parallel \Gamma \vdash A, B, \Delta \quad \mathcal{G} \parallel \Gamma \vdash B, \Delta \parallel \vdash AUB}{\mathcal{G} \parallel \Gamma \vdash AUB, \Delta} \text{uR}_2
\end{array}$$

Figure 1: Inference Rules LNS_{LTL} .

A *linear nested sequent* (LNS) is an expression of the form $\mathcal{G} := \Gamma_0 \vdash \Delta_0 \parallel \dots \parallel \Gamma_n \vdash \Delta_n$ such that (1) $\Gamma_i \vdash \Delta_i$ is a Gentzen sequent for $0 \leq i \leq n$ called the *i-component* of $\mathcal{G}$ (or simply *component* of $\mathcal{G}$ if the index i is not of importance), and (2) at most one component of $\mathcal{G}$ is focused. LNSs are particularly natural for LTL, where every state in a state sequence has a unique successor. Intuitively, an LNS may thus be regarded as a finite ‘snapshot’ of a state sequence, with each Gentzen sequent corresponding to one state and its right neighbor sequent corresponding to its successor state. We use $\mathcal{G}$, $\mathcal{H}$, $\mathcal{K}$ and so on to denote LNSs. For an LNS $\mathcal{G} = \Gamma_0 \vdash \Delta_0 \parallel \dots \parallel \Gamma_n \vdash \Delta_n$ we refer to the $(n-1)$ -component as the *penultimate component*, to the n -component as the *end component*, and to any k -component such that $k < n$ as an *interior component*, that is, an interior component is any component that is *not* the end component. If $\mathcal{G}$ contains a formula in focus, then we call $\mathcal{G}$ *focused* and *unfocused* otherwise. Given a focused LNS $\mathcal{G}$, we write $\mathcal{G}^u$ to denote the LNS obtained by changing the annotation of the focused formula to u.

We define the multiset of components of a linear nested sequent $\mathcal{G} = \Gamma_0 \vdash \Delta_0 \parallel \dots \parallel \Gamma_n \vdash \Delta_n$ to be $\text{cp}(\mathcal{G}) := \{\Gamma_i \vdash \Delta_i \mid 0 \leq i \leq n\}$. The *length* of an LNS $\mathcal{G}$ is the number of components it contains, i.e., $\|\mathcal{G}\| := |\text{cp}(\mathcal{G})|$. For an LNS $\mathcal{G}$, we let $\mathcal{G}(i)$ be the i -component of $\mathcal{G}$, if $0 \leq i < \|\mathcal{G}\|$, and we let $\text{end}(\mathcal{G}) := \mathcal{G}(\|\mathcal{G}\| - 1)$ denote the end component of $\mathcal{G}$. For a Gentzen sequent $\Gamma \vdash \Delta$, let $\text{sufo}(\Gamma \vdash \Delta) := \text{sufo}(\Gamma^- \cup \Delta^-)$. For an LNS $\mathcal{G} = \Gamma_0 \vdash \Delta_0 \parallel \dots \parallel \Gamma_n \vdash \Delta_n$, we define $\text{sufo}(\mathcal{G})$ as follows: $\text{sufo}(\mathcal{G}) := \bigcup_{0 \leq i \leq n} \text{sufo}(\Gamma_i^- \cup \Delta_i^-)$. We can interpret LNSs by means of their *formula interpretation*, defined below:

$$f(\Gamma \vdash \Delta) := \bigwedge \Gamma^- \rightarrow \bigvee \Delta^- \quad f(\Gamma \vdash \Delta \parallel \mathcal{G}) := \bigwedge \Gamma^- \rightarrow (\bigvee \Delta^- \vee \text{xf}(\mathcal{G}))$$

Definition 3.1. Let $\mathcal{G}$ be an LNS, $0 \leq i < \|\mathcal{G}\|$, and $\bar{\sigma}$ be a state sequence. We define $\bar{\sigma}, i \models \mathcal{G}(i)$ iff $\bar{\sigma}, i \models \bigwedge \Gamma^- \rightarrow \bigvee \Delta^-$ with $\mathcal{G}(i) = \Gamma \vdash \Delta$. We define $\bar{\sigma} \models \mathcal{G}$ iff $\bar{\sigma} \models f(\mathcal{G})$, i.e., there exists $0 \leq i < \|\mathcal{G}\|$ such that $\bar{\sigma}, i \models \mathcal{G}(i)$.¹ An LNS $\mathcal{G}$ is valid iff for each state sequence $\bar{\sigma}$, $\bar{\sigma} \models \mathcal{G}$; otherwise, $\mathcal{G}$ is invalid.

¹Note that the annotations do not convey any semantic meaning.

We define LNS_{LTL} to be the set of all rules depicted in Figure 1. The id and $\perp\text{L}$ rules are *initial rules* and we call the conclusion of an initial rule an *initial sequent*. The *focusing rules* u and f , read bottom-up, unfocus and focus an until-formula, respectively. Note the condition that all side formulae in the antecedent of the displayed component are unfocused both in the conclusion and the premise. The remaining rules are *logical rules* that introduce complex formulae into either the antecedent or consequent of a component of an LNS. As discussed in more detail below, the UL_1 and UL_2 rules utilize our focusing mechanism to ensure the soundness of non-wellfounded and cyclic proofs. Moreover, we remark that the UL_1 , UL_2 , UR_1 , and UR_2 rules rewrite an until-formula $A \text{UB}$ into its equivalent unfolding $B \vee (A \wedge \text{X}(A \text{UB}))$. The rules UL_1 and UR_1 operate on until-formulae in an interior component: in the right premise, $A \text{UB}$ is pushed into the next component. The rules UL_2 and UR_2 work similarly but operate on until-formulae in the end component, thereby introducing a new component in the right premise.

The *principal formula* of a logical rule in LNS_{LTL} is the formula explicitly displayed in the conclusion, and we define the *auxiliary formulae* to be those explicitly displayed in the premises. For example, $A \text{UB}$ is principal in UR_1 and A , B , and $A \text{UB}$ are auxiliary. All other formulae are called *side formulae*. The *principal component* (*auxiliary components*) is (are) the component(s) where the principal (auxiliary, resp.) formula(e) occur.

The condition that LNSs contain at most one formula in focus imposes certain restrictions on rule instances, as illustrated by the following two lemmas.

Lemma 3.2. *The following hold.*

- (1) *The premise of an instance of u does not contain a formula in focus.*
- (2) *The conclusion of an instance of f does not contain a formula in focus.*

Proof. For (1) consider an instance of u

$$\frac{\mathcal{G} \parallel \Gamma^u, A \text{UB}^u \vdash \Delta \parallel \mathcal{H}}{\mathcal{G} \parallel \Gamma^u, A \text{UB}^f \vdash \Delta \parallel \mathcal{H}} \text{u}$$

and suppose toward a contradiction that the premise contains a formula C^f in focus. By the shape of the premise, C^f must belong to a component in $\mathcal{G}$ or in $\mathcal{H}$. Therefore the conclusion contains two formulae in focus and is thus not a well-formed LNS, which is a contradiction. The case for (2) is similar. $\square$

Lemma 3.3. *In every instance of UL_1 and UL_2 with the principal formula in focus, the left premise does not contain a formula in focus.*

Proof. For UL_1 consider an instance of UL_1 with the principal formula in focus in component i , as shown below.

$$\frac{\mathcal{G} \parallel \Gamma, B \vdash \Delta \parallel \Sigma \vdash \Pi \parallel \mathcal{H} \quad \mathcal{G} \parallel \Gamma, A \vdash \Delta \parallel \Sigma, A \text{UB}^f \vdash \Pi \parallel \mathcal{H}}{\mathcal{G} \parallel \Gamma, A \text{UB}^f \vdash \Delta \parallel \Sigma \vdash \Pi \parallel \mathcal{H}} \text{UL}_1$$

Suppose toward a contradiction that the left premise contains a formula C^f in focus. By definition of UL_1 , the auxiliary formula B is unfocused, therefore C^f occurs either in $\mathcal{G}$ or $\mathcal{H}$, or belongs to Γ or Σ . In either case, C^f also occurs in the right premise. The right premise contains the displayed focused occurrence in component $i + 1$. Hence any focused side formula would have to coincide with that occurrence; otherwise, the right premise would have two focused formula occurrences. Thus a focused side formula could only be $A \text{UB}^f \in \Sigma$. But then the conclusion would contain focused formula occurrences in both components i and $i + 1$, which is impossible. Therefore there are no focused side occurrences, and the left premise is unfocused. The case for UL_2 is similar. $\square$

Since we are working with sequents based on sets of formulae (where contraction is implicit), rule instances are either preserving or succinct. A rule instance is *preserving* if the principal formula occurs as a side formula on the same side of the principal component in the premise, and *succinct* otherwise. For example, the following shows a preserving instance of $\rightarrow R$ on the left and a succinct instance on the right:

$$\frac{\mathcal{G} \parallel A \vdash B, A \rightarrow B \parallel \mathcal{H}}{\mathcal{G} \parallel \vdash A \rightarrow B \parallel \mathcal{H}} \rightarrow R \quad \frac{\mathcal{G} \parallel A \vdash B \parallel \mathcal{H}}{\mathcal{G} \parallel \vdash A \rightarrow B \parallel \mathcal{H}} \rightarrow R$$

3.1 Non-Wellfounded Proofs

We define a *derivation* π of an LNS $\mathcal{G}$ in LNS_{LTL} to be a (potentially infinite) tree whose nodes are labelled by LNSs such that (1) the root is labelled by $\mathcal{G}$ and (2) every parent node is labelled by the conclusion of a rule of LNS_{LTL} with its children labelled by the corresponding premises. We will usually identify the node with the LNS that labels it, and treat distinct occurrences of the same LNS as distinct nodes. If there is any danger of confusion, we will make the distinction between nodes and their labels explicit. An *index set* is a set $I \subseteq \mathbb{N}$ such that (1) $0 \in I$ and (2) if $n+1 \in I$, then $n \in I$, i.e., an index set is an initial segment of the natural numbers, or the whole set of naturals. We use index sets to define certain kinds of (potentially infinite) sequences below.

A *path* in a derivation π is a (potentially infinite) sequence $\mathcal{P} = \{\mathcal{G}_i\}_{i \in I}$ of LNSs (where I is an index set) such that for each $i \in I$ if $i+1 \in I$, then the LNS $\mathcal{G}_i$ is the parent of $\mathcal{G}_{i+1}$. A path $\mathcal{P} = \{\mathcal{G}_i\}_{i \in I}$ is *finite* if $I \neq \mathbb{N}$ and *infinite* otherwise. Moreover, $\mathcal{P} = \{\mathcal{G}_i\}_{i \in I}$ is *maximal* if $\mathcal{P}$ is finite and $\mathcal{G}_{|I|-1}$ is a leaf or if $\mathcal{P}$ is infinite. A *branch* in a derivation π is a maximal path $\mathcal{B} = \{\mathcal{G}_i\}_{i \in I}$ of LNSs such that $\mathcal{G}_0$ is the root. The *height* of a derivation is defined in the usual way as the maximal length of a branch in the derivation, which may be infinite. A path $\mathcal{P} := \{\mathcal{G}_i\}_{i \in I}$ has a *continuous focus trace* iff for every $i \in I$, $\mathcal{G}_i$ contains a formula in focus. The following lemma follows directly from this definition and Lemma 3.2:

Lemma 3.4. *If a path $\mathcal{P}$ has a continuous focus trace, then neither u nor f are applied along the path.*

Observe that on a continuous focus trace, the underlying until-formula remains the same. An infinite path $\mathcal{P}$ in a derivation π is called *progressing* if (1) $\mathcal{P}$ has a continuous focus trace and (2) $\mathcal{P}$ passes infinitely often through the right premise of an instance of a rule $r \in \{\text{UL}_1, \text{UL}_2\}$ in which the formula in focus is principal. An infinite branch $\mathcal{B}$ in π is called *progressing* if it has a suffix which is a progressing path.

Definition 3.5. *A non-wellfounded proof (or, $\text{LNS}_{\text{LTL}}^\infty$ -proof) of an LNS $\mathcal{G}$ is a derivation π of $\mathcal{G}$ such that (1) every leaf of π is an initial sequent and (2) every infinite branch is progressing.*

We let $\text{LNS}_{\text{LTL}}^\infty$ be the non-wellfounded LNS calculus obtained by letting the set of provable sequents be determined by non-wellfounded proofs and write $\text{LNS}_{\text{LTL}}^\infty \vdash \mathcal{G}$ if $\mathcal{G}$ has a non-wellfounded proof.

Example 3.6. *An example of a non-wellfounded proof π of $p\text{U}q^f \vdash p\text{U}q$ is depicted in Figure 2. The proof consists of one infinite branch (the right-most one indicated by the three dots) and infinitely many finite branches (which eventually branch to the left). Every finite branch ends in an initial sequent, while the infinite branch contains a continuous focus trace which is progressing.*

3.2 Cyclic Proofs

Let $\mathcal{G}$ be an LNS with $\|\mathcal{G}\| = n+1$ and $\mathcal{G}(i) = \Gamma_i \vdash \Delta_i$. For $0 \leq i \leq n$, we define $\mathcal{G}$ to be *i-saturated* iff $\{A \mid A \in (\Gamma_i \cup \Delta_i)^-\} \cap \{\chi A, A \rightarrow B, A \cup B \mid A, B \in \mathcal{L}\} = \emptyset$, i.e., its i -component does not contain any

$$\begin{array}{c}
\pi_1 := \frac{\frac{q \vdash p, q}{q \vdash pUq} \text{id} \quad \frac{q \vdash q // \vdash pUq}{p \vdash q // q \vdash pUq} \text{id}}{p \vdash p, q // pUq^f \vdash} \text{UR}_2 \quad \pi_2 := \frac{\frac{p \vdash q // q \vdash p, q}{p \vdash q // q \vdash pUq} \text{id} \quad \frac{p \vdash q // q \vdash q // \vdash pUq}{p \vdash q // q \vdash pUq} \text{id}}{p \vdash q // p \vdash p, q // pUq^f \vdash} \text{UR}_2 \\
\vdots \\
\frac{\frac{p \vdash q // p \vdash p, q // pUq^f \vdash}{p \vdash q // p \vdash pUq // pUq^f \vdash} \text{id} \quad \frac{p \vdash q // p \vdash pUq // pUq^f \vdash}{p \vdash q // p \vdash pUq // pUq^f \vdash} \text{id}}{p \vdash q // p \vdash pUq // pUq^f \vdash} \text{UR}_1 \\
\frac{\pi_1 \quad \frac{p \vdash p, q // pUq^f \vdash}{p \vdash p, q // pUq^f \vdash} \text{id} \quad \frac{\pi_2 \quad \frac{p \vdash q // p \vdash pUq // pUq^f \vdash}{p \vdash q // p \vdash pUq // pUq^f \vdash} \text{UL}_2}{p \vdash pUq // pUq^f \vdash} \text{UR}_1}{pUq^f \vdash pUq} \text{UL}_2
\end{array}$$

Figure 2: Example of a non-wellfounded proof.

complex formulae. We call $\mathcal{G}$ *saturated* iff $\mathcal{G}$ is not an initial sequent and for each $0 \leq i < n$, $\mathcal{G}$ is i -saturated. Intuitively, saturated sequents serve as checkpoints: since their interior components are free of complex formulae (meaning, no rules are bottom-up applicable to these components), two saturated sequents with matching end components represent identical ‘states’ in a proof, making them natural candidates for the endpoints of a cycle. This serves as the motivation for our notion of cyclic proof given below.

Definition 3.7. A tuple (π, L, c) is a cyclic proof (or, $\text{LNS}_{\text{LTL}}^{\text{cyc}}$ -proof) of an LNS $\mathcal{G}$ iff it satisfies:

- (1) π is a finite derivation of $\mathcal{G}$;
- (2) L is a set containing only leaves of π , which we call buds;
- (3) c is a function mapping each $\mathcal{G} \in L$ to an LNS $c(\mathcal{G})$ strictly below $\mathcal{G}$, called the companion of $\mathcal{G}$, such that $\|c(\mathcal{G})\| < \|\mathcal{G}\|$, the path from $c(\mathcal{G})$ to $\mathcal{G}$ has a continuous focus trace and goes through the right premise of a rule $r \in \{\text{UL}_1, \text{UL}_2\}$ with the principal formula in focus, both $c(\mathcal{G})$ and $\mathcal{G}$ are saturated, and $\text{end}(c(\mathcal{G})) = \text{end}(\mathcal{G})$;²
- (4) every other leaf in π that is not in L is an initial sequent.

We let $\text{LNS}_{\text{LTL}}^{\text{cyc}}$ be the cyclic LNS calculus obtained by letting the set of provable sequents be determined by cyclic proofs and write $\text{LNS}_{\text{LTL}}^{\text{cyc}} \vdash \mathcal{G}$ if $\mathcal{G}$ has a cyclic proof.

Example 3.8. An example of a cyclic proof (π, L, c) is displayed in Figure 3 where $L = \{\mathcal{G}\}$ and $c(\mathcal{G}) = \mathcal{H}$ such that $\mathcal{G} := p \vdash q // pUq^f \vdash pUq$ and $\mathcal{H} := pUq^f \vdash pUq$. It is straightforward to verify that c defines a correct cycle since $\|\mathcal{H}\| < \|\mathcal{G}\|$, the branch from $\mathcal{H}$ to $\mathcal{G}$ has a continuous focus trace and goes through the right premise of UL_2 with the principal formula in focus, both $\mathcal{G}$ and $\mathcal{H}$ are saturated, and $\text{end}(\mathcal{H}) = \text{end}(\mathcal{G})$.

4 Soundness of Non-Wellfounded Proofs

In this section we prove that $\text{LNS}_{\text{LTL}}^\infty$ is sound. We proceed via proof by infinite descent. We assume, toward a contradiction, that some invalid LNS $\mathcal{G}$ has a non-wellfounded proof π . Since $\mathcal{G}$ is invalid, there exists a state sequence $\bar{\sigma}$ such that $\bar{\sigma} \not\models \mathcal{G}$. We then show how to obtain a branch $\mathcal{B} = \{\mathcal{G}_i\}_{i \in \mathbb{N}}$ of π such that $\bar{\sigma} \not\models \mathcal{G}_i$ for each $i \in \mathbb{N}$, by showing that all rules of $\text{LNS}_{\text{LTL}}^\infty$ are locally sound. Note that

²By “strictly below $\mathcal{G}$,” we mean that $c(\mathcal{G})$ occurs on the path from the root of π to the leaf $\mathcal{G}$ and is distinct from $\mathcal{G}$.

$$\begin{array}{c}
\pi_1 := \frac{\frac{\overline{q \vdash p, q} \text{ id} \quad \frac{\overline{q \vdash q // \vdash pUq} \text{ id}}{q \vdash pUq} \text{ UR}_2}{\frac{\frac{\overline{p \vdash p, q // pUq^f \vdash} \text{ id} \quad \frac{p \vdash q // pUq^f \vdash pUq}{p \vdash pUq // pUq^f \vdash} \text{ UR}_1}{p \vdash pUq // pUq^f \vdash} \text{ UL}_2} \text{ UL}_2 \\
\hline
pUq^f \vdash pUq \leftarrow \text{-----} \text{ } c
\end{array}$$

Figure 3: Example of a cyclic proof.

