A Cyclic Proof System for $\mathrm{HFL}_{\mathbb{N}}$

Mayuko Kori
Department of Informatics, The Graduate University for Advanced Studies (SOKENDAI), Hayama, Japan
National Institute of Informatics, Tokyo, Japan
mkori@nii.ac.jp

Takeshi Tsukada
Graduate School of Science, Chiba University, Japan
tsukada@math.s.chiba-u.ac.jp

Naoki Kobayashi
The University of Tokyo, Japan
koba@is.s.u-tokyo.ac.jp

Abstract

A cyclic proof system allows us to perform inductive reasoning without explicit inductions. We propose a cyclic proof system for $\mathrm{HFL}_{\mathbb{N}}$, which is a higher-order predicate logic with natural numbers and alternating fixed-points. Ours is the first cyclic proof system for a higher-order logic, to our knowledge. Due to the presence of higher-order predicates and alternating fixed-points, our cyclic proof system requires a more delicate global condition on cyclic proofs than the original system of Brotherston and Simpson. We prove the decidability of checking the global condition and soundness of this system, and also prove a restricted form of standard completeness for an infinitary variant of our cyclic proof system. A potential application of our cyclic proof system is semi-automated verification of higher-order programs, based on Kobayashi et al.'s recent work on reductions from program verification to $\mathrm{HFL}_{\mathbb{N}}$ validity checking.

2012 ACM Subject Classification Theory of computation → Proof theory

Keywords and phrases Cyclic proof, higher-order logic, fixed-point logic, sequent calculus

Digital Object Identifier 10.4230/LIPIcs.CSL.2021.29

Related Version A full version of the paper is available at `https://arxiv.org/abs/2010.14891`.

Funding This work was supported by JSPS KAKENHI Grant Number JP15H05706, JP20H00577, and JP20H05703.
Mayuko Kori: The first author is supported by ERATO HASUO Metamathematics for Systems Design Project (No. JPMJER1603), JST.

Acknowledgements We would like to thank anonymous referees for useful comments.

1 Introduction

There have recently been extensive studies on cyclic proof systems. They allow a proof to be cyclic, as long as it satisfies a certain sanity condition called the "global trace condition." Cyclic proofs enable inductive reasoning without explicit inductions, which would be useful for proof automation. Various cyclic proof systems have been proposed [3, 7, 13] for first-order logics, and some of them have been applied to automated program verification [1, 2, 14].

In the present paper, we propose a cyclic proof system for $\mathrm{HFL}_{\mathbb{N}}$, a higher-order logic with natural numbers and least/greatest fixpoint operators on higher-order predicates. $\mathrm{HFL}_{\mathbb{N}}$ has been introduced by Kobayashi et al. [9, 10] as an extension of HFL [16], and shown to be useful for higher-order program verification. Verification of various temporal properties of higher-order programs can naturally be reduced to validity checking for $\mathrm{HFL}_{\mathbb{N}}$ formulas. For example, consider the following OCaml program:

```
let rec g n = if n=0 then () else g (n-1) in
let rec f h m = h m; f h (m+1) in f g 0
```

The property "`f` is infinitely often called" is then expressed as the following formula:

$$\left(\nu F.\lambda h.\lambda m.h\ m \wedge F\ h\ (m+1)\right)\ \left(\mu G.\lambda n.(n = 0 \vee (n \neq 0 \wedge G\ (n-1)))\right)\ 0.$$

Here, $\nu F.\lambda h.\cdots$ and $\mu G.\lambda n.\cdots$ respectively represent the greatest and least predicates such that $F = \lambda h.\cdots$ and $G = \lambda n.\cdots$. Notice the close correspondence between the program and the formula: the functions f and g correspond to the predicates F and G, and function calls correspond to applications of the predicates; an interested reader may wish to consult [9, 10] to learn how program verification problems can be translated to HFL$_\mathbb{N}$ formulas. Our cyclic proof system for HFL$_\mathbb{N}$ presented in this paper would, therefore, be useful for semi-automated verification of higher-order programs.

A key issue in the design of our cyclic proof system is how to formalize a decidable "global trace condition," to guarantee the soundness of cyclic proofs in the presence of higher-order predicates and alternating fixed-points. Our global trace condition has been inspired by, and is actually very similar to that of Doumane [7]. The decidability of our global trace condition is, however, non-trivial, due to the presence of higher-order predicates. Inspired by the approach of Brotherston and Simpson [3], we reduce the global trace condition to the containment problem for Büchi automata.

We also consider an infinitary version of our proof system, and prove that the infinitary proof system is complete for sequents without higher-order variables. The restriction to sequents without higher-order variables is sufficient for the aforementioned application of our proof system to higher-order program verification.

The rest of this paper is structured as follows. Section 2 reviews the syntax and semantics of HFL$_\mathbb{N}$. Section 3 defines our cyclic proof system. Sections 4 and 5 respectively prove the decidability of the global trace condition and the soundness of our cyclic proof system. Section 6 discusses the restricted form of completeness of the infinitary variant of our proof system. Section 7 discusses related work, and Section 8 concludes the paper.

2 HFL$_\mathbb{N}$: Higher-Order Fixed-Point Arithmetic

This section introduces the target logic HFL$_\mathbb{N}$, which is a higher-order logic with natural numbers and alternating fixed-points. It has been introduced by Kobayashi et al. [10, 17] as an extension of higher-order modal fixed-point logic (HFL) [16].

2.1 Syntax of HFL$_\mathbb{N}$

HFL$_\mathbb{N}$ is simply typed. The syntax of *types* is given as follows:

$$A ::= \mathbf{N} \mid T \qquad T ::= \mathbf{\Omega} \mid A \to T.$$

$\mathbf{N}$ is the type of natural numbers, $\mathbf{\Omega}$ is the type of propositions and $A \to T$ is a function type. Occurrences of $\mathbf{N}$ are restricted to argument positions for a technical reason (see below).

Let $\mathcal{V}$ be a countably infinite set of *variables*, ranged over by $x, y, z, f, X, Y, Z, \ldots$. The syntax of *terms* and *formulas* is given by:

$$\begin{array}{llll} \text{term} & s,t & ::= & x \mid \mathbf{Z} \mid \mathbf{S}t \\ \text{formula} & \varphi,\psi & ::= & s = t \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid x \mid \lambda x^A.\varphi \mid \varphi\,\psi \mid \varphi\,t \mid \mu x^T.\varphi \mid \nu x^T.\varphi. \end{array}$$

We shall often omit the type annotations. The syntax of terms is standard: $\mathbf{Z}$ represents zero and $\mathbf{S}$ is the successor function. A closed term must be of the form $\mathbf{S}^n\mathbf{Z}$, which is often identified with the natural number n. Constructors of formulas are logical ones ($s = t$, $\varphi \vee \psi$ and $\varphi \wedge \psi$), those from the λ-calculus (variable x, abstraction $\lambda x.\varphi$, and application $\varphi\,\psi$ and $\varphi\,t$), and fixed-point operators (least fixed-point $\mu x.\varphi$ and greatest fixed-point $\nu x.\varphi$). Some standard constructs such as summation $t_1 + t_2 = s$, multiplication $t_1 \times t_2 = s$, truth $\top$ and quantifiers $\forall$ and $\exists$ are definable; see examples later in this subsection. The set of *free variables* is defined as usual; the binders are λx, μx and νx.

A *sequent* is a pair (Γ, Δ) of finite sequences of formulas, written as $\Gamma \vdash \Delta$. For a finite sequence Γ of formulas, $FV(\Gamma)$ denotes the union of the sets of free variables of formulas in Γ. The free variables of a sequent $\Gamma \vdash \Delta$ is $FV(\Gamma) \cup FV(\Delta)$.

The type system is presented in Figure 1. Here $\mathcal{H}$ is a *type environment*, which is a map from a finite subset of variables to the set of types. The typing rules should be easy to understand. The types of fixed-point formulas $\mu x.\varphi$ and $\nu y.\psi$, as well as the types of the variables x and y, are restricted to T (i.e., they cannot be $\mathbf{N}$). A formula φ is *well-typed* if $\mathcal{H} \vdash \varphi : T$ for some $\mathcal{H}$ and T. In the sequel, we shall consider only well-typed formulas.

For a sequent $\Gamma \vdash \Delta$, we write $\mathcal{H} \mid \Gamma \vdash \Delta$ if $\mathcal{H} \vdash \varphi : \mathbf{\Omega}$ for every φ in Γ or Δ. A sequent is *well-typed* if $\mathcal{H} \mid \Gamma \vdash \Delta$ for some $\mathcal{H}$.

We give some examples of formulas and explain their intuitive meaning.

▶ **Example 1.** The truth $\top$ and falsity $\bot$ can be defined as fixed-points:

$$\top \;:=\; \nu x^{\mathbf{\Omega}}.x \qquad \text{and} \qquad \bot \;:=\; \mu x^{\mathbf{\Omega}}.x.$$

The former means that $\top$ is the greatest element in the set of truth values (i.e. elements in the semantic domain of $\mathbf{\Omega}$) such that $x = x$. Similarly $\bot$ is the least truth value. The greatest $\top_T$ and least $\bot_T$ value of type T are defined in a similar way. ⌟

▶ **Example 2.** The quantifiers over natural numbers are definable. Let *forall* be a predicate of type $(\mathbf{N} \to \mathbf{\Omega}) \to \mathbf{N} \to \mathbf{\Omega}$ defined by

$$\mathit{forall} \quad := \quad \nu X.\lambda p^{\mathbf{N}\to\mathbf{\Omega}}.\lambda x^{\mathbf{N}}.p\,x \wedge X\,p\,(\mathbf{S}\,x).$$

Then $\mathit{forall}\,\varphi\,n$ holds if and only if $\varphi\,m$ holds for every $m \geq n$. To see this, notice that $(\mathit{forall}\,\varphi\,n) = (\varphi\,n) \wedge (\mathit{forall}\,\varphi\,(n+1))$ since *forall* is a fixed-point. By iteratively applying this equation, we have

$$(\mathit{forall}\,\varphi\,n) \quad = \quad (\varphi\,n) \wedge (\varphi\,(n+1)) \wedge (\varphi\,(n+2)) \wedge \cdots$$

and thus $\mathit{forall}\,\varphi\,n$ if and only if $\forall m \geq n.\varphi\,m$.[1] Then the universal quantifier over natural numbers is defined by

$$\forall x.\varphi \quad := \quad \mathit{forall}\,(\lambda x.\varphi)\,\mathbf{Z}.$$

The existential quantifier can be defined similarly. A direct definition is

$$\exists x^{\mathbf{N}}.\varphi \quad := \quad \left(\mu Y^{\mathbf{N}\to\mathbf{\Omega}}.\lambda x.\varphi \vee Y\,(\mathbf{S}x)\right)\mathbf{Z}.$$

The quantifiers over T are easier because of monotonicity (see next subsection); we define $\forall x^T.\varphi := (\lambda x.\varphi)\,\bot_T$ and $\exists x^T.\varphi := (\lambda x.\varphi)\,\top_T$. ⌟

[1] The reader may notice that this intuitive argument does not explain why we should use ν instead of μ. Here we skip this subtle issue. An interested reader may compare the interpretations of *forall* and its μ-variant, following the definition in the next subsection.

For terms:

$$\frac{}{\mathcal{H}, x : A \vdash x : A} \qquad \frac{}{\mathcal{H} \vdash \mathbf{Z} : \mathbf{N}} \qquad \frac{\mathcal{H} \vdash t : \mathbf{N}}{\mathcal{H} \vdash \mathbf{S}t : \mathbf{N}}$$

For formulas:

$$\frac{\mathcal{H} \vdash s : \mathbf{N} \qquad \mathcal{H} \vdash t : \mathbf{N}}{\mathcal{H} \vdash s = t : \boldsymbol{\Omega}}$$

$$\frac{\mathcal{H} \vdash \varphi : \boldsymbol{\Omega} \qquad \mathcal{H} \vdash \psi : \boldsymbol{\Omega}}{\mathcal{H} \vdash \varphi \vee \psi : \boldsymbol{\Omega}} \qquad \frac{\mathcal{H} \vdash \varphi : \boldsymbol{\Omega} \qquad \mathcal{H} \vdash \psi : \boldsymbol{\Omega}}{\mathcal{H} \vdash \varphi \wedge \psi : \boldsymbol{\Omega}}$$

$$\frac{\mathcal{H}, x : A \vdash \varphi : T}{\mathcal{H} \vdash \lambda x^A.\varphi : A \to T} \qquad \frac{\mathcal{H} \vdash \varphi : T' \to T \qquad \mathcal{H} \vdash \psi : T'}{\mathcal{H} \vdash \varphi\, \psi : T}$$

$$\frac{\mathcal{H} \vdash \varphi : \mathbf{N} \to T \qquad \mathcal{H} \vdash t : \mathbf{N}}{\mathcal{H} \vdash \varphi\, t : T}$$

$$\frac{\mathcal{H}, x : T \vdash \varphi : T}{\mathcal{H} \vdash \mu x^T.\varphi : T} \qquad \frac{\mathcal{H}, x : T \vdash \varphi : T}{\mathcal{H} \vdash \nu x^T.\varphi : T}$$

Figure 1 Typing Rules for HFL$_{\mathbb{N}}$.

