On Transforming Rewriting-Induction Proofs for Logical-Connective-Free Sequents into Cyclic Proofs

Shujun Zhang and Naoki Nishida

August 12, 2022

On Transforming Rewriting-Induction Proofs for Logical-Connective-Free Sequents into Cyclic Proofs*

Shujun Zhang
Nagoya University
Nagoya, Japan
shujun@trs.css.i.nagoya-u.ac.jp

Naoki Nishida
Nagoya University
Nagoya Japan
nishida@i.nagoya-u.ac.jp

A GSC-terminating and orthogonal inductive definition set (IDS, for short) of first-order predicate logic can be transformed into a many-sorted term rewrite system (TRS, for short) such that a quantifier-free sequent is valid w.r.t. the IDS if and only if a term equation representing the sequent is an inductive theorem of the TRS. Under certain assumptions, it has been shown that a quantifier- and cut-free cyclic proof of a sequent can be transformed into a rewriting-induction (RI, for short) proof of the corresponding term equation. The RI proof can be transformed back into the cyclic proof, but it is not known how to transform arbitrary RI proofs into cyclic proofs. In this paper, we show that for a quantifier- and logical-connective-free sequent, if there exists an RI proof of the corresponding term equation, then there exists a cyclic proof of some sequent w.r.t. some IDS such that the cyclic proof ensures the validity of the initial sequent. To this end, we show a transformation of the RI proof into such a cyclic proof, a sequent, and an IDS.

1 Introduction

Inductive theorem proving is well investigated in functional programming and term rewriting. In the field of term rewriting, *rewriting induction* [11] (RI, for short) is one of the most powerful principles to prove equations to be *inductive theorems*. Here, an equation $s \approx t$ of terms is an inductive theorem of a given (many-sorted) term rewrite system (TRS, for short) $\mathcal{R}$ if the equation is valid w.r.t. the reduction of $\mathcal{R}$, i.e., $s\theta \leftrightarrow^*_{\mathcal{R}} t\theta$ for any ground substitution θ such that $s\theta, t\theta$ are ground. RI has been extended to several kinds of rewrite systems, e.g., logically constrained term rewrite systems [8] (LCTRS, for short) that are models of not only functional but also imperative programs [7].

RI consists of inference rules that are applied to RI processes $(\mathcal{E}, \mathcal{H})$, where $\mathcal{E}$ is a finite set of term equations and $\mathcal{H}$ is a TRS. The application of rewrite rules in $\mathcal{H}$ corresponds to the application of induction hypotheses to subsequent RI processes. Given a TRS $\mathcal{R}$ and a finite set $\mathcal{E}_0$ of term equations, we start with the initial RI process $(\mathcal{E}_0, \emptyset)$, and succeed in proving all equations in $\mathcal{E}_0$ to be inductive theorems of $\mathcal{R}$ if we find an *RI proof* of $\mathcal{E}_0$, which is a sequence of RI processes obtained by the application of RI inference rules and starts with $(\mathcal{E}_0, \emptyset)$ to $(\emptyset, \mathcal{H}')$ for some TRS $\mathcal{H}'$.

A *cyclic proof system* [5] is a proof system in sequent-calculus style for first-order logics with *inductive predicates* which are defined by *inductive definition sets* (IDSs, for short), a set of productions. In contrast to structural proofs which are (possibly infinite) derivation trees, cyclic proofs are finite derivation trees with back-links from *bud* nodes to inner nodes called *companions*. Such back-links allow explicit induction rules, making trees finite. For the last decade, cyclic proof systems have been well investigated for several logics, e.g., *separation logic* [12].

RI and cyclic proof systems seem very similar because they have similar inference rules: Case analysis, the application of rules in given systems, generalization, and so on. For this reason, it is worth comparing them even regarding terminating systems to study their differences from several viewpoints.

[*] This work was partially supported by JSPS KAKENHI Grant Number 18K11160.

Submitted to:
WPTE 2022

For the comparison, we have prepared a common setting for IDSs and TRSs [18, 20]: A (A1) *GSC-terminating* and (A2) *orthogonal* IDS Φ such that

(A3) there is no *ordinary* predicate in Φ, and

(A4) any variable in $A_1,\dots,A_m$ appears in A for any production $\frac{A_1 \ \cdots \ A_m}{A} \in \Phi$,

can be transformed into a GSC-terminating confluent quasi-reductive constructor TRS $\mathcal{R}_\Phi$ such that a quantifier-free sequent $\Gamma \vdash \Delta$ is valid w.r.t. Φ if and only if the corresponding equation $\mathsf{seq}(\widehat{\Gamma},\widecheck{\Delta}) \approx \top$ is an inductive theorem of $\mathcal{R}_\Phi \cup \mathcal{R}_{seq}$. Here, $\widehat{\Gamma}$ and $\widecheck{\Delta}$ are terms representing the conjunction and disjunction of formulas in Γ and Δ, respectively, $\mathcal{R}_{seq}$ is a TRS for seq, a TRS is GSC-terminating if its termination is proved by the *generalized subterm criterion* [17, Theorem 33], and Φ is orthogonal (GSC-terminating) if $\{A \to A_i \mid \frac{A_1 \ \cdots \ A_n}{A} \in \Phi,\ 1 \le i \le n\}$ is orthogonal (GSC-terminating).

An approach to the comparison is to show that there exists a cyclic proof of a sequent $\Gamma \vdash \Delta$ if and only if there exists an RI proof of $\mathsf{seq}(\widehat{\Gamma},\widecheck{\Delta}) \approx \top$. The proof can be achieved by transforming cyclic proofs and RI proofs into each other. For Φ and $\mathcal{R}_\Phi \cup \mathcal{R}_{seq}$ above, it has been shown in [19] that under the following additional assumptions, a cyclic proof of a sequent $F \vdash F'$ w.r.t. Φ can be transformed into an RI proof of $\mathsf{seq}(\widetilde{F},\widetilde{F'}) \approx \top$ w.r.t. $\mathcal{R}_\Phi \cup \mathcal{R}_{seq} \cup \mathcal{R}_{scr}$, where $\widetilde{F}$ is a term representation of F and $\mathcal{R}_{scr}$ is a TRS representing some sequent-calculus rules:

(A5) Cyclic proofs are quantifier-free,

(A6) cyclic proofs are *cut-free*,

(A7) each inductive definition has at most one premise,

(A8) for any sequent $\Gamma \vdash \Delta$ in a cyclic proof, the multiset Γ is empty or singleton and the multiset Δ is singleton,[1]

(A9) cyclic proofs do not include the application of ($\wedge$L$_1$), ($\wedge$L$_2$), ($\vee$R$_1$), ($\vee$R$_2$), and (WL), and

(A10) any companion in cyclic proofs is the conclusion of an instance of the *case* rule which corresponds to a case distinction depending on rules in Φ.

It is possible to transform the obtained RI proof back into the cyclic proof. On the other hand, it is not so easy to transform an arbitrary RI proof of $\mathsf{seq}(\widetilde{F},\widetilde{F'}) \approx \top$ into a cyclic proof of $F \vdash F'$.

In this paper, under the assumptions (A1)–(A4), (A7), we show a transformation of an RI proof of $\mathsf{seq}(A, Q(\vec{u})) \approx \top$ into a cyclic proof that ensures the validity of $A \vdash Q(\vec{u})$, where A is either $\mathtt{true}$ or $P(\vec{t})$. Note that due to the form $A \vdash Q(\vec{u})$, (A11) our target sequents do not have logical connectives—logical-connective-free. In addition, we assume that (A12) RI proofs we deal with are constructed by the main inference rules: EXPAND, SIMPLIFY, and DELETE. Missing proofs can be seen in the appendix.

In [19], POSTULATE is considered for RI proofs. When we add some equations to an intermediate RI process by means of POSTULATE and succeed in proving the initial equations, we can start with the initial RI process together with the added ones to reach the intermediate RI process. For this reason, to simplify discussions, we do not consider the use of POSTULATE in RI proofs.

To facilitate the transformation, we first propose some derived rules of RI, and also a restricted set of RI (derived) rules for sequents (Section 4.1). We call an RI proof over the restricted set an *RI^d proof*. We show that every RI proof can be transformed into an RI$^{\mathrm{d}}$ proof. Then, for cyclic proofs, we define the *Case* rule for the right-hand sides of sequents, called (R-Case P), and show that an RI$^{\mathrm{d}}$ proof of $\mathsf{seq}(A, Q(\vec{u})) \approx \top$ can be transformed into a *quasi pre-proof* of $A \vdash Q(\vec{u})$, which is a pre-proof

[1] This assumption implies that none of ($\neg$L), ($\neg$R), (ContrL), (ContrR), (WR), (PL), and (PR) is used in cyclic proofs.

constructed by cyclic-proof rules and (R-Case P). Finally, we show a transformation of the quasi pre-proof into a cyclic proof of a sequent $Q'(\vec{u}), A \vdash Q(\vec{u})$ w.r.t. an IDS $\overline{\Phi}$ such that $\Phi \subseteq \overline{\Phi}$, Q' is a new predicate symbol that is not defined in Φ but in Φ', no predicate symbol in $\overline{\Phi} \setminus \Phi$ appears in Φ,[2] and $Q'(\vec{x})$ is valid w.r.t. $\overline{\Phi}$ (Section 4.2). The properties above of $\overline{\Phi}$ implies that the validity of $A \vdash Q(\vec{u})$ w.r.t. Φ is equivalent to that of $Q'(\vec{u}), A \vdash Q(\vec{u})$ w.r.t. $\overline{\Phi}$.

Related Work A related work is a comparison of structural proofs with cyclic induction [15]. The comparison may help us in the sense to compare RI and structural proofs; instead of comparing RI and cyclic proof systems, we may compare RI and structural proofs and then we may be able to indirectly compare RI with cyclic proof systems. In [14], the relationship between term- and formula-based induction principles has been discussed. This work would help us e.g., relax the assumptions in this paper.

2 Preliminaries

In this section, we briefly recall basic notions and notations of many-sorted term rewriting [16], RI [11, 2], and cyclic proofs [5]. Basic familiarity with term rewriting is assumed [3, 10].