$\mathcal{B}$ must be an infinite branch of π , since initial sequents are always valid. Simultaneously, we assign a well-founded measure in the form of natural numbers to formulae in focus occurring in LNSs along $\mathcal{B}$ and show that whenever a formula AUB^f is principal in UL_1 or UL_2 and $\mathcal{B}$ passes through the right premise, the measure assigned to AUB^f strictly decreases, and weakly decreases otherwise. Finally, since π is a proof, $\mathcal{B}$ must contain a progressing suffix, meaning that there is a continuous focus trace and the suffix passes infinitely often through the right premise of instances of UL_1 and UL_2 with the formula in focus principal. Thus, we obtain an infinite decreasing sequence of natural numbers which strictly decreases infinitely often, contradicting the well-foundedness of $\mathbb{N}$. This gives the desired contradiction, implying the soundness of LNS_{LTL}^∞ .

Let $\mathcal{G}$ be an LNS and recall that only formulae of the form AUB can occur in focus in an LNS $\mathcal{G}$ and such formulae always occur in the antecedent of a component. A *progress measure* for $\mathcal{G}$ is a partial function which assigns to the formula AUB^f in focus in $\mathcal{G}$ (if such a formula exists) a natural number; if no formula is in focus, the progress measure is undefined.

Theorem 4.1 (Soundness of LNS_{LTL}^∞). *If $LNS_{LTL}^\infty \Vdash \mathcal{G}$, then $\mathcal{G}$ is valid.*

Proof. Let π be a non-wellfounded proof of $\mathcal{G}$ and suppose, for contradiction, that $\mathcal{G}$ is invalid. Then, there exists a state sequence $\bar{\sigma}$ such that $\bar{\sigma} \not\models \mathcal{G}$. We inductively define an infinite branch of LNSs $\mathcal{B} = \{\mathcal{G}_i\}_{i \in \mathbb{N}}$ through π and an infinite sequence of progress measures $\{\mu_i\}_{i \in \mathbb{N}}$ such that for all $i \in \mathbb{N}$ the following hold:

- (1) $\bar{\sigma} \not\models \mathcal{G}_i$;
- (2) μ_i is a progress measure for $\mathcal{G}_i$;
- (3) if a formula AUB^f in focus occurs in the j -component of $\mathcal{G}_i$, then the following hold:
 - (a) $\mu_i(AUB^f) = \min\{k \in \mathbb{N} \mid \bar{\sigma}, j+k \models B \text{ and } \forall k' < k : \bar{\sigma}, j+k' \models A\}$;
 - (b) for $i > 0$ if AUB^f is in focus in $\mathcal{G}_{i-1}$, then $\mu_i(AUB^f) \leq \mu_{i-1}(AUB^f)$;
 - (c) for $i > 0$ if AUB^f is in focus in $\mathcal{G}_{i-1}$ and $\mathcal{G}_{i-1}$ is the conclusion of an instance of $r \in \{UL_1, UL_2\}$ with AUB^f principal, then $\mu_i(AUB^f) < \mu_{i-1}(AUB^f)$.³

We now define $\{\mathcal{G}_i\}_{i \in \mathbb{N}}$ and $\{\mu_i\}_{i \in \mathbb{N}}$. First, let $\mathcal{G}_0 := \mathcal{G}$. By assumption $\bar{\sigma} \not\models \mathcal{G}$, so (1) is satisfied. Moreover, by Definition 3.1, this implies that if a formula AUB^f occurs in the antecedent of the j -th component of $\mathcal{G}_0$ (for $0 \leq j < \|\mathcal{G}_0\|$), $\bar{\sigma}, j \models AUB$ and so there exists a natural number k such that (i) $\bar{\sigma}, j+k \models B$ and (ii) $\bar{\sigma}, j+k' \models A$ for all $0 \leq k' < k$. If a formula in focus occurs in $\mathcal{G}$, then let μ_0 be

³Observe that $\mathcal{G}_i$ must be the right premise of r due to Lemma 3.3.

the partial function that assigns the formula AUB^f the least natural number k such that (i) and (ii) hold. Otherwise let μ_0 be undefined. Observe that conditions (1)–(3) are satisfied.

Suppose we have defined $\mathcal{G}_i$ and μ_i and they satisfy conditions (1)–(3). We define $\mathcal{G}_{i+1}$ and μ_{i+1} by using a case distinction based on the rule r with conclusion $\mathcal{G}_i$ in π . Note that r cannot be an initial rule, since $\bar{\sigma} \not\models \mathcal{G}_i$ by (1). We show the cases for $\rightarrow R$, u , f , XL_2 , UL_1 and UR_2 . Each remaining case is similar to one of these cases.

CASE FOR $\rightarrow R$. Suppose $\mathcal{G}_i$ is the conclusion of an instance of $\rightarrow R$, as shown below.

$$\frac{\overbrace{\mathcal{G} \parallel \Gamma, A \vdash B, \Delta \parallel \mathcal{H}}^{\mathcal{F}}}{\underbrace{\mathcal{G} \parallel \Gamma \vdash A \rightarrow B, \Delta \parallel \mathcal{H}}_{\mathcal{G}_i}} \rightarrow R$$

We assume that the principal formula $A \rightarrow B$ occurs in component j . By condition (1), $\bar{\sigma} \not\models \mathcal{G}_i$, which implies that $\bar{\sigma}, j \models \bigwedge \Gamma^-$, $\bar{\sigma}, j \not\models \bigvee \Delta^-$, and $\bar{\sigma}, j \not\models A \rightarrow B$. The latter implies that $\bar{\sigma}, j \models A$ and $\bar{\sigma}, j \not\models B$. Hence $\bar{\sigma} \not\models \mathcal{F}$. Let $\mathcal{G}_{i+1} := \mathcal{F}$ and note that (1) is satisfied. Note that $\mathcal{G}_{i+1}$ contains a formula occurrence in focus iff $\mathcal{G}_i$ contains the same formula occurrence in focus. Therefore we define $\mu_{i+1} := \mu_i$. Note that (2) and (3) are satisfied.

CASE FOR u . Suppose $\mathcal{G}_i$ is the conclusion of an instance of u , as shown below.

$$\frac{\overbrace{\mathcal{G} \parallel \Gamma^u, AUB^u \vdash \Delta \parallel \mathcal{H}}^{\mathcal{F}}}{\underbrace{\mathcal{G} \parallel \Gamma^u, AUB^f \vdash \Delta \parallel \mathcal{H}}_{\mathcal{G}_i}} u$$

We assume that the formula AUB^f occurs in component j . By condition (1) we have $\bar{\sigma} \not\models \mathcal{G}_i$. Since annotations do not convey any semantic meaning, we have $\bar{\sigma} \not\models \mathcal{F}$. Therefore let $\mathcal{G}_{i+1} := \mathcal{F}$ and note that condition (1) is satisfied. Moreover, by Lemma 3.2, $\mathcal{G}_{i+1}$ does not contain a formula in focus, implying that condition (3) is vacuously satisfied. We define μ_{i+1} to be undefined. Note that (2) is then satisfied as well.

CASE FOR f . Suppose $\mathcal{G}_i$ is the conclusion of an instance of f , as shown below.

$$\frac{\overbrace{\mathcal{G} \parallel \Gamma^u, AUB^f \vdash \Delta \parallel \mathcal{H}}^{\mathcal{F}}}{\underbrace{\mathcal{G} \parallel \Gamma^u, AUB^u \vdash \Delta \parallel \mathcal{H}}_{\mathcal{G}_i}} f$$

We assume that the formula AUB^f occurs in component j . By condition (1) we have that $\bar{\sigma} \not\models \mathcal{G}_i$. Since annotations do not convey any semantic meaning, we obtain $\bar{\sigma} \not\models \mathcal{F}$. Thus let $\mathcal{G}_{i+1} := \mathcal{F}$ and note that condition (1) is satisfied. Moreover, $\bar{\sigma}, j \models AUB$, hence there exists a least natural number k with $\bar{\sigma}, j+k \models B$ and $\bar{\sigma}, j+k' \models A$ for all $k' < k$. Set $\mu_{i+1}(AUB^f) := k$ and note that condition (2) and (3)(a) are satisfied. Finally, by Lemma 3.2–(2) the conclusion $\mathcal{G}_i$ does not have a formula in focus. Hence conditions (3)(b) and (c) vacuously hold.

CASE FOR XL_2 . Suppose $\mathcal{G}_i$ is the conclusion of an instance of XL_2 , as shown below.

$$\frac{\overbrace{\mathcal{G} \parallel \Gamma \vdash \Delta \parallel A \vdash}^{\mathcal{F}}}{\underbrace{\mathcal{G} \parallel \Gamma, \mathbf{x}A \vdash \Delta}_{\mathcal{G}_i}} \text{XL}_2$$

By condition (1) we have that $\bar{\sigma} \not\models \mathcal{G}_i$, which implies that $\bar{\sigma}, j \models \bigwedge \Gamma^-$, $\bar{\sigma}, j \models \mathbf{x}A$ and $\bar{\sigma}, j \not\models \bigvee \Delta^-$, where $j = \|\mathcal{G}_i\| - 1$. Therefore $\bar{\sigma}, j+1 \models A$ and so $\bar{\sigma} \not\models \mathcal{F}$. Let $\mathcal{G}_{i+1} := \mathcal{F}$ and note that condition (1) is satisfied. Moreover $\mathcal{G}_{i+1}$ contains a formula occurrence in focus iff $\mathcal{G}_i$ contains the same formula occurrence in focus. Therefore we let $\mu_{i+1} := \mu_i$. Note that conditions (2) and (3) are satisfied.

CASE FOR UL_1 . Suppose $\mathcal{G}_i$ is the conclusion of an instance of UL_1 , as shown below.

$$\frac{\overbrace{\mathcal{G} \parallel \Gamma, B \vdash \Delta \parallel \Sigma \vdash \Pi \parallel \mathcal{H}}^{\mathcal{F}_1} \quad \overbrace{\mathcal{G} \parallel \Gamma, A \vdash \Delta \parallel \Sigma, AUB^a \vdash \Pi \parallel \mathcal{H}}^{\mathcal{F}_2}}{\underbrace{\mathcal{G} \parallel \Gamma, AUB^a \vdash \Delta \parallel \Sigma \vdash \Pi \parallel \mathcal{H}}_{\mathcal{G}_i}} \text{UL}_1$$

We assume that the principal formula AUB^a occurs in component j and we distinguish two cases. First, suppose that $a = u$. By condition (1) we have that $\bar{\sigma} \not\models \mathcal{G}_i$, which implies that $\bar{\sigma}, j \models \bigwedge \Gamma^-$, $\bar{\sigma}, j \models AUB$ and $\bar{\sigma}, j \not\models \bigvee \Delta$. Hence there exists a natural number k such that $\bar{\sigma}, j+k \models B$ and $\bar{\sigma}, j+k' \models A$ for all $0 \leq k' < k$. If $k = 0$, then $\bar{\sigma}, j \models B$ and so $\bar{\sigma} \not\models \mathcal{F}_1$. In this case let $\mathcal{G}_{i+1} := \mathcal{F}_1$. If $k > 0$, then $\bar{\sigma}, j \models A$ and $\bar{\sigma}, j+1 \models AUB$, implying that $\bar{\sigma} \not\models \mathcal{F}_2$. In that case let $\mathcal{G}_{i+1} := \mathcal{F}_2$. Observe that in both cases (1) holds. There are now two sub-cases to consider: either (i) $\mathcal{G}_i$ contains a formula in focus or (ii) it does not. In case (i), the formula in focus is a side formula of the rule instance and therefore occurs in the same component in both $\mathcal{G}_i$ and $\mathcal{G}_{i+1}$; hence, we take $\mu_{i+1} := \mu_i$ and observe that (2) and (3) hold, where (3)(b) is satisfied with equality. In case (ii), we take μ_{i+1} to be undefined and note that (2) and (3) hold vacuously.

Second, suppose that $a = f$. We make a case distinction on if (i) $\mu_i(AUB^f) = 0$ or (ii) $\mu_i(AUB^f) > 0$.

(i) If $\mu_i(AUB^f) = 0$, then $\bar{\sigma}, j \models B$ by (3)(a). Note that then $\bar{\sigma} \not\models \mathcal{F}_1$ and so let $\mathcal{G}_{i+1} := \mathcal{F}_1$. Note that condition (1) is satisfied. By Lemma 3.3 no formula is in focus in $\mathcal{G}_{i+1}$. We therefore take μ_{i+1} to be undefined, which implies that condition (2) holds. Finally, note that condition (3) is vacuously satisfied.

(ii) If $\mu_i(AUB^f) > 0$, then observe that $\bar{\sigma}, j \models A$ and $\bar{\sigma}, j+1 \models AUB$ by (3)(a). Therefore, $\bar{\sigma} \not\models \mathcal{F}_2$ and so let $\mathcal{G}_{i+1} := \mathcal{F}_2$. Moreover, since $\mu_i(AUB^f)$ is the least natural number k such that $\bar{\sigma}, j+k \models B$ and $\bar{\sigma}, j+k' \models A$ for each $0 \leq k' < k$, we have that $k-1$ is the least natural number such that $\bar{\sigma}, (j+1) + (k-1) \models B$ and $\bar{\sigma}, (j+1) + k' \models A$ for each $0 \leq k' < k-1$. Let $\mu_{i+1}(AUB^f) := \mu_i(AUB^f) - 1$. Note that conditions (1)–(3) are satisfied.

CASE FOR UR_2 . Suppose $\mathcal{G}_i$ is the conclusion of an instance of UR_2 , as shown below.

$$\frac{\overbrace{\mathcal{G} \parallel \Gamma \vdash A, B, \Delta}^{\mathcal{F}_1} \quad \overbrace{\mathcal{G} \parallel \Gamma \vdash B, \Delta \parallel \vdash AUB}^{\mathcal{F}_2}}{\underbrace{\mathcal{G} \parallel \Gamma \vdash AUB, \Delta}_{\mathcal{G}_i}} \text{UR}_2$$

By condition (1) we have that $\bar{\sigma} \not\models \mathcal{G}_i$, which implies that $\bar{\sigma}, j \models \bigwedge \Gamma^-$, $\bar{\sigma}, j \not\models \bigvee \Delta^-$ and $\bar{\sigma}, j \not\models AUB$, where $j = \|\mathcal{G}_i\| - 1$. Note that $\bar{\sigma}, j \not\models AUB$ implies that $\bar{\sigma}, j \not\models B$. We distinguish two cases. (i) If $\bar{\sigma}, j \not\models A$, let $\mathcal{G}_{i+1} := \mathcal{F}_1$. (ii) If $\bar{\sigma}, j \models A$, then $\bar{\sigma}, j+1 \not\models AUB$, and so we let $\mathcal{G}_{i+1} := \mathcal{F}_2$. Note that in either case condition (1) is satisfied. Moreover $\mathcal{G}_{i+1}$ contains a formula occurrence in focus iff the same formula

occurrence is in focus in $\mathcal{G}_i$ and so we let $\mu_{i+1} := \mu_i$. Note that conditions (2) and (3) are satisfied. This concludes the construction of $\mathcal{B} = \{\mathcal{G}_i\}_{i \in \mathbb{N}}$ and of $\{\mu_i\}_{i \in \mathbb{N}}$.

Since π is a proof, the infinite branch $\mathcal{B} = \{\mathcal{G}_i\}_{i \in \mathbb{N}}$ must have a suffix $\mathcal{P} = \{\mathcal{G}_i\}_{i \geq j}$ that is a progressing path. Hence, $\mathcal{P}$ has a continuous focus trace. Moreover, $\mathcal{P}$ passes infinitely often through the right premise of a rule $r \in \{\text{UL}_1, \text{UL}_2\}$ with the principal formula AUB^f in focus. Therefore, $\mu_i(AUB^f)$ is defined for all $i \geq j$. Consider the infinite sequence of natural numbers $\{\mu_i(AUB^f)\}_{i \geq j}$. By (3)(b) and (3)(c) we have that for each $i \geq j$, $\mu_{i+1}(AUB^f) \leq \mu_i(AUB^f)$. Since $\mathcal{P}$ passes infinitely often through the right premise of a rule $r \in \{\text{UL}_1, \text{UL}_2\}$ with the formula AUB^f principal, (3)(c) implies that there are infinitely many $i \geq j$ with $\mu_{i+1}(AUB^f) < \mu_i(AUB^f)$. Therefore, $\{\mu_i(AUB^f)\}_{i \geq j}$ is an infinite decreasing sequence of natural numbers that strictly decreases infinitely often. This contradicts that the natural numbers are well-founded. Hence, $\mathcal{G}$ must be valid. $\square$

In the proof of Theorem 4.1, we have also established that all rules of $\text{LNS}_{\text{LTL}}^\infty$ are locally sound: if the premises are valid, then the conclusion is valid as well. Similarly, we can establish semantic invertibility: if the conclusion is valid, then all premises are valid. The proof of the following lemma follows immediately by the semantic clauses and the interpretation of LNSs.

