On Transforming Inductive Definition Sets into Term Rewrite Systems*

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

In this paper, we transform an inductive definition set—a set of productions for inductive predicates—into a term rewrite system (TRS, for short) such that a quantifier-free sequent over the first-order logic with the inductive definition set is valid if and only if its corresponding equation is an inductive theorem of the TRS. The resulting TRS is composed of three parts: Rewrite rules for logical connectives and a binary symbol for sequents; rewrite rules for productions; rewrite rules for the co-patterns of the second part. For correctness of the resulting TRS, we assume a certain property of the inductive definition set, which is a sufficient condition for ground termination and ground confluence of the resulting TRS. The transformation aims at comparing cyclic proof systems and rewriting induction.

1 Introduction

Inductive theorem proving is well investigated in functional programming and term rewriting. In the field of term rewriting, *rewriting induction* [10] (RI, for short) is one of the most powerful principles to prove equations *inductive theorems*. An equation $s \approx t$ is called an *inductive theorem* of a given (many-sorted) *term rewrite system* (TRS, for short) if the equation is inductively valid under the reduction of the TRS, i.e., all of its ground instances are theorems of the TRS. RI has been extended to several kinds of rewrite systems, e.g., *logically constrained term rewrite systems* [7] (LCTRS, for short) that are models of not only functional but also imperative programs [5].

A *cyclic proof system* [2] is a proof system in sequent-calculus style for first-order logics with *inductive predicates*, where inductive predicates are defined by productions of the form $\frac{A_1 \ \ldots \ A_n}{A}$. In contrast to structural proofs which are (possibly infinite) derivation trees, cyclic proofs are finite derivation trees with back-links from *bud* nodes to *companions*. Such back-links correspond to the application of induction hypotheses, making trees finite. For the last decade, cyclic proof systems are well investigated for several logics, e.g., *separation logic* [11].

RI and cyclic proof systems have similar inference rules such as case analysis, the application of rules in given systems, and generalization. RI is based on *induction* by means of the application of rewrite rules representing induction hypotheses; the measure of the induction is the terminating reduction of the combined system of a given system and the induction hypotheses. Cyclic proofs have bud nodes that are connected with their companion, and the back-link corresponding to induction; the measure of the induction is that every (possibly infinite) path from the root passes infinitely many times through the application of the *case* rule which is based on productions for inductive predicates.

From the above observation, RI and cyclic proof systems seem very similar and we are interested in differences between RI and cyclic proof systems, while the former proves equations to be inductive theorems and the latter proves validity of sequents. If RI and cyclic proof systems have the same proof

*This work was partially supported by JSPS KAKENHI Grant Number JP18K11160.

Submitted to:
WPTE 2021

power, then it would be easy to apply several developed techniques for one to the other; otherwise, we must be able to know something new for inductive theorem proving, e.g., one of them could be improved by the advantage of the other. For this reason, it is worth comparing RI and cyclic proof systems, and the comparison is our ultimate goal. For the comparison, we have to consider the common setting: *Inductive definition sets*, sets of productions for inductive predicates, must be represented by rewrite systems, and formulas by terms.

In this paper, we transform an inductive definition set Φ into a TRS $\mathcal{R}$ such that a quantifier-free sequent $\Gamma \vdash \Delta$ is valid w.r.t. Φ (i.e., $\Phi \models (\bigwedge_{F\in\Gamma} F \Rightarrow \bigvee_{F'\in\Delta} F')$) if and only if its corresponding equation $\mathsf{seq}(\widetilde{\Gamma},\widehat{\Delta}) \approx \mathsf{true}$ is an inductive theorem of $\mathcal{R}$.[1] Given an inductive definition set Φ, the resulting TRS $\mathcal{R}$ is composed of three parts:

- a ground TRS $\mathcal{R}_{PL}$ for $\vdash$ and logical connectives ($\vee$, $\wedge$, and $\neg$);
- a TRS $\mathcal{R}_\Phi$ obtained from Φ by transforming each production in Φ into a rewrite rule;
- rewrite rules for the *co-patterns* [8] of $\mathcal{R}_\Phi$.

Note that $\mathcal{R}_{PL}$ is irrelevant to Φ. We denote by $\mathcal{R}_\Phi^{co}$ the union of $\mathcal{R}_\Phi$ and the set of rules for co-patterns, and $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$ by $\mathcal{R}$ which is the TRS we generate. The co-patterns of left-linear TRSs can be enumerated [8], and thus, $\mathcal{R}_\Phi$ is expected to be left-linear. For this reason, we assume that Φ is *conclusion-linear*, i.e., the conclusions of productions in Φ are linear. In addition, we assume that Φ is *consistent*, i.e., there is no closed formula F such that $\Phi \models F$ and $\Phi \not\models F$. This is not restrictive because we are interested in consistent classes.

For the correctness of the resulting TRS $\mathcal{R}$, we first show that $\mathcal{R}$ is a *quasi-reductive constructor TRS*, and if $\mathcal{R}$ is ground confluent, then both of the following hold:

- a ground formula holds (i.e., $\Phi \models F$) if and only if $F \to_{\mathcal{R}}^* \mathsf{true}$, and
- a ground formula does not hold (i.e., $\Phi \not\models F$) if and only if $F \to_{\mathcal{R}}^* \mathsf{false}$.