2.1 Many-Sorted Term Rewriting

Let $\mathcal{S}$ be a set of *sorts*. Throughout the paper, we use $\mathcal{X}$ as a family of $\mathcal{S}$-sorted sets of variables: $\mathcal{X} = \biguplus_{s \in \mathcal{S}} \mathcal{X}_s$. Each *function symbol* f in a *signature* Σ is equipped with its sort declaration $\alpha_1 \times \cdots \times \alpha_n \to \alpha$, written as $f : \alpha_1 \times \cdots \times \alpha_n \to \alpha$, where $\alpha_1, \ldots, \alpha_n, \alpha \in \mathcal{S}$ and $n \geq 0$. The set of (well-sorted) *terms* is denoted by $T(\Sigma, \mathcal{X})$. The set of *ground* terms, $T(\Sigma, \emptyset)$, is abbreviated to $T(\Sigma)$. The set of variables appearing in any of terms $t_1, \ldots, t_n$ is denoted by $\mathcal{V}ar(t_1, \ldots, t_n)$. Let t be a term. The set of positions of t is denoted by $\mathcal{P}os(t)$. The function symbol at the *root* position ε of t is denoted by $root(t)$. The *domain* and *range* of a substitution σ which is a sort-preserving mapping from variables to terms are denoted by $\mathcal{D}om(\sigma)$ and $\mathcal{R}an(\sigma)$, respectively. A *most general unifier* of terms s, t is denoted by $mgu(s,t)$.

An $\mathcal{S}$-sorted *term rewrite system* (TRS, for short) is a set of rewrite rules of the form $\ell \to r$ such that the sorts of the LHS ℓ and the RHS r coincide, ℓ is not a variable, and $\mathcal{V}ar(\ell) \supseteq \mathcal{V}ar(r)$. The *reduction relation* $\to_{\mathcal{R}}$ of a TRS $\mathcal{R}$ is defined as follows: $s \to_{\mathcal{R}} t$ if and only if there exist a rewrite rule $\ell \to r \in \mathcal{R}$, a position $p \in \mathcal{P}os(s)$, and a substitution θ such that $s|_p = \ell\theta$ and $t = s[r\theta]_p$. We say that a TRS $\mathcal{R}$ is *GSC-terminating* if $\mathcal{R}$ is terminating and its termination can be proved by the *generalized subterm criterion* [17] (cf. [19, 20]).

The sets of *defined symbols* and *constructors* of $\mathcal{R}$ are denoted by $\mathcal{D}_{\mathcal{R}}$ and $\mathcal{C}_{\mathcal{R}}$, respectively: $\mathcal{D}_{\mathcal{R}} = \{root(\ell) \mid \ell \to r \in \mathcal{R}\}$ and $\mathcal{C}_{\mathcal{R}} = \Sigma \setminus \mathcal{D}_{\mathcal{R}}$. Terms in $T(\mathcal{C}_{\mathcal{R}}, \mathcal{X})$ are called *constructor terms* (of $\mathcal{R}$). A substitution σ is called *ground constructor* if $\mathcal{R}an(\sigma) \subseteq T(\mathcal{C}_{\mathcal{R}})$. A term t is called *basic* if t is of the form $f(t_1, \ldots, t_n)$ such that $f \in \mathcal{D}_{\mathcal{R}}$ and $t_1, \ldots, t_n \in T(\mathcal{C}_{\mathcal{R}}, \mathcal{X})$. A position p of a term t is called *basic* if $t|_p$ is basic. The set of basic positions of t is denoted by $\mathcal{B}(t)$. $\mathcal{R}$ is called *quasi-reductive* if every ground basic term is reducible.

2.2 Rewriting Induction

Our formulation of RI [11, 4] with the *generalization* (cf. [7]) follows [1, 13]. In the RI setting below, $\mathcal{R}$ is assumed to be terminating and quasi-reductive.

[2] For all ground formulas F of Φ, $\Phi \models F$ iff $\overline{\Phi} \models F$.

An *equation* (over a signature Σ) is a pair of terms, written as $s \approx t$, such that $s,t \in T(\Sigma,\mathcal{X})$. We write $s \simeq t$ to denote either $s \approx t$ or $t \approx s$. An equation $s \approx t$ is called an *inductive theorem* (of $\mathcal{R}$) if $s\theta \leftrightarrow^*_{\mathcal{R}} t\theta$ for all ground constructor substitutions θ with $\mathcal{V}ar(s,t) \subseteq \mathcal{D}om(\theta)$.

A pair $(\mathcal{E},\mathcal{H})$ consisting of an equation set $\mathcal{E}$ and a TRS $\mathcal{H}$ is called an *RI process*. Inference rules of RI defined below replace an equation by some equations, drop an equation, and insert some equations. From viewpoint of the replacement, to explicitly show descendant/ancestor relationships between equations, we attach to each equation $s \approx t$ a unique label ρ, written $(\rho)\ s \approx t$, that is a sequence of non-zero integers such as positions of terms;[3] We may use negative integers for freshly introduced labels. For case distinctions based on rewrite rules, we assume some fixed order of rewrite rules for each function symbol f.

The basic inference rules of *rewriting induction* are defined over RI processes as follows:

SIMPLIFY $(\mathcal{E} \uplus \{(\rho)\ \ s \simeq t\},\mathcal{H}) \Rrightarrow_{\mathsf{s}} (\mathcal{E} \cup \{(\rho.1)\ s' \approx t\},\mathcal{H})$, where $s \to_{p,\mathcal{R}\cup\mathcal{H}} s'$.

DELETE $(\mathcal{E} \uplus \{(\rho)\ \ s \approx t\},\mathcal{H}) \Rrightarrow_{\mathsf{d}} (\mathcal{E},\mathcal{H})$, where $s \leftrightarrow^*_{\mathcal{R}\cup\mathcal{E}} t$.[4]

EXPAND $(\mathcal{E} \uplus \{(\rho)\ \ s \simeq t\},\mathcal{H}) \Rrightarrow_{\mathsf{e}} (\mathcal{E} \cup \mathit{Expd}_p(s,t),\mathcal{H} \cup \{s \to t\})$, where $p \in \mathcal{B}(s)$, $\mathcal{R} \cup \mathcal{H} \cup \{s \to t\}$ is terminating, and

$$\mathit{Expd}_p(s,t) = \{(\rho.i)\ (s[r]_p)\sigma \approx t\sigma \mid \sigma = \mathit{mgu}(s|_p,\ell),\ \ell \to r \in \mathcal{R} \text{ is the } i\text{-th rule of } \mathit{root}(s|_p)\}$$

GENERALIZE $(\mathcal{E} \uplus \{(\rho)\ \ s\theta \simeq t\theta\},\mathcal{H}) \Rrightarrow_{\mathsf{g}} (\mathcal{E} \cup \{(\rho.1)\ s \approx t\},\mathcal{H})$.

We denote $\Rrightarrow_{\mathsf{s}} \cup \Rrightarrow_{\mathsf{d}} \cup \Rrightarrow_{\mathsf{e}} \cup \Rrightarrow_{\mathsf{g}}$ by $\Rrightarrow_{\mathrm{RI}}$. To make the position p at EXPAND step clear, we write EXPAND-p and $\Rrightarrow_{\mathsf{e}p}$. To make the reduction step $s \to_{p,\mathcal{R}\cup\mathcal{H}} s'$ at SIMPLIFY step clear, we write SIMPLIFY-$p(\mathcal{R}')$ and $\Rrightarrow_{\mathsf{s}p(\mathcal{R}')}$ if $s \to_{p,\mathcal{R}'} s'$ for some $\mathcal{R}' \subseteq \mathcal{R}$, and SIMPLIFY-H and $\Rrightarrow_{\mathsf{sH}}$ if $s \to_{\mathcal{H}} s'$. The proof based on RI starts with the initial RI process $(\mathcal{E},\emptyset)$ and proceeds by applying the inference rules above to RI processes. We attach labels to equations in $\mathcal{E}$ as follows: If $\mathcal{E}$ is singleton, then $\mathcal{E} = \{(\varepsilon)\ s \approx t\}$, and otherwise, $\mathcal{E} = \{(1)\ s_1 \approx t_1, (2)\ s_2 \approx t_2, \ldots\}$. A sequence $(\mathcal{E},\emptyset) = (\mathcal{E}_0,\mathcal{H}_0) \Rrightarrow_{\mathrm{RI}} (\mathcal{E}_1,\mathcal{H}_1) \Rrightarrow_{\mathrm{RI}} \cdots \Rrightarrow_{\mathrm{RI}} (\mathcal{E}_n,\mathcal{H}_n) = (\emptyset,\mathcal{H})$ is called an *RI proof* (of $\mathcal{E}$).