Lemma 4.2 (Semantic invertibility). *Consider an instance of a rule $r \in \text{LNS}_{\text{LTL}}^\infty$ with conclusion $\mathcal{G}$ and premises $\mathcal{H}_i$ for $1 \leq i \leq n$. If $\mathcal{G}$ is valid, then $\mathcal{H}_i$ is valid for every $1 \leq i \leq n$.*

5 Completeness of Non-Wellfounded Proofs

In this section we prove completeness of $\text{LNS}_{\text{LTL}}^\infty$ by a proof-search argument. Given an LNS $\mathcal{G}$, we build a structure π , called a *proof-search tree*, by a phased construction. The construction first exhaustively applies rules bottom-up to formulae in the interior components of LNSs until every open leaf is either initial or saturated. Initial leaves are closed, as are saturated leaves that are not the conclusion of any rule instance. For any other saturated leaf, we first apply a focus rule to one of the until-formulae in the end-component (if any exist). Since there are in general multiple formulae present that may be focused or unfocused, we introduce so-called *branching focus rules* which have one premise for each possible application of u or f . Then we apply a logical rule to one of the complex formulae in the end-component. If we apply a rule that adds a new component, then the premises are in general not saturated, and so the process can be repeated indefinitely. This results in a potentially infinite tree of LNSs called a proof-search tree for $\mathcal{G}$. Since we employ branching focus rules, this proof-search tree is not in general a derivation in $\text{LNS}_{\text{LTL}}^\infty$. However, we can obtain derivations from proof-search trees by choosing a single premise for each branching focus rule instance and deleting the other branches. Derivations obtained in that way satisfy the *saturation recurrence property (SRP)*:

Definition 5.1. *A derivation π satisfies the saturation recurrence property (SRP) if and only if, for any infinite branch $\mathcal{B} := \{\mathcal{G}_i\}_{i \in \mathbb{N}}$ in π , infinitely many $\mathcal{G}_i$ are saturated.*

We then show that given a proof-search tree for an LNS $\mathcal{G}$, we can either extract a derivation from it which is a non-wellfounded proof of $\mathcal{G}$ that satisfies the SRP, or the proof-search tree contains a non-progressing branch which can be used to construct a counter-model for $\mathcal{G}$. Since valid LNSs do not have counter-models, we obtain that every valid LNS has a non-wellfounded proof that satisfies the SRP. Thus we obtain completeness of $\text{LNS}_{\text{LTL}}^\infty$ with respect to SRP-proofs. This property will prove useful in the transformation of non-wellfounded into cyclic proofs in Section 6. We begin by introducing the branching focus rules and proof-search trees.

5.1 Proof-Search Trees

We introduce two new inference rules that will be used in the completeness proof for LNS_{LTL} . These are taken to be *disjunctive rules* (cf. [26, 20]), meaning, the conclusion is provable whenever at least one premise is provable. By contrast, we refer to the remaining rules in LNS_{LTL} as *conjunctive rules*: the conclusion is provable whenever all premises are provable. Our proof-search tree will consist of both conjunctive and disjunctive rule applications, serving as a ‘macro-structure’ from which we can extract either proofs or counter-models (cf. [20, 17]).

Definition 5.2. Let $\mathcal{G} = \Gamma_0 \vdash \Delta_0 // \cdots // \Gamma_n \vdash \Delta_n$ be unfocused and define

$$U(\mathcal{G}) := \{(i, C) \mid 0 \leq i \leq n : C = A \vee B \text{ for some } A, B \in \mathcal{L}, C^u \in \Gamma_i\}.$$

For $(i, C) \in U(\mathcal{G})$, let $\mathcal{G}^{(i, C)}$ be obtained by replacing C^u in the antecedent of component i by C^f , leaving everything else unchanged.

The focused branching rule fb has exactly the following premises

$$\frac{\mathcal{G} \quad \{\mathcal{G}^{(i, C)}\}_{(i, C) \in U(\mathcal{G})}}{\mathcal{G}} \text{fb.}$$

When $U(\mathcal{G}) = \emptyset$, this is a unary identity step.

In essence, this rule takes an unfocused LNS as the conclusion, and via a bottom-up application, focuses all formulae of the form $A \vee B$ in antecedents of components in all possible ways (as well as copying the conclusion as a premise).

Definition 5.3. The unfocused branching rule ub is given by

$$\frac{\mathcal{G}^u \quad \mathcal{G}}{\mathcal{G}} \text{ub}$$

where $\mathcal{G}$ is focused.

This rule takes an LNS in focus, and via a bottom-up application, returns one premise with the formula unfocused and another premise identical to the conclusion.

We define the calculus $\text{LNS}_{\text{LTL}}^+ := (\text{LNS}_{\text{LTL}} \setminus \{u, f\}) \cup \{\text{fb}, \text{ub}\}$. A *pseudo-derivation* is defined to be a derivation in $\text{LNS}_{\text{LTL}}^+$.

Definition 5.4. The trivial rule triv is defined by

$$\frac{\mathcal{G}}{\mathcal{G}} \text{triv.}$$

Observe that if a trivial rule application occurs in a proof, then it can be ‘contracted’ to a single copy of $\mathcal{G}$, showing its admissibility in LNS_{LTL} .

Let us call XL_2 , XR_2 , UL_2 and UR_2 *expansion* rules and all other rules in $\text{LNS}_{\text{LTL}}^+$ *length-preserving*. An LNS $\mathcal{G}$ is called *fully saturated* iff $\mathcal{G}$ is not initial and for $0 \leq i < \|\mathcal{G}\|$, $\mathcal{G}$ is *i-saturated*, i.e., $\mathcal{G}$ is not the conclusion of any instance of a rule in $\text{LNS}_{\text{LTL}}^\infty$. We formalize proof-search by means of *proof-search trees* which represent a systematic search for a proof of a given LNS. We provide a phased construction of proof-search trees, and show that it produces a pseudo-derivation.

Definition 5.5. The phased construction is given as follows. Every logical rule instance used in the construction is succinct.

▷ **Saturation Phase.** At the current node $\mathcal{H}$:

- (1) If $\mathcal{H}$ is initial, close the node.

- (2) If $\mathcal{H}$ is not saturated, choose a complex formula in an interior component and apply the corresponding logical rule bottom-up. Restart the saturation phase at every premise.
- (3) Otherwise $\mathcal{H}$ is saturated.
 - (a) If $\mathcal{H}$ is fully saturated, close the node.
 - (b) If $\mathcal{H}$ is not fully saturated, enter the focusing phase.
- ▷ **Focusing Phase.** Apply exactly one instance of fb when $\mathcal{H}$ is unfocused, and exactly one instance of ub when $\mathcal{H}$ is focused. At every premise, enter the logical phase.
- ▷ **Logical Phase.** Apply exactly one logical rule to a complex formula in the end component of $\mathcal{H}$. Enter the saturation phase at every premise.

A few comments about this definition. First, note that the focusing phase always terminates, even if $\mathcal{H}$ does not contain a formula of the form $A \vee B$. In this case $\mathcal{H}$ is unfocused, the rule fb is applied and the instance has a single premise which is a copy of $\mathcal{H}$. Second, if the construction enters the logical phase, it is guaranteed that the end component of $\mathcal{H}$ contains a complex formula, since $\mathcal{H}$ is not fully saturated. Third, the logical phase may apply an expansion rule or an implication rule to a formula in the end component. An expansion rule may produce a premise whose newly interior component contains complex formulae, since the previous end component is now an interior component. An implication rule leaves the interior components unchanged; consequently, every non-initial premise remains saturated.

Lemma 5.6 (Local-wellformedness). *Let $\mathcal{H}$ be an LNS.*

1. Every complex formula in $\mathcal{H}$ admits a succinct bottom-up application of its corresponding logical rule, using the interior or end component variant as appropriate. Every premise of this application is an LNS.
2. If $\mathcal{H}$ is unfocused, then fb is applicable and all its premises are LNSs.
3. If $\mathcal{H}$ is focused, then ub is applicable and all its premises are LNSs.

Proof. By inspection of the rules. □

We distinguish between two types of open leaves in the construction. An *S-leaf* is an open leaf at which the saturation phase is started. An *L-leaf* is an open leaf at which the logical phase is started.

Proposition 5.7. *For every LNS $\mathcal{G}$, the phased construction initiated with $\mathcal{G}$ produces a pseudo-derivation of $\mathcal{G}$, in which every leaf is either initial or fully saturated.*

Proof. Let π_0 be the tree consisting of a single node labelled $\mathcal{G}$, with its root marked S. We construct an infinite sequence of finite trees

$$\pi_0 \subseteq \pi_1 \subseteq \pi_2 \subseteq \dots$$

by processing all open leaves at each stage. We prove by induction on n that the construction of π_n is (I) well-defined and satisfies the following invariant:

- (1) Every node of π_n is labelled by an LNS.
- (2) Every node which is not a leaf is the conclusion of an instance of a rule of $\text{LNS}_{\text{LTL}}^+$ with its children the corresponding premises.
- (3) Every open leaf marked L is non-initial, saturated, and not fully saturated.

BASE CASE. By definition π_0 consists of a single node labelled with $\mathcal{G}$, and $\mathcal{G}$ is an LNS by assumption. Moreover $\mathcal{G}$ is an open leaf marked S, so (I) holds.

INDUCTION STEP. Suppose π_n satisfies (I). If π_n does not contain open leaves, set $\pi_{n+1} := \pi_n$. Note that π_{n+1} satisfies (I). Otherwise let $\mathcal{H}$ be an open leaf of π_n . We distinguish the following cases. (i) If $\mathcal{H}$ is an S-leaf and initial or fully saturated, the construction closes $\mathcal{H}$, which introduces no new nodes and therefore preserves the invariant (I). (ii) If $\mathcal{H}$ is an S-leaf, non-initial and not saturated, then some interior component contains a complex formula. By Lemma 5.6, the procedure can bottom-up apply the corresponding logical rule to said formula and the obtained premises are LNSs. Hence appending the premises and marking them S preserves (I). (iii) If $\mathcal{H}$ is an S-leaf, non-initial, saturated, and not fully saturated, then either $\mathcal{H}$ is unfocused or it contains a single focused formula. If $\mathcal{H}$ is unfocused, then by Lemma 5.6 the procedure can bottom-up apply fb and the resulting premises are LNSs. If $\mathcal{H}$ contains a formula in focus, then by Lemma 5.6 the procedure can bottom-up apply ub and the resulting premises are LNSs. Thus in either case appending the corresponding premises gives a well-formed tree. Moreover, every premise remains non-initial, saturated and not fully saturated, and so we mark each premise with L. Note that (I) is preserved. (iv) If $\mathcal{H}$ is marked L, then $\mathcal{H}$ is non-initial, saturated and not fully saturated by (I)(3). This implies that $\mathcal{H}$ does not contain a complex formula in an interior component and thus must contain a complex formula in the end component. By Lemma 5.6 the procedure can bottom-up apply the corresponding logical rule to the complex formula and the resulting premises are LNSs. Thus appending these premises and marking them S preserves invariant (I).

Since each processed leaf has only finitely many premises, and π_n has only finitely many open leaves, the resulting tree π_{n+1} is finite. This finishes the induction.

Now define

$$\pi := \bigcup_{n \in \mathbb{N}} \pi_n$$

and delete all phase markings. Note that the root of π is labelled by $\mathcal{G}$ and every node by an LNS. If a node is not a leaf, then it is a non-leaf node of some π_n and therefore, by (I)(2) the node and its children form a correct rule instance of a rule of $\text{LNS}_{\text{LTL}}^+$. Since the construction never alters the children at any subsequent stage, the node and its children form a correct rule instance of a rule of $\text{LNS}_{\text{LTL}}^+$ in π . Therefore π is a pseudo-derivation of $\mathcal{G}$. Finally, every open leaf is processed at some stage of the construction, implying that every leaf of π is closed, and therefore initial or fully saturated. $\square$

Definition 5.8. Let $\mathcal{G}$ be an LNS. A pseudo-derivation of $\mathcal{G}$ obtained by the phased construction is called a proof-search tree for $\mathcal{G}$.

We will now show that every proof-search tree satisfies the SRP. The *length* $\text{len}(A)$ of a formula A is defined to be the number of symbols it contains. We define $\text{len}(A^a) := \text{len}(A)$ for $A \in \mathcal{L}$ and $a \in \{u, f\}$. For a set Γ of annotated formulae we define $\text{len}(\Gamma) := \sum_{A^a \in \Gamma} \text{len}(A^a)$. Given a Gentzen sequent $\Gamma \vdash \Delta$, we define the *size* of $\Gamma \vdash \Delta$ to be $s(\Gamma \vdash \Delta) := \text{len}(\Gamma) + \text{len}(\Delta)$. Now let $\mathcal{G} = \Gamma_0 \vdash \Delta_0 // \dots // \Gamma_{N-1} \vdash \Delta_{N-1}$ and define the vector

$$s_N(\mathcal{G}) := (s(\Gamma_0 \vdash \Delta_0), \dots, s(\Gamma_{N-1} \vdash \Delta_{N-1})).$$

For the following lemma, $<_l$ denotes the standard lexicographical order on $\mathbb{N}^N$. It is well-known that $<_l$ is a well-order.

Lemma 5.9. Let $\mathcal{H}$ be a premise and $\mathcal{G}$ the conclusion of a succinct logical rule instance of a rule r of $\text{LNS}_{\text{LTL}}^+$. If $\|\mathcal{H}\| = \|\mathcal{G}\| = N$, then $s_N(\mathcal{H}) <_l s_N(\mathcal{G})$.

Proof. Suppose the principal formula occurs in component i . By inspection of the rules, note that all components $j < i$ remain unchanged in the rule instance of r . In component i , since the rule instance is succinct, the principal formula is removed in the step from $\mathcal{G}$ to $\mathcal{H}$. If the principal formula is of the form $A \rightarrow B$ and r is an implication rule, then the auxiliary formulae introduced in component i of $\mathcal{H}$ are A and/or B and $\text{len}(A) + \text{len}(B) < \text{len}(A \rightarrow B)$. If the principal formula is of the form XA , and r is a next rule, then $r \in \{\text{XL}_1, \text{XR}_1\}$ and the auxiliary formula A is introduced in component $i + 1$. If the principal formula is of the form $A \text{UB} B$ and r is an until rule, then the auxiliary formulae introduced in component i are A and/or B and $\text{len}(A) + \text{len}(B) < \text{len}(A \text{UB} B)$. Therefore $s(\mathcal{H}(j)) = s(\mathcal{G}(j))$ for all $j < i$ and $s(\mathcal{H}(i)) < s(\mathcal{G}(i))$, which implies $s_N(\mathcal{H}) <_l s_N(\mathcal{G})$. $\square$

Corollary 5.10. *Every infinite branch $\mathcal{B}$ of a proof-search tree π passes through the logical phase infinitely often (and hence through the focusing phase infinitely often).*

Proof. Suppose otherwise. Then the phased construction of π passes only finitely many times through the logical phase, implying that a suffix of $\mathcal{B} = \{\mathcal{G}_i\}_{i \in \mathbb{N}}$ exists that contains only saturation steps. Note that the saturation phase only applies succinct length-preserving logical rules. Therefore there exists a suffix $\mathcal{P} = \{\mathcal{G}_i\}_{i \geq j}$ with $\mathcal{G}_i$ the conclusion and $\mathcal{G}_{i+1}$ a premise of a succinct logical rule and $\|\mathcal{G}_i\| = \|\mathcal{G}_{i+1}\| = N$ for all $i \geq j$. By Lemma 5.9 the sequence $\{s_N(\mathcal{G}_i)\}_{i \geq j}$ is an infinite and strictly decreasing sequence in $<_l$, contradicting the fact that $<_l$ is a well-order. $\square$

Corollary 5.11. *For every infinite branch $\mathcal{B} = \{\mathcal{G}_i\}_{i \in \mathbb{N}}$ of a proof-search tree π , $\{\|\mathcal{G}_i\|\}_{i \in \mathbb{N}}$ is unbounded.*

Proof. Suppose $\{\|\mathcal{G}_i\|\}_{i \in \mathbb{N}}$ is bounded. Then there is a suffix $\mathcal{P} = \{\mathcal{G}_i\}_{i \geq j}$ of $\mathcal{B}$ with $\|\mathcal{G}_i\| = \|\mathcal{G}_{i+1}\| = N$ for all $i \geq j$. If $\mathcal{G}_i$ is the conclusion and $\mathcal{G}_{i+1}$ the premise of a logical rule instance, then $s_N(\mathcal{G}_{i+1}) <_l s_N(\mathcal{G}_i)$ by Lemma 5.9. If $\mathcal{G}_i$ is the conclusion and $\mathcal{G}_{i+1}$ the premise of a branching focus rule, then $s_N(\mathcal{G}_{i+1}) \leq_l s_N(\mathcal{G}_i)$. Note that after every instance of a branching focus rule, there is at least one instance of a logical rule. Thus $\{s_N(\mathcal{G}_i)\}_{i \geq j}$ is a decreasing infinite sequence in $<_l$ which strictly decreases infinitely often, contradicting that $<_l$ is a well-order. $\square$

Lemma 5.12. *Every proof-search tree satisfies the SRP.*

Proof. Let $\mathcal{B} = \{\mathcal{G}_i\}_{i \in \mathbb{N}}$ be an infinite branch of π . By Corollary 5.10 $\mathcal{B}$ passes infinitely often through the logical phase. In each logical phase the underlying LNS is saturated. Hence $\mathcal{B}$ contains infinitely many saturated sequents. $\square$

Let A^a be a complex formula occurrence in an LNS $\mathcal{G}$ on a branch $\mathcal{B}$ of a proof-search tree. We say that A^a is *decomposed at $\mathcal{G}$* if $\mathcal{G}$ is the conclusion of a logical rule instance with A^a principal. Moreover, A^a is *eventually decomposed on $\mathcal{B}$* if there is a finite segment $\{\mathcal{G}_i\}_{i \leq k}$ of $\mathcal{B}$, with $\mathcal{G}_0 = \mathcal{G}$, along which the occurrence is carried through side occurrences and the annotation changes induced by the chosen premises of fb and ub, until its resulting occurrence is decomposed at $\mathcal{G}_k$. Before this decomposition, the occurrence remains in the same component and side of each LNS.

Lemma 5.13. *On an infinite branch $\mathcal{B}$ of a proof-search tree, every complex formula is eventually decomposed on $\mathcal{B}$. The same holds for every finite branch ending in a fully saturated sequent.*

Proof. We only consider the case for an infinite branch $\mathcal{B} = \{\mathcal{G}_i\}_{i \in \mathbb{N}}$. Let $i \in \mathbb{N}$ and suppose $\mathcal{G}_i$ contains a complex formula A^a in component j . By Corollary 5.11 and Lemma 5.12 there exists $k > i$ with $\mathcal{G}_k$ saturated and $\|\mathcal{G}_k\| > j + 1$. The j -component of $\mathcal{G}_k$ is therefore free of complex formulae, implying that A^a was decomposed at some node between $\mathcal{G}_i$ and $\mathcal{G}_k$. $\square$

We now define two dual notions of subtree within a proof-search tree. Intuitively, a *CB-structure* represents a potential proof of $\mathcal{G}$, while a *DB-structure* represents a potential counter-model.

Definition 5.14 (CB- and DB-Closure). *Let π be a proof-search tree for $\mathcal{G}$. A CB-structure is a subtree $(T, \prec)$ of π such that $\mathcal{G} \in T$ and for each non-leaf node $\mathcal{H} \in \pi$ belonging to T :*

- (1) *If $\mathcal{H}$ is the conclusion of a conjunctive rule in π , then all premises are in T ;*
- (2) *If $\mathcal{H}$ is the conclusion of a disjunctive rule in π , then exactly one premise is in T .*

A DB-structure is a subtree $(T, \prec)$ of π such that $\mathcal{G} \in T$ and for each non-leaf node $\mathcal{H} \in \pi$ belonging to T :

- (1) *If $\mathcal{H}$ is the conclusion of a conjunctive rule in π , then exactly one premise is in T ;*
- (2) *If $\mathcal{H}$ is the conclusion of a disjunctive rule in π , then all premises are in T .*

The CB-closure $\text{CB}(\pi)$ is the set of all CB-structures of π , and the DB-closure $\text{DB}(\pi)$ is the set of all DB-structures of π .