Then, we show that if $\mathcal{R}$ is ground terminating, then

- $\mathcal{R}$ is ground confluent, and
- a quantifier-free sequent $\Gamma \vdash \Delta$ is valid w.r.t. Φ if and only if its corresponding equation $\mathsf{seq}(\widetilde{\Gamma},\widehat{\Delta}) \approx \mathsf{true}$ is an inductive theorem of $\mathcal{R}$.

Note that we assume ground termination of $\mathcal{R}$ as a sufficient condition for ground confluence of $\mathcal{R}$. Finally, we show that the TRS $\{A \to A_i \mid \frac{A_1 \ \cdots \ A_n}{A} \in \Phi,\ 1 \le i \le n\}$ is GSC-terminating if and only if $\mathcal{R}$ is so, where a TRS $\mathcal{R}'$ is said to be *GSC-terminating* if $\mathcal{R}'$ is terminating and its termination can be proved by the *generalized subterm criterion* [13, Theorem 33]; in other words, GSC-termination of $\{A \to A_i \mid \frac{A_1 \ \cdots \ A_n}{A} \in \Phi,\ 1 \le i \le n\}$ is a sufficient condition for termination of the resulting TRS $\mathcal{R}$. In summary, if Φ is conclusion-linear and consistent, and the TRS $\{A \to A_i \mid \frac{A_1 \ \cdots \ A_n}{A} \in \Phi,\ 1 \le i \le n\}$ is GSC-terminating, then the resulting TRS $\mathcal{R}$ has the expected property.

This paper is organized as follows. In Section 2, we briefly recall many-sorted term rewriting and first-order logics with inductive predicates. In Section 3, we show a transformation of an inductive definition set into an equivalent TRS. In Section 4, we discuss termination of the resulting TRS. In Section 5, we conclude this paper and discuss future work of this research.

2 Preliminaries

In this section, we briefly recall basic notions and notations of many-sorted term rewriting [12] and first-order logics with inductive predicates [4, 3]. Basic familiarity with term rewriting is assumed [1, 9].

[1] $\widetilde{\Gamma},\widehat{\Delta}$ are terms representing Γ,Δ, respectively, and we represent a sequent $\Gamma \vdash \Delta$ by a term $\mathsf{seq}(\widetilde{\Gamma},\widehat{\Delta})$.

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 an $\mathcal{S}$-sorted *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 $\mathcal{V}ar(t_1,\ldots,t_n)$. For a term t, the set of positions of t is denoted by $\mathcal{P}os(t)$. For a term t and a position p of t, the *subterm* of t at p is denoted by $t|_p$; we write $t \trianglerighteq t|_p$, and $t \triangleright t|_p$ if $p \neq \varepsilon$. The function symbol at the *root* position ε of a term t is denoted by $\mathrm{root}(t)$. Given terms $s,t_1,\ldots,t_n$ and parallel positions $p_1,\ldots,p_n$ of s, we denote by $s[t_1,\ldots,t_n]_{p_1,\ldots,p_n}$ the term obtained from s by replacing the subterm $s|_{p_i}$ at p_i by t_i for each $i \in \{1,\ldots,n\}$.

A *substitution* σ is a sort-preserving mapping from variables to terms such that the number of variables x with $\sigma(x) \neq x$ is finite, and is naturally extended over terms. The *domain* and *range* of σ are denoted by $\mathcal{D}om(\sigma)$ and $\mathcal{R}an(\sigma)$, respectively. We may denote σ by $\{x_1 \mapsto t_1, \ldots, x_n \mapsto t_n\}$ if $\mathcal{D}om(\sigma) = \{x_1,\ldots,x_n\}$ and $\sigma(x_i) = t_i$ for all $1 \leq i \leq n$. A substitution σ is called *ground* if $\mathcal{R}an(\sigma) \subseteq T(\Sigma)$. The application of a substitution σ to a term t, $\sigma(t)$, is abbreviated to $t\sigma$, and $t\sigma$ is called an *instance* of t. A *most general unifier* of terms s,t is denoted by $\mathrm{mgu}(s,t)$.

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 $\mathcal{V}ar(\ell) \supseteq \mathcal{V}ar(r)$. In the following, we use $\mathcal{R}$ as a TRS over an $\mathcal{S}$-sorted signature Σ without notice. The *reduction relation* $\to_\mathcal{R}$ of $\mathcal{R}$ 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 $t = s[r\theta]_p$. $\mathcal{R}$ is called *left-linear* if all rules in $\mathcal{R}$ are left-linear (i.e., have linear terms as their lhss), and *ground* if all rules in $\mathcal{R}$ are ground (i.e., have ground terms as their lhss and rhss).

The set of *defined symbols* of $\mathcal{R}$ is denoted by $\mathcal{D}_\mathcal{R}$: $\mathcal{D}_\mathcal{R} = \{\mathrm{root}(\ell) \mid \ell \to r \in \mathcal{R}\}$. The set of *constructors* of $\mathcal{R}$ is denoted by $\mathcal{C}_\mathcal{R}$: $\mathcal{C}_\mathcal{R} = \Sigma \setminus \mathcal{D}_\mathcal{R}$. Terms in $T(\mathcal{C}_\mathcal{R},\mathcal{X})$ are called *constructor terms* (of $\mathcal{R}$). A 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 $t_1,\ldots,t_n \in T(\mathcal{C}_\mathcal{R},\mathcal{X})$. $\mathcal{R}$ is called a *constructor system* if every rule in $\mathcal{R}$ has a basic term as its lhs. 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 $\mathcal{B}(t)$. TRS $\mathcal{R}$ is called *quasi-reductive* if every ground basic term is reducible. A substitution σ is called *constructor* if $\mathcal{R}an(\sigma) \subseteq T(\mathcal{C}_\mathcal{R},\mathcal{X})$.