Theorem 2.1 ([11, 1]) *Let $\mathcal{R}$ be a quasi-reductive and terminating TRS. For a finite set $\mathcal{E}$ of equations, if $(\mathcal{E},\emptyset) \Rrightarrow^*_{\mathrm{RI}} (\emptyset,\mathcal{H})$ for some TRS $\mathcal{H}$, then every equation in $\mathcal{E}$ is an inductive theorem of $\mathcal{R}$.*

Example 2.2 Let us consider the signature $\Sigma_1 = \{+ : \mathit{nat} \times \mathit{nat} \to \mathit{nat},\ \mathsf{s} : \mathit{nat} \to \mathit{nat},\ \mathsf{0} : \mathit{nat}\}$ and the TRS $\mathcal{R}_1 = \{\ \mathsf{0}+y \to y,\ \ \mathsf{s}(x)+y \to \mathsf{s}(x+y)\ \}$ representing addition over natural numbers. Using RI, we can prove that $x+\mathsf{0} \approx x$ is an inductive theorem of $\mathcal{R}_1$ as follows:

$$\begin{array}{rllll}
(\{(\varepsilon)\ \ x+\mathsf{0} \approx x\},\emptyset) & \Rrightarrow_{\mathsf{e}} & (\{\ (1)\ \ \mathsf{0} \approx \mathsf{0}, & (2)\ \ \mathsf{s}(x_1+\mathsf{0}) \approx \mathsf{s}(x_1)\ \}, & \{(\varepsilon)\ \ x+\mathsf{0} \to x\}) \\
& \Rrightarrow_{\mathsf{d}} & (\{ & (2)\ \ \mathsf{s}(x_1+\mathsf{0}) \approx \mathsf{s}(x_1)\ \}, & \{(\varepsilon)\ \ x+\mathsf{0} \to x\}) \\
& \Rrightarrow_{\mathsf{sH}} & (\{ & (2.1)\ \ \mathsf{s}(x_1) \approx \mathsf{s}(x_1)\ \ \ \}, & \{(\varepsilon)\ \ x+\mathsf{0} \to x\}) \\
& \Rrightarrow_{\mathsf{d}} & (& \emptyset & ,\{(\varepsilon)\ \ x+\mathsf{0} \to x\})
\end{array}$$

2.3 First-Order Formulas with Inductive Definition Sets

In the rest of this paper, we consider a signature Σ with sorts $\mathcal{S} \supseteq \{\mathit{bool}\}$ such that $\mathsf{true},\mathsf{false} : \mathit{bool} \in \Sigma$. A symbol $P : \alpha_1 \times \cdots \times \alpha_n \to \mathit{bool} \in \Sigma$ is called a *predicate symbol*. A term $P(t_1,\ldots,t_n)$ with predicate

[3] $\mathcal{E}$ of an RI process $(\mathcal{E},\mathcal{H})$ is a set, but due to attached labels, $\mathcal{E}$ may contain an equation $s \approx t$ as $\rho_1 : s \approx t$ and $\rho_2 : s \approx t$ such that $\rho_1 \neq \rho_2$. This is not a problem because we can apply the same inference rules to both equations in order.

[4] We use a simplified side condition, which is enough for our purpose.

symbol $P : \alpha_1 \times \cdots \times \alpha_n \to \mathit{bool}$ is called an *atomic formula*. For brevity, we assume that $\mathcal{S} = \{\alpha, \mathit{bool}\}$, and every predicate symbol P has sort $\alpha \times \cdots \times \alpha \to \mathit{bool}$. In addition, we do not deal with *ordinary* predicates but *inductive* predicates as in [18, 20].

An *inductive definition set* (IDS, for short) Φ over Σ is a finite set of *productions* of the form $\dfrac{A_1 \quad \ldots \quad A_m}{A}$, where $A, A_1, \ldots, A_m$ are atomic formulas over Σ. We denote the set of productions for a predicate symbol P by $\Phi|_P$: $\Phi|_P = \{\frac{A_1 \ \ldots \ A_m}{A} \in \Phi \mid \mathit{root}(A) = P\}$. We say that Φ is *orthogonal* (*GSC-terminating*, resp.) if $\{A \to A_i \mid \frac{A_1 \ \ldots \ A_n}{A} \in \Phi,\ 1 \leq i \leq n\}$ is orthogonal (GSC-terminating, resp.). In the rest of this paper, we assume that (A4) and (A7) hold, i.e., every production in Φ is of the form either $\frac{}{A}$ or $\frac{A'}{A}$ such that $\mathcal{V}ar(A) \supseteq \mathcal{V}ar(A')$. For brevity, we allow A' to be true, writing both $\frac{}{A}$ and $\frac{A'}{A}$.

Example 2.3 ([5]) Let us consider the signature $\Sigma_1 = \{\ \mathsf{0} : \mathit{nat},\ \mathsf{s} : \mathit{nat} \to \mathit{nat},\ \mathsf{true}, \mathsf{false} : \mathit{bool},\ \mathsf{E}, \mathsf{O}, \mathsf{N} : \mathit{nat} \to \mathit{bool}\ \}$ and the following inductive definition set:

$$\Phi_1 = \left\{ \quad \frac{}{\mathsf{N}(\mathsf{0})} \qquad \frac{\mathsf{N}(x)}{\mathsf{N}(\mathsf{s}(x))} \qquad \frac{}{\mathsf{E}(\mathsf{0})} \qquad \frac{\mathsf{O}(x)}{\mathsf{E}(\mathsf{s}(x))} \qquad \frac{\mathsf{E}(x)}{\mathsf{O}(\mathsf{s}(x))} \quad \right\}$$

Note that the symbols E, O, and N stand for predicates *Even*, *Odd*, and *Nat*, respectively. This IDS is orthogonal and GSC-terminating.

We now consider standard first-order quantifier-free formulas over Σ. Structures for Σ are irrelevant because we do not deal with any *ordinary predicate*. For this reason, we do not deal with any structure for Σ, and define the semantics of formulas over the term structure for Σ in the syntactic way as usual: For an IDS Φ and a ground formula F, we write $\Phi \models F$ if F holds w.r.t. Φ (cf. [18, 20]). We say that a formula F' is *valid w.r.t.* Φ if $\Phi \models F'\theta$ for all ground substitutions θ with $\mathcal{D}om(\theta) \supseteq \mathcal{V}ar(F')$.

A sequent is a pair $\Gamma \vdash \Delta$ such that Γ, Δ are finite multisets of formulas, which can be written like lists of formulas. The application of a substitution θ to a finite multiset Γ of formulas is defined as $\Gamma\theta = \{F\theta \mid F \in \Gamma\}$. A sequent $\Gamma \vdash \Delta$ is said to be *valid* (w.r.t. Φ) if $\neg(\bigwedge_{F \in \Gamma} F) \vee (\bigvee_{F' \in \Delta} F')$ is valid w.r.t. Φ.

Example 2.4 For Φ_1 in Example 2.3, the sequent $\mathsf{E}(x) \vee \mathsf{O}(x) \vdash \mathsf{N}(x)$ is valid w.r.t. Φ_1 because $\neg(\mathsf{E}(x) \vee \mathsf{O}(x)) \vee \mathsf{N}(x)$ is valid w.r.t. Φ_1.

2.4 Cyclic Proofs

In this section, we consider proofs in the sequent-calculus style. We follow the formulation in [5] to define cyclic proofs. As in [19], we consider some simplified rules for sequent calculus illustrated in Figure 1.

The rule of applying a production in Φ is defined as follows:

$$\frac{}{\Gamma \vdash P(t_1, \ldots, t_n)\theta, \Delta}\ (P_i\text{-App}) \qquad \frac{\Gamma \vdash A\theta, \Delta}{\Gamma \vdash P(u_1, \ldots, u_n)\theta, \Delta}\ (P_j\text{-App})$$

where $\frac{}{P(t_1, \ldots, t_n)}, \frac{A}{P(u_1, \ldots, u_n)} \in \Phi$ are the i- and j-th rules of P, respectively. The rule for case distinctions based on productions in Φ is defined as follows:

$$\frac{\Gamma\theta_1, A_1 \vdash \Delta\theta_1 \quad \ldots \quad \Gamma\theta_n, A_k \vdash \Delta\theta_n}{\Gamma, P(y_1, \ldots, y_n) \vdash \Delta}\ (\text{Case } P)$$

$$\frac{\Gamma \cap \Delta \neq \emptyset}{\Gamma \vdash \Delta}\ (\text{Axiom}) \qquad \frac{\Gamma \vdash \Delta}{\Gamma\theta \vdash \Delta\theta}\ (\text{Subst}) \qquad \frac{\Gamma, F_1 \vdash \Delta \quad \Gamma, F_2 \vdash \Delta}{\Gamma, F_1 \vee F_2 \vdash \Delta}\ (\vee\text{L}) \qquad \frac{\Gamma \vdash F_1, \Delta \quad \Gamma \vdash F_2, \Delta}{\Gamma \vdash F_1 \wedge F_2, \Delta}\ (\wedge\text{R})$$

Figure 1: Sequent-calculus rules considered in this paper.

where $\Phi|_P = \{\frac{A_i}{P(t_{i,1},\ldots,t_{i,n})} \mid 1 \leq i \leq k\}$ for some k, $\frac{A_i}{P(t_{i,1},\ldots,t_{i,n})}$ is renamed as $\mathcal{V}ar(t_{i,1},\ldots,t_{i,n},A_i) \cap \mathcal{V}ar(F) = \emptyset$, and $\theta_i = \{y_j \mapsto t_{i,j} \mid 1 \leq j \leq n\}$ for $1 \leq i \leq k$.

Next, for cyclic proofs, we define some notions. Note that given a function f, we write $f : X \rightharpoonup Y$ and $f : X \to Y$ if f is partial and total, respectively.

Definition 2.5 (derivation tree and bud/companion nodes [5]) *Let $\mathcal{S}eqs$ be the set of well-formed sequents in some language, $\mathcal{R}ules$ some set of rules, and n the maximum number of premises of any rule in $\mathcal{R}ules$. A* derivation tree *is a rooted tree $\mathcal{T}$ represented by a quadruple (V,s,r,p) such that V is a set of nodes, $s : V \to \mathcal{S}eqs$ is a mapping that assigns a sequent to a node, $r : V \rightharpoonup \mathcal{R}ules$ is a mapping that assigns a rule to a node, $p : V \rightharpoonup V^n$ is a mapping that assigns premises nodes to a node, where $p_j(v)$ denotes the j-th component of $p(v)$, and for all nodes $v \in V$, $p_j(v)$ is defined just in case $r(v)$ is a rule with m premises ($1 \leq j \leq m$), and $\frac{s(p_1(v))\ \ldots\ s(p_m(v))}{s(v)}$ is an instance of rule $r(v)$. Note that the edges of the derivation tree is $\{(v, p_j(v)) \mid v \in V,\ p_j(v)$ is defined$\}$. A node $v \in V$ is called a* bud node *(of $\mathcal{T}$) if $r(v)$ is undefined, i.e., v is not the conclusion of any proof-rule instance in $\mathcal{T}$, and a* companion *for a bud node v' if $r(v)$ is defined and $s(v) = s(v')$.*

Note that a companion does not have to be an ancestor of its bud nodes. For readability, in illustrating a derivation tree, we attach to each node a label as well as RI proofs; such labels are sequences of positive integers indicating positions in the tree.