We remark that since a CB-structure selects exactly one premise at each disjunctive node, and since both fb and ub include a premise identical to the conclusion, a CB-structure may contain instances of *triv*. As noted above, such instances are admissible and can be eliminated by contracting the premise and conclusion into a single node. For the remaining (non-trivial) instances, a pruned fb application reduces to an instance of *f* (focusing a single formula), and a pruned ub application reduces to an instance of *u* (unfocusing a formula). For a CB-structure β , we write $N(\beta)$ for the tree obtained by contracting all unary instances of *triv*, and we identify the remaining pruned fb and ub applications with *f* and *u* instances, respectively.

A branch in a proof-search tree is *successful* if it either terminates at an initial sequent or is infinite and progressing; it is *failed* if it either terminates at a fully saturated sequent or is infinite and not progressing. Note that every branch is either successful or failed (but not both). The proof of the following lemma is then immediate.

Lemma 5.15. *Let π be a proof-search tree for $\mathcal{G}$, and let $\beta \in \text{CB}(\pi)$. Then $N(\beta)$ is an $\text{LNS}_{\text{LTL}}^\infty$ -derivation of $\mathcal{G}$ satisfying the SRP. Moreover, $N(\beta)$ is a proof if and only if every branch of β is successful.*

5.2 Completeness

We now show how to use CB- and DB-structures to construct proofs and countermodels, respectively, which yields completeness. We first prove some auxiliary lemmas.

Lemma 5.16. *Let π be a proof-search tree for $\mathcal{G}$. Then, there exists $\beta \in \text{CB}(\pi)$ such that $N(\beta)$ is a $\text{LNS}_{\text{LTL}}^\infty$ -proof iff every DB-structure in $\text{DB}(\pi)$ contains a successful branch.*

Proof. We interpret the proof-search tree π as the arena of a two-player game between *Prover* and *Refuter*:

- At conjunctive (binary) nodes, Refuter chooses a premise;
- At disjunctive (fb or ub) nodes, Prover chooses a premise;
- At unary nodes, the move is forced.

A *play* is a maximal path through π determined by the players' choices. Prover wins a play if it is successful; Refuter wins otherwise. We observe the following correspondence between strategies and closures:

- A *Prover strategy* determines a CB-structure: at disjunctive nodes, one premise is retained (Prover's choice), while at conjunctive nodes, both premises are retained. A Prover strategy is *winning* iff every branch in the corresponding CB-structure is successful.
- A *Refuter strategy* determines a DB-structure: at conjunctive nodes, one premise is retained (Refuter's choice), while at disjunctive nodes, all premises are retained. A Refuter strategy is *winning* iff the corresponding DB-structure has no successful branch.

Under this correspondence, we obtain:

$$\begin{aligned}
\exists \beta \in \text{CB}(\pi) : N(\beta) \text{ is a proof} &\iff \exists \beta \in \text{CB}(\pi) : \text{every branch of } \beta \text{ is successful} \\
&\iff \text{Prover has a winning strategy} \\
&\iff \text{Refuter has no winning strategy} \\
&\iff \text{every DB-structure has a successful branch}
\end{aligned}$$

The first equivalence follows from Lemma 5.15. The second equivalence follows from the strategy–structure correspondence: a winning Prover strategy is precisely a CB-structure in which every branch is successful. The fourth equivalence is analogous: a winning Refuter strategy is precisely a DB-structure in which every branch is failed. It remains to justify the third equivalence, which is the determinacy of the game.

The winning condition for Prover is Borel: whether a finite play is successful depends only on the leaf reached (a condition determined by a finite prefix), while whether an infinite play is progressing is determined by countable conditions on the play's steps—specifically, the existence of a starting index from which (i) every step preserves a formula in focus and (ii) infinitely many steps pass through the right premise of UL_1 or UL_2 with that formula principal. Each of these is expressible as a countable intersection or union of conditions depending on finitely many steps of the play. By Martin's Borel Determinacy Theorem [21], the game is determined, which completes the proof. $\square$

A *trace value* is a pair $\tau = \langle \text{AUB}^a, i \rangle$ where $\text{AUB} \in \mathcal{L}$, $a \in \{f, u\}$, and $i \in \mathbb{N}$. Given a linear nested sequent $\mathcal{G} = \Gamma_0 \vdash \Delta_0 // \dots // \Gamma_n \vdash \Delta_n$, a trace value $\langle \text{AUB}^a, i \rangle$ is a *trace value of* $\mathcal{G}$ if $\text{AUB}^a \in \Gamma_i$ for some $0 \leq i \leq n$.

Definition 5.17. Let r be a rule of $\text{LNS}_{\text{LTL}}^+$, and fix an instance with conclusion $\mathcal{G}$ and premise $\mathcal{H}$. A pair $(\langle \text{AUB}^a, i \rangle, \langle \text{AUB}^b, j \rangle)$ is a *trace pair for* $(\mathcal{G}, \mathcal{H})$ iff $\langle \text{AUB}^a, i \rangle$ is a trace value of $\mathcal{G}$, $\langle \text{AUB}^b, j \rangle$ is a trace value of $\mathcal{H}$, and one of the following holds:

- (1) r is a logical rule, the occurrence of AUB^a in the antecedent of component i is not principal in the rule instance, $a = b$, and $j = i$;
- (2) $r \in \{\text{UL}_1, \text{UL}_2\}$, the occurrence of AUB^a in the antecedent of component i is principal in the rule instance, $\mathcal{H}$ is its right premise, $a = b$, and $j = i + 1$;
- (3) $r = \text{fb}$, $a = u$, $b \in \{u, f\}$, and $j = i$;
- (4) $r = \text{ub}$, $j = i$, and either $\mathcal{H} = \mathcal{G}$ and $a = b$, or $\mathcal{H} = \mathcal{G}^u$ and $b = u$.

A trace pair is *progressing* if $j = i + 1$. A *trace* along a path $\mathcal{P} = \{\mathcal{G}_i\}_{i \in I}$ is a sequence of trace values $\{\tau_i\}_{i \in I}$ such that (τ_i, τ_{i+1}) is a trace pair for $(\mathcal{G}_i, \mathcal{G}_{i+1})$ for each i . A trace along an infinite path is *strongly progressing* if infinitely many of its trace pairs are progressing. An infinite path is *strongly progressing* if it admits a strongly progressing trace, and an infinite branch is *strongly progressing* if it contains a suffix that is a strongly progressing path.

Lemma 5.18. *Let π be a proof-search tree. If a DB-structure in $\text{DB}(\pi)$ does not contain a progressing branch, then it contains a branch that is not strongly progressing.*

Proof. Let $\mathcal{G}$ be the root of π , and let $\alpha \in \text{DB}(\pi)$ contain no progressing branch. If α contains a finite branch, then that branch is not strongly progressing, and we are done. Hence assume that α contains only infinite branches. In particular, every node of α has at least one child. Let

$$\mathcal{U} := \{C \in \text{sufo}(\mathcal{G}) \mid C = A \cup B \text{ for some } A, B \in \mathcal{L}\}.$$

This set is finite. Moreover, inspection of the rules shows that every formula occurring in π , after erasing its annotation, belongs to $\text{sufo}(\mathcal{G})$. Consequently, the underlying formula of every trace value occurring in π belongs to $\mathcal{U}$.

If $\mathcal{U} = \emptyset$, then no node of π has a trace value, so any branch of α is not strongly progressing. Thus assume that

$$\mathcal{U} = \{C_0, \dots, C_{M-1}\}, \quad M \geq 1,$$

where the C_j are pairwise distinct.

We recursively construct an infinite branch

$$\mathcal{B} = (\mathcal{G}_n)_{n \in \mathbb{N}}$$

of α , starting with $\mathcal{G}_0 := \mathcal{G}$. Simultaneously, we maintain a counter $q_n \in \mathbb{N}$ recording the number of fb applications encountered in the branch before reaching $\mathcal{G}_n$. Initially, let $q_0 := 0$.

Suppose that $\mathcal{G}_n$ and q_n have been defined. We distinguish three cases according to the rule applied at $\mathcal{G}_n$.

- (1) If the rule is conjunctive, let $\mathcal{G}_{n+1}$ be its unique premise retained in α , and set $q_{n+1} := q_n$.
- (2) If the rule is ub, let $\mathcal{G}_{n+1}$ be the unchanged premise, so that the current focus is retained, and set $q_{n+1} := q_n$.
- (3) If the rule is fb, let $j := q_n \bmod M$. If C_j^u occurs in an antecedent of $\mathcal{G}_n$, choose the premise that focuses this occurrence. Otherwise, choose the unchanged premise. In either case, set $q_{n+1} := q_n + 1$.

The choice in the third case is unambiguous. Indeed, the conclusion of an fb instance is unfocused and, by the phased construction, saturated. Its interior components therefore contain no until-formulae. Hence C_j^u , if present in an antecedent, occurs only in the end component.

All these choices are available in α : it retains exactly one premise at every nonterminal conjunctive node and all premises at every disjunctive node. Since α contains no finite branch, the construction continues indefinitely and defines an infinite branch $\mathcal{B}$. Every node of this branch has the same rule application as in π , so $\mathcal{B}$ is also an infinite branch of π .

We establish three claims.

Claim 1. $\forall n \in \mathbb{N} \exists k \geq n \mathcal{G}_k \text{ is unfocused.}$

Suppose otherwise. Then there exists $n \in \mathbb{N}$ such that every $\mathcal{G}_k$, for $k \geq n$, contains a focused formula. Let τ_k be the unique focused trace value of $\mathcal{G}_k$. For every $k \geq n$, the pair (τ_k, τ_{k+1}) is a trace pair: fb cannot be applied to a focused sequent, and at every ub instance our construction selects the unchanged premise. Moreover, in a logical rule instance, the focused occurrence either remains a side occurrence or is principal in an instance of UL_1 or UL_2 . In the latter case, the branch must take the right premise, since the left premise is unfocused by Lemma 3.3.

By Lemma 5.13, the focused occurrence at any $\mathcal{G}_k$, with $k \geq n$, is eventually decomposed. Until that decomposition, it remains focused. Thus, for every $k \geq n$, there exists $\ell \geq k$ such that the focused occurrence is principal in an instance of UL_1 or UL_2 at $\mathcal{G}_\ell$. As observed above, the branch takes the right premise of that instance. Consequently, the suffix $(\mathcal{G}_k)_{k \geq n}$ has a continuous focus trace and passes infinitely often through the right premise of an until-left rule with the focused formula principal. Therefore $\mathcal{B}$ is progressing, contrary to the assumption on α . This proves Claim 1.

Claim 2. The branch $\mathcal{B}$ passes through infinitely many fb instances.

Fix an arbitrary $n \in \mathbb{N}$. By Corollary 5.10, there is a focusing-phase application at some $\mathcal{G}_s$ with $s \geq n$. If that application is fb, we have found an fb instance at an index at least n . Otherwise, the application is ub, so $\mathcal{G}_s$ is focused. By Claim 1, there exists $k > s$ such that $\mathcal{G}_k$ is unfocused. Let $\mathcal{G}_t$, with $t \geq k$, be the first subsequent conclusion of a focusing-phase application. Such a node exists by Corollary 5.10.

Between $\mathcal{G}_k$ and $\mathcal{G}_t$, only logical rules are applied. A logical rule cannot introduce a focused formula into an unfocused LNS. Hence $\mathcal{G}_t$ is unfocused, and the focusing rule applied there must be fb. We have therefore found an fb instance at an index at least n . Since n was arbitrary, there are infinitely many such instances.

Claim 3. No suffix of $\mathcal{B}$ admits an infinite trace.

Suppose, toward a contradiction, that $\tau = (\tau_n)_{n \geq m}$ is an infinite trace along a suffix of $\mathcal{B}$. By the definition of trace pairs, the underlying until-formula is constant along τ . Thus, for some $j < M$,

$$\tau_n = \langle C_j^{a_n}, i_n \rangle \quad \text{for every } n \geq m.$$

By Claim 2, we can enumerate the indices of the fb applications along $\mathcal{B}$ as

$$s_0 < s_1 < s_2 < \dots$$

By construction, $q_{s_r} = r$ for every $r \in \mathbb{N}$. Choose r such that

$$s_r \geq m \quad \text{and} \quad r \bmod M = j.$$

Such an r exists because there are infinitely many natural numbers congruent to j modulo M . The sequent $\mathcal{G}_{s_r}$ is unfocused and saturated. Since τ_{s_r} is a trace value of this sequent, we have

$$a_{s_r} = \mathbf{u}, \quad i_{s_r} = \|\mathcal{G}_{s_r}\| - 1.$$

In particular, C_j^u occurs in the antecedent of the end component of $\mathcal{G}_{s_r}$.

At this fb application, our construction of $\mathcal{B}$ selects C_j . Therefore $\mathcal{G}_{s_r+1}$ is the premise obtained by replacing C_j^u with C_j^f . Hence, by the definition of trace pairs,

$$\tau_{s_r+1} = \langle C_j^f, \|\mathcal{G}_{s_r+1}\| - 1 \rangle.$$

We show by induction on $n \geq s_r + 1$ that τ_n is focused. The base case was just established. For the inductive step, suppose that τ_n is focused. Then $\mathcal{G}_n$ is focused, so the rule applied there cannot be fb. If it is ub, our construction selects the unchanged premise, and the trace-pair definition preserves the annotation. If it is a logical rule, every trace pair preserves the annotation as well. Thus τ_{n+1} is focused. Consequently, every node of the suffix $(\mathcal{G}_n)_{n \geq s_r+1}$ is focused, contradicting Claim 1. This proves Claim 3.

In particular, no suffix of $\mathcal{B}$ admits a strongly progressing trace. Therefore $\mathcal{B}$ is not strongly progressing, as required. $\square$

We now show that every LNS either has an $\text{LNS}_{\text{LTL}}^\infty$ -proof satisfying the SRP or is falsifiable, from which completeness follows immediately.

Lemma 5.19. *Let $\mathcal{G}$ be an LNS. Then, either $\mathcal{G}$ has an $\text{LNS}_{\text{LTL}}^\infty$ -proof satisfying the SRP or there exists a state sequence $\bar{\sigma}$ such that $\bar{\sigma} \not\models \mathcal{G}$.*

Proof. Let $\mathcal{G}$ be an LNS and let π be a proof-search tree for $\mathcal{G}$. If there exists $\alpha \in \text{CB}(\pi)$ such that $N(\alpha)$ is a $\text{LNS}_{\text{LTL}}^\infty$ -proof of $\mathcal{G}$, then it satisfies the SRP by Lemma 5.12. Suppose otherwise that for every $\alpha \in \text{CB}(\pi)$, $N(\alpha)$ is not a $\text{LNS}_{\text{LTL}}^\infty$ -proof. By Lemma 5.16, some DB-structure α in $\text{DB}(\pi)$ contains only failed branches. In particular, α contains no progressing branch, so by Lemma 5.18, α contains a branch $\mathcal{B}$ that is not strongly progressing. Either $\mathcal{B}$ is finite, in which case failure of the branch implies it must end in a fully saturated leaf, or infinite and not strongly progressing. We treat only the latter case, as the former is argued similarly. Let $\mathcal{B} := \{\mathcal{G}_i\}_{i \in \mathbb{N}}$ be an infinite branch of α that is *not* strongly progressing.

We use $\mathcal{B}$ to construct a state sequence $\bar{\sigma}$ such that $\bar{\sigma} \not\models \mathcal{G}$. Define:

$$\Gamma_i := \bigcup_{j \in \mathbb{N}, \mathcal{G}_j(i) = (\Gamma \vdash \Delta)} \Gamma^- \quad \Delta_i := \bigcup_{j \in \mathbb{N}, \mathcal{G}_j(i) = (\Gamma \vdash \Delta)} \Delta^-$$

That is, Γ_i collects all unannotated formulae appearing in the antecedent of the i -component along $\mathcal{B}$, and Δ_i collects those in the consequent. We define $\sigma_i := \{p \mid p \in \Gamma_i\}$ and $\bar{\sigma} := (\sigma_0, \sigma_1, \dots)$. We prove the following by mutual induction on the length of A :

- (1) If $A \in \Gamma_i$, then $\bar{\sigma}, i \models A$;
- (2) If $A \in \Delta_i$, then $\bar{\sigma}, i \not\models A$.

CASE $A = \perp \in \Gamma_i$. There exists $m \in \mathbb{N}$ such that $\perp \in \mathcal{G}_m(i)$, making $\mathcal{G}_m$ an initial sequent. Therefore, once the phased construction enters the saturation phase for the first time after $\mathcal{G}_m$, the open leaf is closed as an initial node, contradicting the assumption that $\mathcal{B}$ is infinite. Hence $\perp \notin \Gamma_i$.

CASE $A = \perp \in \Delta_i$. By definition, $\bar{\sigma}, i \not\models \perp$.

CASE $A = p \in \Gamma_i$. By definition, $p \in \sigma_i$, so $\bar{\sigma}, i \models p$.

CASE $A = p \in \Delta_i$. Suppose for contradiction that $p \in \Gamma_i$ as well. Then there exist $m_0, m_1 \in \mathbb{N}$ with $\mathcal{G}_{m_0}(i) = \Sigma \vdash \Pi$ and $\mathcal{G}_{m_1}(i) = \Sigma' \vdash \Pi'$ such that $p \in \Sigma$ and $p \in \Pi'$. Without loss of generality, assume $m_0 \leq m_1$. Since no rule in $\text{LNS}_{\text{LTL}}^\infty$ deletes atoms, $p \in \Sigma$ implies $p \in \Sigma'$. Therefore $p \in \Sigma' \cap \Pi'$, making $\mathcal{G}_{m_1}$ an instance of *id*. By the same argument as in case $\perp \in \Gamma_i$, this implies that the branch $\mathcal{B}$ is finite, contradicting our assumption. Hence $p \notin \Gamma_i$, and so $\bar{\sigma}, i \not\models p$.

CASE $A = B \rightarrow C \in \Delta_i$. There exists $m \in \mathbb{N}$ such that $B \rightarrow C \in \Pi$ with $\mathcal{G}_m(i) = \Sigma \vdash \Pi$. Lemma 5.13 implies that $B \rightarrow C$ is eventually decomposed. Hence, for some $l > m$ with $\mathcal{G}_l(i) = \Sigma' \vdash \Pi'$, we have $B \in \Sigma'$ and $C \in \Pi'$, giving $B \in \Gamma_i$ and $C \in \Delta_i$. By IH, $\bar{\sigma}, i \models B$ and $\bar{\sigma}, i \not\models C$, so $\bar{\sigma}, i \not\models B \rightarrow C$. The case for $A = B \rightarrow C \in \Gamma_i$ is similar and omitted.