▶ **Example 3.** Let *sum* be the summation, i.e. the predicate such that $sum\, n\, m\, k$ holds if and only if $n + m = k$. This predicate can be defined as follows:

$$\mu sum.\, \lambda x^{\mathbf{N}}.\lambda y^{\mathbf{N}}.\lambda z^{\mathbf{N}}.(x = \mathbf{Z} \wedge y = z) \vee (\exists x'.\exists z'.x = \mathbf{S}x' \wedge sum\, x'\, y\, z' \wedge z = \mathbf{S}z').$$

This formula represents the standard inductive definition of the summation: $\mathbf{Z} + y = y$ and $x' + y = z' \Longrightarrow \mathbf{S}x' + y = \mathbf{S}z'$. One can define the multiplication in a similar way, representing the standard inductive definition of the multiplication using *sum*. ⌟

▶ **Example 4.** The inequality $s < t$ on terms is also definable. The idea is to appeal to the following fact: if $n < m$, then $n + 1 = m$ or $(n + 1) < m$. This justifies

$$(s < t) \quad := \quad \big(\mu X.\lambda y^{\mathbf{N}}.(\mathbf{S}y = t) \vee X\,(\mathbf{S}y)\big)\, s.$$

Here X is a variable of type $\mathbf{N} \to \boldsymbol{\Omega}$ and $X\, s'$ means $s' < t$. Then the inequality $s \neq t$ can be defined as $(s < t) \vee (t < s)$. ⌟

▶ Remark 5. HFL$_{\mathbb{N}}$ does not have negation $\neg$. The absence of negation plays an important role in the interpretation of fixed-point operators, as we shall see in the next subsection. For a formula φ with no free variables except for those of type $\mathbf{N}$, the negation $\neg\varphi$ can be obtained by replacing each logical connective with its De Morgan dual,

$$\wedge \leftrightsquigarrow \vee, \qquad \mu \leftrightsquigarrow \nu, \quad \text{and} \quad = \leftrightsquigarrow \neq,$$

as discussed in [12]. ⌟

2.2 Semantics of HFL$_{\mathbb{N}}$

This subsection introduces the interpretations of types and formulas.

The interpretation of a type A is a poset $[\![A]\!] = ([\![A]\!], \leq_A)$. It is inductively defined by

$$\begin{aligned}
[\![\mathbf{N}]\!] &:= \mathbb{N} & x \leq_{\mathbf{N}} y &:\Leftrightarrow x = y \\
[\![\boldsymbol{\Omega}]\!] &:= \{\top, \bot\} & x \leq_{\boldsymbol{\Omega}} y &:\Leftrightarrow x = \bot \text{ or } y = \top \\
[\![A \to T]\!] &:= \{f : [\![A]\!] \to [\![T]\!] \mid f \text{ is monotone}\} & f \leq_{A \to T} g &:\Leftrightarrow \forall x \in [\![A]\!].f(x) \leq_T g(x).
\end{aligned}$$

$$
\begin{aligned}
[\![\mathcal{H} \vdash x : A]\!](\rho) &= \rho(x)\\
[\![\mathcal{H} \vdash \mathbf{Z} : \mathbf{N}]\!](\rho) &= 0\\
[\![\mathcal{H} \vdash \mathbf{S}t : \mathbf{N}]\!](\rho) &= 1 + [\![\mathcal{H} \vdash t : \mathbf{N}]\!](\rho)\\
[\![\mathcal{H} \vdash s = t : \boldsymbol{\Omega}]\!](\rho) &= ([\![\mathcal{H} \vdash s : \mathbf{N}]\!](\rho) = [\![\mathcal{H} \vdash t : \mathbf{N}]\!](\rho))\\
[\![\mathcal{H} \vdash \varphi \vee \psi : \boldsymbol{\Omega}]\!](\rho) &= ([\![\mathcal{H} \vdash \varphi : \boldsymbol{\Omega}]\!](\rho) \vee [\![\mathcal{H} \vdash \psi : \boldsymbol{\Omega}]\!](\rho))\\
[\![\mathcal{H} \vdash \varphi \wedge \psi : \boldsymbol{\Omega}]\!](\rho) &= ([\![\mathcal{H} \vdash \varphi : \boldsymbol{\Omega}]\!](\rho) \wedge [\![\mathcal{H} \vdash \psi : \boldsymbol{\Omega}]\!](\rho))\\
[\![\mathcal{H} \vdash \lambda x^A.\varphi : A \to T]\!](\rho) &= \lambda v \in [\![A]\!].[\![\mathcal{H}, x : A \vdash \varphi : T]\!](\rho[x \mapsto v])\\
[\![\mathcal{H} \vdash \varphi\, \psi : T]\!](\rho) &= ([\![\mathcal{H} \vdash \varphi : A \to T]\!](\rho))\,([\![\mathcal{H} \vdash \psi : A]\!](\rho))\\
[\![\mathcal{H} \vdash \varphi\, t : T]\!](\rho) &= ([\![\mathcal{H} \vdash \varphi : \mathbf{N} \to T]\!](\rho))\,([\![\mathcal{H} \vdash t : \mathbf{N}]\!](\rho))\\
[\![\mathcal{H} \vdash \mu x^T.\varphi : T]\!](\rho) &= \mathrm{lfp}([\![\mathcal{H} \vdash \lambda x^T.\varphi : T \to T]\!](\rho))\\
[\![\mathcal{H} \vdash \nu x^T.\varphi : T]\!](\rho) &= \mathrm{gfp}([\![\mathcal{H} \vdash \lambda x^T.\varphi : T \to T]\!](\rho))
\end{aligned}
$$

Figure 2 Interpretation of terms and formulas.

Note that

- $[\![A \to T]\!]$ is the space of *monotone* functions, and
- $[\![T]\!]$ is a complete lattice for every T.

On the contrary $[\![A]\!]$ is not necessarily a complete lattice since $[\![\mathbf{N}]\!]$ is not.

The interpretation $[\![\mathcal{H}]\!]$ of a type environment $\mathcal{H}$ is the set of mappings ρ such that $\rho(x) \in [\![\mathcal{H}(x)]\!]$ for every x in the domain of $\mathcal{H}$. The set $[\![\mathcal{H}]\!]$ forms a poset with respect to the point-wise ordering. An element of $[\![\mathcal{H}]\!]$ is called a *valuation*. We write $\rho[x \mapsto v]$ for the mapping defined by $\rho[x \mapsto v](x) = v$ and $\rho[x \mapsto v](y) = \rho(y)$ for $y \neq x$.

Assume $\mathcal{H} \vdash \varphi : A$ (where φ is a formula or a term). Its interpretation $[\![\mathcal{H} \vdash \varphi : A]\!]$ is a monotone function $[\![\mathcal{H}]\!] \longrightarrow [\![A]\!]$. The definition is in Figure 2. Here $\mathit{lfp}(f)$ and $\mathit{gfp}(f)$ are the least and greatest fixed-points of the function f. The interpretations of the fixed-point operators are well-defined since every monotone function $f : P \longrightarrow P$ on a complete lattice P has both the least and greatest fixed-points. We shall write $[\![\mathcal{H} \vdash \varphi : A]\!](\rho)$ simply $[\![\varphi]\!]_\rho$ when no confusion can arise.

▶ **Definition 6.** *Let $\mathcal{H} \mid \Gamma \vdash \Delta$ be a sequent and ρ be a valuation in $[\![\mathcal{H}]\!]$. Then we write $\Gamma \models_\rho \Delta$ if $\bigwedge_{\varphi\in\Gamma}[\![\varphi]\!]_\rho \leq \bigvee_{\psi\in\Delta}[\![\psi]\!]_\rho$, or equivalently, if $[\![\varphi]\!]_\rho = \bot$ for some $\varphi \in \Gamma$ or $[\![\psi]\!]_\rho = \top$ for some $\psi \in \Delta$. A sequent $\mathcal{H} \mid \Gamma \vdash \Delta$ is* valid *if $\Gamma \models_\rho \Delta$ for all valuations ρ, and we denote it briefly by $\Gamma \models \Delta$.*

3 A Cyclic Proof System for $\mathrm{HFL}_{\mathbb{N}}$

In this section, we introduce a cyclic proof system for $\mathrm{HFL}_{\mathbb{N}}$. A cyclic proof is a proof diagram which can contain cycles and should satisfy a certain condition in order to ensure the soundness. Subsection 3.1 describes derivations and the soundness condition to define cyclic proofs and Subsection 3.2 shows that the induction rule based on prefixed-points is admissible. In Subsection 3.3, we show some examples of cyclic proofs.

3.1 Definition of the Cyclic Proof System

Figure 3 shows our deduction rules, which are based on Gentzen's sequent calculus. Here $\varphi[\psi/x]$ represents the capture-avoiding substitution and $\Gamma[\varphi/x]$ represents the sequence of formulas which is the result of applying the substitution $[\varphi/x]$ to all formulas in Γ. The rules

should be easy to understand since most of the rules are standard. We explain uncommon rules. The rule (Mono) is based on the fact that each formula defines a monotone function, i.e. $[\![\psi]\!] \sqsubseteq [\![\chi]\!]$ implies $[\![\varphi[\psi/x]]\!] \sqsubseteq [\![\varphi[\chi/x]]\!]$. Since only a formula of type $\mathbf{\Omega}$ can appear in a sequent, $[\![\psi]\!] \sqsubseteq [\![\chi]\!]$ is expressed as $\psi\,\vec{y} \vdash \chi\,\vec{y}$ for fresh $\vec{y}$. The rules (λL) and (λR) are justified by the fact that the β-equivalence $(\lambda x.\varphi)\,\psi = \varphi[\psi/x]$ preserves semantics. The rules (σL) and (σR) express the fact that $\sigma x.\varphi$ (where σ is μ or ν) is a fixed-point and thus $\sigma x.\varphi = \varphi[\sigma x.\varphi/x]$. The rule (Nat) says that a variable of type $\mathbf{N}$ indeed represents a natural number, and (P1) and (P2) correspond to the axioms $(\mathbf{Z} = \mathbf{S}x) \to \bot$ and $(\mathbf{S}x = \mathbf{S}y) \to (x = y)$.