Definition 2.6 (pre-proof [5]) *A* pre-proof *of a sequent $\Gamma \vdash \Delta$ is a pair $(\mathcal{T}, \xi)$ of a finite derivation tree $\mathcal{T} = (V,s,r,p)$ (with v_0 the root node) and a mapping $\xi : V \rightharpoonup V$ such that the codomain of s is the set of well-formed sequents, $s(v_0) = (\Gamma \vdash \Delta)$, the codomain of r comprises the sequent calculus rules in this section, and every bud node v of $\mathcal{T}$ is assigned by ξ a companion, i.e., $\xi(v)$ is a companion.*

Definition 2.7 (trace) *Let $\mathcal{P}$ be a pre-proof. The* pre-proof graph *of $\mathcal{P}$, written as $\mathcal{G}_{\mathcal{P}}$, is a diredcted graph $(V', \{(v,v_i) \mid v \in V',\ p(v) = v_1 \ldots v_m,\ 1 \leq i \leq m\})$, where V' is obtained from V by identifying each bud node v in $\mathcal{T}$ with its companion $\xi(v)$. A* trace *following a (possibly infinite) path $v_1 v_2 \ldots$ in $\mathcal{G}_{\mathcal{P}}$ is a (possibly infinite) sequence $\tau_1 \tau_2 \ldots$ such that, for all $i > 0$:*

- *$\tau_i = F_i \in \Gamma_i$, where $s(v_i) = (\Gamma_i \vdash \Delta_i)$,*
- *if $r(v_i)$ is (Subst), then $\tau_{i+1}\theta = \tau_i$, where θ is the substitution associated with the rule instance,*
- *if $r(v_i)$ is (Case P), then either*
  - *$\tau_{i+1} = \tau_i\theta$, where θ is the substitution associated with the case distinction at $s(v_{i+1})$, or*
  - *τ_i is the principal formula $P(y_1,\ldots,y_n)$ of the rule instance and τ_{i+1} is a case descendant of $P(y_1,\ldots,y_n)$—i is said to be a* progress point *of the trace,*

  and
- *if $r(v_i)$ is neither (Subst) nor (Case P), then $\tau_{i+1} = \tau_i$.*

An infinite trace having infinitely may progress points is called an infinitely progressing *trace.*

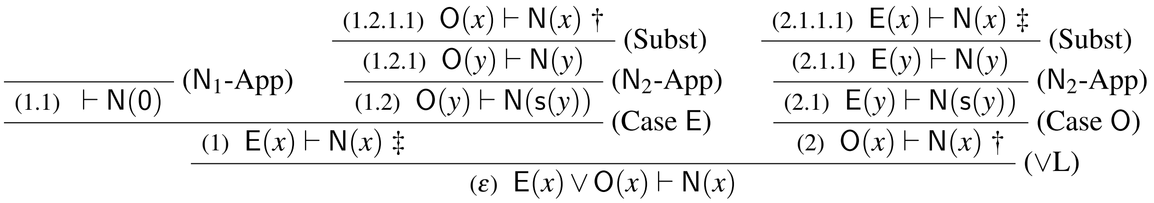
(1.2.1.1) $\mathsf{O}(x) \vdash \mathsf{N}(x)$ †
(Subst)
(1.2.1) $\mathsf{O}(y) \vdash \mathsf{N}(y)$
(N$_2$-App)
(1.1) $\vdash \mathsf{N}(0)$
(N$_1$-App)
(1.2) $\mathsf{O}(y) \vdash \mathsf{N}(\mathsf{s}(y))$
(Case E)
(1) $\mathsf{E}(x) \vdash \mathsf{N}(x)$ ‡
(2.1.1.1) $\mathsf{E}(x) \vdash \mathsf{N}(x)$ ‡
(Subst)
(2.1.1) $\mathsf{E}(y) \vdash \mathsf{N}(y)$
(N$_2$-App)
(2.1) $\mathsf{E}(y) \vdash \mathsf{N}(\mathsf{s}(y))$
(Case O)
(2) $\mathsf{O}(x) \vdash \mathsf{N}(x)$ †
($\vee$L)
(ε) $\mathsf{E}(x) \vee \mathsf{O}(x) \vdash \mathsf{N}(x)$

Figure 2: A cyclic proof for $\mathsf{E}(x) \vee \mathsf{O}(x) \vdash \mathsf{N}(x)$ [5].

Definition 2.8 (cyclic proof [5]) *A pre-proof $\mathcal{P}$ is said to be a* cyclic proof *if, for every infinite path in the pre-proof graph of $\mathcal{P}$, there is an infinitely progressing trace following some tail of the path.*

Theorem 2.9 ([5]) *If there exists a cyclic proof of a sequent $\Gamma \vdash \Delta$, then $\Gamma \vdash \Delta$ is valid w.r.t. Φ.*

Example 2.10 Consider the IDS Φ_1 in Example 2.3 again. Figure 2 illustrates a cyclic proof for $\mathsf{E}(x) \vee \mathsf{O}(x) \vdash \mathsf{N}(x)$ [5]. Therefore, $\mathsf{E}(x) \vee \mathsf{O}(x) \models_{\Phi_1} \mathsf{N}(x)$ holds.

3 From Orthogonal IDSs to Confluent TRSs

In this section, we briefly recall the transformation of orthogonal IDSs into confluent quasi-reductive constructor TRSs [18, 20, 19].

Given an orthogonal IDS Φ, $\mathcal{R}_\Phi$ is defined as $\mathcal{R}_\Phi^{base} \cup \mathcal{R}_\Phi^{ind} \cup \mathcal{R}_\Phi^{co} \cup \mathcal{R}_{pl} \cup \mathcal{R}_{seq}$, where

- $\mathcal{R}_\Phi^{base} = \{\, A \to \mathsf{true} \mid \frac{}{A} \in \Phi \,\}$,
- $\mathcal{R}_\Phi^{ind} = \{\, A \to A' \mid \frac{A'}{A} \in \Phi,\ A' \neq \mathsf{true} \,\}$,
- $\mathcal{R}_\Phi^{co} = \{\, t \to \mathsf{false} \mid t \in \mathit{Cop} \,\}$ with Cop a finite set of *co-patterns* [9] of $\{A \to A' \mid \frac{A'}{A} \in \Phi\}$,
- $\mathcal{R}_{pl} = \left\{ \begin{array}{lll} \mathsf{and}(\mathsf{false},\mathsf{false}) \to \mathsf{false}, & \mathsf{or}(\mathsf{false},\mathsf{false}) \to \mathsf{false}, & \mathsf{not}(\mathsf{false}) \to \mathsf{true}, \\ \mathsf{and}(\mathsf{false},\mathsf{true}) \to \mathsf{false}, & \mathsf{or}(\mathsf{false},\mathsf{true}) \to \mathsf{true}, & \mathsf{not}(\mathsf{true}) \to \mathsf{false}, \\ \mathsf{and}(\mathsf{true},\mathsf{false}) \to \mathsf{false}, & \mathsf{or}(\mathsf{true},\mathsf{false}) \to \mathsf{true}, & \\ \mathsf{and}(\mathsf{true},\mathsf{true}) \to \mathsf{true}, & \mathsf{or}(\mathsf{true},\mathsf{true}) \to \mathsf{true} & \end{array} \right\}$, and
- $\mathcal{R}_{seq} = \left\{ \begin{array}{llll} \mathsf{seq}(\mathsf{false},\mathsf{false}) \to \top, & \mathsf{seq}(\mathsf{false},\mathsf{true}) \to \top, & \bot \mathbin{\&} \bot \to \bot, & \bot \mathbin{\&} \top \to \bot, \\ \mathsf{seq}(\mathsf{true},\mathsf{false}) \to \bot, & \mathsf{seq}(\mathsf{true},\mathsf{true}) \to \top, & \top \mathbin{\&} \bot \to \bot, & \top \mathbin{\&} \top \to \top \end{array} \right\}$.

Note that $\top, \bot$ mean validity and invalidity of sequents, respectively.

Example 3.1 ([20, 19]) We transform Φ_1 in Example 2.3 into the following TRS:

$$\mathcal{R}_{\Phi_1} = \left\{ \begin{array}{lll} \mathsf{N}(0) \to \mathsf{true}, & \mathsf{E}(0) \to \mathsf{true}, & \mathsf{O}(0) \to \mathsf{false}, \\ \mathsf{N}(\mathsf{s}(x)) \to \mathsf{N}(x), & \mathsf{E}(\mathsf{s}(x)) \to \mathsf{O}(x), & \mathsf{O}(\mathsf{s}(x)) \to \mathsf{E}(x) \end{array} \right\} \cup \mathcal{R}_{pl} \cup \mathcal{R}_{seq}$$

For sequent-calculus rules (Axiom), ($\vee$L), and ($\wedge$R) in Figure 1, we define the following TRS [19]:

$$\mathcal{R}_{scr} = \left\{ \begin{array}{ll} \text{(Axiom)}\ \ \mathsf{seq}(x,x) \to \mathsf{true}, & (\vee\text{L})\ \ \mathsf{seq}(\mathsf{or}(x,y),z) \to \mathsf{seq}(x,z) \mathbin{\&} \mathsf{seq}(y,z), \\ & (\wedge\text{R})\ \ \mathsf{seq}(x,\mathsf{and}(y,z)) \to \mathsf{seq}(x,y) \mathbin{\&} \mathsf{seq}(x,z) \end{array} \right\}$$

Theorem 3.2 ([20, 19]) *For a GSC-terminating and orthogonal IDS Φ, all of the following hold:*

- *$\mathcal{R}_\Phi \cup \mathcal{R}_{seq}$ is a GSC-terminating, confluent, and quasi-reductive constructor TRS, and*
- *a sequent $A \vdash A'$ is valid w.r.t. Φ iff $\mathsf{seq}(A, A') \approx \top$ is an inductive theorem of $\mathcal{R}_\Phi \cup \mathcal{R}_{seq} \cup \mathcal{R}_{scr}$.*

4 Transformation of RI Proofs into Cyclic Proofs

In this section, given an IDS Φ, we consider to prove the validity of a quantifier- and logical-connective-free sequent $A \vdash Q(\vec{u})$ such that A is either $P(\vec{t})$ or true. We transform an RI proof of $\mathsf{seq}(A, Q(\vec{u})) \approx \top$[5] w.r.t. $\mathcal{R}_\Phi \cup \mathcal{R}_{pl} \cup \mathcal{R}_{seq}$ into a cyclic proof of $Q'(\vec{u}), A \vdash Q(\vec{u})$ w.r.t. $\overline{\Phi}$ such that

- Q' is a newly introduced predicate not appearing in Φ,
- $\overline{\Phi} \setminus \Phi$ defines only Q' and other related predicates not defined in Φ,
- $Q'(\vec{x})$ is valid w.r.t. $\overline{\Phi} \setminus \Phi$, and
- the patterns of Q' in $\overline{\Phi} \setminus \Phi$ are the same as those of Q in Φ.