CASE $A = XB \in \Gamma_i$. There exists $m \in \mathbb{N}$ such that $XB \in \Sigma$ with $\mathcal{G}_m(i) = \Sigma \vdash \Pi$. By Lemma 5.13, XB is eventually decomposed. Hence, for some $l > m$ with $\mathcal{G}_l(i+1) = \Sigma' \vdash \Pi'$, we have $B \in \Sigma'$, giving $B \in \Gamma_{i+1}$. By IH, $\bar{\sigma}, i+1 \models B$, and so $\bar{\sigma}, i \models XB$. The case for $A = XB \in \Delta_i$ is similar and omitted.

CASE $A = BUC \in \Gamma_i$. There exist $m \in \mathbb{N}$ and an annotation $a_m \in \{u, f\}$ such that $(BUC)^{a_m}$ occurs in the antecedent of component i of $\mathcal{G}_m$. We follow this particular occurrence along $\mathcal{B}$, allowing its annotation to change at branching focusing steps. More precisely, starting with

$$\tau_m := \langle (BUC)^{a_m}, i \rangle,$$

we recursively construct trace values

$$\tau_n = \langle (BUC)^{a_n}, j_n \rangle \quad (n \geq m)$$

until the tracked occurrence is principal in an until-left inference whose left premise is selected by $\mathcal{B}$. At each stage for which τ_n is defined, we maintain the following invariant:

- (i) $(\tau_k)_{m \leq k \leq n}$ is a trace along $(\mathcal{G}_k)_{m \leq k \leq n}$;
- (ii) $j_n \geq i$ and $B \in \Gamma_\ell$ for every $i \leq \ell < j_n$.

The invariant holds initially: τ_m is a trace value of $\mathcal{G}_m$, $j_m = i$, and the last condition is vacuous.

Suppose τ_n has been defined. We distinguish the following cases according to the rule applied at $\mathcal{G}_n$. If the rule is logical and the occurrence represented by τ_n is not principal, then it is carried as a side occurrence into $\mathcal{G}_{n+1}$. Set $\tau_{n+1} := \tau_n$. This gives a trace pair by the side-occurrence clause of the trace-pair definition. The component index is unchanged, so the invariant is preserved.

If the rule is fb or ub, follow the annotation change, if any, induced by the premise selected by $\mathcal{B}$. Let a_{n+1} be the resulting annotation and set $\tau_{n+1} := \langle (BUC)^{a_{n+1}}, j_n \rangle$. If unfocusing merges the tracked occurrence with an already-present unfocused entry, continue with the resulting entry. In every case, (τ_n, τ_{n+1}) is a trace pair by the corresponding branching-focusing clause. Since the component index remains unchanged, the invariant is preserved.

Finally, suppose the tracked occurrence is principal in a logical rule instance. Since the formula is of the form BUC and occurs in the antecedent, the rule is UL_1 or UL_2 . If $\mathcal{B}$ takes the left premise, then C occurs in the antecedent of component j_n of $\mathcal{G}_{n+1}$. Consequently, $C \in \Gamma_{j_n}$, and we stop the construction.

If $\mathcal{B}$ takes the right premise, then B occurs in the antecedent of component j_n of $\mathcal{G}_{n+1}$, and $(BUC)^{a_n}$ occurs in the antecedent of component $j_n + 1$. Hence $B \in \Gamma_{j_n}$. Set $\tau_{n+1} := \langle (BUC)^{a_n}, j_n + 1 \rangle$. The pair (τ_n, τ_{n+1}) is progressing. Moreover, the previous invariant, together with $B \in \Gamma_{j_n}$, gives

$$B \in \Gamma_\ell \quad \text{for every } i \leq \ell < j_{n+1}.$$

Thus the invariant is again preserved.

We claim that this construction stops after finitely many steps. Suppose otherwise. It then defines an infinite trace

$$\tau = (\tau_n)_{n \geq m}$$

along a suffix of $\mathcal{B}$. This trace must have infinitely many progressing pairs. Indeed, suppose it had only finitely many. Choose $N \geq m$ such that no pair (τ_n, τ_{n+1}) , for $n \geq N$, is progressing. The tracked occurrence cannot be principal in any logical inference at or after $\mathcal{G}_N$: a right-premise until-left step would be progressing, while a left-premise until-left step would stop the construction. Consequently, the occurrence represented by τ_N would never be logically decomposed, contrary to Lemma 5.13.

Thus τ is strongly progressing. This contradicts the choice of $\mathcal{B}$ as a branch that is not strongly progressing. The construction therefore stops at some index $s \geq m$, where the tracked occurrence is principal in an until-left rule instance and $\mathcal{B}$ takes its left premise.

Let $j := j_s$. By the invariant and the stopping condition,

$$B \in \Gamma_\ell \quad \text{for every } i \leq \ell < j, \quad C \in \Gamma_j.$$

Hence the induction hypotheses yield

$$\bar{\sigma}, \ell \models B \quad \text{for every } i \leq \ell < j, \quad \bar{\sigma}, j \models C.$$

Since $j \geq i$, the state index j witnesses $\bar{\sigma}, i \models BUC$.

CASE $A = BUC \in \Delta_i$. There exists $m \in \mathbb{N}$ such that BUC occurs in the consequent of component i of $\mathcal{G}_m$. Again, we follow this particular occurrence and its successive residuals. Since trace values are defined only for antecedent until-occurrences, we describe the consequent-occurrence construction directly, without calling it a trace.

Initially, set $j_m := i$ and select the given occurrence in $\mathcal{G}_m$. At each stage n , the selected occurrence is an occurrence of BUC in the consequent of component j_n of $\mathcal{G}_n$. We maintain the following invariant:

- (i) $j_n \geq i$;
- (ii) $C \in \Delta_\ell$ for every $i \leq \ell < j_n$.

The invariant holds initially, since $j_m = i$. Suppose the selected occurrence has been defined at $\mathcal{G}_n$. If it is not principal in the rule applied there, follow its unchanged occurrence in the consequent of the same component of $\mathcal{G}_{n+1}$, and set $j_{n+1} := j_n$. This includes branching focusing steps, since fb and ub leave all consequents unchanged. The invariant is preserved.

If the selected occurrence is principal, then the rule is UR_1 or UR_2 . If $\mathcal{B}$ takes the left premise, then both B and C occur in the consequent of component j_n of $\mathcal{G}_{n+1}$. Thus $B, C \in \Delta_{j_n}$, and we stop the construction. If $\mathcal{B}$ takes the right premise, then C occurs in the consequent of component j_n of $\mathcal{G}_{n+1}$, and BUC occurs in the consequent of component $j_n + 1$. Hence $C \in \Delta_{j_n}$. Select this residual occurrence of BUC and set $j_{n+1} := j_n + 1$. If the residual coincides with an already-present consequent entry, continue with that entry. The invariant is preserved, since it now also holds at the additional index j_n . There are two possibilities.

Subcase 1: the construction continues indefinitely. In this case the selected occurrence is principal infinitely often. Otherwise, after its last decomposition, a selected residual would never be decomposed, contrary to Lemma 5.13. Every such decomposition is an until-right inference whose right premise is selected, since selecting its left premise would stop the construction. Therefore the sequence $(j_n)_{n \geq m}$ increases by one infinitely often and is otherwise constant. Since $j_m = i$, it is unbounded. Fix any $\ell \geq i$. Choose $n \geq m$ such that $j_n > \ell$. By the invariant, $C \in \Delta_\ell$. Since ℓ was arbitrary, this holds for every $\ell \geq i$. Applying the induction hypothesis to the proper subformula C , we obtain

$$\bar{\sigma}, \ell \not\models C \quad \text{for every } \ell \geq i.$$

Thus there is no future state witnessing C , and consequently $\bar{\sigma}, i \not\models BUC$.

Subcase 2: the construction stops after finitely many steps. Let $s \geq m$ be the stopping index and put $j := j_s$. The branch takes the left premise of the until-right inference applied to the selected occurrence at $\mathcal{G}_s$. By the invariant and the stopping condition,

$$C \in \Delta_\ell \quad \text{for every } i \leq \ell \leq j, \quad B \in \Delta_j.$$

The induction hypotheses therefore give

$$\bar{\sigma}, \ell \not\models C \quad \text{for every } i \leq \ell \leq j, \quad \bar{\sigma}, j \not\models B.$$

Suppose, toward a contradiction, that $\bar{\sigma}, i \models BUC$. Then there exists $h \geq i$ such that

$$\bar{\sigma}, h \models C \quad \text{and} \quad \bar{\sigma}, \ell \models B \quad \text{for every } i \leq \ell < h.$$

If $h \leq j$, the first condition contradicts $\bar{\sigma}, h \not\models C$. If $h > j$, then $i \leq j < h$, so the second condition implies $\bar{\sigma}, j \models B$, contradicting $\bar{\sigma}, j \not\models B$. Both possibilities lead to a contradiction. Hence $\bar{\sigma}, i \not\models BUC$.

Finally, let $\mathcal{G}(i) = \Sigma_i \vdash \Pi_i$ for $0 \leq i < \|\mathcal{G}\|$. Since $\Sigma_i^- \subseteq \Gamma_i$ and $\Pi_i^- \subseteq \Delta_i$ for each $0 \leq i < \|\mathcal{G}\|$, it follows that $\bar{\sigma}, i \not\models \mathcal{G}(i)$ for all $0 \leq i < \|\mathcal{G}\|$. Hence, by Definition 3.1, $\bar{\sigma} \not\models \mathcal{G}$. $\square$

Theorem 5.20 (Completeness of $\text{LNS}_{\text{LTL}}^\infty$). *If $\mathcal{G}$ is valid, then $\mathcal{G}$ has an $\text{LNS}_{\text{LTL}}^\infty$ -proof satisfying the SRP.*

Proof. Suppose $\mathcal{G}$ is valid. By Lemma 5.19 either (i) $\mathcal{G}$ has a $\text{LNS}_{\text{LTL}}^\infty$ -proof which satisfies the SRP or (ii) there exists a state sequence $\bar{\sigma}$ such that $\bar{\sigma} \not\models \mathcal{G}$. Case (ii) cannot happen since $\mathcal{G}$ is valid. Hence $\mathcal{G}$ has a $\text{LNS}_{\text{LTL}}^\infty$ -proof which satisfies the SRP. $\square$

As a consequence of soundness, completeness and semantic invertibility (Lemma 4.2), we obtain the invertibility of all rules of $\text{LNS}_{\text{LTL}}^\infty$.

Lemma 5.21 (Invertibility). *All rules r of $\text{LNS}_{\text{LTL}}^\infty$ are invertible: for every instance of r with conclusion $\mathcal{G}$ and premises $\mathcal{H}_i$ for $1 \leq i \leq n$, if $\mathcal{G}$ has a $\text{LNS}_{\text{LTL}}^\infty$ -proof, then $\mathcal{H}_i$ has an $\text{LNS}_{\text{LTL}}^\infty$ -proof for every $1 \leq i \leq n$.*

6 Correspondence between Non-Wellfounded and Cyclic Proofs

We now establish bi-directional proof transformations between non-wellfounded and cyclic proofs. We first address cycle recognition in non-wellfounded proofs. In order to turn non-wellfounded proofs into cyclic proofs, we will make use of the SRP, which guarantees that infinite branches contain suitable ‘repetitions’ where they can be pruned and replaced by cycles.

Transforming cyclic proofs into non-wellfounded ones is non-trivial in the LNS setting. The key idea is captured by the Shifting Lemma (Lemma 6.13). Given a cyclic proof (π, L, c) whose conclusion $c(\mathcal{G})$ is the companion of a bud $\mathcal{G}$, we construct a new cyclic proof of $\mathcal{G}$ by shifting inferences forward in (π, L, c) , yielding a cyclic proof (π', L', c') . Iterating this construction allows one to unravel cycles ad infinitum, thereby producing a non-wellfounded derivation from a cyclic one. By carefully handling annotations in the construction, we can then show that the derivation is a non-wellfounded proof which satisfies the SRP. Together, these bi-directional transformations establish the soundness and completeness of $\text{LNS}_{\text{LTL}}^{\text{cyc}}$.

6.1 From Non-Wellfounded Proofs to Cyclic Proofs

This section shows how non-wellfounded proofs which satisfy the SRP can be transformed into cyclic proofs. We call a rule $r \in \{\text{XL}_2, \text{XR}_2, \text{UL}_2, \text{UR}_2\}$ an *expansion rule* (cf. [19]). We first show that every infinite branch in a non-wellfounded proof passes infinitely often through an expansion rule. We then show that every infinite branch in a non-wellfounded proof which satisfies the SRP contains a suitable ‘repetition’ and finally use these two results to establish the translation.

Lemma 6.1. *If π is an $\text{LNS}_{\text{LTL}}^\infty$ -proof and $\mathcal{B}$ is an infinite branch of π , then $\mathcal{B}$ passes infinitely often through XL_2 , XR_2 , or through the right premise of UL_2 or UR_2 .*

Proof. Let π be an $\text{LNS}_{\text{LTL}}^\infty$ -proof and suppose, toward a contradiction, that $\mathcal{B} = \{\mathcal{G}_i\}_{i \in \mathbb{N}}$ is an infinite branch of π , which does not pass infinitely often through an expansion rule. Then, there exists a $j \in \mathbb{N}$ such that the suffix $\mathcal{P} = \{\mathcal{G}_i\}_{i \geq j}$ of $\mathcal{B}$ starting at $\mathcal{G}_j$ never passes through instances of XL_2 , XR_2 or the right premises of UL_2 or UR_2 . Therefore, $\|\mathcal{G}_i\| = \|\mathcal{G}_k\|$ for all $i, k \geq j$. Since on a continuously focused path, the component index of the focused formula is unchanged by non-progressing steps and increases by one at each progressing step, a progressing suffix cannot have bounded LNS length. Therefore $\mathcal{B}$ does not have a progressing suffix and is thus not progressing. This contradicts our assumption that π is a proof. $\square$

Lemma 6.2. *Let π be a non-wellfounded proof which satisfies the SRP and let $\mathcal{B}$ be an infinite branch. Then, $\mathcal{B}$ contains two saturated LNSs $\mathcal{G}$ and $\mathcal{H}$ such that $\mathcal{G}$ occurs strictly below $\mathcal{H}$ in $\mathcal{B}$, $\|\mathcal{G}\| < \|\mathcal{H}\|$, the path from $\mathcal{G}$ to $\mathcal{H}$ contains a continuous focus trace and goes through the right premise of $r \in \{\text{UL}_1, \text{UL}_2\}$ with the formula in focus principal, and $\text{end}(\mathcal{G}) = \text{end}(\mathcal{H})$.*

Proof. Let π be an $\text{LNS}_{\text{LTL}}^\infty$ -proof of $\mathcal{F}$ satisfying the SRP. All rules of $\text{LNS}_{\text{LTL}}^\infty$ are *analytic*, i.e., if $\mathcal{H}$ is the conclusion and $\mathcal{H}'$ a premise of a rule, then $\text{sufo}(\mathcal{H}') \subseteq \text{sufo}(\mathcal{H})$. Thus, there are at most $2^{3 \times |\text{sufo}(\mathcal{F})|}$ many Gentzen sequents that can occur as components of an LNS in π . Since π satisfies the SRP, every infinite branch encounters infinitely many saturated sequents. Since there are at most $2^{3 \times |\text{sufo}(\mathcal{F})|}$ many Gentzen sequents that can serve as an end component, every infinite branch must contain infinitely many saturated sequents with identical end components. Moreover, since π is a proof, every infinite branch $\mathcal{B}$ is progressing and thus contains a suffix with a continuous focus trace that passes infinitely often through the right premise of UL_1 or UL_2 with the formula in focus principal. Finally, by Lemma 6.1, for every infinite branch $\mathcal{B} = \{\mathcal{G}_i\}_{i \in \mathbb{N}}$ there are infinitely many $i \in \mathbb{N}$ such that $\|\mathcal{G}_i\| < \|\mathcal{G}_{i+1}\|$. Thus, every infinite branch must contain a pair of saturated sequents $\mathcal{G}$ and $\mathcal{H}$ such that $\|\mathcal{G}\| < \|\mathcal{H}\|$, the path from $\mathcal{G}$ to $\mathcal{H}$ contains a continuous focus trace and goes through the right premise of $r \in \{\text{UL}_1, \text{UL}_2\}$ with the formula in focus principal, and $\text{end}(\mathcal{G}) = \text{end}(\mathcal{H})$. $\square$

Lemma 6.2 implies that every infinite branch of a proof π which satisfies the SRP contains a suitable candidate for a cycle, which allows us to extract a cyclic proof.

Theorem 6.3. *Every non-wellfounded proof satisfying the SRP can be transformed into a cyclic proof.*

Proof. Let π be an $\text{LNS}_{\text{LTL}}^\infty$ -proof of $\mathcal{G}$ which satisfies the SRP. Let π_0 be the subtree of π obtained by pruning every infinite branch of π at the lowermost sequent $\mathcal{H}$ such that $\mathcal{H}$ is saturated, there exists a saturated sequent $\mathcal{K}$ strictly below $\mathcal{H}$ with $\|\mathcal{K}\| < \|\mathcal{H}\|$, the path from $\mathcal{K}$ to $\mathcal{H}$ contains a continuous focus trace and passes through the right premise of $r \in \{\text{UL}_1, \text{UL}_2\}$ with the formula in focus principal, and $\text{end}(\mathcal{K}) = \text{end}(\mathcal{H})$. For each such $\mathcal{H}$ we call $\mathcal{K}$ its companion. Since π is finitely branching, König's Lemma and Lemma 6.2 imply that π_0 is finite and therefore a finite derivation of $\mathcal{G}$. Let L be the set of leaves of π_0 that were obtained by pruning a branch and let c be the function which maps each such leaf to its companion. Observe that every other leaf which is not contained in L is also a leaf of π and therefore an initial sequent. Hence, the tuple (π_0, L, c) is a cyclic proof of $\mathcal{G}$. $\square$

We obtain the completeness of the cyclic LNS system as a corollary of Theorems 5.20 and 6.3.