The *marked symbol* of a defined symbol $f : \alpha_1 \times \cdots \times \alpha_n \to \alpha \in \mathcal{D}_\mathcal{R}$ is denoted by $f^\#$ and the set of marked symbols for $\mathcal{D}_\mathcal{R}$ is denoted by $\mathcal{D}^\#_\mathcal{R}$. The marked symbol $f^\#$ has sort $\alpha_1 \times \cdots \times \alpha_n \to \mathit{dpsort}$, where *dpsort* is a newly introduced sort for marked symbols. For a term t of the form $f(t_1,\ldots,t_n)$ with $f \in \mathcal{D}_\mathcal{R}$, the term $f^\#(t_1,\ldots,t_n)$ is denoted by $t^\#$. For each rule $\ell \to r \in \mathcal{R}$, the *dependency pairs* (DP, for short) of $\mathcal{R}$ contains all rules $\ell^\# \to u^\#$ such that u is a subterm of r and $\mathrm{root}(u) \in \mathcal{D}_\mathcal{R}$. The set of DPs of $\mathcal{R}$ is denoted by $DP(\mathcal{R})$. Let $\mathcal{P} \subseteq DP(\mathcal{R})$. A reduction sequence $s_1^\# \to_\mathcal{P} t_1^\# \to^*_\mathcal{R} s_2^\# \to_\mathcal{P} t_2^\# \to^*_\mathcal{R} \cdots$ with $s_1,t_1,s_2,t_2,\ldots \in T(\Sigma,\mathcal{X})$ is called a *dependency chain* of $\mathcal{P}$ ($\mathcal{P}$-chain, for short). The *dependency graph* (DG, for short) of $\mathcal{R}$ is denoted by $DG(\mathcal{R})$: $DG(\mathcal{R}) = (DP(\mathcal{R}), \{(s^\# \to t^\#,\, u^\# \to v^\#) \mid s^\# \to t^\#,\, u^\# \to v^\# \in DP(\mathcal{R}),\ \exists\theta,\sigma.\ t^\#\theta \to^*_\mathcal{R} u^\#\sigma\})$.

As a termination criterion, we use a simplified variant of the *generalized subterm criterion* [13]. A *multi-projection* π for a set $\mathcal{F}$ of function symbols is a mapping that assigns every symbol $f \in \mathcal{F}$ a non-empty multiset of its argument positions. We extend π for terms as follows:

- $\pi(t) = \pi(t_{i_1}) \oplus \cdots \oplus \pi(t_{i_m})$ if $t = f(t_1,\ldots,t_n)$, $f \in \mathcal{F}$, and $\pi(f) = \{i_1,\ldots,i_m\}$, and
- $\pi(t) = \{t\}$, otherwise,

where $\oplus$ is the union of multisets. For a binary relation $\sqsupset$ on terms, we denote the multiset extension of $\sqsupset$ by $\sqsupset^{mul}$, and we write $s \sqsupset^\pi t$ if $\pi(s) \sqsupset^{mul} \pi(t)$.

Theorem 2.1 (cf. [6, Theorem 3.3] and [13, Theorem 33]) *A TRS $\mathcal{R}$ is terminating if for every cycle $\mathcal{P}$ in $DG(\mathcal{R})$ there exists a multi-projection π for $\mathcal{D}^\#$ such that $\mathcal{P} \subseteq \trianglerighteq^\pi$ and $\mathcal{P} \cap \triangleright^\pi \neq \emptyset$.*

A TRS $\mathcal{R}$ is said to be *GSC-terminating* if $\mathcal{R}$ is terminating and its termination can be proved by the generalized subterm criterion (Theorem 2.1).

An *equation* (over an $\mathcal{S}$-sorted signature Σ) is a pair of terms, written as $s \approx t$, such that $s,t \in T(\Sigma,\mathcal{X})$ and s,t have the same sort. An equation $s \approx t$ is called an *inductive theorem* (of $\mathcal{R}$) if $s\theta \leftrightarrow^*_\mathcal{R} t\theta$ for all ground substitutions θ with $\mathcal{V}ar(s,t) \subseteq \mathcal{D}om(\theta)$ and $\mathcal{R}an(\theta) \subseteq T(\Sigma)$. Note that if $\mathcal{R}$ is quasi-reductive, then we can assume θ in the above definition to be a ground constructor substitution.

2.2 First-Order Logics with Inductive Predicates

In the rest of this paper, we consider a signature Σ with sorts $\mathcal{S} \supseteq \{bool\}$ such that $\mathsf{true}, \mathsf{false} : bool \in \Sigma$. A symbol $P : \alpha_1 \times \cdots \times \alpha_n \to bool \in \Sigma$ is called a *predicate symbol*. A term $P(t_1,\ldots,t_n)$ with predicate symbol $P : \alpha_1 \times \cdots \times \alpha_n \to bool$ is called an *atomic formula*. For brevity, we do not deal with *ordinary* predicates but *inductive* predicates.