Note that the validity of $Q'(\vec{u}), A \vdash Q(\vec{u})$ w.r.t. $\overline{\Phi}$ is equivalent to that of $A \vdash Q(\vec{u})$ w.r.t. Φ.

4.1 Derived Rules of RI Inference Rules

In applying rules in $\mathcal{R}_\Phi^{base} \cup \mathcal{R}_\Phi^{ind}$, the applied position p is either 1 or 2 because equations are of the form $\mathsf{seq}(A, A') \approx \top$ and defined predicates may appear only at 1 or 2. In applying rules in $\mathcal{R}_{scr} \cup \mathcal{H}$, p is ε because rules in $\mathcal{R}_{scr} \cup \mathcal{H}$ are of the form $\mathsf{seq}(A, A') \to \top$. To facilitate the transformation of RI proofs into cyclic proofs, we propose the following derived rules of RI w.r.t. $\mathcal{R}_\Phi \cup \mathcal{R}_{scr}$:

$\textsc{Simplify-1}(\mathcal{R}_\Phi^{co})$ & $\textsc{Delete}$

$$(\mathcal{E} \uplus \{(\rho)\ \mathsf{seq}(s,t) \simeq \top\}, \mathcal{H}) \Rrightarrow_{\mathsf{s1}(\mathcal{R}_\Phi^{co})\mathsf{\&d}} (\mathcal{E}, \mathcal{H}) \ \text{ if } \mathsf{seq}(s,t) \to_{1,\mathcal{R}_\Phi^{co}} \mathsf{seq}(\mathsf{false}, t)$$

$\textsc{Simplify-2}(\mathcal{R}_\Phi^{base})$ & $\textsc{Delete}$

$$(\mathcal{E} \uplus \{(\rho)\ \mathsf{seq}(s,t) \simeq \top\}, \mathcal{H}) \Rrightarrow_{\mathsf{s2}(\mathcal{R}_\Phi^{base})\mathsf{\&d}} (\mathcal{E}, \mathcal{H}) \ \text{ if } \mathsf{seq}(s,t) \to_{2,\mathcal{R}_\Phi^{base}} \mathsf{seq}(s, \mathsf{true})$$

$\textsc{Simplify-}\varepsilon(\mathcal{R}_{scr})$ & $\textsc{Delete}$

$$(\mathcal{E} \uplus \{(\rho)\ \mathsf{seq}(s,t) \simeq \top\}, \mathcal{H}) \Rrightarrow_{\mathsf{s}\varepsilon(\mathcal{R}_{scr})\mathsf{\&d}} (\mathcal{E}, \mathcal{H}) \ \text{ if } \mathsf{seq}(s,t) \to_{\varepsilon,\mathcal{R}_{scr}} \top$$

$\textsc{Simplify-H}$ & $\textsc{Delete}$

$$(\mathcal{E} \uplus \{(\rho)\ \mathsf{seq}(s,t) \simeq \top\}, \mathcal{H}) \Rrightarrow_{\mathsf{sH\&d}} (\mathcal{E}, \mathcal{H}) \ \text{ if } \mathsf{seq}(s,t) \to \top \in \mathcal{H}$$

As described in [19], at the $\textsc{Expand-1}$ step, any equation generated by a rule in $\mathcal{R}_\Phi^{co}$ can be deleted by $\textsc{Delete}$. For this reason, we use the following derived rule instead of $\textsc{Expand-1}$:

$\textsc{Expand}_1^+$ $(\mathcal{E} \uplus \{(\rho)\ \mathsf{seq}(P(s_1,\dots,s_n),t) \simeq \top\}, \mathcal{H}) \Rrightarrow_{\mathsf{e}} (\mathcal{E} \cup \mathcal{E}', \mathcal{H} \cup \{\mathsf{seq}(P(s_1,\dots,s_n),t) \to \top\})$ where
$\mathcal{E}' = \{(\rho.i)\ \mathsf{seq}(r\sigma, t\sigma) \approx \top \mid \sigma = mgu(P(s_1,\dots,s_n), \ell),\ \ell \to r \in \mathcal{R}_\Phi \text{ is the } i\text{-th rule of } P, \}$

In the following, $\Rrightarrow_{\mathsf{e1}}$ denotes the application of $\textsc{Expand}_1^+$.

An RI proof is called an *RI*^{d} *proof* if it is a sequence of $\Rrightarrow_{\mathrm{RId}}$ which denotes the following relation:

$$\Rrightarrow_{\mathsf{e1}} \cup \Rrightarrow_{\mathsf{e2}} \cup \Rrightarrow_{\mathsf{s1}(\mathcal{R}_\Phi^{base} \cup \mathcal{R}_\Phi^{ind})} \cup \Rrightarrow_{\mathsf{s1}(\mathcal{R}_\Phi^{co})\mathsf{\&d}} \cup \Rrightarrow_{\mathsf{s2}(\mathcal{R}_\Phi^{ind})} \cup \Rrightarrow_{\mathsf{s2}(\mathcal{R}_\Phi^{base})\mathsf{\&d}} \cup \Rrightarrow_{\mathsf{s}\varepsilon(\mathcal{R}_{scr})\mathsf{\&d}} \cup \Rrightarrow_{\mathsf{sH\&d}} \cup \Rrightarrow_{\mathsf{g}}$$

Every RI proof for sequents of the form $A \vdash Q(\vec{u})$ can be transformed into an RI$^{\mathrm{d}}$ proof.

Theorem 4.1 *Let $\mathcal{E} = \{A_i \vdash Q_i(\vec{u}_i) \mid 1 \le i \le n\}$ for some $n > 0$. For an RI proof of $\mathcal{E}$ w.r.t. $\mathcal{R}_\Phi \cup \mathcal{R}_{seq} \cup \mathcal{R}_{scr}$, there exists an RI^d proof of $\mathcal{E}$ w.r.t. $\mathcal{R}_\Phi \cup \mathcal{R}_{seq} \cup \mathcal{R}_{scr}$.*

In the rest of this section, we only deal with RI$^{\mathrm{d}}$ proofs.

[5] No rule in $\mathcal{R}_{pl} \cup \{(\vee\mathrm{L}), (\wedge\mathrm{R})\}$ is used in any RI proof for logical-connective-free sequents.

$$
\begin{array}{llll}
\multicolumn{4}{l}{(\{\ (\varepsilon)\ \mathsf{seq}(\mathsf{E}(x),\mathsf{N}(x))) \approx \top\ \},\emptyset)} \\
\Rrightarrow_{\mathsf{e2}} & (\{\ (1)\ \mathsf{seq}(\mathsf{E}(0),\mathsf{true}) \approx \top, & (2)\ \mathsf{seq}(\mathsf{E}(\mathsf{s}(y)),\mathsf{N}(y)) \approx \top & \},\{\ (\varepsilon)\ \}) \\
\Rrightarrow_{\mathsf{s2}(\mathcal{R}_{\Phi}^{base})\mathsf{\&d}} & (\{ & (2)\ \mathsf{seq}(\mathsf{E}(\mathsf{s}(y)),\mathsf{N}(y)) \approx \top & \},\{\ (\varepsilon)\ \}) \\
\Rrightarrow_{\mathsf{s1}(\mathcal{R}_{\Phi}^{ind})} & (\{ & (2.1)\ \mathsf{seq}(\mathsf{O}(y),\mathsf{N}(y)) \approx \top & \},\{\ (\varepsilon)\ \}) \\
\Rrightarrow_{\mathsf{e2}} & (\{\ (2.1.1)\ \mathsf{seq}(\mathsf{O}(0),\mathsf{true}) \approx \top, & (2.1.2)\ \mathsf{seq}(\mathsf{O}(\mathsf{s}(z)),\mathsf{N}(z)) \approx \top & \},\{\ (\varepsilon), (2.1)\ \}) \\
\Rrightarrow_{\mathsf{s1}(\mathcal{R}_{\Phi}^{co})\mathsf{\&d}} & (\{ & (2.1.2)\ \mathsf{seq}(\mathsf{O}(\mathsf{s}(z)),\mathsf{N}(z)) \approx \top & \},\{\ (\varepsilon), (2.1)\ \}) \\
\Rrightarrow_{\mathsf{s1}(\mathcal{R}_{\Phi}^{ind})} & (\{ & (2.1.2.1)\ \mathsf{seq}(\mathsf{E}(z),\mathsf{N}(z)) \approx \top & \},\{\ (\varepsilon), (2.1)\ \}) \\
\Rrightarrow_{\mathsf{g}} & (\{ & (2.1.2.1.1)\ \mathsf{seq}(\mathsf{E}(x),\mathsf{N}(x)) \approx \top & \},\{\ (\varepsilon), (2.1)\ \}) \\
\Rrightarrow_{\mathsf{sH\&d}} & (& \emptyset & ,\{\ (\varepsilon), (2.1)\ \})
\end{array}
$$

Figure 3: An RI^{d} proof for $\mathsf{E}(x) \vdash \mathsf{N}(x)$.

4.2 Transformation of RI^{d} Proofs into Cyclic Proof

In this section, we transform an RI^{d} proof of $\mathsf{seq}(A, Q(\vec{u})) \approx \top$ into a cyclic proof of a sequent, the validity of which is equivalent to that of $A \vdash Q(\vec{u})$.