Corollary 6.4 (Completeness of $\text{LNS}_{\text{LTL}}^{\text{cyc}}$). *If $\mathcal{G}$ is valid, then $\mathcal{G}$ has an $\text{LNS}_{\text{LTL}}^{\text{cyc}}$ -proof.*

6.2 From Cyclic Proofs to Non-Wellfounded Proofs

In this section, we prove the soundness of cyclic proofs by means of a proof transformation. We introduce a technique for unraveling cyclic linear nested sequent proofs; in particular, we show how a cyclic proof rooted at the companion of a cycle can be transformed into a cyclic proof of the corresponding bud (Lemma 6.13). This is achieved by shifting rule applications forward from the source proof to the target proof while maintaining a suitable invariant, ensuring that the transformation can be iterated indefinitely. By iterating this shifting procedure, we obtain an infinite sequence of growing cyclic proofs, whose limit is a non-wellfounded derivation. We then show that the limit is in fact a non-wellfounded proof which satisfies the SRP. The main difficulty for the last step is to prove that every infinite branch of the limit derivation is progressing. To show this, we will identify every infinite branch in the limit with an infinite *looping path* of the original cyclic proof. Intuitively, a looping path is a path through a cyclic

proof that, upon reaching a bud, may continue at its companion. Due to the use of annotations, we can guarantee that every infinite looping path in a cyclic proof is progressing. This property transfers to the corresponding infinite branch in the limit derivation, thus showing that the limit is a non-wellfounded proof. We begin by introducing looping paths and by showing that every infinite looping path has a suffix with a continuous focus trace.

Definition 6.5. Let (π, L, c) be a cyclic proof. A looping path of (π, L, c) is a finite or infinite sequence $\mathcal{P}^\circ = \{\mathcal{G}_i\}_{i \in I}$ (where I is an index set) of LNSs in π such that the following hold:

- (1) $\mathcal{G}_0$ is the root of π ;
- (2) If $\mathcal{G}_i$ is a leaf of π and $\mathcal{G}_i \notin L$, then $\mathcal{P}^\circ$ ends at $\mathcal{G}_i$;
- (3) If $\mathcal{G}_i \in L$, then either $\mathcal{P}^\circ$ ends at $\mathcal{G}_i$ or $\mathcal{G}_{i+1}$ is a child of $c(\mathcal{G}_i)$;
- (4) Otherwise, $\mathcal{G}_{i+1}$ is a child of $\mathcal{G}_i$.

If $\mathcal{G}_i \in L$ and $\mathcal{G}_{i+1}$ is a child of $c(\mathcal{G}_i)$, then we say that $\mathcal{P}^\circ$ passes through the leaf $\mathcal{G}_i$. Note that if $\mathcal{P}^\circ$ is an infinite looping path, then there exists a leaf $\mathcal{G}_i \in L$ through which $\mathcal{P}^\circ$ passes infinitely often. We denote by $\mathcal{P}_\infty^\circ$ the set of leaves in L through which $\mathcal{P}^\circ$ passes infinitely often. Note that $\mathcal{P}_\infty^\circ$ is finite since π is finite. Given a cyclic proof $C = (\pi, L, c)$, let

$$\text{src}_C(v) := \begin{cases} c(v) & \text{if } v \in L \\ v & \text{else.} \end{cases}$$

We call $\text{src}(v)$ the *source* of v . In order to distinguish looping paths from ordinary paths through derivations, we also call the latter *upward paths*. We introduce a pre-order on the set of buds of π as follows.

Definition 6.6. Let (π, L, c) be a cyclic proof. The dependency order $\preceq$ on L is given by: $\mathcal{G}_i \preceq \mathcal{G}_j$ if and only if there exists an upward path from $c(\mathcal{G}_j)$ to $c(\mathcal{G}_i)$. A $\preceq$ -maximal leaf of $L' \subseteq L$ is a leaf $\mathcal{G}_i \in L'$ such that for every leaf $\mathcal{G}_j \in L'$, if $\mathcal{G}_i \preceq \mathcal{G}_j$, then $\mathcal{G}_j \preceq \mathcal{G}_i$.

The following lemma was proven in [37, Lemma 6.10].

Lemma 6.7. Let (π, L, c) be a cyclic proof and $\mathcal{P}^\circ$ an infinite looping path. Then, every $\preceq$ -maximal leaf of $\mathcal{P}_\infty^\circ$ has the same companion.

Lemma 6.8. Let (π, L, c) be a cyclic proof and $\mathcal{P}^\circ$ an infinite looping path. Let $\mathcal{G}_0$ be $\preceq$ -maximal in $\mathcal{P}_\infty^\circ$. Then, for every $\mathcal{G}_i \in \mathcal{P}_\infty^\circ$ the upward path from $c(\mathcal{G}_0)$ to $\mathcal{G}_i$ has a continuous focus trace.

Proof. Recall that by Lemma 6.7 every $\preceq$ -maximal bud in $\mathcal{P}_\infty^\circ$ has the same companion and therefore an upward path from $c(\mathcal{G}_0)$ to $\mathcal{G}_i$ exists for every $\mathcal{G}_i \in \mathcal{P}_\infty^\circ$. Let $\mathcal{G}_i \in \mathcal{P}_\infty^\circ$ be arbitrary and let $\mathcal{H}$ be any LNS on the upward path from $c(\mathcal{G}_0)$ to $\mathcal{G}_i$. We show that $\mathcal{H}$ is focused.

Suppose that $\mathcal{H}$ lies on the upward path from $c(\mathcal{G}_0)$ to $\mathcal{G}_0$. Since this path has a continuous focus trace, $\mathcal{H}$ is focused. Otherwise $\mathcal{H}$ does not lie on the upward path from $c(\mathcal{G}_0)$ to $\mathcal{G}_0$. Consider the subtree T of π rooted at $\mathcal{H}$. By assumption, $\mathcal{G}_i \in T$ and $\mathcal{G}_0 \notin T$. Since $\mathcal{P}_\infty^\circ$ visits both $\mathcal{G}_0$ and $\mathcal{G}_i$ infinitely often, it must leave T after some visit of $\mathcal{G}_i$. An ordinary upward path cannot leave T , therefore the exit must occur at some bud $\mathcal{F}$ of T by moving to its companion $c(\mathcal{F})$ which occurs outside of T . In fact, $c(\mathcal{F})$ must lie below $\mathcal{H}$, since T is rooted at $\mathcal{H}$, and so $\mathcal{H}$ lies on the upward path from $c(\mathcal{F})$ to $\mathcal{F}$. This implies that $\mathcal{H}$ is focused. Since $\mathcal{G}_i$ and $\mathcal{H}$ were arbitrary, this finishes the proof. $\square$

Lemma 6.9. Let (π, L, c) be a cyclic proof and $\mathcal{P}^\circ$ an infinite looping path. Then, $\mathcal{P}^\circ$ has a suffix with a continuous focus trace.

Proof. Let $\mathcal{P}^\circ = \{\mathcal{G}_i\}_{i \in \mathbb{N}}$ and let $\mathcal{G}$ be $\preceq$ -maximal in $\mathcal{P}^\circ$. Let j be the least natural number such that $\mathcal{G}_j = \mathcal{G}$ and the suffix $\mathcal{P}$ of $\mathcal{P}^\circ$ starting at $\mathcal{G}_j$ only passes through leaves in $\mathcal{P}^\circ$. Note that $\mathcal{P}$ remains in the subtree T of π rooted at $c(\mathcal{G})$: ordinary upward steps remain in T and, by Lemma 6.7, each companion of a bud through which $\mathcal{P}$ passes lies in T as well. Moreover, every LNS occurring in $\mathcal{P}$ must lie on the upward path from $c(\mathcal{G})$ to $\mathcal{F}$ for some $\mathcal{F} \in \mathcal{P}^\circ$: otherwise either $\mathcal{P}$ ends in an axiom contradicting our assumption that $\mathcal{P}^\circ$ is infinite, or $\mathcal{P}$ passes through a bud outside of $\mathcal{P}^\circ$, contradicting our construction of $\mathcal{P}$. By Lemma 6.8, every LNS in $\mathcal{P}$ has a formula in focus and so $\mathcal{P}$ has a continuous focus trace. $\square$

Definition 6.10. Let $C = (\pi, L, c)$ be a cyclic proof and $\mathcal{P}^\circ = (v_i)_{i \in \mathbb{N}}$ an infinite looping path. A progress point of $\mathcal{P}^\circ$ is an index i such that v_{i+1} is the right premise of an instance of UL_1 or UL_2 , and the principal formula in $\text{src}_C(v_i)$ is in focus.

Lemma 6.11. Let (π, L, c) be a cyclic proof and $\mathcal{P}^\circ$ an infinite looping path. There are infinitely many progress points in $\mathcal{P}^\circ$.

Proof. Mark every edge in the finite tree π that passes from the conclusion of a rule instance of UL_1 or UL_2 with focused principal formula to the right premise. For each node v of π , define

$$\mu(v) := \text{the number of marked edges on the upward path from the root to } v.$$

Note that $\mu(v) \in \mathbb{N}$ for every v of π . For every bud $b \in L$, the companion $c(b)$ is a proper ancestor of b and the upward path from $c(b)$ to b passes through at least one progress point. Therefore (1) $\mu(c(b)) \leq \mu(b) - 1$. Suppose, toward a contradiction, that there exists an infinite looping path $\mathcal{P}^\circ = \{v_i\}_{i \in \mathbb{N}}$ which contains only finitely many progress points. Let $N \in \mathbb{N}$ such that no index $i \geq N$ is a progress point. Then for every $i \geq N$ the edge $\text{src}_C(v_i) \rightarrow v_{i+1}$ is unmarked. Therefore (2) $\mu(\text{src}_C(v_i)) = \mu(v_{i+1})$. If $v_i \notin L$, then $\text{src}_C(v_i) = v_i$ and therefore (2) gives $\mu(v_i) = \mu(v_{i+1})$. If $v_i \in L$, then $\text{src}_C(v_i) = c(v_i)$ and so (1) and (2) together give

$$\mu(v_{i+1}) = \mu(c(v_i)) \leq \mu(v_i) - 1.$$

Note that $\mathcal{P}^\circ$ passes infinitely often through a bud. Hence $\{\mu(v_i)\}_{i \geq N}$ is a decreasing sequence of natural numbers that strictly decreases infinitely often. This contradicts the well-foundedness of the natural numbers. Hence $\mathcal{P}^\circ$ contains infinitely many progress points. $\square$

Now we turn toward proving the Shifting Lemma (see Lemma 6.13 below). We begin by introducing a notion of isomorphism between cyclic proofs. Let π and π' be derivations in LNS_{LTL} . A *homomorphism* from π to π' is a function h mapping occurrences of LNSs in π to occurrences of LNSs in π' such that whenever a rule r occurs in π with premises $\mathcal{G}_1, \dots, \mathcal{G}_n$ and conclusion $\mathcal{G}$, the same rule r occurs in π' with premises $h(\mathcal{G}_1), \dots, h(\mathcal{G}_n)$ and conclusion $h(\mathcal{G})$.⁴ We say that π and π' are *isomorphic* if there exists a homomorphism $h : \pi \rightarrow \pi'$ whose inverse h^{-1} exists and is also a homomorphism, and call h an *isomorphism*. Isomorphisms preserve the structure of derivations, but not necessarily their *content*: isomorphic derivations may be derivations of different LNSs. A *strict isomorphism* is an isomorphism $h : \pi \rightarrow \pi'$ which satisfies the following two properties for any $\mathcal{G}$ in π : (1) $\mathcal{G}$ is saturated iff $h(\mathcal{G})$ is saturated and (2) $\text{end}(\mathcal{G}) = \text{end}(h(\mathcal{G}))$; in such a case, we call π and π' *strictly isomorphic*. We call cyclic proofs (π, L, c) and (π', L', c') *strictly isomorphic* if there exists a strict isomorphism $h : \pi \rightarrow \pi'$ such that (1) $\mathcal{F} \in L$ iff $h(\mathcal{F}) \in L'$ and (2) for any $\mathcal{F} \in L$, $h(c(\mathcal{F})) = c'(h(\mathcal{F}))$.

⁴Note that homomorphisms map *occurrences* of linear nested sequents, so two distinct occurrences of the same LNS $\mathcal{G}$ may map to different LNSs $\mathcal{H}$ and $\mathcal{F}$.

CASE FOR $\rightarrow R$. Suppose $\rightarrow R$ is applied bottom-up (shown below left) to the i -component of $\mathcal{F}$ in π with d the depth of $\mathcal{F}$ in π . By IH, we have constructed the output derivation and map h up to depth d and know that the i -component of $\mathcal{F}$ is equal to the j -component of $h(\mathcal{F})$. We extend the output derivation by bottom-up applying $\rightarrow R$ to the j -component of $h(\mathcal{F})$, as shown below right.

$$\frac{\mathcal{F}\{\Sigma, A \vdash B, \Pi\}_i}{\mathcal{F}\{\Sigma \vdash A \rightarrow B, \Pi\}_i} \rightarrow R \quad \frac{h(\mathcal{F})\{\Sigma, A \vdash B, \Pi\}_j}{h(\mathcal{F})\{\Sigma \vdash A \rightarrow B, \Pi\}_j} \rightarrow R$$

We set $h(\mathcal{F}\{\Sigma, A \vdash B, \Pi\}_i) := h(\mathcal{F})\{\Sigma, A \vdash B, \Pi\}_j$. Note that h is an isomorphism between the input and output derivations up to depth $d + 1$, and so (a) is satisfied. Moreover, for all $m \neq i$ with $m \geq n$, the m -component has been unaffected by the $\rightarrow R$ application, so the m -component of $\mathcal{F}\{\Sigma, A \vdash B, \Pi\}_i$ and the $k + (m - n)$ -component of $h(\mathcal{F})\{\Sigma, A \vdash B, \Pi\}_j$ are equal. Also, observe that the i -component of $\mathcal{F}\{\Sigma, A \vdash B, \Pi\}_i$ is equal to the j -component of $h(\mathcal{F})\{\Sigma, A \vdash B, \Pi\}_j$. This implies that (b) and (c) hold as well and so that (I) holds.

CASE FOR UL_1 . Suppose UL_1 is applied bottom-up to the i -component of $\mathcal{F}$ in π with d the depth of $\mathcal{F}$ in π , where $a \in \{u, f\}$.

$$\frac{\mathcal{F}\{\Gamma, B \vdash \Delta // \Sigma \vdash \Pi\}_i \quad \mathcal{F}\{\Gamma, A \vdash \Delta // \Sigma, AUB^a \vdash \Pi\}_i}{\mathcal{F}\{\Gamma, AUB^a \vdash \Delta // \Sigma \vdash \Pi\}_i}$$

By IH we have constructed the output derivation and map h up to depth d and know that the i - and $i + 1$ -components of $\mathcal{F}$ are equal to the j - and $j + 1$ -components of $h(\mathcal{F})$, respectively. We extend the output derivation by bottom-up applying UL_1 to the j -component of $h(\mathcal{F})$, as shown below.

$$\frac{h(\mathcal{F})\{\Gamma, B \vdash \Delta // \Sigma \vdash \Pi\}_j \quad h(\mathcal{F})\{\Gamma, A \vdash \Delta // \Sigma, AUB^a \vdash \Pi\}_j}{h(\mathcal{F})\{\Gamma, AUB^a \vdash \Delta // \Sigma \vdash \Pi\}_j}$$

Note that if $a = f$, then both output premises are well-formed, since the corresponding input premises are well-formed and the construction preserves their annotations while inserting only unfocused components. We set $h(\mathcal{F}\{\Gamma, B \vdash \Delta // \Sigma \vdash \Pi\}_i) := h(\mathcal{F})\{\Gamma, B \vdash \Delta // \Sigma \vdash \Pi\}_j$ and $h(\mathcal{F}\{\Gamma, A \vdash \Delta // \Sigma, AUB^a \vdash \Pi\}_i) := h(\mathcal{F})\{\Gamma, A \vdash \Delta // \Sigma, AUB^a \vdash \Pi\}_j$. Note that h is an isomorphism between the input and output derivations up to depth $d + 1$, and so (a) is satisfied. Moreover, by an argument similar to the $\rightarrow R$ -case, one can verify that conditions (b) and (c) are satisfied as well and so that (I) holds. This concludes the construction of the output derivation π' and of h .

We now show how to define a cyclic proof of $\mathcal{G}'$ based on π' that is strictly isomorphic to (π, L, c) . First, observe that (I) implies that h is a strict isomorphism between π and π' . Let $(c(\mathcal{F}), \mathcal{F})$ form a cycle in (π, L, c) . By the definition of h , we know that a path occurs from $h(c(\mathcal{F}))$ to $h(\mathcal{F})$ in π' with $h(\mathcal{F})$ a leaf of π' and that since $\|c(\mathcal{F})\| < \|\mathcal{F}\|$, the path from $c(\mathcal{F})$ to $\mathcal{F}$ passes through an expansion rule⁶ and so the path from $h(c(\mathcal{F}))$ to $h(\mathcal{F})$ passes through an expansion rule as well, implying that $\|h(c(\mathcal{F}))\| < \|h(\mathcal{F})\|$. Moreover, by the invariant (I), we know that $h(\mathcal{F})$ and $h(c(\mathcal{F}))$ are saturated since $\mathcal{F}$ and $c(\mathcal{F})$ are saturated, $\text{end}(\mathcal{F}) = \text{end}(h(\mathcal{F}))$, and $\text{end}(c(\mathcal{F})) = \text{end}(h(c(\mathcal{F})))$. Therefore, since $\text{end}(c(\mathcal{F})) = \text{end}(\mathcal{F})$, we have $\text{end}(h(c(\mathcal{F}))) = \text{end}(h(\mathcal{F}))$. Since $(c(\mathcal{F}), \mathcal{F})$ is a cycle in a cyclic proof, the path from $c(\mathcal{F})$ to $\mathcal{F}$ has a continuous focus trace and passes through the right premise of a rule $r \in \{UL_1, UL_2\}$ with the formula in focus principal. By construction of π' , h and by (I), the path from $h(c(\mathcal{F}))$ to $h(\mathcal{F})$ therefore has a continuous focus trace and passes through the right premise of a rule $r \in \{UL_1, UL_2\}$ with the formula in focus principal. Let $L' := \{h(\mathcal{F}) \mid \mathcal{F} \in L\}$ and let c' map each $h(\mathcal{F}) \in L'$ to $h(c(\mathcal{F}))$. For any other leaf $h(\mathcal{H})$ of π' , $\mathcal{H}$ is a leaf of π which does not belong to L and

⁶If the expansion rule is UL_1 or UL_2 , then the path passes through the right premise. We will not mention this again.

therefore is an initial sequent. By (I), $h(\mathcal{H})$ is an initial sequent as well; therefore, (π', L', c') is a cyclic proof of $\mathcal{G}'$ that is strictly isomorphic to (π, L, c) . $\square$

Next, we use Lemma 6.13 to obtain an infinite sequence of ‘growing’ cyclic proofs and define the limit of this sequence. Let (π_0, L_0, c_0) be a cyclic proof of $\mathcal{G}$ with $L_0 \neq \emptyset$. Let $\mathcal{H}_0 = c_0(\mathcal{H}_1)$ be a companion in π_0 with *least depth*, i.e., a companion node such that no other companion occurs in π with depth strictly smaller (where depth means the distance of $\mathcal{H}$ from the root). Note that a companion of least depth has no companion as a proper ancestor, so its rooted subtree is a self-contained cyclic proof. Thus let (π, L, c) denote the cyclic proof rooted at $\mathcal{H}_0$ in (π_0, L_0, c_0) .