Definition 2.2 (inductive definition set [2, 3]) *An* inductive definition set Φ *over* Σ *is a finite set of* productions *of the form*

$$\frac{A_1 \quad \ldots \quad A_m}{A}$$

where $A, A_1, \ldots, A_m$ are atomic formulas over Σ, and $\mathcal{V}ar(A_1,\ldots,A_m) \subseteq \mathcal{V}ar(A)$. We denote the set of productions for a predicate symbol P by $\Phi|_P$: $\Phi|_P = \{\frac{A_1 \ \ldots \ A_m}{A} \in \Phi \mid \mathrm{root}(A) = P\}$. A production $\frac{A_1 \ \ldots \ A_m}{A}$ is called conclusion-linear *if its conclusion A is linear. We say that Φ is* conclusion-linear *if all productions in Φ are conclusion-linear.*

Example 2.3 ([2]) Let us consider the signature $\Sigma_1 = \{\, \mathsf{0} : nat,\ \mathsf{s} : nat \to nat,\ \mathsf{true}, \mathsf{false} : bool,\ \mathsf{E}, \mathsf{O}, \mathsf{N} : nat \to bool \,\}$ and the following inductive definition set:

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

Note that the symbols E, O, and N stand for predicates *Even*, *Odd*, and *Nat*, respectively. This inductive definition set is conclusion-linear.

This paper considers standard first-order formulas over Σ. Structures for the signature are irrelevant because we do not deal with any *ordinary predicate*. For this reason, we do not deal with any structure for the signature, and define the semantics of formulas over the term structure for the signature in the syntactic way as usual.

Definition 2.4 (semantics of formulas) *Let Φ be an inductive definition set (over Σ) for predicate symbols $P_1,\ldots,P_n$ where k_i denotes the arity of P_i. The* semantics *of ground formula F—we write $\Phi \models F$ if F holds w.r.t. Φ—is inductively defined as follows:*

- $\Phi \models \mathsf{true}$,
- *$\Phi \models A$ if and only if there exists a production $\frac{A'_1 \ \ldots \ A'_m}{A'} \in \Phi$ and a substitution θ such that $A = A'\theta$ and $\Phi \models A'_j\theta$ for all $1 \leq j \leq m$,*
- *$\Phi \models \neg F'$ if and only if $\Phi \not\models F'$,*

- $\Phi \models F_1 \vee F_2$ *if and only if* $\Phi \models F_1$ *or* $\Phi \models F_2$*, and*
- $\Phi \models F_1 \wedge F_2$ *if and only if* $\Phi \models F_1$ *and* $\Phi \models F_2$*.*

We say that Φ *is* consistent *if there is no ground formula* F *such that* $\Phi \models F$ *and* $\Phi \not\models F$*. We say that a formula* F *is* valid w.r.t. Φ *if* $\Phi \models F\rho$ *for all ground substitutions* ρ *with* $\mathcal{D}om(\rho) \supseteq \mathcal{V}ar(F)$*.*

Note that we are interested in consistent inductive definition sets only.

A (multi-conclusion) *sequent* (over Σ) 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 M of formulas is defined as $M\theta = \{F\theta \mid F \in M\}$. In the following, we use Γ, Δ for finite multisets of formulas, and F for formulas.

For a sequent $\Gamma \vdash \Delta$, the formulas in Γ are considered conjunctively (all the formulas are assumed to hold at the same time), and the formulas in Δ are considered disjunctively (at least one of the formulas must hold for any substitution). To be more precise, a sequent $\Gamma \vdash \Delta$ is *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 2.5 The sequent $\mathsf{E}(x) \vee \mathsf{O}(x) \vdash \mathsf{N}(x)$ is valid w.r.t. Φ_1 in Example 2.3 because $\neg(\mathsf{E}(x) \vee \mathsf{O}(x)) \vee \mathsf{N}(x)$ is valid w.r.t. Φ_1.

3 Transformation of Inductive Definition Sets into TRSs

In this section, we show a transformation of a conclusion-linear consistent inductive definition set Φ into a TRS $\mathcal{R}$ such that a sequent $\Gamma \vdash \Delta$ is valid w.r.t. Φ if and only if $\mathsf{seq}(\widetilde{\Gamma}, \widehat{\Delta}) \approx \mathsf{true}$ is an inductive theorem of $\mathcal{R}$.

3.1 Term Representation of Formulas and Sequents

To represent formulas and sequents as terms, we prepare function symbols for the truth values, logical connectives, and $\vdash$ as follows:

- the truth values are represented by constants $\mathsf{true}, \mathsf{false} : \mathit{bool}$ which are included in Σ,
- logical connectives $\wedge, \vee, \neg$ are represented by $\mathsf{and}, \mathsf{or} : \mathit{bool} \times \mathit{bool} \to \mathit{bool}$, $\mathsf{not} : \mathit{bool} \to \mathit{bool}$, respectively, and
- $\vdash$ is represented by $\mathsf{seq} : \mathit{bool} \times \mathit{bool} \to \mathit{bool}$.