- Identity rules

$$\frac{}{\varphi \vdash \varphi}\ (\text{Axiom}) \qquad \frac{\Gamma \vdash \varphi, \Delta \qquad \Gamma, \varphi \vdash \Delta}{\Gamma \vdash \Delta}\ (\text{Cut})$$

- Structural rules

$$\frac{\Gamma \vdash \Delta}{\Gamma, \varphi \vdash \Delta}\ (\text{Wk L}) \qquad \frac{\Gamma \vdash \Delta}{\Gamma \vdash \varphi, \Delta}\ (\text{Wk R})$$

$$\frac{\Gamma, \varphi, \varphi \vdash \Delta}{\Gamma, \varphi \vdash \Delta}\ (\text{Ctr L}) \qquad \frac{\Gamma \vdash \varphi, \varphi, \Delta}{\Gamma \vdash \varphi, \Delta}\ (\text{Ctr R})$$

$$\frac{\Gamma, \psi, \varphi, \Gamma' \vdash \Delta}{\Gamma, \varphi, \psi, \Gamma' \vdash \Delta}\ (\text{Ex L}) \qquad \frac{\Gamma \vdash \Delta, \psi, \varphi, \Delta'}{\Gamma \vdash \Delta, \varphi, \psi, \Delta'}\ (\text{Ex R})$$

$$\frac{\Gamma \vdash \Delta}{\Gamma[\varphi/x] \vdash \Delta[\varphi/x]}\ (\text{Subst})$$

$$\frac{\Gamma, \psi\,\vec{y} \vdash \chi\,\vec{y}, \Delta}{\Gamma, \varphi[\psi/x^T] \vdash \varphi[\chi/x^T], \Delta}\ \vec{y} \cap FV(\Gamma, \psi, \chi, \Delta) = \emptyset,\ (\text{Mono})$$

- Logical rules $(\sigma = \mu, \nu)$

$$\frac{\Gamma[t/x, s/y] \vdash \Delta[t/x, s/y]}{\Gamma[s/x, t/y], s = t \vdash \Delta[s/x, t/y]}\ (= L) \qquad \frac{}{\Gamma \vdash t = t, \Delta}\ (= R)$$

$$\frac{\Gamma, \varphi \vdash \Delta \qquad \Gamma, \psi \vdash \Delta}{\Gamma, \varphi \vee \psi \vdash \Delta}\ (\vee L) \qquad \frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \varphi \vee \psi, \Delta}\ (\vee R)$$

$$\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta}\ (\wedge L) \qquad \frac{\Gamma \vdash \varphi, \Delta \qquad \Gamma \vdash \psi, \Delta}{\Gamma \vdash \varphi \wedge \psi, \Delta}\ (\wedge R)$$

$$\frac{\Gamma, \varphi[\psi/x]\,\vec{\psi} \vdash \Delta}{\Gamma, (\lambda x.\varphi)\,\psi\,\vec{\psi} \vdash \Delta}\ (\lambda L) \qquad \frac{\Gamma \vdash \varphi[\psi/x]\,\vec{\psi}, \Delta}{\Gamma \vdash (\lambda x.\varphi)\,\psi\,\vec{\psi}, \Delta}\ (\lambda R)$$

$$\frac{\Gamma, \varphi[\sigma x.\varphi/x]\,\vec{\psi} \vdash \Delta}{\Gamma, (\sigma x.\varphi)\,\vec{\psi} \vdash \Delta}\ (\sigma L) \qquad \frac{\Gamma \vdash \varphi[\sigma x.\varphi/x]\,\vec{\psi}, \Delta}{\Gamma \vdash (\sigma x.\varphi)\,\vec{\psi}, \Delta}\ (\sigma R)$$

- Natural number rules
  $N \equiv \mu X.\lambda x.(x = \mathbf{Z}) \vee (\exists x'.x = \mathbf{S}x' \wedge X\,x')$

$$\frac{\Gamma, N\,x^{\mathbf{N}} \vdash \Delta}{\Gamma \vdash \Delta}\ (\text{Nat}) \qquad \frac{}{\mathbf{S}s = \mathbf{Z} \vdash}\ (\text{P1}) \qquad \frac{\Gamma, s = t \vdash \Delta}{\Gamma, \mathbf{S}s = \mathbf{S}t \vdash \Delta}\ (\text{P2})$$

Figure 3 Deduction Rules.

Although every leaf of an ordinary proof tree is an axiom, this is not the case in cyclic proof systems. A *(finite) derivation tree* is a tree obtained by using the rules in Figure 3, whose leaves are not necessarily axioms. A leaf that is not an axiom is called *open*.

▶ **Definition 7** (Pre-proof). *A* pre-proof *consists of a finite derivation tree $\mathcal{D}$ and a function $\mathcal{R}$ that assigns to each open leaf n in $\mathcal{D}$ a non-leaf node $\mathcal{R}(n)$ that has the same sequent as n.*

A pre-proof induces the infinite derivation tree by iteratively replacing an open leaf n with $\mathcal{R}(n)$. This correspondence would be helpful to understand the definitions below.

A pre-proof is unsound in general, i.e., the root sequent of a pre-proof may be invalid. We introduce a sanity condition called the *global trace condition*, and define cyclic proofs as pre-proofs that satisfy this condition.

Given a pre-proof $(\mathcal{D}, \mathcal{R})$, its *path* is a (finite or infinite) sequence $(n_i)_{i=1,2,\ldots}$ of nodes of $\mathcal{D}$ such that, for every i,

- if n_i is an open leaf, then $n_{i+1} = \mathcal{R}(n_i)$, and
- otherwise n_{i+1} is a premise of n_i.

Given a path $(n_i)_{i=1,2,\ldots}$, a *pre-trace* in this path is a sequence $(\tau_i)_{i=1,2,\ldots}$ of occurrences of formulas such that, for every i, τ_i is an occurrence of a formula in the sequent n_i and τ_{i+1} is a "relevant occurrence" of τ_i in the sequent n_{i+1}. The latter condition means that τ_{i+1} originates from τ_i in a bottom-up construction of the proof. For example, if n_i and n_{i+1} are respectively the conclusion and premise of $(\wedge L)$, i.e., if n_i is the sequent $\Gamma, \varphi \wedge \psi \vdash \Delta$ and n_{i+1} is $\Gamma, \varphi, \psi \vdash \Delta$, then (i) φ and ψ in n_{i+1} are relevant occurrences of $\varphi \wedge \psi$ in n_i, and (ii) each formula in Γ (resp. Δ) of n_{i+1} is a relevant occurrence of the corresponding formula in Γ (resp. Δ) of n_i. For another example, if n_i is the conclusion of $(\vee L)$, n_{i+1} is the left premise and τ_i is the occurrence of $\varphi \vee \psi$, then τ_{i+1} is the occurrence of φ. The concrete definition, which we omit here, is lengthy but straightforward; a possible exception is (Mono), in which $\psi\,\vec{y}$ and $\chi\,\vec{y}$ are defined as relevant occurrences of $\varphi[\psi/x]$ and $\varphi[\chi/x]$ respectively. See Appendix A for more detail. A pre-trace is a *trace* if, for infinitely many i, τ_i is the principal occurrence[2] of a logical rule.

The global trace condition requires existence of a "good" trace for each infinite path. The appropriate notion of "good" traces depends on the logic. In [3], a trace is "good" if it contains infinitely many principal occurrences of (μL) or (νR). In other words, a "good" trace contains infinitely many expansions of μ. This fairly simple condition comes from the restriction of usage of fixed-points: their logic does not allow alternation of fixed-points, e.g. $\mu P.\varphi$ is allowed only if the free predicate variables of φ are bound by μ. Allowing nested fixed-points makes the definition of "good" traces more complicated; the definition in [7] refers to the most significant fixed-point operator among those that are expanded infinitely many times. The higher-order nature of $\mathrm{HFL}_{\mathbb{N}}$ requires us to more precisely track the usage of fixed-point operators.

The following definition is inspired by the winning criterion of game semantics of $\mathrm{HFL}_{\mathbb{N}}$ [4, 10, 15]. The idea is to track which occurences of fixed-point operators are unfolded infinitely in depth by annotating each occurrence with a sequence that grows with each unfolding. By abuse of notation, a path is written as a sequence $(\Gamma_i \vdash \Delta_i)_{i=1,2,\ldots}$ of sequents. We often identify an occurrence of a formula with the formula. For example, $\tau_i \equiv \varphi$ means that τ_i is an occurrence of φ.

[2] An occurrence of a formula in the conclusion of a rule in Figure 3 is *principal* if it belongs to neither Γ nor Δ.

▶ **Definition 8** (μ-trace, ν-trace). *Let $(\tau_i)_{i\geq 0}$ be a trace. We assign a sequence of natural numbers to every fixed-point operator in τ_i for all i by the following algorithm. We use σ as a metavariable of fixed-point operators $\{\mu, \nu\}$. We write σ_p for the fixed-point operator to which the sequence p is assigned.*

- *For every σ in τ_0, we assign ϵ to σ.*
- *If τ_i is the principal occurrence of $(\sigma L/R)$, then τ_i with annotation is $\sigma_p x.\varphi$ (where fixed-point operators in φ are also annotated). Then τ_{i+1} with annotation is $\varphi[\sigma_{p.k} x.\varphi/x]$ where k is a natural number that has not been used in this annotation process.*[3]
- *Otherwise, the sequence of a fixed-point operator in τ_{i+1} comes from the corresponding operator in τ_i. For example, if τ_i is the principal occurrence of $(\lambda L/R)$ and τ_i with annotation is $(\lambda x.\varphi)\,\psi$, then τ_{i+1} with annotation is $\varphi[\psi/x]$.*

Let $p[0:n]$ denote the sequence consisting of the first n elements of p. We call $(\tau_i)_{i\geq 0}$ a μ-trace (resp. ν-trace) *if there is an infinite sequence p such that $p[0:n]$ is assigned to μ (resp. ν) in some τ_i for every natural number n.*

In the definiton above, for each trace $(\tau_i)_{i\geq 0}$, there is at most one infinite trace p that satisfies the condition above; hence, no trace can be both a μ-trace and a ν-trace; see Lemma 22.

▶ **Example 9.** Let us consider the following pre-proof $(\mathcal{D}, \mathcal{R})$.

$$\dfrac{\dfrac{\dfrac{\dfrac{(\star)\ \vdash (\nu f.\lambda g.g\,(f\,g))\,(\mu x.\lambda a.a)}{\vdash (\lambda a.a)\,((\nu f.\lambda g.g\,(f\,g))\,(\mu x.\lambda a.a))}\,(\lambda R)}{\vdash (\mu x.\lambda a.a)\,((\nu f.\lambda g.g\,(f\,g))\,(\mu x.\lambda a.a))}\,(\mu R)}{\vdash (\lambda h.h\,((\nu f.\lambda g.g\,(f\,g))\,h))\,(\mu x.\lambda a.a)}\,(\lambda R)}{(\star)\ \vdash (\nu f.\lambda g.g\,(f\,g))\,(\mu x.\lambda a.a)}\,(\nu R)$$

In the diagram, the function $\mathcal{R}$ is indicated by the $\star$ marks: the open leaf $(\star)$ is mapped to the other node marked $(\star)$. This pre-proof has a unique path, and the path has a unique trace $(\tau_i)_{i\geq 0}$ (since each sequent consists of a single formula).

We assign a sequence of natural numbers to each occurrence of a fixed-point operator in the trace $(\tau_i)_{i\geq 0}$. Both fixed-point operators in τ_0 are annotated by the empty sequence:

$$\tau_0 \quad\equiv\quad (\nu_\epsilon f.\lambda g.g\,(f\,g))\,(\mu_\epsilon x.\lambda a.a).$$

The first rule expands ν, and we annotate the recursive occurrence of this ν with a fresh natural number, say 0:

$$\tau_1 \quad\equiv\quad (\lambda h.h\,((\nu_0 f.\lambda g.g\,(f\,g))\,h))\,(\mu_\epsilon x.\lambda a.a).$$

The next rule is (λR) and we just substitute the annotated formula $(\mu_\epsilon x.\lambda a.a)$ for h:

$$\tau_2 \quad\equiv\quad (\mu_\epsilon x.\lambda a.a)\,((\nu_0 f.\lambda g.g\,(f\,g))\,(\mu_\epsilon x.\lambda a.a)).$$

Then we expand μ and annotate its recursive occurrences (if there were any) with 1; actually, since x does not appear in the body $\lambda a.a$, the resulting formula does not have label 1.

$$\tau_3 \quad\equiv\quad (\lambda a.a)\,((\nu_0 f.\lambda g.g\,(f\,g))\,(\mu_\epsilon x.\lambda a.a)).$$

Applying the β-reduction, we have

$$\tau_4 \quad\equiv\quad (\nu_0 f.\lambda g.g\,(f\,g))\,(\mu_\epsilon x.\lambda a.a).$$

[3] One can weaken the freshness requirement for k: the minimal requirement is that the sequence $p.k$ has not been used.

The current node is the open leaf, and the next node is determined by $\mathcal{R}$. The annotation is copied: $\tau_5 \equiv \tau_4$. The next rule is (νR) and we name the recursive occurrences 0.2, extending the annotation 0 by a fresh number 2:

$$\tau_6 \quad \equiv \quad (\lambda h.h\,((\nu_{0.2} f.\lambda g.g\,(f\,g))\,h))\,(\mu_\epsilon x.\lambda a.a).$$

By continuing this argument, we have

$$\tau_{1+5k} \quad \equiv \quad (\lambda h.h\,((\nu_p f.\lambda g.g\,(f\,g))\,h))\,(\mu_\epsilon x.\lambda a.a)$$

where $p = 0.2.4.\ldots.(2k)$. Note that the annotation of ν grows but that of μ does not. Hence this trace is a ν-trace but not a μ-trace. ⌟

A trace $(\tau_i)_{i\geq 0}$ is a *left trace* (resp. *right trace*) if τ_0 occurs on the left (resp. right) side of $\vdash$. Note that every τ_i occurs on the same side as τ_0.