By Lemma 6.13, we can transform (π, L, c) into a strictly isomorphic cyclic proof (π', L', c') of $\mathcal{H}_1$ such that $\mathcal{H}_1 = c'(\mathcal{H}_2)$ for some leaf $\mathcal{H}_2$ in L' . We construct a new cyclic proof (π_1, L_1, c_1) of $\mathcal{G}$ as follows: first, we ‘paste’ π' above $\mathcal{H}_1$ in π_0 to obtain a new derivation π_1 . Second, we set $L_1 := (L_0 \setminus \{\mathcal{H}_1\}) \cup L'$. Third, we define c_1 as follows:

$$c_1(\mathcal{F}) = \begin{cases} c_0(\mathcal{F}) & \text{if } \mathcal{F} \in L_0 \setminus \{\mathcal{H}_1\}, \\ c'(\mathcal{F}) & \text{if } \mathcal{F} \in L'. \end{cases}$$

We first argue that (π_1, L_1, c_1) is a cyclic proof of $\mathcal{G}$. Indeed, π_1 is a finite derivation of $\mathcal{G}$ since π_0 is a finite derivation of $\mathcal{G}$ and π' is a finite derivation of $\mathcal{H}_1$. For condition (3) of Definition 3.7, consider first a bud $\mathcal{F} \in L_0 \setminus \{\mathcal{H}_1\}$. Since the path from the root to $\mathcal{F}$ in π_1 is identical to the corresponding path in π_0 , the cycle $(c_0(\mathcal{F}), \mathcal{F})$ satisfies all required conditions in π_1 . Next, consider a bud $\mathcal{F} \in L'$. Since (π', L', c') is a cyclic proof, the path from $c'(\mathcal{F})$ to $\mathcal{F}$ in π' is preserved in π_1 and satisfies all required conditions. Finally, every leaf of π_1 not in L_1 is either a leaf of π_0 not in L_0 or a leaf of π' not in L' , and is therefore an initial sequent in either case.

Note that the cycle from $\mathcal{H}_1$ to $\mathcal{H}_0$ in (π_0, L_0, c_0) has been shifted upward and replaced by a cycle from $\mathcal{H}_2$ to $\mathcal{H}_1$ in (π_1, L_1, c_1) . In general, $\mathcal{H}_0$ may be the companion of several buds, so after repeating the above process for each bud, $\mathcal{H}_0$ is no longer a companion in the generated cyclic proof. By successively repeating the above process for companions of least depth, we obtain an infinite sequence $(\pi_0, L_0, c_0), (\pi_1, L_1, c_1), \dots$ of cyclic proofs.

By construction, each π_i is a tree of LNSs with root $\mathcal{G}$, i.e., $\pi_i = (T_i, \leq_i, \mathcal{G})$ where T_i is a set of nodes (labeled by LNSs) and $(T_i, \leq_i)$ is a poset, such that for each $t \in T_i$, the down-set of t in T_i is well-ordered. Observe that for all $i \in \mathbb{N}$, $T_i \subseteq T_{i+1}$ and $\leq_i \subseteq \leq_{i+1}$. We define the *limit derivation* of the sequence of cyclic proofs to be $\pi_\omega := (T_\omega, \leq_\omega, \mathcal{G})$ where $T_\omega := \bigcup_{i \in \mathbb{N}} T_i$ and $\leq_\omega := \bigcup_{i \in \mathbb{N}} \leq_i$. Note that π_ω is a well-defined derivation of $\mathcal{G}$ in $\text{LNS}_{\text{LTL}}^\infty$.

Note that in each step of the unfolding construction, one bud $\mathcal{G}$ is unfolded, and all other existing buds whose companion is $c(\mathcal{G})$ remain unchanged. If the selected companion has depth d , every newly introduced companion lies in the subtree rooted at $\mathcal{G}$ and therefore has depth strictly greater than d . Together with the least-depth scheduling rule, this yields the following result.

Lemma 6.14. *Every leaf of π_ω is an initial LNS.*

Proof. Let u be a leaf of π_ω , labelled by $\mathcal{H}$, and choose $i \in \mathbb{N}$ such that u is a leaf of π_i . Then either $\mathcal{H}$ is initial or $u \in L_i$. Suppose $u \in L_i$. Since u remains a leaf of π_ω , it is never unfolded. Consequently, the definition of the construction implies

$$u \in L_j \quad \text{and} \quad c_j(u) = c_i(u) \quad \text{for every } j \geq i.$$

Let D be the depth of $c_i(u)$, and let d_j be the least depth of a companion in (π_j, L_j, c_j) . Since $c_i(u)$ remains a companion, $d_j \leq D$ for every $j \geq i$. At each step, existing nodes retain their depths, and every

newly introduced companion has depth strictly greater than that of the selected companion. Since the selected companion has least depth, it follows that $d_{j+1} \geq d_j$. Thus $\{d_j\}_{j \geq i}$ is a bounded non-decreasing sequence of natural numbers. Therefore there exists $J \geq i$ and $d \in \mathbb{N}$ such that $d_j = d$ for every $j \geq J$. There are only finitely many buds in (π_J, L_J, c_J) whose companions have depth d . From stage J onwards, each unfolding step removes exactly one such bud and introduces no new bud whose companion has depth d . Hence, after finitely many steps, no companion at depth d remains, contradicting $d_j = d$ for every $j \geq J$. Therefore $u \notin L_i$, and so $\mathcal{H}$ is initial. $\square$

In order to show that every infinite branch $\mathcal{B}$ of π_ω is progressing, we show that every such $\mathcal{B}$ corresponds to a looping path of (π_0, L_0, c_0) in the following sense.

Definition 6.15. Given $j, n \in \mathbb{N}$ and finite looping paths $\mathcal{P}_j^\circ = \{\mathcal{G}_i\}_{i \leq k}$ of $C_j = (\pi_j, L_j, c_j)$ and $\mathcal{P}_n^\circ = \{\mathcal{H}_i\}_{i \leq k}$ of $C_n = (\pi_n, L_n, c_n)$ of equal length, the looping path $\mathcal{P}_n^\circ$ corresponds to $\mathcal{P}_j^\circ$ iff for all $i \leq k$

- (1) $\text{src}_{C_j}(\mathcal{G}_i)$ is the conclusion of an instance of $r \in \text{LNS}_{\text{LTL}}^\infty$ iff $\text{src}_{C_n}(\mathcal{H}_i)$ is the conclusion of an instance of r .
- (2) for $i < k$, $\text{src}_{C_j}(\mathcal{G}_i)$ is the conclusion of a binary rule r and $\mathcal{G}_{i+1}$ is the right (resp. left) premise if and only if $\text{src}_{C_n}(\mathcal{H}_i)$ is the conclusion of r and $\mathcal{H}_{i+1}$ is the right (resp. left) premise;
- (3) $\mathcal{G}_i$ contains a formula in focus iff $\mathcal{H}_i$ contains a formula in focus, and if $\text{src}_{C_j}(\mathcal{G}_i)$ is the conclusion of a logical rule r whose principal formula is in focus, then the principal formula of the rule applied to $\text{src}_{C_n}(\mathcal{H}_i)$ is in focus (and vice versa);
- (4) $\mathcal{G}_i$ is saturated iff $\mathcal{H}_i$ is saturated, and if so, $\text{end}(\mathcal{G}_i) = \text{end}(\mathcal{H}_i)$.

For π_ω define $\text{src}_C(v) := v$ for all nodes $v \in \pi_\omega$. An infinite branch $\mathcal{B} = \{\mathcal{H}_i\}_{i \in \mathbb{N}}$ of π_ω then corresponds to an infinite looping path $\mathcal{P}^\circ = \{\mathcal{G}_i\}_{i \in \mathbb{N}}$ of (π_0, L_0, c_0) iff for all $i \in \mathbb{N}$ conditions (1)–(4) are satisfied.

Lemma 6.16. For any $j \in \mathbb{N}$ and every finite looping path $\mathcal{P}_{j+1}^\circ$ of $(\pi_{j+1}, L_{j+1}, c_{j+1})$, there exists a looping path $\mathcal{P}_j^\circ$ of (π_j, L_j, c_j) of equal length, such that $\mathcal{P}_{j+1}^\circ$ corresponds to $\mathcal{P}_j^\circ$.

Proof. Write $C_i := (\pi_i, L_i, c_i)$. Suppose that C_{j+1} is obtained from C_j by unfolding the bud $\mathcal{H} \in L_j$. Let $C = (\pi, L, c)$ be the cyclic subproof rooted at $c_j(\mathcal{H})$, and let $C' = (\pi', L', c')$ be the cyclic proof pasted above $\mathcal{H}$. Let $h : \pi \rightarrow \pi'$ be the strict isomorphism constructed in the proof of Lemma 6.13. In particular,

$$h(c(\mathcal{H})) = \mathcal{H}, \quad L' = h[L], \quad c'(h(\mathcal{F})) = h(c(\mathcal{F})) \quad \text{for every } \mathcal{F} \in L. \quad (*)$$

Recall that besides being a strict isomorphism, the constructed h in the proof of Lemma 6.13 preserves whether a node contains a formula in focus and whether the principal formula of a logical rule instance is in focus. Indeed, the fixed prefixes in the input and output derivations are unfocused, while invariant (I)(c) and the construction preserve the annotations in all remaining components and copy each inference with the same annotated principal formula.

Let $\mathcal{P}_{j+1}^\circ = (\mathcal{G}_i)_{i \leq k}$ be a finite looping path of C_{j+1} . Recall that looping paths start at the root. Therefore the only way for $\mathcal{P}_{j+1}^\circ$ to reach a node in C' is to pass through $\mathcal{H}$. Moreover, after passing through $\mathcal{H}$, the path remains in C' : ordinary steps remain in that subtree, and all companions of its buds belong to that subtree, since $c(\mathcal{H})$ has least depth.

If $\mathcal{H}$ does not occur in $\mathcal{P}_{j+1}^\circ$, set $\mathcal{P}_j^\circ := \mathcal{P}_{j+1}^\circ$. Since $\mathcal{P}_{j+1}^\circ$ does not enter C' and C_j and C_{j+1} are otherwise identical, $\mathcal{P}_j^\circ$ is a looping path of C_j which corresponds to $\mathcal{P}_{j+1}^\circ$.

If $\mathcal{H}$ occurs in $\mathcal{P}_{j+1}^\circ$, let $t \leq k$ be the least natural number such that $\mathcal{G}_t = \mathcal{H}$. Define

$$\mathcal{P}_j^\circ = (\mathcal{F}_i)_{i \leq k}, \quad \mathcal{F}_i := \begin{cases} \mathcal{G}_i & i \leq t, \\ h^{-1}(\mathcal{G}_i) & t < i \leq k. \end{cases}$$

We first verify that $\mathcal{P}_j^\circ$ is a looping path of C_j . The initial segment $(\mathcal{F}_i)_{i \leq t}$ only passes through nodes in π_{j+1} outside of π' which also belong to π_j . Hence $(\mathcal{F}_i)_{i \leq t}$ is an initial segment of a looping path of C_j . If $t < k$, then $\mathcal{G}_{t+1}$ is a child of the root $\mathcal{H}$ of π' . Since $h(c_j(\mathcal{H})) = \mathcal{H}$, $\mathcal{F}_{t+1} = h^{-1}(\mathcal{G}_{t+1})$ is a child of $c_j(\mathcal{H})$. Thus the step from $\mathcal{F}_t = \mathcal{H}$ to $\mathcal{F}_{t+1}$ is permitted by Definition 6.5(3). For every $t < i \leq k$, we have

$$\text{src}_{C_j}(\mathcal{F}_i) = h^{-1}(\text{src}_{C_{j+1}}(\mathcal{G}_i)). \quad (**)$$

For a non-bud this follows directly from the definition of src ; for a bud $\mathcal{F}_i$ we have $c'(h(\mathcal{F}_i)) = h(c(\mathcal{F}_i))$ by (*). Hence by applying h^{-1} to both sides and taking $\mathcal{F}_i = h^{-1}(\mathcal{G}_i)$, we obtain

$$h^{-1}(c'(\mathcal{G}_i)) = c(h^{-1}(\mathcal{G}_i)).$$

Finally, since $c'(\mathcal{G}_i) = \text{src}_{C_{j+1}}(\mathcal{G}_i)$, $h^{-1}(\mathcal{G}_i) = \mathcal{F}_i \in L_i$ and $c(h^{-1}(\mathcal{G}_i)) = \text{src}_{C_j}(\mathcal{F}_i)$, (**) follows.

Consequently, if $t < i < k$, then $\mathcal{F}_{i+1}$ is a child of $\text{src}_{C_j}(\mathcal{F}_i)$, because $\mathcal{G}_{i+1}$ is a child of $\text{src}_{C_{j+1}}(\mathcal{G}_i)$ and h^{-1} is a homomorphism. This handles both ordinary steps and bud jumps. Moreover, h preserves buds and initial leaves. Thus $\mathcal{P}_j^\circ$ is a looping path of C_j of the required length. It remains to check correspondence. We distinguish three cases.

CASE $i < t$. By definition $\mathcal{F}_i = \mathcal{G}_i$ and $\text{src}_{C_j}(\mathcal{F}_i) = \text{src}_{C_{j+1}}(\mathcal{G}_i)$. When $i < k$, the selected premise is unchanged as well. Therefore conditions (1)–(4) hold.

CASE $i = t$. Then $\mathcal{F}_i = \mathcal{G}_i = \mathcal{H}$, but their sources are

$$\text{src}_{C_{j+1}}(\mathcal{H}) = \mathcal{H}, \quad \text{src}_{C_j}(\mathcal{H}) = c_j(\mathcal{H}).$$

By construction of π' in the Shifting Lemma, the same rule is applied to $\mathcal{H}$ in π' and to $c_j(\mathcal{H})$ in π . This implies that condition (1) is satisfied. If $t < k$, $\mathcal{H}$ is the conclusion of a binary rule in π' and $\mathcal{G}_{t+1}$ is the left (right, respectively) premise, then $h^{-1}(\mathcal{G}_t)$ is the conclusion of the same binary rule by (1) and $h^{-1}(\mathcal{G}_{t+1})$ is the left (right, respectively) premise by construction of h . This implies that condition (2) holds. If $t = k$, condition (2) is vacuous. For (3), $\mathcal{F}_t$ contains a formula in focus iff $\mathcal{G}_t$ contains a formula in focus since $\mathcal{F}_t = \mathcal{G}_t = \mathcal{H}$. Moreover, if $\mathcal{H}$ is the conclusion of a logical rule r with the principal formula in focus in π' , then $c_j(\mathcal{H})$ is the conclusion of r with the principal formula in focus in π , due to the construction of h , and vice versa. Therefore, (3) holds. Finally, (4) holds trivially since $\mathcal{F}_t = \mathcal{G}_t = \mathcal{H}$.

CASE $t < i \leq k$. Then $\mathcal{F}_i = h^{-1}(\mathcal{G}_i)$ and $\text{src}_{C_j}(\mathcal{F}_i) = h^{-1}(\text{src}_{C_{j+1}}(\mathcal{G}_i))$ by (**). Thus $\text{src}_{C_j}(\mathcal{F}_i)$ is the conclusion of an instance of r iff $\text{src}_{C_{j+1}}(\mathcal{G}_i)$ is the conclusion of an instance of r since h^{-1} is a homomorphism. Thus, condition (1) holds. Conditions (2) and (3) follow by similar arguments. Finally, since h is a strict isomorphism,

$$\mathcal{G}_i \text{ is saturated} \iff \mathcal{F}_i \text{ is saturated}, \quad \text{end}(\mathcal{G}_i) = \text{end}(\mathcal{F}_i),$$

so condition (4) holds. Therefore, $\mathcal{P}_{j+1}^\circ$ corresponds to $\mathcal{P}_j^\circ$. $\square$

Lemma 6.17. *For any $j \in \mathbb{N}$ and every finite looping path $\mathcal{P}_j^\circ$ of (π_j, L_j, c_j) , there exists a finite looping path $\mathcal{P}_0^\circ$ of (π_0, L_0, c_0) of equal length, such that $\mathcal{P}_j^\circ$ corresponds to $\mathcal{P}_0^\circ$.*

Proof. By induction on j . The base case for $j = 0$ is trivial. Thus, suppose $j = n + 1$ and consider a finite looping path $\mathcal{P}_{n+1}^\circ$ of $(\pi_{n+1}, L_{n+1}, c_{n+1})$. By Lemma 6.16, there exists a finite looping path $\mathcal{P}_n^\circ$ of (π_n, L_n, c_n) of equal length, such that $\mathcal{P}_{n+1}^\circ$ and $\mathcal{P}_n^\circ$ are corresponding. By the induction hypothesis, there exists a finite looping path $\mathcal{P}_0^\circ$ of (π_0, L_0, c_0) of equal length, such that $\mathcal{P}_n^\circ$ and $\mathcal{P}_0^\circ$ are corresponding. Since the notion of correspondence between looping paths is transitive (each condition is a biconditional that composes), $\mathcal{P}_{n+1}^\circ$ and $\mathcal{P}_0^\circ$ are corresponding as well. $\square$

We are now ready to prove the main theorem of this section.

Theorem 6.18. *Every $\text{LNS}_{\text{LTL}}^{\text{cyc}}$ -proof can be transformed into an $\text{LNS}_{\text{LTL}}^\infty$ -proof satisfying the SRP.*

Proof. Let (π_0, L_0, c_0) be a cyclic proof of $\mathcal{G}$. If $L_0 = \emptyset$, then π_0 is a finite $\text{LNS}_{\text{LTL}}^\infty$ -proof of $\mathcal{G}$ which vacuously satisfies the SRP. Thus, suppose that $L_0 \neq \emptyset$ and let π_ω be the infinite $\text{LNS}_{\text{LTL}}^\infty$ -derivation obtained by successively applying the Shifting Lemma to least-depth companions as described above. We will now argue that π_ω is a $\text{LNS}_{\text{LTL}}^\infty$ -proof. By Lemma 6.14 every leaf of π_ω is an initial sequent. Thus it remains to show that every infinite branch is progressing.