A formula F is transformed by $\check{\cdot}$ into a term as follows:

- $\check{b} = b$ for $b \in \{\mathsf{true}, \mathsf{false}\}$,
- $\check{A} = A$ for an atomic formula A,
- $\check{F} = \mathsf{not}(\check{F}')$ if $F = \neg F'$,
- $\check{F} = \mathsf{and}(\check{F}_1, \check{F}_2)$ if $F = F_1 \wedge F_2$, and
- $\check{F} = \mathsf{or}(\check{F}_1, \check{F}_2)$ if $F = F_1 \vee F_2$.

To transform multisets of formulas into terms, we prepare $\widetilde{\cdot}$ and $\widehat{\cdot}$ as follows:

- $\widetilde{\emptyset} = \mathsf{true}$,
- $\widehat{\emptyset} = \mathsf{false}$,

- $\widetilde{\{F\}} = \widehat{\{F\}} = \check{F}$,
- $\widetilde{\{F_1,\ldots,F_n\}} = \mathsf{and}(\check{F_1}, \mathsf{and}(\check{F_2},\ldots,\mathsf{and}(\check{F}_{n-1},\check{F_n})\ldots)$ for $n > 1$, and
- $\widehat{\{F_1,\ldots,F_n\}} = \mathsf{or}(\check{F_1}, \mathsf{or}(\check{F_2},\ldots,\mathsf{or}(\check{F}_{n-1},\check{F_n})\ldots)$ for $n > 1$,

Roughly speaking, $\widetilde{M}$ and $\widehat{M}$ are the conjunction and disjunction of the formulas in M, respectively. We represent a sequent $\Gamma \vdash \Delta$ by a term $\mathsf{seq}(\widetilde{\Gamma},\widehat{\Delta})$. In the following, we abuse Σ for $\Sigma \cup \{\mathsf{and},\mathsf{or} : \mathit{bool} \times \mathit{bool} \to \mathit{bool},\ \mathsf{not} : \mathit{bool} \to \mathit{bool},\ \mathsf{seq} : \mathit{bool} \times \mathit{bool} \to \mathit{bool}\}$.

Example 3.1 The sequent $\mathsf{E}(x) \vee \mathsf{O}(x) \vdash \mathsf{N}(x)$ is transformed into $\mathsf{seq}(\mathsf{or}(\mathsf{E}(x),\mathsf{O}(x)),\mathsf{N}(x))$.

3.2 Rewriting Rules for Logical Connectives

For logical connectives (and, or, and not) and seq, we prepare the following rewrite rules:

$$\mathcal{R}_{PL} = \left\{ \begin{array}{rr} \mathsf{and}(\mathsf{false},\mathsf{false}) \to \mathsf{false}, & \mathsf{or}(\mathsf{false},\mathsf{false}) \to \mathsf{false}, \\ \mathsf{and}(\mathsf{false},\mathsf{true}) \to \mathsf{false}, & \mathsf{or}(\mathsf{false},\mathsf{true}) \to \mathsf{true}, \\ \mathsf{and}(\mathsf{true},\mathsf{false}) \to \mathsf{false}, & \mathsf{or}(\mathsf{true},\mathsf{false}) \to \mathsf{true}, \\ \mathsf{and}(\mathsf{true},\mathsf{true}) \to \mathsf{true}, & \mathsf{or}(\mathsf{true},\mathsf{true}) \to \mathsf{true}, \\ \mathsf{not}(\mathsf{false}) \to \mathsf{true}, & \mathsf{not}(\mathsf{true}) \to \mathsf{false}, \\ \mathsf{seq}(\mathsf{false},\mathsf{false}) \to \mathsf{true}, & \mathsf{seq}(\mathsf{false},\mathsf{true}) \to \mathsf{true}, \\ \mathsf{seq}(\mathsf{true},\mathsf{false}) \to \mathsf{false}, & \mathsf{seq}(\mathsf{true},\mathsf{true}) \to \mathsf{true}, \end{array} \right\}$$

Note that $\mathcal{D}_{\mathcal{R}_{PL}} = \{\mathsf{and},\ \mathsf{or},\ \mathsf{not},\ \mathsf{seq}\}$ and $\mathcal{R}_{PL}$ is a *ground* TRS, i.e., both sides of all rules are ground.

3.3 Transformation of Productions into Rewrite Rules

We transform an inductive definition set Φ over Σ into a TRS $\mathcal{R}_\Phi$ as follows:

$$\mathcal{R}_\Phi = \{\ A \to \widetilde{\{A_1,\ldots,A_n\}} \mid \frac{A_1 \ \ \ldots \ \ A_n}{A} \in \Phi\ \}$$

Example 3.2 We transform Φ_1 in Example 2.3 into the following TRS:

$$\mathcal{R}_{\Phi_1} = \{\ \mathsf{N}(0) \to \mathsf{true},\ \ \mathsf{N}(\mathsf{s}(x)) \to \mathsf{N}(x),\ \ \mathsf{E}(0) \to \mathsf{true},\ \ \mathsf{E}(\mathsf{s}(x)) \to \mathsf{O}(x),\ \ \mathsf{O}(\mathsf{s}(x)) \to \mathsf{E}(x)\ \}$$

3.4 Generation of Rewrite Rules for Co-Patterns

An inductive definition set implicitly defines that some atomic formulas do not hold, e.g., $\Phi_1 \not\models \mathsf{O}(0)$.[2] To represent such unsatisfaction, we need rewrite rules, e.g., $\mathsf{O}(0) \to \mathsf{false}$ for $\Phi_1 \not\models \mathsf{O}(0)$. Undefined atomic formulas of Φ are *co-patterns* (cf. [8]) of $\mathcal{R}_\Phi$. Roughly speaking, co-patterns of a TRS $\mathcal{R}$ are strongly irreducible[3] basic terms of $\mathcal{R}$. If $\mathcal{R}$ is left-linear, then the finite set $CO_\mathcal{R}$ of co-patterns of $\mathcal{R}$ is computable [8]: A ground basic term s is irreducible if and only if there exists a term t in $CO_\mathcal{R}$ such that $s = t\theta$ for some substitution θ.