Let us consider $\mathcal{R}_{\Phi_1}$, a sequent $\mathsf{E}(x) \vdash \mathsf{N}(x)$, and its RI^{d} proof in Figure 3. The first and fourth steps by $\Rrightarrow_{\mathsf{e2}}$, the second step by $\Rrightarrow_{\mathsf{s2}(\mathcal{R}_{\Phi}^{base})\mathsf{\&d}}$, and fifth step by $\Rrightarrow_{\mathsf{s1}(\mathcal{R}_{\Phi}^{co})\mathsf{\&d}}$ do not correspond any rule of cyclic proof systems because (Case P) is not applicable to the right-hand side of $\vdash$ and the second and fifth steps are complements of the first and fourth steps to delete (1) and (2.1.1), respectively. The third and sixth steps by $\Rrightarrow_{\mathsf{s1}(\mathcal{R}_{\Phi}^{ind})}$ correspond to (Case E) and (Case O), respectively.[6] The last step by $\Rrightarrow_{\mathsf{sH\&d}}$ corresponds to a bud node in cyclic proofs. For this reason, we define a rule for cyclic proofs, which corresponds to $\Rrightarrow_{\mathsf{e2}}$.

Definition 4.2 *We define the rule for case distinction in the right-hand side of sequents as follows:*

$$\frac{\Gamma\theta_1 \vdash A_1 \quad \dots \quad \Gamma\theta_k \vdash A_k \quad \Gamma\theta_{k+1} \vdash \mathsf{false} \quad \dots \quad \Gamma\theta_{k'} \vdash \mathsf{false}}{\Gamma \vdash P(y_1,\dots,y_n)} \textit{ (R-Case P)}$$

where $\Phi|_P \setminus \{\frac{}{P(\dots)} \in \Phi\} = \{\frac{A_i}{P(t_{i,1},\dots,t_{i,n})} \mid 1 \le i \le k\}$ *for some* k,[7] $\{\ell \mid \ell \to r \in \mathcal{R}_{\Phi}^{co}\} = \{P(t_{i,1},\dots,t_{i,n}) \mid k+1 \le i \le k'\}$ *for some* $k' \ge k$*, and* $\theta_i = \{y_{i,j} \mapsto t_{i,j} \mid 1 \le i \le k',\ 1 \le j \le n\}$ *for* $1 \le i \le k'$*. We call a pre-proof with (R-Case P) a* quasi pre-proof.

In the following, we consider false a constant predicate that is not defined by any IDS.

The correspondence between RI^{d} rules and quasi pre-proof rules is summarized in Table 1. Referring to Table 1, an RI^{d} proof can be transformed into a quasi pre-proof.

Theorem 4.3 *For an RI^{d} proof of $A \vdash Q(\vec{u})$ w.r.t. Φ, there exists a quasi pre-proof of $A \vdash Q(\vec{u})$ w.r.t. Φ such that each companion is the conclusion of the application of either (Case P) or (R-Case P) for some predicate P.*

Example 4.4 The RI^{d} proof in Figure 3 is transformed into a quasi pre-proof in Figure 4. Both (1) and (2.1.1) disappear in the quasi pre-proof because the corresponding sequents are implicitly deleted by (R-Case N).

[6] The application of production rules to the left-hand side is not introduced but (Case P) takes the place of it [6]. On the other hand, (Case P) in this paper is not enough for this point, but, for brevity, we use the current formulation.

[7] A_i is not true, and $\frac{A_i}{P(t_{i,1},\dots,t_{i,n})}$ is renamed as $\mathcal{V}ar(t_{i,1},\dots,t_{i,n},A_i) \cap \mathcal{V}ar(A) = \emptyset$.

Table 1: The correspondence between RI$^{\mathrm{d}}$ rules and quasi pre-proof rules.

RI$^{\mathrm{d}}$ rules		quasi pre-proof rules
EXPAND$^{+}_{1}$	$(\Rrightarrow_{\mathsf{e1}})$	(Case P)
EXPAND-2	$(\Rrightarrow_{\mathsf{e2}})$	(R-Case P)
SIMPLIFY-1$(\mathcal{R}_{\Phi}^{base})$	$(\Rrightarrow_{\mathsf{s1}(\mathcal{R}_{\Phi}^{base})})$	(Case P) for leaves
SIMPLIFY-1$(\mathcal{R}_{\Phi}^{ind})$	$(\Rrightarrow_{\mathsf{s1}(\mathcal{R}_{\Phi}^{ind})})$	(Case P) for inner nodes
SIMPLIFY-1$(\mathcal{R}_{\Phi}^{co})$ & DELETE	$(\Rrightarrow_{\mathsf{s1}(\mathcal{R}_{\Phi}^{co})\&\mathsf{d}})$	(Case P)
SIMPLIFY-2$(\mathcal{R}_{\Phi}^{ind})$ & DELETE	$(\Rrightarrow_{\mathsf{s2}(\mathcal{R}_{\Phi}^{ind})})$	(P_i-App) for inner nodes
SIMPLIFY-2$(\mathcal{R}_{\Phi}^{base})$ & DELETE	$(\Rrightarrow_{\mathsf{s2}(\mathcal{R}_{\Phi}^{base})\&\mathsf{d}})$	(P_i-App) for leaves
SIMPLIFY-$\varepsilon(\mathcal{R}_{scr})$ & DELETE	$(\Rrightarrow_{\mathsf{s}\varepsilon(\mathcal{R}_{scr})\&\mathsf{d}})$	(Axiom)
SIMPLIFY-H & DELETE	$(\Rrightarrow_{\mathsf{sH\&d}})$	bud nodes
GENERALIZE	$(\Rrightarrow_{\mathsf{g}})$	(Subst)

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{(2.1.2.1.1)\ \ \mathsf{E}(x)\vdash\mathsf{O}(x)\ \dagger}{(2.1.2.1)\ \ \mathsf{E}(z)\vdash\mathsf{N}(z)}\,\text{(Subst)}}{(2.1.2)\ \ \mathsf{O}(\mathsf{s}(z))\vdash\mathsf{N}(z)}\,\text{(Case }\mathsf{O}\text{)}}{(2.1)\ \ \mathsf{O}(y)\vdash\mathsf{N}(y)}\,\text{(R-Case }\mathsf{N}\text{)}}{(2)\ \ \mathsf{E}(\mathsf{s}(y))\vdash\mathsf{N}(y)}\,\text{(Case }\mathsf{E}\text{)}}{(\varepsilon)\ \ \mathsf{E}(x)\vdash\mathsf{N}(x)\ \dagger}\,\text{(R-Case }\mathsf{N}\text{)}$$

Figure 4: A quasi pre-proof obtained from the RI$^{\mathrm{d}}$ proof in Figure 3.

Finally, we transform a quasi pre-proof obtained from an RI$^{\mathrm{d}}$ proof into a cyclic proof. To this end, we simulate (R-Case P) by the original rules by introducing a dummy predicate P' for P, replacing (R-Case P) by (Case P'). Given an IDS Φ, we define $\overline{\Phi}$ as follows:

$$\overline{\Phi}=\{\ \tfrac{Q'(\vec{u})}{P'(\vec{t})}\mid\tfrac{Q(\vec{u})}{P(\vec{t})}\in\Phi\ \}\cup\{\ \tfrac{}{P'(\vec{t})}\mid\tfrac{}{P(\vec{t})}\in\Phi\ \}\cup\{\ \tfrac{}{P'(\vec{t})}\mid P(\vec{t})\to\mathsf{false}\in\mathcal{R}_{\Phi}^{co}\ \}$$

Since Φ is GSC-terminating, by definition, it is clear that $\overline{\Phi}$ and $\overline{\Phi}\setminus\Phi$ are GSC-terminating. GSC-termination of $\overline{\Phi}\setminus\Phi$ implies that for any predicate P, $P'(\vec{x})$ is valid w.r.t. $\overline{\Phi}$. In the following, we use P' as a generated dummy symbol that does not appear in Φ.

Lemma 4.5 *Let $A\vdash Q(\vec{u})$ be a sequent that A is either* true *or $P(\vec{t})$. Then, $A\vdash Q(\vec{u})$ is valid w.r.t. Φ if and only if $Q'(\vec{u}),A\vdash Q(\vec{u})$ is valid w.r.t. $\overline{\Phi}$.*

Example 4.6 The IDS Φ_1 in Example 2.3 is transformed into the following one:

$$\overline{\Phi_1}=\Phi_1\cup\left\{\quad\frac{}{\mathsf{N}'(0)}\qquad\frac{\mathsf{N}'(x)}{\mathsf{N}'(\mathsf{s}(x))}\qquad\frac{}{\mathsf{E}'(0)}\qquad\frac{\mathsf{O}'(x)}{\mathsf{E}'(\mathsf{s}(x))}\qquad\frac{\mathsf{E}'(x)}{\mathsf{O}'(\mathsf{s}(x))}\qquad\frac{}{\mathsf{O}'(0)}\quad\right\}$$

Let us consider the quasi pre-proof in Figure 4 again. We transform the quasi pre-proof into a pre-proof of $\mathsf{N}'(x),\mathsf{E}(x)\vdash\mathsf{N}(x)$. We use $\mathsf{N}'(x)$ instead of $\mathsf{N}(x)$ in applying (R-Case N), that is, we replace (R-Case N) by (Case N'). Using this idea, the quasi pre-proof in Figure 4 is transformed into a pre-proof in Figure 5. Nodes without labels are newly introduced to simulate the application of productions at (R-Case N). The transformation is formulated as follows.

$$
\dfrac{\dfrac{}{\mathsf{E}(0)\vdash\mathsf{N}(0)}\,(\mathsf{N}_1\text{-App}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{\mathsf{E}(0)\vdash\mathsf{N}(0)}\,(\mathsf{N}_1\text{-App}) \qquad \dfrac{\dfrac{\dfrac{\text{(2.1.2.1.1)}\ \mathsf{N}'(x),\mathsf{E}(x)\vdash\mathsf{N}(x)\ \dagger}{\text{(2.1.2.1)}\ \mathsf{N}'(z),\mathsf{E}(z)\vdash\mathsf{N}(z)}\,(\text{Subst})}{\text{(2.1.2)}\ \mathsf{N}'(z),\mathsf{O}(\mathsf{s}(z))\vdash\mathsf{N}(z)}\,(\text{Case }\mathsf{O})}{\mathsf{N}'(z),\mathsf{O}(\mathsf{s}(z))\vdash\mathsf{N}(\mathsf{s}(z))}\,(\mathsf{N}_2\text{-App})}{\text{(2.1)}\ \mathsf{N}'(y),\mathsf{O}(y)\vdash\mathsf{N}(y)}\,(\text{Case }\mathsf{N}')}{\text{(2)}\ \mathsf{N}'(y),\mathsf{E}(\mathsf{s}(y))\vdash\mathsf{N}(y)}\,(\text{Case }\mathsf{E})}{\mathsf{N}'(y),\mathsf{E}(\mathsf{s}(y))\vdash\mathsf{N}(\mathsf{s}(y))}\,(\mathsf{N}_2\text{-App})}{(\varepsilon)\ \mathsf{N}'(x),\mathsf{E}(x)\vdash\mathsf{N}(x)\ \dagger}\,(\text{Case }\mathsf{N}')
$$

Figure 5: A cyclic proof obtained from the quasi pre-proof in Figure 4.