Suppose $\mathcal{B} = \{\mathcal{G}_m\}_{m \in \mathbb{N}}$ is an infinite branch of π_ω . Note that by construction of π_ω , for every $j \in \mathbb{N}$ there exists a natural number m_j such that $\mathcal{G}_{m_j}$ is a bud of $C_j = (\pi_j, L_j, c_j)$ and the initial segment $\mathcal{B}_j = \{\mathcal{G}_m\}_{m \leq m_j}$ of $\mathcal{B}$ is a branch of π_j . We now define an infinite looping path $\mathcal{P}^\circ = \{\mathcal{H}_i\}_{i \in \mathbb{N}}$ of $C_0 = (\pi_0, L_0, c_0)$ such that $\mathcal{P}^\circ$ corresponds to $\mathcal{B}$. To do so, we construct an infinite sequence of finite looping paths $\mathcal{Q}_i^\circ$ of C_0 . We proceed by induction on i and show that invariant (II) holds, which is a conjunction of two statements:

- (i) $\mathcal{Q}_i^\circ$ contains $\mathcal{Q}_{i-1}^\circ$ as an initial segment (for $i > 0$) and
- (ii) $\mathcal{Q}_i^\circ$ corresponds to $\mathcal{B}_i$.

Let $\mathcal{Q}_0^\circ := \mathcal{B}_0$. By our previous observation $\mathcal{Q}_0^\circ$ is a branch of π_0 ending in a bud. Since $i = 0$, (i) holds vacuously, and (ii) holds since $\mathcal{Q}_0^\circ = \mathcal{B}_0$. Therefore, invariant (II) holds. Now suppose we have constructed $\mathcal{Q}_j^\circ$ and (II) holds. Since $\mathcal{B}_{j+1}$ is a branch of π_{j+1} ending in a bud, it is also a looping path of C_{j+1} and therefore corresponds to a looping path $\mathcal{Q}^\circ$ of C_0 by Lemma 6.17. Set $\mathcal{Q}_{j+1}^\circ := \mathcal{Q}^\circ$. Note that (ii) holds. For (i) we show that $\mathcal{Q}_j^\circ = \{\mathcal{H}_m\}_{m \leq m_j}$ is an initial segment of $\mathcal{Q}_{j+1}^\circ = \{\mathcal{F}_m\}_{m \leq m_{j+1}}$. First note that both $\mathcal{Q}_j^\circ$ and $\mathcal{Q}_{j+1}^\circ$ start at the root of π_0 . Therefore $\mathcal{H}_0 = \mathcal{F}_0$. Now suppose inductively that $\mathcal{H}_m = \mathcal{F}_m$ for $m < m_j$. Therefore $\text{src}_{C_0}(\mathcal{H}_m) = \text{src}_{C_0}(\mathcal{F}_m)$. Recall that $\mathcal{Q}_j^\circ$ and $\mathcal{Q}_{j+1}^\circ$ correspond to $\mathcal{B}_j = \{\mathcal{G}_m\}_{m \leq m_j}$ and $\mathcal{B}_{j+1} = \{\mathcal{G}'_m\}_{m \leq m_{j+1}}$, respectively. Thus at $\mathcal{H}_m$ the looping path $\mathcal{Q}_j^\circ$ chooses the same premise as the transition in $\mathcal{B}_j$ from $\mathcal{G}_m$ to $\mathcal{G}_{m+1}$, and similarly for $\mathcal{Q}_{j+1}^\circ$ and $\mathcal{B}_{j+1}$. Since $m < m_j$ and $\mathcal{B}_j$ is an initial segment of $\mathcal{B}_{j+1}$, $\mathcal{G}_{m+1} = \mathcal{G}'_{m+1}$ and therefore $\mathcal{H}_{m+1} = \mathcal{F}_{m+1}$. Thus $\mathcal{H}_m = \mathcal{F}_m$ for all $m \leq m_j$, which implies that $\mathcal{Q}_j^\circ$ is an initial segment of $\mathcal{Q}_{j+1}^\circ$ and so (i) holds. We therefore conclude that for any $i \in \mathbb{N}$, invariant (II) holds. Finally, we let

$$\mathcal{P}^\circ := \bigcup_{i \in \mathbb{N}} \mathcal{Q}_i^\circ.^7$$

Note that $\mathcal{P}^\circ$ is a well-defined infinite looping path of (π_0, L_0, c_0) by (i) and corresponds to $\mathcal{B}$ by (ii). We now argue that $\mathcal{P}^\circ$ is progressing.

Write $\mathcal{P}^\circ = \{\mathcal{H}_m\}_{m \in \mathbb{N}}$. By Lemma 6.9 there exists $N \in \mathbb{N}$ such that the suffix $\{\mathcal{H}_m\}_{m \geq N}$ of $\mathcal{P}^\circ$ has a continuous focus trace. By Lemma 6.11, $\mathcal{P}^\circ$ has infinitely many progress points. Thus the suffix $\{\mathcal{H}_m\}_{m \geq N}$ of $\mathcal{P}^\circ$ has infinitely many progress points as well. Therefore $\{\mathcal{H}_m\}_{m \geq N}$ is a progressing suffix, which implies that $\mathcal{P}^\circ$ is progressing.

⁷We view a sequence $\{\mathcal{G}_i\}_{i \in I}$ of LNSs as a function which maps natural numbers in I to LNSs.

Since $\mathcal{P}^\circ$ corresponds to $\mathcal{B}$, we can transfer the progressing property from $\mathcal{P}^\circ$ to $\mathcal{B}$ using Definition 6.15. Let $\mathcal{P}_{\geq j}^\circ = \{\mathcal{H}_i\}_{i \geq j}$ be the progressing suffix of $\mathcal{P}^\circ$ and let $\mathcal{B}_{\geq j} = \{\mathcal{G}_i\}_{i \geq j}$ be the corresponding suffix of $\mathcal{B}$. By condition (3), $\mathcal{G}_i$ contains a formula in focus if and only if $\mathcal{H}_i$ contains a formula in focus for each $i \geq j$. Since $\mathcal{P}_{\geq j}^\circ$ has a continuous focus trace, $\mathcal{B}_{\geq j}$ has a continuous focus trace as well. By conditions (1) and (2), whenever $\text{src}_{C_0}(\mathcal{H}_i)$ is the conclusion of a rule $r \in \{\text{UL}_1, \text{UL}_2\}$ and $\mathcal{H}_{i+1}$ is the right premise, we have that $\mathcal{G}_i$ is the conclusion of the same rule r and $\mathcal{G}_{i+1}$ is the right premise. Moreover, by condition (3), if the principal formula of r applied to $\text{src}_{C_0}(\mathcal{H}_i)$ is in focus, then the principal formula of r applied to $\mathcal{G}_i$ is in focus. Since $\mathcal{P}^\circ$ passes infinitely often through the right premise of $r \in \{\text{UL}_1, \text{UL}_2\}$ with the formula in focus principal, the same holds for $\mathcal{B}$. We thus conclude that $\mathcal{B}$ is progressing.

Finally, we argue that π_ω satisfies the SRP. By condition (4), whenever $\mathcal{H}_i$ is saturated, $\mathcal{G}_i$ is saturated as well. Since $\mathcal{P}^\circ$ passes infinitely often through the buds $\mathcal{H} \in \mathcal{P}_\infty^\circ$ which are saturated by Definition 3.7–(3), $\mathcal{P}^\circ$ passes through infinitely many saturated LNSs. Therefore, $\mathcal{B}$ passes through infinitely many saturated LNSs as well. Since this holds for every infinite branch $\mathcal{B}$ of π_ω , π_ω satisfies the SRP. We conclude that π_ω is an $\text{LNS}_{\text{LTL}}^\infty$ -proof of $\mathcal{G}$ which satisfies the SRP. $\square$

We obtain the soundness of the cyclic LNS system as a corollary of Theorems 4.1 and 6.18.

Corollary 6.19 (Soundness of $\text{LNS}_{\text{LTL}}^{\text{cyc}}$). *If $\mathcal{G}$ has an $\text{LNS}_{\text{LTL}}^{\text{cyc}}$ -proof, then $\mathcal{G}$ is valid.*

7 Concluding Remarks

There are several directions for future research. First, it would be natural to investigate whether—and under what conditions—the proof transformations between non-wellfounded and cyclic proofs developed here can be extended to more expressive multisequent formalisms, such as nested and labeled sequents. Such generalizations would deepen our understanding of cycle recognition and unraveling beyond the linear nested setting. Second, while non-wellfounded Gentzen-style sequent calculi for LTL were given in [5, 10, 15], the problem of syntactic cut-elimination was left open in all three works. We conjecture that the additional structural expressiveness of $\text{LNS}_{\text{LTL}}^\infty$ enables a syntactic cut-elimination proof, which we aim to investigate. Third, it would be interesting to study proof transformations between our linear nested sequent calculi and the existing Gentzen and tableau systems of [5, 10, 15, 30]. Such translations would clarify the relationship between these formalisms and allow results to be transferred between them—in particular, should cut-elimination be established for $\text{LNS}_{\text{LTL}}^\infty$, such translations could provide a pathway to resolving the analogous open problem for the Gentzen systems of [5, 10, 15]. Finally, while our contribution is primarily theoretical, the invertible and formula-driven nature of our cyclic system suggests applications to automated proof-search and proof certificates for LTL.

References

- [1] Bahareh Afshari, Lide Grotenhuis, Graham E. Leigh & Lukas Zenger (2023): *Ill-Founded Proof Systems for Intuitionistic Linear-Time Temporal Logic*. In Revantha Ramanayake & Josef Urban, editors: *Automated Reasoning with Analytic Tableaux and Related Methods*, Springer Nature Switzerland, Cham, pp. 223–241, doi:10.1007/978-3-031-43513-3_13.
- [2] Bahareh Afshari & Graham E. Leigh (2017): *Cut-free Completeness for Modal μ -Calculus*. In: *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pp. 1–12, doi:10.1109/LICS.2017.8005088.

- [3] Bahareh Afshari & Graham E. Leigh (2022): *Lyndon Interpolation for Modal μ -Calculus*. In Aybüke Özgün & Yulia Zinova, editors: *Language, Logic, and Computation*, Springer International Publishing, Cham, pp. 197–213, doi:10.1007/978-3-030-98479-3_10.
- [4] Juan P. Aguilera & Leonardo Pacheco (2025): *Intuitionistic Gödel-Löb without Sharps*. *ACM Trans. Comput. Logic* 26(4), doi:10.1145/3748649.
- [5] R. Alonderis, R. Pliuškevičius, A. Pliuškevičienė & H. Giedra (2020): *Loop-Type Sequent Calculi for Temporal Logic*. *Journal of Automated Reasoning* 64(8), pp. 1663–1684, doi:10.1007/s10817-020-09544-1.
- [6] Arnon Avron (1996): *The Method of Hypersequents in the Proof Theory of Propositional Non-Classical Logics*. In Wilfrid Hodges, Martin Hyland, Charles Steinhorn & John Truss, editors: *Logic: From Foundations to Applications: European Logic Colloquium*, Clarendon Press, USA, p. 1–32, doi:10.1093/oso/9780198538622.003.0001.
- [7] Bianca Boretti (2008): *Proof Analysis in Temporal Logic*. Ph.D. thesis, University of Milan.
- [8] James Brotherston (2006): *Sequent calculus proof systems for inductive definitions*. Ph.D. thesis, University of Edinburgh.
- [9] Kai Brünnler (2009): *Deep Sequent Systems for Modal Logic*. *Archive for Mathematical Logic* 48(6), pp. 551–577, doi:10.1007/s00153-009-0137-3.
- [10] Kai Brünnler & Martin Lange (2008): *Cut-free Sequent Systems for Temporal Logic*. *The Journal of Logic and Algebraic Programming* 76(2), pp. 216–225, doi:10.1016/j.jlap.2008.02.004. *Logic and Information: From Logic to Constructive Reasoning*.
- [11] Anupam Das, Iris van der Giessen & Sonia Marin (2024): *Intuitionistic Gödel-Löb Logic, à la Simpson: Labelled Systems and Birelational Semantics*. In Aniello Murano & Alexandra Silva, editors: *32nd EACSL Annual Conference on Computer Science Logic (CSL 2024), Leibniz International Proceedings in Informatics (LIPIcs)* 288, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 22:1–22:18, doi:10.4230/LIPIcs.CSL.2024.22.
- [12] Anupam Das & Marianna Girlando (2023): *Cyclic Hypersequent System for Transitive Closure Logic*. *Journal of Automated Reasoning* 67(3), p. 27, doi:10.1007/s10817-023-09675-1.
- [13] Simon Docherty & Reuben N. S. Rowe (2019): *A Non-wellfounded, Labelled Proof System for Propositional Dynamic Logic*. In Serenella Cerrito & Andrei Popescu, editors: *Automated Reasoning with Analytic Tableaux and Related Methods*, Springer International Publishing, Cham, pp. 335–352, doi:10.1007/978-3-030-29026-9_19.
- [14] Jérôme Fortier & Luigi Santocanale (2013): *Cuts for circular proofs: semantics and cut-elimination*. In Simona Ronchi Della Rocca, editor: *Computer Science Logic 2013 (CSL 2013), Leibniz International Proceedings in Informatics (LIPIcs)* 23, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 248–262, doi:10.4230/LIPIcs.CSL.2013.248.
- [15] Ioannis Kokkinis & Thomas Studer (2016): *Cyclic Proofs for Linear Temporal Logic*. *Concepts of Proof in Mathematics, Philosophy, and Computer Science* 6, pp. 171–192, doi:10.1515/9781501502620-011.
- [16] Björn Lellmann (2015): *Linear Nested Sequents, 2-Sequents and Hypersequents*. In Hans De Nivelle, editor: *Automated Reasoning with Analytic Tableaux and Related Methods, Lecture Notes in Computer Science* 9323, Springer, Cham, pp. 135–150, doi:10.1007/978-3-319-24312-2_10.
- [17] Tim S. Lyon (2026): *Loop-Checking and Counter-Model Extraction for Intuitionistic Tense Logics via Nested Sequents*. *CoRR* abs/2603.29424, doi:10.48550/arXiv.2603.29424.
- [18] Tim S. Lyon, Agata Ciabattoni, Didier Galmiche, Marianna Girlando, Dominique Larchey-Wendling, Daniel Méry, Nicola Olivetti & Revantha Ramanayake (2025): *Internal and External Calculi: Ordering the Jungle without Being Lost in Translations*. *Bulletin of the Section of Logic* 54(1), p. 59–151, doi:10.18778/0138-0680.2025.02.
- [19] Tim S. Lyon & Piotr Ostropolski-Nalewaja (2026): *Foundations for an Abstract Proof Theory in the Context of Horn Rules*. *ACM Transactions on Computational Logic* 27(4), doi:10.1145/3821208.

- [20] Tim S. Lyon & Omar Taher (2026): *Optimizing Proof-Search via Linearization for Gödel-Löb Logic with Tree-Hypersequents*. In: *Proceedings of the Seventeenth International Symposium on Games, Automata, Logics, and Formal Verification*, EPTCS. To Appear.
- [21] Donald A. Martin (1975): *Borel Determinacy*. *Annals of Mathematics* 102(2), pp. 363–371. Available at <http://www.jstor.org/stable/1971035>.
- [22] Andrea Masini (1992): *2-Sequent Calculus: A Proof Theory of Modalities*. *Annals of Pure and Applied Logic* 58(3), pp. 229–246, doi:10.1016/0168-0072(92)90029-Y.
- [23] Andrea Masini (1993): *2-Sequent Calculus: Intuitionism and Natural Deduction*. *Journal of Logic and Computation* 3(5), pp. 533–562, doi:10.1093/logcom/3.5.533.
- [24] Guillermo Menéndez Turata (2024): *Cyclic Proof Systems for Modal Fixpoint Logics*. Ph.D. thesis, Universiteit van Amsterdam. Available at <https://eprints.illc.uva.nl/id/eprint/2285/>. ILLC Dissertation Series DS-2023-11.
- [25] G. E. Mints (1978): *Finite Investigations of Transfinite Derivations*. *Journal of Soviet Mathematics* 10(4), pp. 548–596, doi:10.1007/BF01091743.
- [26] Grigori Mints (2000): *A Short Introduction to Intuitionistic Logic*. Springer New York, NY, doi:<https://doi.org/10.1007/b115304>.
- [27] Barbara Paech (1989): *Gentzen-Systems for Propositional Temporal Logics*. In Egon Börger, Hans Kleine Büning & Michael M. Richter, editors: *CSL '88*, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 240–253, doi:10.1007/BFb0026305.
- [28] Amir Pnueli (1977): *The Temporal Logic of Programs*. In: *18th Annual Symposium on Foundations of Computer Science*, pp. 46–57, doi:10.1109/SFCS.1977.32.
- [29] Garrel Pottinger (1983): *Uniform, Cut-free Formulations of T, S4 and S5*. *Journal of Symbolic Logic* 48(3), p. 900.
- [30] Mark Reynolds (2016): *A New Rule for LTL Tableaux*. In Domenico Cantone & Giorgio Delzanno, editors: *Proceedings of the Seventh International Symposium on Games, Automata, Logics and Formal Verification, GandALF 2016, Catania, Italy, 14-16 September 2016*, EPTCS 226, pp. 287–301, doi:10.4204/EPTCS.226.20.
- [31] Jan Rooduijn (2021): *Cyclic Hypersequent Calculi for Some Modal Logics with the Master Modality*. In Anupam Das & Sara Negri, editors: *Automated Reasoning with Analytic Tableaux and Related Methods*, Springer International Publishing, Cham, pp. 354–370, doi:10.1007/978-3-030-86059-2_21.
- [32] Jan Rooduijn (2024): *Fragments and Frame Classes: Towards a Uniform Proof Theory for Modal Fixed Point Logics*. Ph.D. thesis, Universiteit van Amsterdam. Available at <https://www.illc.uva.nl/Research/Publications/Dissertations/DS/publication/5395/>.
- [33] Jan Rooduijn & Lukas Zenger (2022): *An Analytic Proof System for Common Knowledge Logic over S5*. In David Fernández-Duque, Alessandra Palmigiano & Sophie Pinchinat, editors: *Advances in Modal Logic 14*, College Publications, pp. 659–679. Available at <http://www.aiml.net/volumes/volume14/37-Rooduijn-Zenger.pdf>.
- [34] D. S. Shamkanov (2014): *Circular Proofs for the Gödel-Löb Provability Logic*. *Mathematical Notes* 96(3), pp. 575–585, doi:10.1134/S0001434614090326.
- [35] Alex K. Simpson (1994): *The Proof Theory and Semantics of Intuitionistic Modal Logic*. Ph.D. thesis, University of Edinburgh.
- [36] Luca Viganò (2000): *Labelled Non-Classical Logics*. Springer Science & Business Media, doi:10.1007/978-1-4757-3208-5.
- [37] Lukas Zenger (2026): *Intuitionistic Common Knowledge*. CoRR abs/2605.00523, doi:10.48550/arXiv.2605.00523.