▶ **Definition 10** (Cyclic proof). *A* cyclic proof *is a pre-proof that satisfies the* global trace condition*: for every infinite path, a tail of the path has a left μ-trace or right ν-trace.*

▶ **Example 11.** The pre-proof in Example 9 is a cyclic proof. ⌟

▶ Remark 12. One may find our global trace condition (cf. Definitions 8 and 10) complicated and wonder if it is possible to replace it with a simpler conidition such as the parity condition. A recent result [15, Theorem 25] suggests a negative answer: It shows that the validity of $\mathrm{HFL}_{\mathbb{N}}$ formulas cannot be captured by parity games, but games with more complicated winning criteria. Our global trace condition is inspired by the criteria. ⌟

3.2 Some Admissible Rules

This subsection shows that some familiar rules for quantifiers and inductions are admissible in our cyclic proof system.

As we saw in Subsection 2.1, formulas with quantifiers can be expressed by fixed-points. The proposition below enables us to use quantifier rules in our cyclic proof system.

▶ **Proposition 13.** *If there is a cyclic proof of $\Gamma \vdash \Delta$ derived by the rules in Figure 3 plus the following quantifier rules, then there exists a cyclic proof of $\Gamma \vdash \Delta$ without the quantifier rules.*

$$\frac{\Gamma, \varphi[\psi/x] \vdash \Delta}{\Gamma, \forall x.\varphi \vdash \Delta}\ (\forall L) \qquad \frac{\Gamma \vdash \varphi, \Delta}{\Gamma \vdash \forall x.\varphi, \Delta}\ x \notin FV(\Gamma, \Delta)\ (\forall R)$$

$$\frac{\Gamma, \varphi \vdash \Delta}{\Gamma, \exists x.\varphi \vdash \Delta}\ x \notin FV(\Gamma, \Delta)\ (\exists L) \qquad \frac{\Gamma \vdash \varphi[\psi/x], \Delta}{\Gamma \vdash \exists x.\varphi, \Delta}\ (\exists R)$$

Proof. See [11]. ◀

We can also embed explicit induction rules, so-called Park's fixed-point rules:

$$\frac{\Gamma, \varphi[\chi/x]\,\vec{y} \vdash \chi\,\vec{y}, \Delta \qquad \Gamma, \chi\,\vec{\psi} \vdash \Delta}{\Gamma, (\mu x.\varphi)\,\vec{\psi} \vdash \Delta}\ \vec{y} \cap FV(\Gamma, \varphi[\chi/x], \Delta) = \emptyset\ \text{(Pre)}$$

$$\frac{\Gamma, \chi\,\vec{y} \vdash \varphi[\chi/x]\,\vec{y}, \Delta \qquad \Gamma \vdash \chi\,\vec{\psi}, \Delta}{\Gamma \vdash (\nu x.\varphi)\,\vec{\psi}, \Delta}\ \vec{y} \cap FV(\Gamma, \varphi[\chi/x], \Delta) = \emptyset\ \text{(Post)}$$

They are inspired by Knaster-Tarski's fixed-point theorem: these rules replace pre/postfixed-points with least/greatest fixed-points. The rule (Pre) is sound because χ is a prefixed-point by the left premise and $\mu x.\varphi$ is the least one. The same argument holds for (Post).

▶ **Proposition 14.** *If there is a cyclic proof of* $\Gamma \vdash \Delta$ *derived by the rules in Figure 3 + (Pre) + (Post), then there exists a cyclic proof of* $\Gamma \vdash \Delta$ *without (Pre) and (Post).*

Proof. Given a cyclic proof Π that may use (Pre) and (Post), we first construct a pre-proof Π' by removing (Pre) and (Post). For every instance of (Pre) of the form:

$$\frac{\Gamma, \varphi[\chi/x]\,\vec{y} \vdash \chi\,\vec{y}, \Delta \qquad \Gamma, \chi\,\vec{\psi} \vdash \Delta}{\Gamma, (\mu x.\varphi)\,\vec{\psi} \vdash \Delta}\ (\text{Pre})$$

we will replace it with the following diagram.

$$\dfrac{\Pi \qquad \dfrac{\Gamma, \chi\,\vec{\psi} \vdash \Delta}{\Gamma, (\mu x.\varphi)\,\vec{\psi}, \chi\,\vec{\psi} \vdash \Delta}\ (\text{Wk L})}{\Gamma, (\mu x.\varphi)\,\vec{\psi} \vdash \Delta}\ (\text{Cut})$$

$$\Pi := \dfrac{\dfrac{\Gamma, \varphi[\chi/x]\,\vec{y} \vdash \chi\,\vec{y}, \Delta}{\Gamma, (\mu x.\varphi)\,\vec{y}, \varphi[\chi/x]\,\vec{y} \vdash \chi\,\vec{y}, \Delta}\ (\text{Wk L}) \qquad \dfrac{\dfrac{(\star)\ \Gamma, (\mu x.\varphi)\,\vec{y} \vdash \chi\,\vec{y}, \Delta}{\Gamma, \varphi[\mu x.\varphi/x]\,\vec{y} \vdash \varphi[\chi/x]\,\vec{y}, \Delta}\ (\text{Mono})}{\Gamma, \varphi[\mu x.\varphi/x]\,\vec{y} \vdash \chi\,\vec{y}, \varphi[\chi/x]\,\vec{y}, \Delta}\ (\text{Wk R})}{\Gamma, (\mu x.\varphi)\,\vec{y} \vdash \chi\,\vec{y}, \varphi[\chi/x]\,\vec{y}, \Delta}\ (\mu\text{L})}{\dfrac{(\star)\ \Gamma, (\mu x.\varphi)\,\vec{y} \vdash \chi\,\vec{y}, \Delta}{\Gamma, (\mu x.\varphi)\,\vec{\psi} \vdash \chi\,\vec{\psi}, \Delta}\ (\text{Subst})}\ (\text{Cut})$$

(Post) can also be removed in the same manner.

We show that the resulting pre-proof Π' is a cyclic proof. For each infinite path π in Π', if some tail of π goes through only one cycle as the above one from $(\star)$ to $(\star)$ then we can trace the left μ in $\mu x.\varphi$ or the right ν in $\nu x.\varphi$. Otherwise, there exists a corresponding infinite path in Π, which satisfies the global trace condition. Therefore Π' is a cyclic proof of $\Gamma \vdash \Delta$. ◀

3.3 Examples

This subsection presents two examples of cyclic proofs.

▶ **Example 15** (Well-foundedness of a tree)**.** In this example, we write $\mathbf{N}^*$ for the type of finite sequences of natural numbers. We also use the concatenation operation $(-) \cdot (-)$. This $\mathbf{N}^*$ can be expressed by $\mathbf{N}$ and thus this additional type does not increase the expressivity.

A *tree* is a subset of finite sequences of natural numbers that represents the complement of the tree; the idea is to regard ϵ as the root and p as the parent of $p \cdot i$. Let $f^{\mathbf{N}^* \to \boldsymbol{\Omega}}$ be a term representing a tree. We define Φ and Ψ as follows:

$$\Phi := \mu w^{(\mathbf{N}^* \to \boldsymbol{\Omega}) \to \boldsymbol{\Omega}}.\lambda k^{\mathbf{N}^* \to \boldsymbol{\Omega}}.k\,\epsilon \vee \forall i^{\mathbf{N}}.w\,(\lambda z^{\mathbf{N}^*}.k\,(i \cdot z))$$
$$\Psi := \mu v^{\mathbf{N}^* \to \boldsymbol{\Omega}}.\lambda z^{\mathbf{N}^*}.f\,z \vee \forall i^{\mathbf{N}}.v\,(z \cdot i)$$

Then both $\Phi\,f$ and $\Psi\,\epsilon$ represent well-foundedness of f, i.e. whether there is no infinite path in f. Φ checks whether the tree k satisfies well-foundedness. This returns true if k has only one node or all the immediate children of the root satisfy well-foundedness. Ψ checks whether the subtree of f whose root node is z satisfies well-foundedness. This returns true if z is a leaf or all subtrees under z satisfy well-foundedness.

A cyclic proof of $\Phi\,f \vdash \Psi\,\epsilon$ is given as follows:

$$\dfrac{\dfrac{\dfrac{}{f\,l \vdash f\,l}\,(\text{Axiom}) \qquad \dfrac{\dfrac{(\dagger)\quad \Phi\,(\lambda z.f\,(l\cdot n\cdot z)) \vdash \Psi\,(l\cdot n) \qquad (\star)\quad Y_\Phi\,(\mathbf{S}n) \vdash Y_\Psi\,(\mathbf{S}n)}{\Phi\,(\lambda z.f\,(l\cdot n\cdot z)) \wedge Y_\Phi\,(\mathbf{S}n) \vdash \Psi\,(l\cdot n) \wedge Y_\Psi\,(\mathbf{S}n)}\,(\nu L, R) \qquad \dfrac{(\star)\quad Y_\Phi\,n \vdash Y_\Psi\,n}{Y_\Phi\,\mathbf{Z} \vdash Y_\Psi\,\mathbf{Z}}\,(\text{Subst})}{f\,l \vee \forall i.\Phi\,(\lambda z.f\,(l\cdot i\cdot z)) \vdash f\,l \vee \forall i.\Psi\,(l\cdot i)}\,(\vee L, \vee R, Wk)}{\dfrac{(\dagger)\quad \Phi\,(\lambda z.f\,(l\cdot z)) \vdash \Psi\,l}{\Phi\,f \vdash \Psi\,\epsilon}\,(\text{Subst})}\,(\mu L, R)$$

and $f \equiv \lambda z.f\,(\epsilon\cdot z)$

where $Y_\Phi \equiv (\nu Y.\lambda i.\Phi\,(\lambda z.f\,(l\cdot i\cdot z)) \wedge Y\,(\mathbf{S}i))$ and $Y_\Psi \equiv (\nu Y.\lambda i.\Psi\,(l\cdot i) \wedge Y\,(\mathbf{S}i))$. Here we omit (Subst) for open leaves; hence cycles of $(\star)$ and $(\dagger)$ are valid, although two nodes for each label have different sequents. Note that $Y_\Phi\,\mathbf{Z} \equiv \forall i.\Phi\,(\lambda z.f\,(l\cdot i\cdot z))$ and $Y_\Psi \equiv \forall i.\Psi\,(l\cdot i)$.

For all infinite paths, if the path includes $(\dagger) \to (\dagger)$ infinitely then we can trace the left μ in Φ and otherwise we can trace the right ν in $\forall i.\Psi\,(l\cdot i)$. ⌟

The example below demonstrates an application of our cyclic proof system to program verification, based on the reduction of Kobayashi et al. [10, 17] from program verification to $\mathrm{HFL}_{\mathbb{N}}$ validity checking.

▶ **Example 16** (Example 2.4 and 3.3 in [17])**.** Consider the following OCaml-like program.

```
let rec repeat f x =
  if x = 0 then () else if * then repeat f (f x) else repeat f (x-1)
in let y = input() in
  repeat (fun x -> x-y) n
```

In this program, `*` represents a non-deterministic Boolean value and `input()` means a user input. We aim to verify that an appropriate input `y` makes this program eventually terminate.

To verify this, we have to check $\vdash \mathrm{input}\,0\,(g\,n)$ where

$$\begin{aligned}
\mathrm{repeat} &:= \mu R.\lambda f.\lambda x.(x=0) \vee (\exists x'.x = x'+1 \wedge f\,x\,(R\,f) \wedge R\,f\,x')\\
\mathrm{sub} &:= \lambda y.\lambda x.\lambda k.k\,(x-y)\\
g &:= \lambda z.\lambda y.\mathrm{repeat}\,(\mathrm{sub}\,y)\,z\\
\mathrm{input} &:= \mu I.\lambda x.\lambda k.(k\,x) \vee (I\,(x+1)\,k)
\end{aligned}$$

where $(-)$ is defined naturally by using μ. Here, functions on integers of type $\mathbf{N} \to \mathbf{N}$ in the program have been turned into predicates of type $\mathbf{N} \to (\mathbf{N} \to \boldsymbol{\Omega}) \to \boldsymbol{\Omega}$, which are obtained by CPS translation; for example, the function `fun x->x-y` has been turned into $\mathrm{sub}\,y$ $(\equiv \lambda x.\lambda k.k(x-y))$. Note that $\mathrm{input}\,0\,(g\,n)$ is equivalent to $\exists y.g\,n\,y$, which models an angelic non-determinsm of `input()` in the program.