Definition 4.7 *Let $\mathcal{P} = ((V,s,r,p),\xi)$ be a quasi pre-proof. We define $\mathcal{P}' = ((V \cup V',s',r',p'),\xi)$ such that V' is the set of nodes introduced during the construction of s',r',p' and for a node $v \in V$ with $s(v) = (A \vdash Q(\vec{u}))$ and $p(v) = v_1 \dots v_n$, we define s',r',p' as follows:*

- *$s'(v) = Q'(\vec{u}),A \vdash Q(\vec{u})$,*
- *if $r(v)$ is (Case P), then $r'(v) =$ (Case P) and $p'(v) = p(v)$,*
- *if $r(v)$ is (Q_i-App), then $r'(v) =$ (Q_i-App), $p'(v) = v'$, $p'(v') = p(v)$, $s'(v') = (Q'(\vec{u'}),A \vdash R(\vec{u'}))$, and $r'(v') =$ (Case Q'), where $v' \notin V$ is a fresh node and $s(v_1) = (A \vdash R(\vec{u'}))$,*
- *if $r(v_i)$ is (Subst), then $r'(v) =$ (Subst) and $p'(v) = p(v)$, and*
- *if $r(v_i)$ is (R-Case Q), then $p'(v) = v'_1 \dots v'_n$, $r'(v) =$ (R-Case Q'), $s'(v'_i) = Q'_i(\vec{u}_i),A_i \vdash Q(\vec{u}_i)\theta_i$ for $1 \le i \le n$, $p'(v'_i) = v_i$ for $1 \le i \le n$, and $r'(v'_i) =$ (Q_i-App) for $1 \le i \le n$, where $s(v_i) = (A_i \vdash Q_i(\vec{u}_i))$ and θ_i is the matching substitution for the application of the i-th rule of Q.*

Theorem 4.8 *Let $\mathcal{P}'$ be a tree constructed in Definition 4.7, v_0 the root node of $\mathcal{P}'$ such $s(v_0) = (A \vdash Q(\vec{u}))$. Then, $\mathcal{P}'$ is a cyclic proof of $Q'(\vec{u}),A \vdash Q(\vec{u})$ w.r.t. $\overline{\Phi}$.*

Finally, we show the existence of a cyclic proof for an RI$^{\mathrm{d}}$ proof. The following is a direct consequence of Theorems 4.1 and 4.8.

Theorem 4.9 (main result) *For a sequent $A \vdash Q(\vec{u})$, if there exists an RI proof of $\mathsf{seq}(A,Q(\vec{u})) \approx \top$, then there exists a cyclic proof for a sequent $Q'(\vec{u}),A \vdash Q(\vec{u})$ w.r.t. $\overline{\Phi}$.*

Recall that the validity of $Q'(\vec{u}),A \vdash Q(\vec{u})$ w.r.t. $\overline{\Phi}$ is equivalent to that of $A \vdash Q(\vec{u})$ w.r.t. Φ (Lemma 4.5).

5 Conclusion

In this paper, under the assumptions (A1)–(A4), (A7), we showed a transformation of an RI proof of $\mathsf{seq}(A,Q(\vec{u})) \approx \top$ into a cyclic proof of $Q'(\vec{u}),A \vdash Q(\vec{u})$ such that the validity $A \vdash Q(\vec{u})$ w.r.t. Φ is equivalent to that of $Q'(\vec{u}),A \vdash Q(\vec{u})$ w.r.t. $\overline{\Phi}$. This result indicates that under the assumptions (A1)–(A12), for a sequent $A \vdash Q(\vec{u})$ such that A is either true or $P(\vec{t})$, there exists a cyclic proof of $A \vdash Q(\vec{u})$ or $Q'(\vec{u}),A \vdash Q(\vec{u})$ w.r.t. $\overline{\Phi}$ if and only if there exists an RI proof of $\mathsf{seq}(A,Q(\vec{u})) \approx \top$ w.r.t. $\mathcal{R}_\Phi \cup \mathcal{R}_{seq} \cup \mathcal{R}_{scr}$.

Our future work is to relax the assumptions (A1)–(A4), (A7), (A11) as much as possible. For the relaxation of (A7) and (A11), we will start by ensuring the global trace condition which is one of the most difficult characteristics of cyclic proofs.

Acknowledgements We gratefully acknowledge the anonymous reviewers for their useful comments to improve the paper.

References

[1] Takahito Aoto (2007): *Rewriting Induction Using Termination Checkers.* In: *Proceedings of the JSSST 24th Annual Conference*, 3C-C, pp. 1–4. Available at `http://www.nue.ie.niigata-u.ac.jp/~aoto/research/papers/report/itp.pdf`. In Japanese.

[2] Takahito Aoto & Yoshihito Toyama (2016): *Ground Confluence Prover based on Rewriting Induction.* In Delia Kesner & Brigitte Pientka, editors: *Proceedings of the 1st International Conference on Formal Structures for Computation and Deduction*, *LIPIcs* 52, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 33:1–33:12, doi:10.4230/LIPIcs.FSCD.2016.33.

[3] Franz Baader & Tobias Nipkow (1998): *Term Rewriting and All That.* Cambridge University Press, doi:10.1017/CBO9781139172752.

[4] Adel Bouhoula (1997): *Automated Theorem Proving by Test Set Induction. Journal of Symbolic Computation* 23(1), pp. 47–77, doi:10.1006/jsco.1996.0076.

[5] James Brotherston (2005): *Cyclic Proofs for First-Order Logic with Inductive Definitions.* In Bernhard Beckert, editor: *Proceedings of the 14th International Conference on Automated Reasoning with Analytic Tableaux and Related Methods*, *Lecture Notes in Computer Science* 3702, Springer, pp. 78–92, doi:10.1007/11554554_8.

[6] James Brotherston (2006): *Sequent Calculus Proof Systems for Inductive Definitions.* Ph.D. thesis, University of Edinburgh.

[7] Carsten Fuhs, Cynthia Kop & Naoki Nishida (2017): *Verifying Procedural Programs via Constrained Rewriting Induction. ACM Transactions on Computational Logic* 18(2), pp. 14:1–14:50, doi:10.1145/3060143.

[8] Cynthia Kop & Naoki Nishida (2013): *Term Rewriting with Logical Constraints.* In Pascal Fontaine, Christophe Ringeissen & Renate A. Schmidt, editors: *Proceedings of the 9th International Symposium on Frontiers of Combining Systems*, *Lecture Notes in Artificial Intelligence* 8152, pp. 343–358, doi:10.1007/978-3-642-40885-4_24.

[9] Azeddine Lazrek, Pierre Lescanne & Jean-Jacques Thiel (1990): *Tools for Proving Inductive Equalities, Relative Completeness, and omega-Completeness. Information and Computation* 84(1), pp. 47–70, doi:10.1016/0890-5401(90)90033-E.

[10] Enno Ohlebusch (2002): *Advanced Topics in Term Rewriting.* Springer, doi:10.1007/978-1-4757-3661-8.

[11] Uday S. Reddy (1990): *Term Rewriting Induction.* In Mark E. Stickel, editor: *Proceedings of the 10th International Conference on Automated Deduction*, *Lecture Notes in Computer Science* 449, Springer, pp. 162–177, doi:10.1007/3-540-52885-7_86.

[12] John C. Reynolds (2002): *Separation Logic: A Logic for Shared Mutable Data Structures.* In: *Proceedings of the 17th IEEE Symposium on Logic in Computer Science*, IEEE Computer Society, pp. 55–74, doi:10.1109/LICS.2002.1029817.

[13] Haruhiko Sato & Masahito Kurihara (2010): *Multi-Context Rewriting Induction with Termination Checkers. IEICE Transactions on Information & Systems* 93-D(5), pp. 942–952, doi:10.1587/transinf.E93.D.942.

[14] Sorin Stratulat (2012): *A Unified View of Induction Reasoning for First-Order Logic.* In Andrei Voronkov, editor: *Turing-100. The Alan Turing Centenary*, *EPiC Series in Computing* 10, EasyChair, pp. 326–352.

[15] Sorin Stratulat (2016): *Structural vs. Cyclic Induction: A Report on Some Experiments with Coq.* In James H. Davenport, Viorel Negru, Tetsuo Ida, Tudor Jebelean, Dana Petcu, Stephen M. Watt & Daniela Zaharie, editors: *Proceedings of the 18th International Symposium on Symbolic and Numeric Algorithms for Scientific Computing*, IEEE, pp. 29–36, doi:10.1109/SYNASC.2016.018.

[16] Terese (2003): *Term Rewriting Systems*. *Cambridge Tracts in Theoretical Computer Science* 55, Cambridge University Press.

[17] Akihisa Yamada, Christian Sternagel, René Thiemann & Keiichirou Kusakari (2016): *AC Dependency Pairs Revisited*. In Jean-Marc Talbot & Laurent Regnier, editors: *Proceedings of the 25th EACSL Annual Conference on Computer Science Logic*, *LIPIcs* 62, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 8:1–8:16, doi:10.4230/LIPIcs.CSL.2016.8.