We add rewrite rules for the co-patterns of $\mathcal{R}_\Phi$ as follows:

$$\mathcal{R}^{co}_\Phi = \mathcal{R}_\Phi \cup \{\ t \to \mathsf{false} \mid t \in CO_\mathcal{R}\ \}$$

Example 3.3 From $\mathcal{R}_{\Phi_1}$ in Example 3.2, we obtain the TRS $\mathcal{R}^{co}_{\Phi_1} = \mathcal{R}_{\Phi_1} \cup \{\ \mathsf{O}(0) \to \mathsf{false}\ \}$.

[2] We have $\Phi_1 \models \neg\mathsf{O}(0)$, and thus, we need a reduction for $\Phi_1 \not\models \mathsf{O}(0)$.

[3] A term is *strongly irreducible* w.r.t. a TRS $\mathcal{R}$ if no ground normalized instance of the term is reducible [9].

3.5 Properties of the Resulting TRSs

The resulting TRS $\mathcal{R}_\Phi$ for a conclusion-linear inductive definition set Φ has the following properties.

Proposition 3.4 *All of the following hold:*

- $\mathcal{R}_\Phi^{co}$ *and* $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$ *are left-linear constructor systems such that* $\mathcal{D}_{\mathcal{R}_\Phi^{co}} = \{P \mid P$ *is an inductive predicate of* $\Phi\}$ *and* $\mathcal{D}_{\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}} = \mathcal{D}_{\mathcal{R}_\Phi^{co}} \cup \mathcal{D}_{\mathcal{R}_{PL}}$*, respectively, and*
- $\mathcal{R}_\Phi^{co}$ *and* $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$ *are quasi-reductive.*

Proof. Trivial by definition. □

Lemma 3.5 *For any ground atomic formula A, both of the following hold:*

(a) $\Phi \models A$ *if and only if* $A \to^*_{\mathcal{R}_\Phi \cup \mathcal{R}_{PL}} \mathsf{true}$ *(i.e.,* $A \to^*_{\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}} \mathsf{true}$*), and*

(b) $\Phi \not\models A$ *if A has no normal form of* $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$*.*

Proof.

(a) (Sketch) The *if* and *only-if* parts can straightforwardly be proved by induction on the length of reduction sequences and the depth of recursion of $\Phi \models$, respectively.

(b) We proceed by contradiction. Assume that $\Phi \models A$ and A has no normal form of $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$. It follows from (a) that $A \to^*_{\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}} \mathsf{true}$, and hence A has a normal form of $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$. This contradicts the assumption. □

Lemma 3.6 *Suppose that* $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$ *is ground confluent. Then, for any ground atomic formula A, both of the following hold:*

(a) $\Phi \not\models A$ *if* $A \to^*_{\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}} \mathsf{false}$*, and*

(b) if $\Phi \not\models A$*, then either A has no normal form of* $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$ *or* $A \to^*_{\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}} \mathsf{false}$.

Proof.

(a) Assume that $A \to^*_{\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}} \mathsf{false}$. It follows from ground confluence of $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$ that $A \not\to^*_{\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}} \mathsf{true}$. Therefore, it follows from Lemma 3.5 (a) that $\Phi \not\models A$.

(b) Assume that $\Phi \not\models A$. Then, it follows from Lemma 3.5 (a) that $A \not\to^*_{\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}} \mathsf{true}$. Since $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$ is quasi-reductive (Proposition 3.4), the normal forms with sort *bool* are true and false. Therefore, $A \to^*_{\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}} \mathsf{false}$ or A has no normal form of $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$. □

Lemma 3.7 *Suppose that* $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$ *is ground terminating and ground confluent. Then, for any ground formula F, both of the following hold:*

- $\Phi \models F$ *if and only if* $\check{F} \to^*_{\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}} \mathsf{true}$*, and*
- $\Phi \not\models F$ *if and only if* $\check{F} \to^*_{\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}} \mathsf{false}$.

Proof. (Sketch) Using Lemmas 3.5 and 3.6, the *if* and *only-if* parts of both claims can straightforwardly be proved mutually by structural induction on F. □

Lemma 3.8 *Suppose that* Φ *is conclusion-linear and consistent. Then, all of the following hold:*

- *every ground term has at most a normal form of* $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$*, and*
- *if* $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$ *is ground terminating, then* $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$ *is ground confluent.*

Proof. Since the second claim is an immediate consequence of the first claim, we only show the first claim. Assume that there exists a ground term t that has two or more normal forms. Since we have only rewrite rules for *bool*, the ground term t has the sort *bool*. Since $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$ is quasi-reductive, the normal forms of t are true and false. Let F be a formula such that $\check{F} = t$. Then, it follows from Lemma 3.7 that $\Phi \models F$ and $\Phi \not\models F$. This contradicts consistency of Φ. □

Note that if Φ is conclusion-linear and $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$ is ground confluent, then Φ is consistent.