The goal sequent can be proved by the following cyclic proof:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash 0=0}\,(=\mathrm{R})}{\vdash \mathrm{repeat}\,(\mathrm{sub}\,1)\,0}\,(\mu\mathrm{R},\vee\mathrm{R})}{n=0 \vdash \mathrm{repeat}\,(\mathrm{sub}\,1)\,n}\,(=\mathrm{L}) \qquad \Pi}{(\star,\dagger)\ \mathbf{N}\,n \vdash \mathrm{repeat}\,(\mathrm{sub}\,1)\,n}\,(\mu\mathrm{L})}{\vdash g\,n\,1}\,(\lambda\mathrm{R},\ \mathrm{Nat})}{\vdash g\,n\,1, \mathrm{input}\,2\,(g\,n)}\,(\mathrm{Wk\ R})}{\vdash \mathrm{input}\,1\,(g\,n)}\,(\mu\mathrm{R},\vee\mathrm{R})}{\vdash g\,n\,0, \mathrm{input}\,1\,(g\,n)}\,(\mathrm{Wk\ R})}{\vdash \mathrm{input}\,0\,(g\,n)}\,(\mu\mathrm{R},\vee\mathrm{R})$$

$$\Pi := \dfrac{\dfrac{\dfrac{\dfrac{(\star)\ \mathbf{N}\,n' \vdash \mathrm{repeat}\,(\mathrm{sub}\,1)\,n'}{\mathbf{N}\,n' \vdash (\mathrm{sub}\,1)\,(n'+1)\,(\mathrm{repeat}\,(\mathrm{sub}\,1))}\,(\lambda\mathrm{R}) \qquad (\dagger)\ \mathbf{N}\,n' \vdash \mathrm{repeat}\,(\mathrm{sub}\,1)\,n'}{\mathbf{N}\,n' \vdash (\mathrm{sub}\,1)\,(n'+1)\,(\mathrm{repeat}\,(\mathrm{sub}\,1)) \wedge \mathrm{repeat}\,(\mathrm{sub}\,1)\,n'}\,(\wedge\mathrm{R})}{\mathbf{N}\,n' \vdash \mathrm{repeat}\,(\mathrm{sub}\,1)\,(n'+1)}\,(\mu\mathrm{R})}{\mathbf{N}\,n', n = n'+1 \vdash \mathrm{repeat}\,(\mathrm{sub}\,1)\,n}\,(=\mathrm{L})}{\exists n'.\mathbf{N}\,n' \wedge n = n'+1 \vdash \mathrm{repeat}\,(\mathrm{sub}\,1)\,n}\,(\exists\mathrm{L},\ \wedge\mathrm{L})$$

This satisfies the global trace condition because we can trace the left μ in $\mathbf{N}$ for every infinite path. ⌟

4 Decidability of the Global Trace Condition

For our cyclic proof system to be useful, there should exist an algorithm to check whether a given proof candidate is indeed a cyclic proof. It is easy to check whether a given candidate is a pre-proof, but it is non-trivial to check whether the pre-proof also satisfies the global trace condition. In this section we prove that the global condition is indeed decidable. We follow the approach in [3], which reduces the global trace condition of a pre-proof to Büchi automata containment. One automaton accepts all infinite paths of the pre-proof and the other accepts those that satisfy the global trace condition. Then whether the pre-proof is a cyclic proof corresponds to the inclusion between these automata.

Let us briefly recall the definition of Büchi automata. A *(nondeterministic) Büchi automaton* is a tuple $(Q, \Sigma, \delta, Q_0, F)$ where Q is a finite set of elements called *states*, Σ is a finite set of symbols, $\delta : Q \times \Sigma \times Q$ is a *transition relation*, $Q_0 \subseteq Q$ is a set of *initial states*, and $F \subseteq \delta$ is a set of accepting transition rules.[4] Given an infinite word $w = (a_i)_{i\geq 0} \in \Sigma^\omega$, a *run* over w is a sequence $(q_i)_{i\geq 0}$ of states such that $q_0 \in Q_0$ and $(q_i, a_i, q_{i+1}) \in \delta$ for all $i \geq 0$. This run is *accepting* if $(q_i, a_i, q_{i+1}) \in F$ for infinitely many i. The automaton *accepts* an infinite word if there is an accepting run over the word.

Let $(\mathcal{D}, \mathcal{R})$ be a given pre-proof. The alphabet Σ is the set of nodes of $\mathcal{D}$. Then an infinite path is represented as an infinite word, and it is easy to construct an automaton $\mathcal{A}_{path}$ that accepts all the paths of $\mathcal{D}$. We define an automaton $\mathcal{A}_{gtc}$ that checks if a given path satisfies the global trace condition.

The idea of the automaton is as follows. Recall Definition 8, in which we assign a sequence of natural numbers to each occurrence of a fixed-point operator. One cannot directly simulate the annotation process by an automaton since the set of sequences of natural numbers is infinite. However, at the end of Definition 8, we focus on an infinite sequence p of natural numbers, and sequences that are not prefixes of p can be safely ignored. Furthermore, it suffices to remember exactly one finite sequence by the following argument. Consider the case that $\tau_i \equiv \sigma_q x.\varphi$ where q is a prefix of p, and $\tau_{i+1} \equiv \varphi[\sigma_{q.k} x.\varphi/x]$.

- If $q.k$ is not a prefix of p, then we can safely forget the annotation $q.k$.
- If $q.k$ is a prefix of p, then we can safely forget the annotation q.

To see the latter, observe that other expansions of σ_q generate annotations $q.k'$ with $k' \neq k$ by freshness of k'. Hence $q.k'$ is not a prefix of p and thus we can forget it. So any extension of q that will be generated afterward can be safely ignored, and thus we can forget q itself.

The above argument motivates the following definition.

[4] This differs from the standard definition, in which the acceptance condition is specified by the set of accepting *states*. It is not difficult to see that this change does not affect the expressive power.

▶ **Definition 17.** *A* marked formula $\breve{\varphi}$ *is a formula in which some occurrences of fixed-point operators* σ *are* marked*; a marked fixed-point operator is written as* $\sigma_\bullet$*. We write* $|\breve{\varphi}|$ *for the (standard) formula obtained by removing marks. An* occurrence-with-marks $\breve{\tau}$ *is a pair* $(\tau, \breve{\varphi})$ *of an occurrence* τ *and a marked formula* $\breve{\varphi}$ *such that* $\tau \equiv |\breve{\varphi}|$*.*

We define $\mathcal{A}_{gtc}$. The set of states of $\mathcal{A}_{gtc}$ consists of occurrences-with-marks of $\mathcal{D}$ and a distinguished (initial) state $*$. Most rules just simulate Definition 8. For example, if τ is the principal occurrence in the conclusion of (λL), $(\lambda x.\breve{\varphi})\,\breve{\psi}$ is a marked formula such that $\tau \equiv (\lambda x.|\breve{\varphi}|)\,|\breve{\psi}|$ and n is the premise, then $((\tau, (\lambda x.\breve{\varphi})\,\breve{\psi}), n, (\tau', \breve{\varphi}[\breve{\psi}/x]))$ where τ' is the unique occurrence in v that is relevant to τ. Important transition rules are those dealing with $(\sigma L/R)$. Consider the state $\breve{\tau} = (\tau, \breve{\varphi})$ where τ is the principal occurrence of $(\sigma L/R)$. Let n be the premise and τ' be the unique relevant occurrence in n.

- Case $\breve{\varphi} \equiv (\sigma x.\breve{\psi})\,\breve{\vec{\chi}}$: The automaton just unfolds the fixed-point operator, i.e.,

  $$((\tau, (\sigma x.\breve{\psi})\,\breve{\vec{\chi}}),\ n,\ (\tau', \breve{\psi}[(\sigma x.\breve{\psi})/x]\,\breve{\vec{\chi}})) \in \delta.$$

- Case $\breve{\varphi} \equiv (\sigma_\bullet x.\breve{\psi})\,\breve{\vec{\chi}}$: Note that there may be other copies of $\sigma_\bullet x.\breve{\psi}$ in $\breve{\vec{\chi}}$ or $\breve{\psi}$. So the automaton has to choose which copy should be tracked, and the transition is non-deterministic. If the automaton decides to track the occurrence of $\sigma_\bullet x.\breve{\psi}$ being unfolded, it removes all marks in $\breve{\vec{\chi}}$ and tracks the recursive calls of the head occurrence by marking them:

  $$((\tau, (\sigma_\bullet x.\breve{\psi})\,\breve{\vec{\chi}}),\ n,\ (\tau', |\breve{\psi}|[(\sigma_\bullet x.|\breve{\psi}|)/x]\,|\breve{\vec{\chi}}|)) \in \delta.$$

  Otherwise it removes the mark of the fixed-point operator being unfolded and unfolds it:

  $$((\tau, (\sigma_\bullet x.\breve{\psi})\,\breve{\vec{\chi}}),\ n,\ (\tau', \breve{\psi}[(\sigma x.\breve{\psi})/x]\,\breve{\vec{\chi}})), \in \delta.$$

  If the rule is (μL) or (νR), then the former transition is an accepting transition. All other transitions for $(\sigma L/R)$ and other rules are not accepting.

The initial state either ignores the input ($(*, n, *) \in \delta$) or nondeterministically chooses an occurrence-with-marks $((*, n, \breve{\tau}) \in \delta$ if $\breve{\tau}$ is an occurrence in n and has exactly one marked fixed-point operator).

▶ **Lemma 18.** *Let* $(\mathcal{D}, \mathcal{R})$ *be a pre-proof. For every infinite path* π*,* $\pi \in \mathcal{L}(\mathcal{A}_{gtc})$ *if and only if* π *satisfies the global trace condition.*

Proof. Assume that π satisfies the global trace condition. Let $((\tau_i)_{i\geq 0}, p)$ be the pair of a trace (with annotations) and an infinite sequence of natural numbers that witnesses the global trace condition. For each i, let q_i be the longest prefix of p among the annotations in τ_i. Let us mark σ_{q_i} in τ_i and write $\breve{\varphi}_i$ for the resulting marked formula. Then $(\tau_i, \breve{\varphi}_i)_{i\geq 0}$ is an accepting run.

We prove the converse. Assume a (possibly non-accepting) run, which determines a trace $(\tau_i)_{i\geq 0}$. We annotate $(\tau_i)_{i\geq 0}$ following Definition 8. Let q_i be the sequence assigned to the marked operator in τ_i and p be the limit of $(q_i)_{i\geq 0}$. The transition rules ensure the well-definedness of q_i and p. If the run is accepting, p is infinite since $(q_i)_{i\geq 0}$ must grow infinitely many times. Furthermore it is a left μ-trace or right ν-trace as all accepting transitions are for (μL) or (νR). ◀

We have the following theorem as a corollary.

▶ **Theorem 19.** *The validity checking of a pre-proof* $(\mathcal{D}, \mathcal{R})$ *is decidable.*

5 Soundness

The soundness proof follows the proof strategy of [3]. We prove the claim by contraposition. Assume that the conclusion $\Gamma \vdash \Delta$ of a cyclic proof is invalid. Then there exists a valuation ρ such that $\Gamma \not\models_\rho \Delta$. Since all rules are *locally sound* (i.e. if the premises are valid under a valuation, the conclusion is also valid under the valuation), there exists an infinite path whose sequents are invalid under ρ. Using the global trace condition, we construct an infinite decreasing chain of ordinals, a contradiction.

To be precise, we impose on the infinite path a condition slightly stronger than the invalidity under ρ. To describe the condition, we need fixed-point operators $\mu^\alpha x.\varphi$ and $\nu^\alpha x.\varphi$ annotated by ordinals. The semantics of $\mu^\alpha x.\varphi$ is given by

$$[\![\mu^0 x^T.\varphi]\!](\rho) := \bot_T \quad \text{and} \quad [\![\mu^\alpha x^T.\varphi]\!](\rho) := \bigsqcup_{\beta<\alpha} [\![(\lambda x.\varphi)\,(\mu^\beta x^T.\varphi)]\!](\rho).$$

The definition of $\nu^\alpha x.\varphi$ is similar (but uses the dual operations).