[18] Shujun Zhang & Naoki Nishida (2021): *On Transforming Inductive Definition Sets into Term Rewrite Systems*. In: *Informal Proceedings of the 8th International Workshop on Rewriting Techniques for Program Transformations and Evaluation*, pp. 1–10. Available at `https://www.ipl.riec.tohoku.ac.jp/wpte2021/Zhang21wpte.pdf`.

[19] Shujun Zhang & Naoki Nishida (2022): *On Transforming Cut- and Quantifier-Free Cyclic Proofs into Rewriting-Induction Proofs*. In Michael Hanus & Atsushi Igarashi, editors: *Proceedings of the 16th International Symposium on Functional and Logic Programming*, *Lecture Notes in Computer Science* 13215, p. 262–281, doi:10.1007/978-3-030-99461-7_15.

[20] Shujun Zhang & Naoki Nishida (2022): *Transforming Orthogonal Inductive Definition Sets into Confluent Term Rewrite Systems*. *Journal of Logical and Algebraic Methods in Programming*, doi:10.1016/j.jlamp.2022.100779. To appear.

A Missing Proofs

Theorem 4.1 *Let $\mathcal{E} = \{A_i \vdash Q_i(\vec{u}_i) \mid 1 \leq i \leq n\}$ for some $n > 0$. For an RI proof of $\mathcal{E}$ w.r.t. $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{seq}$, there exists an RI^d proof of $\mathcal{E}$ w.r.t. $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{seq}$.*

Proof. Under the assumptions in Section 4, we have that

$$\begin{aligned}
\Rrightarrow_{\mathrm{RI}} \;=\; & \Rrightarrow_{\mathsf{e1}} \cup \Rrightarrow_{\mathsf{e2}} \\
& \cup \Rrightarrow_{\mathsf{s1}(\mathcal{R}_\Phi^{base} \cup \mathcal{R}_\Phi^{ind})} \cup \Rrightarrow_{\mathsf{s1}(\mathcal{R}_\Phi^{co})} \cup \Rrightarrow_{\mathsf{s2}(\mathcal{R}_\Phi^{ind})} \cup \Rrightarrow_{\mathsf{s2}(\mathcal{R}_\Phi^{co})} \cup \Rrightarrow_{\mathsf{s2}(\mathcal{R}_\Phi^{base})} \cup \Rrightarrow_{\mathsf{s}\varepsilon(\mathcal{R}_{seq})} \cup \Rrightarrow_{\mathsf{s}\varepsilon(\mathcal{R}_{scr})} \\
& \cup \Rrightarrow_{\mathsf{sH}} \\
& \cup \Rrightarrow_{\mathsf{d}} \\
& \cup \Rrightarrow_{\mathsf{g}}
\end{aligned}$$

By definition, we have that Recall that $\Rrightarrow_{\mathrm{RId}}$ is defined as

$$\begin{aligned}
\Rrightarrow_{\mathrm{RId}} \;=\; & \Rrightarrow_{\mathsf{e1}} \cup \Rrightarrow_{\mathsf{e2}} \\
& \cup \Rrightarrow_{\mathsf{s1}(\mathcal{R}_\Phi^{base} \cup \mathcal{R}_\Phi^{ind})} \cup \Rrightarrow_{\mathsf{s1}(\mathcal{R}_\Phi^{co})\&\mathsf{d}} \cup \Rrightarrow_{\mathsf{s2}(\mathcal{R}_\Phi^{ind})} \cup \Rrightarrow_{\mathsf{s2}(\mathcal{R}_\Phi^{base})\&\mathsf{d}} \cup \Rrightarrow_{\mathsf{s}\varepsilon(\mathcal{R}_{scr})\&\mathsf{d}} \\
& \cup \Rrightarrow_{\mathsf{sH\&d}} \\
& \cup \Rrightarrow_{\mathsf{g}}
\end{aligned}$$

It suffices to show that steps of $\Rrightarrow_{\mathsf{s1}(\mathcal{R}_\Phi^{co})} \cup \Rrightarrow_{\mathsf{s2}(\mathcal{R}_\Phi^{co})} \cup \Rrightarrow_{\mathsf{s2}(\mathcal{R}_\Phi^{base})} \cup \Rrightarrow_{\mathsf{s}\varepsilon(\mathcal{R}_{seq})} \cup \Rrightarrow_{\mathsf{s}\varepsilon(\mathcal{R}_{scr})} \cup \Rrightarrow_{\mathsf{sH}} \cup \Rrightarrow_{\mathsf{d}}$ are simulated by $\Rrightarrow_{\mathrm{RId}}$.

- The equation modified at the $\Rrightarrow_{\mathsf{s1}(\mathcal{R}_\Phi^{co})}$ is of the form $\mathsf{seq}(\mathsf{false}, A) \approx \top$. To delete it, SIMPLIFY-1($\mathcal{R}_\Phi^{co}$) & DELETE can be used.
- The equation modified at the $\Rrightarrow_{\mathsf{s2}(\mathcal{R}_\Phi^{co})}$ is of the form $\mathsf{seq}(A, \mathsf{false}) \approx \top$. To delete the equation, A have to be simplified to false, and thus, SIMPLIFY-1($\mathcal{R}_\Phi^{ind}$) and SIMPLIFY-1($\mathcal{R}_\Phi^{co}$) & DELETE can be used.
- The equation modified at the $\Rrightarrow_{\mathsf{s2}(\mathcal{R}_\Phi^{base})}$ is of the form $\mathsf{seq}(A, \mathsf{true}) \approx \top$. To delete it, SIMPLIFY-2($\mathcal{R}_\Phi^{base}$) & DELETE can be used.

- The equation modified at the $\Rrightarrow_{\mathsf{s}\varepsilon(\mathcal{R}_{seq})}$ is of the form $\mathsf{seq}(\mathsf{true},\mathsf{true}) \approx \top$ because $\mathcal{R}_\Phi^{co}$ is not used at SIMPLIFY steps. The equation is not in $\mathcal{E}$ and its parent is obtained by SIMPLIFY-2$(\mathcal{R}_\Phi^{base})$. The equation is deleted by DELETE, and thus, SIMPLIFY-2$(\mathcal{R}_\Phi^{base})$ & DELETE can simulate the steps.
- The equation modified at the $\Rrightarrow_{\mathsf{s}\varepsilon(\mathcal{R}_{scr})}$ step is of the form $\top \approx \top$ which cannot be oriented. The equation is deleted by DELETE, and thus, SIMPLIFY-$\varepsilon(\mathcal{R}_{scr})$ & DELETE can simulate the steps.
- The equation modified at the $\Rrightarrow_{\mathsf{sH}}$ step is of the form $\top \approx \top$ which cannot be oriented. The equation is deleted by DELETE, and thus, GENERALIZE and SIMPLIFY-H & DELETE can simulate the steps.
- Any step of $\Rrightarrow_{\mathsf{d}}$ follows another as mentioned above, and no single step of $\Rrightarrow_{\mathsf{d}}$ appears. □

Theorem 4.3 *For an RI^d proof of $A \vdash Q(\vec{u})$ w.r.t. Φ, there exists a quasi pre-proof of $A \vdash Q(\vec{u})$ w.r.t. Φ such that each companion is the conclusion of the application of either (Case P) or (R-Case P) for some predicate P.*

Proof. Let the RI$^\mathrm{d}$ proof be $(\{(\varepsilon) : \mathsf{seq}(A, Q(\vec{u})) \approx \top\}, \emptyset) \Rrightarrow_{\mathrm{RI}} \cdots \Rrightarrow_{\mathrm{RI}} (\emptyset, \mathcal{H})$ for some TRS $\mathcal{H}$. A quasi pre-proof $((V,s,r,p),\xi)$ can be constructed as follows:

- V is the set of labels in the RI$^\mathrm{d}$ proof, except for labels whose equations are of the form either $\mathsf{seq}(\ldots,\mathsf{true}) \approx \top$ or $\mathsf{seq}(\mathsf{false},\ldots) \approx \top$,
- $s(v)$ is the corresponding sequent of the equation v represents,
- $p(v)$ is the sequence of child equations of v, and
- $r(v)$ and ξ are determined by referring to Table 1: If an equation with label v is deleted by SIMPLIFY-H & DELETE using a rewrite rule in $\mathcal{H}$ with label v', then $r(v)$ is undefined (i.e., v is a bud node) and $\xi(v) = v'$ (i.e., v' is a companion connected with v by means of ξ).

By construction, the equation of a companion v' is oriented at the application of either EXPAND$_1^+$ or EXPAND-2. Therefore, the companion v' is the conclusion of the application of either (Case P) or (R-Case P) for some predicate P. □

Lemma 14.5 *Let $A \vdash Q(\vec{u})$ be a sequent that A is either* true *or $P(\vec{t})$. Then, $A \vdash Q(\vec{u})$ is valid w.r.t. Φ if and only if $Q'(\vec{u}), A \vdash Q(\vec{u})$ is valid w.r.t. $\overline{\Phi}$.*

Proof. The *only-if* part is trivial, and the *if* part follows from the construction of $\overline{\Phi}$ and the validity of $Q'(\vec{u})$ w.r.t. $\overline{\Phi'}$. □

Theorem 4.8 *Let $\mathcal{P}'$ be a tree constructed in Definition 4.7, v_0 the root node of $\mathcal{P}'$ such $s(v_0) = (A \vdash Q(\vec{u}))$. Then, $\mathcal{P}'$ is a cyclic proof of $Q'(\vec{u}), A \vdash Q(\vec{u})$ w.r.t. $\overline{\Phi}$.*

Proof. By definition, it is clear that $\mathcal{P}'$ is a pre-proof for $Q'(\vec{u}), A \vdash Q(\vec{u})$ w.r.t. $\overline{\Phi}$. It suffices to show the trace condition. $\mathcal{P}$ has at most two traces: One is along the left-hand side of sequents, and the other is along the right-hand side of sequents. The latter is a trace along the left-hand side of sequents in $\mathcal{P}'$. It follows from Theorem 4.3 that any companion in $\mathcal{P}$ is the conclusion of the application of either (Case P) or (R-Case P) for some predicate P, and thus, any companion in $\mathcal{P}'$ is the conclusion of the application of either (Case P) or (Case P'). Therefore, $\mathcal{P}'$ satisfies the trace condition. □