Theorem 3.9 *Suppose that* Φ *is consistent and* $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$ *is ground terminating. A sequent* $\Gamma \vdash \Delta$ *is valid w.r.t.* Φ *if and only if* $\mathsf{seq}(\widetilde{\Gamma}, \widehat{\Delta}) \approx \mathsf{true}$ *is an inductive theorem of* $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$*.*

Proof. It follows from Proposition 3.4 that $\mathcal{D}_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} = \mathcal{D}_{\mathcal{R}^{co}_{\Phi}} \cup \{\mathsf{and},\ \mathsf{or},\ \mathsf{not},\ \mathsf{seq}\}$, and thus, ρ is a ground substitution for formulas if and only if ρ is a ground constructor substitution for $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$.

We first show the *only-if* part. Let ρ be a ground constructor substitution with $\mathcal{D}om(\rho) \supseteq \mathcal{V}ar(\widetilde{\Gamma}, \widehat{\Delta})$. Then, ρ is a ground substitution for $\Gamma \vdash \Delta$ with $\mathcal{D}om(\rho) \supseteq \mathcal{V}ar(\Gamma, \Delta)$. Since $\Gamma \vdash \Delta$ is valid w.r.t. Φ, we have that $\Phi \models \neg(\bigwedge_{F \in \Gamma} F)\rho \vee (\bigvee_{F' \in \Delta} F')\rho$. We make a case distinction depending on whether $\Phi \models \neg(\bigwedge_{F \in \Gamma} F)\rho$ holds or not.

- Consider the case where $\Phi \models \neg(\bigwedge_{F \in \Gamma} F)\rho$. Then, we have that $\Phi \not\models (\bigwedge_{F \in \Gamma} F)\rho$. It follows from Lemma 3.7 that $\widetilde{\Gamma}\rho \to^*_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} \mathsf{false}$, and hence $\mathsf{seq}(\widetilde{\Gamma}, \widehat{\Delta})\rho \to^*_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} \mathsf{seq}(\mathsf{false}, t_2) \to_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} \mathsf{true}$.
- Consider the remaining case where $\Phi \not\models \neg(\bigwedge_{F \in \Gamma} F)\rho$. Then, we have that $\Phi \models (\bigwedge_{F \in \Gamma} F)\rho$ and $\Phi \models (\bigvee_{F' \in \Delta} F')\rho$. It follows from Lemma 3.7 that $\widetilde{\Gamma}\rho \to^*_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} \mathsf{true}$ and $\widehat{\Delta}\rho \to^*_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} \mathsf{true}$, and hence $\mathsf{seq}(\widetilde{\Gamma}, \widehat{\Delta})\rho \to^*_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} \mathsf{seq}(\mathsf{true}, \mathsf{true}) \to_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} \mathsf{true}$.

Therefore, $\mathsf{seq}(\widetilde{\Gamma}, \widehat{\Delta}) \approx \mathsf{true}$ is an inductive theorem of $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$.

Next, we show the *if* part. Let ρ be a ground substitution for $\Gamma \vdash \Delta$ with $\mathcal{D}om(\rho) \supseteq \mathcal{V}ar(\Gamma, \Delta)$. Then, ρ is a ground constructor substitution with $\mathcal{D}om(\rho) \supseteq \mathcal{V}ar(\widetilde{\Gamma}, \widehat{\Delta})$. Since $\mathsf{seq}(\widetilde{\Gamma}, \widehat{\Delta}) \approx \mathsf{true}$ is an inductive theorem of $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$, we have that $\mathsf{seq}(\widetilde{\Gamma}, \widehat{\Delta})\rho \leftrightarrow^*_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} \mathsf{true}$. It follows from Lemma 3.8 that $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$ is ground confluent, and hence, $\mathsf{seq}(\widetilde{\Gamma}, \widehat{\Delta})\rho \to^*_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} \mathsf{seq}(t_1, t_2) \to_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} \mathsf{true}$, where either $t_1 = \mathsf{false}$ or $t_1 = t_2 = \mathsf{true}$. Thus, $\widetilde{\Gamma}\rho \to^*_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} t_1$ and $\widehat{\Delta}\rho \to^*_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} t_2$. We make a case distinction depending on t_1.

- Consider the case where $t_1 = \mathsf{false}$. Since $\widetilde{\Gamma}\rho \to^*_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} t_1 = \mathsf{false}$, it follows from Lemma 3.7 that $\Phi \not\models (\bigwedge_{F \in \Gamma} F)\rho$, and hence $\Phi \models \neg(\bigwedge_{F \in \Gamma} F)\rho \vee (\bigvee_{F' \in \Delta} F')\rho$.
- Consider the remaining case where $t_1 = t_2 = \mathsf{true}$. Since $\widehat{\Delta}\rho \to^*_{\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}} t_2 = \mathsf{true}$, it follows from Lemma 3.7 that $\Phi \models (\bigvee_{F' \in \Delta} F')\rho$, and hence $\Phi \models \neg(\bigwedge_{F \in \Gamma} F)\rho \vee (\bigvee_{F' \in \Delta} F')\rho$.