During the construction of the path, we give ordinal annotations to occurrences μ on the left side of $\vdash$ and to occurrences of ν on the right side. The annotation processes are essentially the same as that in Definition 8. Again, the most important case is that τ_i is the principal occurrence of (μL) or (νR). Consider the (μL) case. Then $\tau_i \equiv (\mu^\alpha x.\varphi)\,\vec{\psi}$ (where μ in φ and $\vec{\psi}$ are also annotated). By the assumption of the invalidity of the sequent under ρ, we have $[\![(\mu^\alpha x.\varphi)\,\vec{\psi}]\!](\rho) = \top$. The annotation to τ_{i+1} is $\varphi[\mu^\beta x.\varphi/x]\,\vec{\psi}$ where β is the minimum ordinal such that $[\![\varphi[\mu^\beta x.\varphi]\,\vec{\psi}]\!](\rho) = \top$. By this process, fixed-point operators annotated with the same sequence of natural numbers have the same ordinal annotation. In other words, it defines a function from sequences q of natural numbers appearing in the trace to ordinal numbers α_q. Furthermore one can show that $\alpha_{q.k} < \alpha_q$ (provided that $q.k$ appears in the trace). Therefore, if the path has a left μ-trace witnessed by an infinite sequence p, then $\alpha_{p[0:1]} > \alpha_{p[0:2]} > \alpha_{p[0:3]} > \dots$ is an infinite decreasing chain of ordinals.

The above argument shows the following theorem. A detailed proof is in [11].

▶ **Theorem 20** (Soundness). *If there is a cyclic proof of $\Gamma \vdash \Delta$, then $\Gamma \vdash \Delta$ is valid.*

6 Completeness of Infinitary Variant

Our cyclic proof system is incomplete. Existence of a proof in our cyclic proof system is computably enumerable, but the valid sequents in $\mathrm{HFL}_{\mathbb{N}}$ are not because $\mathrm{HFL}_{\mathbb{N}}$ includes Peano arithmetic. The aim of this section is to find a complete proof system. Following [3], we study the infinitary variant of our cyclic proof system, in which a proof is an infinite tree instead of a finite tree with cycles. More precisely, an *infinitary proof* is a (possibly) infinite derivation tree (without open leaves) of which all infinite paths satisfy the global trace condition. Soundness proof of the previous section is applicable to the infinitary system as well. Here we discuss its completeness.

▶ **Theorem 21.** *Let $\Gamma \vdash \Delta$ be a sequent without higher-order free variables. That means, every free variable of the sequent is of type $\mathbf{N}$ or of type $\mathbf{N}^k \to \mathbf{\Omega}$ for some $k \geq 0$. If $\Gamma \models \Delta$, then $\Gamma \vdash \Delta$ is provable in the infinitary proof system.*

We give a sketch of the proof. A detailed proof is given in Appendix B. We can assume without loss of generality that the sequent has no free variable of type $\mathbf{N}$. We start from a (finite) pre-proof consisting only of the root node, which is an open leaf, and iteratively expand each open leaf by applying rules $(\vee L/R)$, $(\wedge L/R)$, $(\lambda L/R)$, and $(\sigma L/R)$. The only

restriction is that the expansion must be "fair": the fairness condition we impose is that (i) each open leaf will eventually be expanded, and (ii) for every path, each formula in a sequent will eventually appear in the principal position (unless the path is terminated by the axiom rule). This process generates a growing sequence of (finite) pre-proofs, and the infinitary proof is defined as its limit.

Now it suffices to show the global trace condition of the above constructed candidate proof, but this is the hardest part of the proof. Assume an infinite path π. Since $\Gamma \models \Delta$, for each valuation ρ, there exists $\varphi \in \Gamma$ such that $[\![\varphi]\!](\rho) = \bot$ or $\psi \in \Delta$ such that $[\![\varphi]\!](\rho) = \top$. Then, by a similar argument to the proof of soundness, existence of a left ν-trace or a right μ-trace leads to a contradiction. It is worth noting that the construction is "dual": whereas in the soundness proof we ensure $[\![\tau_i]\!](\rho) = \top$ for a left-trace $(\tau_i)_{i\geq 0}$, here $[\![\tau_i]\!](\rho) = \bot$ is kept. This difference allows us to construct a trace of the intended path π. By appropriately choosing ρ, one can ensure that the generated trace is infinite; hence we have constructed an infinite trace that is not left ν- nor right μ-trace. The proof is completed by the following lemma, which is technical but often used in the analysis of HFL (cf. [4, Lemma 6 & 7], [10, Lemma 26, Appendix E.2] and [15, Lemma 14]):

▶ **Lemma 22.** *Every infinite trace $(\tau_i)_{i\leq 0}$ is either a μ-trace or a ν-trace but not both.*

▶ Remark 23. We are not sure if the proof can be extended to sequents with higher-order free variables. The problem is the construction of ρ. In the current setting, for each free variable f of type $\mathbf{N}^k \to \mathbf{\Omega}$, the valuation ρ is defined for follows: for each k-tuple $\vec{m}$ of natural numbers, (i) if $f\,\vec{m}$ appears on the left side of a sequent in the path, then $\rho(f)(\vec{m}) := \top$; (ii) if $f\,\vec{m}$ appears on the right side of a sequent in the path, then $\rho(f)(\vec{m}) := \bot$; and (iii) otherwise the value of $\rho(f)(\vec{m})$ is arbitrary. The point is that the values of arguments of each fully-applied occurrence of f is canonically determined. This construction of ρ is no longer possible in the presence of a higher-order free variable, say $g : \mathbf{\Omega} \to \mathbf{\Omega}$. When $g\,(f\,\vec{m})$ appears on the left side, there are two assignments that make this formula true, namely $\rho_1(g)(\bot) = \top$ with $\rho_1(f)(\vec{m}) = \bot$ and $\rho_2(g)(\top) = \top$ with $\rho_2(f)(\vec{m}) = \top$. ⌟

7 Related Work

7.1 Cyclic Proof Systems

The idea of cyclic proof can at least be traced back to the work of Sprenger and Dam [13] on a cyclic proof system called S_{glob}, where proofs are explicitly annotated with ordinals. The ordinal annotations are not convenient for automated theorem proving.

Brotherston and Simpson [3] proposed a cyclic proof system called CLKID for a first-order predicate logic with inductive definitions, which does not require ordinal annotations. They have proved soundness of CLKID, and also shown that a proof system with explicit induction rules (called LKID) can be embedded into CLKID; we have shown analogous results in Theorem 20 and Proposition 14 for $\mathrm{HFL}_{\mathbb{N}}$. Doumane [7] formalized a cyclic proof system for the linear-time (propositional) μ-calculus, which features alternating fixed-points. Our global trace condition has been inspired by her trace condition.

Besides cyclic proof systems for ordinary first-order logics, cyclic proof systems have also been proposed for separation logics [1, 2, 14], and automated tools have been developed based on those cyclic proof systems.

7.2 Proof Systems and Games for HFL

HFL was originally proposed by Viswanathan and Viswanathan [16], and its extensions with arithmetic (such as $\mathrm{HFL}_{\mathbb{N}}$) have recently been drawing attention in the context of higher-order program verification [10, 17].

For pure HFL (without natural numbers), Kobayashi et al. [8] proposed a type-based inference system for proving the validity of HFL formulas.[5] It is left for future work to study the relationship between their type system and our cyclic proof system; it would be interesting to see whether their type system can be embedded in our cyclic proof system.

Tsukada [15] gave a game-based characterization of $\mathrm{HFL}_{\mathbb{N}}$. His game may be considered a special case of the infinitary version of our proof system, where formulas are restricted to those whose free variables have only natural number types. Burn et al. [5] proposed a refinement type system for HoCHC, which is closely related to the ν-only fragment of $\mathrm{HFL}_{\mathbb{N}}$. Their proof system is incomplete, and does not support fixed-point alternations.

8 Conclusion

We have proposed a cyclic proof system for $\mathrm{HFL}_{\mathbb{N}}$, a higher-order logic with natural numbers and alternating fixed-points, which we expect to be useful for higher-order program verification. Our proof system has been inspired by previous cyclic proof systems for first-order logics [3, 7]. We have shown the soundness of the proof system and the decidability of the global trace condition. We have also shown a restricted form of standard completeness for the infinitary version of our proof system.

Constructing a (semi-)automated tool based on our cyclic proof system is left for future work. On the theoretical side, we plan to study whether Henkin completeness and the cut elimination property hold for (possibly a variation of) our cyclic proof system.

References

1 James Brotherston, Dino Distefano, and Rasmus Lerchedahl Petersen. Automated cyclic entailment proofs in separation logic. In Nikolaj Bjørner and Viorica Sofronie-Stokkermans, editors, *Automated Deduction - CADE-23 - 23rd International Conference on Automated Deduction, Wroclaw, Poland, July 31 - August 5, 2011. Proceedings*, volume 6803 of *Lecture Notes in Computer Science*, pages 131–146. Springer, 2011. doi:10.1007/978-3-642-22438-6_12.

2 James Brotherston, Nikos Gorogiannis, and Rasmus Lerchedahl Petersen. A generic cyclic theorem prover. In Ranjit Jhala and Atsushi Igarashi, editors, *Programming Languages and Systems*, pages 350–367, Berlin, Heidelberg, 2012. Springer Berlin Heidelberg.

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

4 Florian Bruse. Alternating parity Krivine automata. In Erzsébet Csuhaj-Varjú, Martin Dietzfelbinger, and Zoltán Ésik, editors, *Mathematical Foundations of Computer Science 2014 - 39th International Symposium, MFCS 2014, Budapest, Hungary, August 25-29, 2014. Proceedings, Part I*, volume 8634 of *Lecture Notes in Computer Science*, pages 111–122. Springer, 2014. doi:10.1007/978-3-662-44522-8_10.

5 Toby Cathcart Burn, C.-H. Luke Ong, and Steven J. Ramsay. Higher-order constrained Horn clauses for verification. *PACMPL*, 2(POPL):11:1–11:28, 2018. doi:10.1145/3158099.

[5] Actually, they formalized a type system for model checking, which can also be used as a proof system for proving validity of $\mathrm{HFL}_{\mathbb{N}}$ formulas without natural numbers.

6 Patrick Cousot and Radhia Cousot. Constructive versions of Tarski's fixed point theorems. *Pacific J. Math.*, 82(1):43–57, 1979. URL: https://projecteuclid.org:443/euclid.pjm/1102785059.

7 Amina Doumane. *On the infinitary proof theory of logics with fixed points*. Theses, Université Sorbonne Paris Cité, June 2017.

8 Naoki Kobayashi, Étienne Lozes, and Florian Bruse. On the relationship between higher-order recursion schemes and higher-order fixpoint logic. *ACM SIGPLAN Notices*, 52(1):246–259, 2017.

9 Naoki Kobayashi, Takeshi Nishikawa, Atsushi Igarashi, and Hiroshi Unno. Temporal verification of programs via first-order fixpoint logic. In *International Static Analysis Symposium*, pages 413–436. Springer, 2019.

10 Naoki Kobayashi, Takeshi Tsukada, and Keiichi Watanabe. Higher-order program verification via HFL model checking. In *European Symposium on Programming*, pages 711–738. Springer, 2018.

11 Mayuko Kori, Takeshi Tsukada, and Naoki Kobayashi. A cyclic proof system for $\mathrm{HFL}_{\mathbf{N}}$. *CoRR*, 2020. A longer version. arXiv:2010.14891.

12 Étienne Lozes. A type-directed negation elimination. In Ralph Matthes and Matteo Mio, editors, *Proceedings Tenth International Workshop on Fixed Points in Computer Science, FICS 2015, Berlin, Germany, September 11-12, 2015*, volume 191 of *EPTCS*, pages 132–142, 2015. doi:10.4204/EPTCS.191.12.

13 Christoph Sprenger and Mads Dam. On the structure of inductive reasoning: Circular and tree-shaped proofs in the μ-calculus. In *Foundations of Software Science and Computation Structures*, pages 425–440. Springer Berlin Heidelberg, 2003.

14 Gadi Tellez and James Brotherston. Automatically verifying temporal properties of pointer programs with cyclic proof. *J. Autom. Reasoning*, 64(3):555–578, 2020. doi:10.1007/s10817-019-09532-0.

15 Takeshi Tsukada. On computability of logical approaches to branching-time property verification of programs. In *Proceedings of the 35th Annual ACM/IEEE Symposium on Logic in Computer Science*, LICS '20, page 886–899, New York, NY, USA, 2020. Association for Computing Machinery. doi:10.1145/3373718.3394766.

16 Mahesh Viswanathan and Ramesh Viswanathan. A higher order modal fixed point logic. In *International Conference on Concurrency Theory*, pages 512–528. Springer, 2004.

17 Keiichi Watanabe, Takeshi Tsukada, Hiroki Oshikawa, and Naoki Kobayashi. Reduction from branching-time property verification of higher-order programs to HFL validity checking. In *Proceedings of the 2019 ACM SIGPLAN Workshop on Partial Evaluation and Program Manipulation*, PEPM 2019, page 22–34, New York, NY, USA, 2019. Association for Computing Machinery. doi:10.1145/3294032.3294077.

A Definition of relevant occurrences

Here we give a detailed definition of the notion of *relevant occurrences* introduced after Definition 7. We have already defined relevant occurrences for the rules $(\wedge L)$, $(\vee L)$, and (Mono). We give the definition for the other rules. Below, n_i refers to the conclusion of each rule, and n_{i+1} refers to one of the premises. In all the rules, each formula in Γ or Δ in n_{i+1} is a relevant occurrence of the corresponding formula in Γ or Δ in n_i.

- In (Cut), φ in n_{i+1} is not relevant to any formula in n_i.
- In (Ctr L) and (Ctl R), both occurrences of φ are relevant to φ in n_i.
- In (Ex L) and (Ex R), ψ (φ, resp.) in n_{i+1} is a relevant occurrence of ψ (φ, resp.) in n_i.
- In (Subst), each formula ψ in Γ (Δ, resp.) of n_{i+1} is relevant to $\psi[\varphi/x]$ in $\Gamma[\varphi/x]$ ($\Delta[\varphi/x]$, resp.) of n_i.

- In (=L), each formula $\psi[t/x, s/y]$ in $\Gamma[t/x, s/y]$ ($\Delta[t/x, s/y]$, resp.) of n_{i+1} is relevant to $\psi[s/x, t/y]$ in $\Gamma[s/x, t/y]$ ($\Delta[s/x, t/y]$, resp.) of n_i.
- In ($\vee$R), φ and ψ in n_{i+1} are relevant to $\varphi \vee \psi$ in n_i.
- In ($\wedge$R), φ and ψ in n_{i+1} are relevant to $\varphi \wedge \psi$ in n_i.
- In (λL) and (λR), $\varphi[\psi/x]\,\vec{\psi}$ in n_{i+1} is relevant to $(\lambda x.\varphi)\,\psi\,\vec{\psi}$ in n_i.
- In (σL) and (σR), $\varphi[\sigma x.\varphi/x]\,\vec{\psi}$ in n_{i+1} is relevant to $(\sigma x.\varphi)\,\vec{\psi}$ in n_i.
- In (Nat), $N\,x$ in n_{i+1} is not relevant to any formula in n_i.
- In (P2), $s = t$ in n_{i+1} is relevant to $\mathbf{S}s = \mathbf{S}t$ in n_i.

B Proof of Theorem 21

Let f be a function on a complete lattice. For each ordinal α, we define $f^\alpha(\bot)$ by $f^0(\bot) = \bot$ and $f^\alpha(\bot) = \bigsqcup_{\beta<\alpha} f(f^\beta(\bot))$. Similarly $f^\alpha(\top)$ is defined by $f^0(\top) = \top$ and $f^\alpha(\top) = \bigsqcap_{\beta<\alpha} f(f^\beta(\top))$.

▶ **Lemma 24** (Cousot-Cousot [6]). *Let $(C, \leq)$ be a complete lattice, f be a monotone function and γ be an ordinal number greater than the cardinality of C. Then $f^\gamma(\bot)$ is the least fixed-point of f and $f^\gamma(\top)$ is the greatest fixed-point of f.*

We extend the definition of formulas by allowing μ and ν to have ordinal numbers like μ^α, ν^α. The definition of $[\![.]\!]$ for μ^α, ν^α is as follows.

$$[\![\mathcal{H} \vdash (\mu^\alpha x^T.\varphi)\,\vec{\psi}]\!](\rho) = ((\lambda v.[\![\mathcal{H}, x : T \vdash \varphi]\!](\rho[x \mapsto v]))^\alpha\,(\bot_n))\,\rho(\vec{\psi})$$
$$[\![\mathcal{H} \vdash (\nu^\alpha x^T.\varphi)\,\vec{\psi}]\!](\rho) = ((\lambda v.[\![\mathcal{H}, x : T \vdash \varphi]\!](\rho[x \mapsto v]))^\alpha\,(\top_n))\,\rho(\vec{\psi})$$

▶ **Lemma 25.** *Let $\Gamma \vdash \Delta$ be a sequent and $x^{\mathbf{N}} \in FV(\Gamma, \Delta)$. If $(\Gamma[\mathbf{S}^n\mathbf{Z}/x] \vdash \Delta[\mathbf{S}^n\mathbf{Z}/x])$ is cut-free provable for all $n \in \mathbb{N}$ then $\Gamma \vdash \Delta$ is cut-free provable.*

Proof. Assume $(\Gamma[\mathbf{S}^n\mathbf{Z}/x] \vdash \Delta[\mathbf{S}^n\mathbf{Z}/x])$ is cut-free provable for all $n \in \mathbb{N}$, and let Π_n be the proof. We define $(\Pi^i)_{i\geq 0}$ recursively whose root node is $(N\,x, \Gamma[\mathbf{S}^i x/x] \vdash \Delta[\mathbf{S}^i x/x])$ as follows:

$$\Pi^i := \dfrac{\dfrac{\dfrac{\Pi_i}{\Gamma[\mathbf{S}^i\mathbf{Z}/x] \vdash \Delta[\mathbf{S}^i\mathbf{Z}/x]}\,(=\text{L})}{x = \mathbf{Z}, \Gamma[\mathbf{S}^i x/x] \vdash \Delta[\mathbf{S}^i x/x]} \qquad \dfrac{\dfrac{\dfrac{\Pi^{i+1}}{N\,x, \Gamma[\mathbf{S}^{i+1}x/x] \vdash \Delta[\mathbf{S}^{i+1}x/x]}\,(\text{Subst})}{N\,x', \Gamma[\mathbf{S}^i\mathbf{S}x'/x] \vdash \Delta[\mathbf{S}^i\mathbf{S}x'/x]}\,(=\text{L})}{x = \mathbf{S}x', N\,x', \Gamma[\mathbf{S}^i x/x] \vdash \Delta[\mathbf{S}^i x/x]}\,(\exists\text{L},\ \wedge\text{L})}{\exists x'.x = \mathbf{S}x' \wedge N\,x', \Gamma[\mathbf{S}^i x/x] \vdash \Delta[\mathbf{S}^i x/x]}\,(\mu\text{L})$$

Then $\Pi := \dfrac{\dfrac{\Pi^0}{N\,x, \Gamma \vdash \Delta}\,(\text{Nat})}{\Gamma \vdash \Delta}$ is a pre-proof of $\Gamma \vdash \Delta$. For all infinite paths π in the pre-proof Π, if π goes through some Π_i then there exists left μ-trace or right ν-trace because Π_i is a proof, and otherwise, we can trace the left μ in N. Therefore, Π is a proof of $\Gamma \vdash \Delta$. ◀

▶ **Corollary 26.** *Let $\Gamma \vdash \Delta$ be a sequent and $\vec{x}$ are all natural number free variables in $\Gamma \cup \Delta$. If $(\Gamma[\vec{n}/\vec{x}] \vdash \Delta[\vec{n}/\vec{x}])$ is cut-free provable for all $\vec{n} \in \vec{\mathbb{N}}$ then $\Gamma \vdash \Delta$ is cut-free provable.*

Thanks to this corollary, it suffices to prove completeness of valid sequents whose free variables have type $\mathbf{N}^k \to \mathbf{\Omega}$ for some $k \geq 0$. In other words, we can assume without loss of generality that the sequent of interest has no free variable of type $\mathbf{N}$.

The next two definitions construct a tree T_ω of $\Gamma \vdash \Delta$ without natural number free variables. The first definition is needed to deal with all formulas in a fair manner.

▶ **Definition 27** (Schedule). A schedule element *is a formula of the form* $\varphi \vee \psi, \varphi \wedge \psi, (\lambda x.\varphi)\,\psi\,\vec{\psi}, (\sigma x.\varphi)\,\vec{\psi}$. *We call* $(E_i)_{i\geq 0}$ a schedule *if* E_i *is a schedule element for all* i *and every schedule element appears infinitely often in* $(E_i)_{i\geq 0}$.

There exists a schedule and we fix one.

▶ **Definition 28** (T_ω). *Let* $\Gamma \vdash \Delta$ *be a valid sequent that does not have natural number or higher-order free variables. That is, the type of each free variable in* $\Gamma \vdash \Delta$ *is* $\mathbf{N}^n \to \mathbf{\Omega}$ *for some* $n \in \mathbb{N}$.

Then we will define trees $(T_i)_{i\geq 0}$ *whose roots are* $\Gamma \vdash \Delta$ *inductively by using a schedule* $(E_i)_{i\geq 0}$. *First,* T_0 *is defined by* $T_0 := \Gamma \vdash \Delta$.

Assume T_i *is already defined. Then we define* T_{i+1} *in* T_i *by replacing each open leaf* $\Gamma' \vdash \Delta'$ *with the following tree:*

- *If there exists a formula* $\varphi \in \Gamma' \cap \Delta'$*:*
$$\dfrac{\dfrac{}{\varphi \vdash \varphi}\,(Axiom)}{\Gamma' \vdash \Delta'}\,(Wk)$$
- *If* $(\mathbf{S}^n\mathbf{Z} = \mathbf{S}^m\mathbf{Z}) \in \Gamma'$ *for some different natural numbers* n, m*:*
$$\dfrac{\dfrac{\dfrac{}{\mathbf{Z} = \mathbf{S}^{|n-m|}\mathbf{Z} \vdash}\,(P1)}{\mathbf{S}^n\mathbf{Z} = \mathbf{S}^m\mathbf{Z} \vdash}\,(P2)}{\Gamma' \vdash \Delta'}\,(Wk)$$
- *If* $(\mathbf{S}^n\mathbf{Z} = \mathbf{S}^n\mathbf{Z}) \in \Delta'$ *for some* n*:*
$$\dfrac{\dfrac{}{\vdash \mathbf{S}^n\mathbf{Z} = \mathbf{S}^n\mathbf{Z}}\,(=R)}{\Gamma' \vdash \Delta'}\,(Wk)$$
- *Otherwise: This replacement is performed in such a way that each formula is chosen as the target of expansion in a fair manner, so that every formula is expanded at some point.*
  - *Case* $(E_i \equiv \varphi \vee \psi)$*:*
    - *if* $E_i \in \Gamma'$*:*
$$\dfrac{\dfrac{\Gamma', \varphi \vdash \Delta' \qquad \Gamma', \psi \vdash \Delta'}{\Gamma', \varphi \vee \psi \vdash \Delta'}\,(\vee L)}{\Gamma' \vdash \Delta'}\,(Ctr)$$
    - *if* $E_i \in \Delta'$*:*
$$\dfrac{\dfrac{\Gamma' \vdash \varphi, \psi, \Delta'}{\Gamma' \vdash \varphi \vee \psi, \Delta'}\,(\vee R)}{\Gamma' \vdash \Delta'}\,(Ctr)$$
  - *Case* $(E_i \equiv \varphi \wedge \psi)$*: The tree is defined in the similar way to the case* $E_i \equiv \varphi \vee \psi$.
  - *Case* $(E_i \equiv (\lambda x.\varphi)\,\psi\,\vec{\psi})$*:*
    - *if* $E_i \in \Gamma'$*:*
$$\dfrac{\dfrac{\Gamma', \varphi[\psi/x]\,\vec{\psi} \vdash \Delta'}{\Gamma', (\lambda x.\varphi)\,\psi\,\vec{\psi} \vdash \Delta'}\,(\lambda L)}{\Gamma' \vdash \Delta'}\,(Ctr)$$
    - *if* $E_i \in \Delta'$*:*
$$\dfrac{\dfrac{\Gamma' \vdash \varphi[\psi/x]\,\vec{\psi}, \Delta'}{\Gamma' \vdash (\lambda x.\varphi)\,\psi\,\vec{\psi}, \Delta'}\,(\lambda R)}{\Gamma' \vdash \Delta'}\,(Ctr)$$
  - *Case* $(E_i \equiv (\sigma x.\varphi)\,\vec{\psi})$*:*

* *if* $E_i \in \Gamma'$*:*

$$\dfrac{\dfrac{\Gamma', \varphi[\sigma x.\varphi/x]\,\vec{\psi} \vdash \Delta'}{\Gamma', (\sigma x.\varphi)\,\vec{\psi} \vdash \Delta'}\,(\sigma L)}{\Gamma' \vdash \Delta'}\,(Ctr)$$

* *if* $E_i \in \Delta'$*:*

$$\dfrac{\dfrac{\Gamma' \vdash \varphi[\sigma x.\varphi/x]\,\vec{\psi}, \Delta'}{\Gamma' \vdash (\sigma x.\varphi)\,\vec{\psi}, \Delta'}\,(\sigma R)}{\Gamma' \vdash \Delta'}\,(Ctr)$$

Note that for all $i \geq 0$, $T_i \subseteq T_{i+1}$ *and each sequent of an open leaf in* T_{i+1} *includes the sequent of the corresponding leaf in* T_i*. We define* T_ω *to be* $\lim_{i\to\infty} T_i$*.*

We aim to prove that T_ω is a proof of $\Gamma \vdash \Delta$.

▶ **Definition 29** ($\Gamma_\omega \vdash_{n/\pi} \Delta_\omega, \rho_\omega$)**.** *For all open leaves* n *of* T_ω*, we define* $\Gamma_\omega \vdash_n \Delta_\omega$ *as the leaf sequent. For all infinite paths* $\pi = (\pi_i)_{i\geq 0}$ *in* T_ω*, we define* $\Gamma_\omega \vdash_\pi \Delta_\omega$ *as* $\lim_{i\to\infty}(\Gamma_i \vdash \Delta_i)$ *where* $\Gamma_i \vdash \Delta_i$ *is the sequent of* π_i*.*

A valuation ρ_ω *of* $\Gamma_\omega \vdash_{n/\pi} \Delta_\omega$ *is defined as below:*

$$\rho_\omega(x^{\mathbf{\Omega}}) := \begin{cases} \top & \text{if } x \in \Gamma_\omega \\ \bot & \text{otherwise} \end{cases}$$

$$\rho_\omega(f^{\mathbf{N}^n \to \mathbf{\Omega}}) := \lambda \vec{x}^{\mathbf{N}^n}. \begin{cases} \top & \text{if } f\,\vec{x} \in \Gamma_\omega \\ \bot & \text{otherwise} \end{cases}$$

▶ **Lemma 30.** *T_ω is a proof.*

Proof. We aim to show that T_ω is a pre-proof and T_ω satisfies the global trace condition.

Assume T_ω is not a pre-proof. There exists an open leaf n in T_ω, and all formulas of the leaf are of the form $x^{\mathbf{\Omega}}$, $f\,\vec{t}$ or $\mathbf{S}^m\mathbf{Z} = \mathbf{S}^n\mathbf{Z}$. Then the definition of ρ_ω of $\Gamma_\omega \vdash_n \Delta_\omega$ induces $[\![\varphi]\!]_{\rho_\omega} = \top$ for all $\varphi \in \Gamma_\omega$ and $[\![\varphi]\!]_{\rho_\omega} = \bot$ for all $\varphi \in \Delta_\omega$. This contradicts to $\Gamma \models \Delta$ because $\Gamma \subseteq \Gamma_\omega$ and $\Delta \subseteq \Delta_\omega$. Therefore T_ω is a pre-proof.

We next show that T_ω satisfies the global trace condition. For all infinite paths π in T_ω, we define ρ_ω of $\Gamma_\omega \vdash_\pi \Delta_\omega$ as Definition 29. We have $\Gamma \models \Delta$, $\Gamma \subseteq \Gamma_\omega$ and $\Delta \subseteq \Delta_\omega$ by the assumption and the construction of T_ω. Therefore, it follows that there exists either (1) a formula $\varphi \in \Gamma_\omega$ such that $[\![\varphi]\!]_{\rho_\omega} = \bot$ or (2) a formula $\varphi \in \Delta_\omega$ such that $[\![\varphi]\!]_{\rho_\omega} = \top$. We now give the proof only for the case (1). The other case can be also proved by the same method.

Let j be a number such that $\varphi \in \pi_j$. We will show that there exists a left μ-trace $(\tau_i)_{i\geq j}$ in π starting from φ.

We define $(\tau_i)_{i\geq j}$ and $(\tau'_i)_{i\geq j}$ which satisfy the following conditions:

1. $\tau_i \in \Gamma_i$;
2. we can get τ_i by forgetting ordinal numbers of τ'_i;
3. each ν in τ'_i has an ordinal number;
4. $[\![\tau'_i]\!]_{\rho_\omega} = \bot$;
5. each ordinal number in τ'_{i+1} is less than or equal to the corresponding ordinal number in τ'_i.

The definition of $(\tau_i)_{i\geq j}$ and $(\tau'_i)_{i\geq j}$ as below:

- $\tau_j := \varphi$. Because $[\![\varphi]\!]_{\rho_\omega} = \bot$, we can get φ' which satisfies $[\![\varphi']\!]_{\rho_\omega} = \bot$ by assigning ordinal numbers to all ν in φ.
- Assume τ_i and τ'_i are defined. Below, ψ' means the corresponding subformula of τ'_i for each subformula ψ of τ_i.

- If $\tau_i \equiv x$:
  Since $x \in \Gamma_i$, $[\![x]\!]_{\rho_\omega} = \top$ because of the definition of ρ_ω. By the assumption of τ'_i, $[\![(x)']\!]_{\rho_\omega} = [\![x]\!]_{\rho_\omega} = \bot$. This is a contradiction.
- If $\tau_i \equiv (\mathbf{S}^n\mathbf{Z} = \mathbf{S}^m\mathbf{Z})$:
  By the assumption of τ'_i, $[\![(\mathbf{S}^n\mathbf{Z} = \mathbf{S}^m\mathbf{Z})']\!]_{\rho_\omega} = [\![\mathbf{S}^n\mathbf{Z} = \mathbf{S}^m\mathbf{Z}]\!]_{\rho_\omega} = \bot$. Then $n \neq m$ holds and π is not infinite because of the construction of T_ω. This contradicts the fact that π is an infinite path.
- If $\tau_i \equiv f\,\vec{t}$:
  Since $f\,\vec{t} \in \Gamma_i$, $[\![f\,\vec{t}]\!]_{\rho_\omega} = \top$ because of the definition of ρ_ω. By the assumption of τ'_i, $[\![(f\,\vec{t})']\!]_{\rho_\omega} = [\![f\,\vec{t}]\!]_{\rho_\omega} = \bot$. This is a contradiction.
- If $E_i \not\equiv \tau_i$ then $\tau_{i+1} := \tau_i$ and $\tau'_{i+1} := \tau'_i$.
- Otherwise:
  * Case $\tau_i \equiv (\psi \vee \chi)$:
    Since $[\![\psi' \vee \chi']\!]_{\rho_\omega} = \bot$, $[\![\psi']\!]_{\rho_\omega} = [\![\chi']\!]_{\rho_\omega} = \bot$. $\tau_{i+1} := \psi, \tau'_{i+1} := \psi'$ if π goes to the left leaf, and otherwise $\tau_{i+1} := \chi, \tau'_{i+1} := \chi'$. In each case, $[\![\tau'_{i+1}]\!]_{\rho_\omega} = \bot$ holds.
  * Case $\tau_i \equiv (\psi \wedge \chi)$:
    Since $[\![\psi' \wedge \chi']\!]_{\rho_\omega} = \bot$, either $[\![\psi']\!]_{\rho_\omega}$ or $[\![\chi']\!]_{\rho_\omega}$ is $\bot$.
    $\tau_{i+1} := \psi, \tau'_{i+1} := \psi'$ if $[\![\psi']\!]_{\rho_\omega} = \bot$, and otherwise $\tau_{i+1} := \chi, \tau'_{i+1} := \chi'$.
  * Case $\tau_i \equiv (\lambda x.\psi)\,\chi\,\vec{\theta}$:
    $\tau_{i+1} := \psi[\chi/x]\,\vec{\theta}, \tau'_{i+1} := \psi'[\chi'/x]\,\vec{\theta'}$
    then $[\![\psi'[\chi'/x]\,\vec{\theta'}]\!]_{\rho_\omega} = [\![(\lambda x.\psi')\,\chi'\,\vec{\theta'}]\!]_{\rho_\omega} = \bot$.
  * Case $\tau_i \equiv (\mu x.\psi)\,\vec{\theta}$:
    $\tau_{i+1} := \psi[\mu x.\psi/x]\,\vec{\theta}$, $\tau'_{i+1} := \psi'[\mu x.\psi'/x]\,\vec{\theta'}$
    then $[\![(\psi'[\mu x.\psi'/x])\,\vec{\theta'}]\!]_{\rho_\omega} = [\![(\mu x.\psi')\,\vec{\theta'}]\!]_{\rho_\omega} = \bot$.
  * Case $\tau_i \equiv (\nu x.\psi)\,\vec{\theta}$:
    Let α be the ordinal number assigned to the head ν of $(\nu x.\varphi)\,\vec{\psi}$. If $\alpha = 0$ then $[\![(\nu^0 x^T.\varphi')\,\vec{\psi'}]\!]_{\rho_\omega} = [\![(\top_T)\,\vec{\psi'}]\!]_{\rho_\omega} = \top$ and this contradicts the assumption.

$$\begin{aligned}[\![(\nu^\alpha x.\psi')\,\vec{\theta'}]\!]_{\rho_\omega} &= ((\lambda v.[\![\psi']\!]_{\rho_\omega[x\mapsto v]})^\alpha\,(\top_n))\,[\![\vec{\theta'}]\!]_{\rho_\omega}\\ &= (\prod_{\beta<\alpha}(\lambda v.[\![\psi']\!]_{\rho_\omega[x\mapsto v]})\,((\lambda v.[\![\psi']\!]_{\rho_\omega[x\mapsto v]})^\beta\,(\top_n)))\,[\![\vec{\theta'}]\!]_{\rho_\omega}\\ &= \bot\end{aligned}$$

There exists $\beta < \alpha$ such that $((\lambda v.[\![\psi']\!]_{\rho_\omega[x\mapsto v]})\,((\lambda v.[\![\psi']\!]_{\rho_\omega[x\mapsto v]})^\beta\,(\top_n)))\,[\![\vec{\theta'}]\!]_{\rho_\omega} = \bot$.
$\tau_{i+1} := \psi[\nu x.\psi/x]\,\vec{\theta}$, $\tau'_{i+1} := \psi'[\nu^\beta x.\psi'/x]\,\vec{\theta'}$ then

$$\begin{aligned}[\![(\psi'[\nu^\beta x.\psi'/x])\,\vec{\theta'}]\!]_{\rho_\omega} &= ((\lambda v.[\![\psi']\!]_{\rho_\omega[x\mapsto v]})\,([\![\nu^\beta x.\psi']\!]_{\rho_\omega}))\,[\![\vec{\theta'}]\!]_{\rho_\omega}\\ &= ((\lambda v.[\![\psi']\!]_{\rho_\omega[x\mapsto v]})\,((\lambda v.[\![\psi']\!]_{\rho_\omega[x\mapsto v]})^\beta\,(\top_n)))\,[\![\vec{\theta'}]\!]_{\rho_\omega}\\ &= \bot\end{aligned}$$

If this $(\tau_i)_{i\geq j}$ is a ν-trace, there is an infinite sequence p of the trace by the definition of ν-trace. For all $n \geq 1$, there is a $\nu_{p[0:n]}$ in some τ_i and we denote by α_n the ordinal number assigned to this ν in τ'_i. By the definition of $(\tau'_i)_{i\geq j}$, all ordinal numbers assigned to $\nu_{p[0:n]}$ is equal to α_n. Then for all $n \geq 1$, there exists i such that $(\tau_i \equiv (\nu_{p[0:n]}x.\psi)\,\vec{\theta}) \to (\tau_{i+1} \equiv (\psi[\nu_{p[0:n+1]}x.\psi/x])\,\vec{\theta})$. Hence $\alpha_n > \alpha_{n+1}$ holds for all $n \geq 1$ so $(\alpha_n)_{n\geq 1}$ is a decreasing sequence of ordinal numbers. However, such a sequence does not exist, hence a contradiction. By Lemma 22, it follows that $(\tau_i)_{i\geq j}$ is a left μ-trace of π.

Therefore T_ω satisfies the global trace condition. ◀

Proof (Theorem 21). Let $\vec{x}$ be all natural number free variables of $\Gamma \vdash \Delta$. By Corollary 26, it suffices to show that $\Gamma[\vec{n}/\vec{x}] \vdash \Delta[\vec{n}/\vec{x}]$ is cut-free provable for all $\vec{n} \in \vec{\mathbb{N}}$. For each $\vec{n} \in \vec{\mathbb{N}}$, we construct T_ω for $\Gamma[\vec{n}/\vec{x}] \vdash \Delta[\vec{n}/\vec{x}]$ as Definition 28, then Lemma 30 concludes that T_ω is a proof. Besides, T_ω is cut-free by the construction of T_ω. ◀