Therefore, $\neg(\bigwedge_{F \in \Gamma} F) \vee (\bigvee_{F' \in \Delta} F')$ is valid w.r.t. Φ, and hence $\Gamma \vdash \Delta$ is valid w.r.t. Φ. □

4 Termination of the Resulting TRSs

In this section, we discuss termination of $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$, which is necessary for the correctness of $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$.

A given inductive definition set Φ defines inductive predicates *inductively* and must often be terminating in the sense of the computation of $\Phi \models \cdot$, while Φ is not necessary to be so. In our setting, $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$ is expected to be terminating. Using termination proof techniques for TRSs, termination of $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$ can be examined after the transformation. For our ultimate goal which is the comparison of RI and cyclic proof systems, we would like to ensure termination of $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$ for arbitrary Φ that is terminating. On the other hand, one may specify a non-terminating inductive definition set. For this reason, as a criterion to ensure termination of $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$, we consider an inductive definition set Φ such that the TRS $\{A \to A_i \mid \frac{A_1 \ \cdots \ A_n}{A} \in \Phi,\ 1 \le i \le n\}$ is GSC-terminating.

The dependency pairs of $\mathcal{R}_\Phi^{co}$ have the following property w.r.t. Φ.

Lemma 4.1 *$DP(\{A \to A_i \mid \frac{A_1 \ \cdots \ A_n}{A} \in \Phi,\ 1 \le i \le n\}) = DP(\mathcal{R}_\Phi^{co})$ up to variable-renaming.*

Proof. By definition, we have that

- $DP(\{A \to A_i \mid \frac{A_1 \ \cdots \ A_n}{A} \in \Phi,\ 1 \le i \le n\}) = \{A^\# \to A_i^\# \mid \frac{A_1 \ \cdots \ A_n}{A} \in \Phi,\ 1 \le i \le n\}$, and
- $DP(\mathcal{R}_\Phi^{co}) = DP(\mathcal{R}_\Phi) = \{A^\# \to A_i^\# \mid A \to \{\widetilde{A_1,\ldots,A_n}\} \in \mathcal{R}_\Phi, 1 \le i \le n\}$.

Therefore, the claim holds. □

By definition, it is clear that $DP(\mathcal{R}_{PL}) = \emptyset$. For a directed graph $\mathcal{G} = (V,E)$, we denote by $Cycles(\mathcal{G})$ the family of node sets that are nodes of cycles of $\mathcal{G}$:

$$Cycles(\mathcal{G}) = \{V' \mid \text{there exists a set of edges } E' \text{ such that } (V',E') \text{ is a cycle of } \mathcal{G}\ \}$$

The dependency graphs of $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$ have the following property w.r.t. Φ.

Lemma 4.2 *$Cycles(DG(\{A \to A_i \mid \frac{A_1 \ \cdots \ A_n}{A} \in \Phi,\ 1 \le i \le n\})) = Cycles(DG(\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}))$ up to variable-renaming.*

Proof. It follows from Lemma 4.1 that $DG(\{A \to A_i \mid \frac{A_1 \ \cdots \ A_n}{A} \in \Phi,\ 1 \le i \le n\}) = DG(\mathcal{R}_\Phi^{co})$. It follows from $DP(\mathcal{R}_{PL}) = \emptyset$ that $Cycles(DG(\mathcal{R}_\Phi^{co})) = Cycles(DG(\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}))$. Therefore, the claim holds. □

Lemma 4.3 *Let $\mathcal{R}_1$ and $\mathcal{R}_2$ be TRSs such that $Cycles(DG(\mathcal{R}_1)) = Cycles(DG(\mathcal{R}_2))$. $\mathcal{R}_1$ is GSC-terminating if and only if $\mathcal{R}_2$ is so.*

Proof. Let $\mathcal{D}_i^\#$ be the set of marked symbols in $Cycles(DG(\mathcal{R}_i))$ for $i = 1,2$. It follows from the assumption that $\mathcal{D}_1^\# = \mathcal{D}_2^\#$. It suffices to show the *if* part. Assume that $\mathcal{R}_2$ is GSC-terminating. Let $\mathcal{P}$ be a cycle in $DG(\mathcal{R}_1)$. Then, by the assumption, $\mathcal{P}$ is a cycle of $DG(\mathcal{R}_2)$. Since $\mathcal{R}_2$ is GSC-terminating, there exists a multi-projection π for $\mathcal{D}_2^\#$ $(= \mathcal{D}_1^\#)$ such that $\mathcal{P} \subseteq \unrhd^\pi$ and $\mathcal{P} \cap \rhd^\pi \neq \emptyset$. Therefore, $\mathcal{R}_1$ is GSC-terminating. □

The following theorem is a direct consequence of Lemmas 4.2 and 4.3.

Theorem 4.4 *The TRS $\{A \to A_i \mid \frac{A_1 \ \cdots \ A_n}{A} \in \Phi,\ 1 \le i \le n\}$ is GSC-terminating if and only if $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$ is so.*

Example 4.5 Since $\{\mathsf{E}(\mathsf{s}(x)) \to \mathsf{O}(x),\ \mathsf{O}(\mathsf{s}(x)) \to \mathsf{E}(x),\ \mathsf{N}(\mathsf{s}(x)) \to \mathsf{N}(x)\}$ is GSC-terminating, it follows from Theorem 4.4 that $\mathcal{R}_{\Phi_1}^{co} \cup \mathcal{R}_{PL}$ is GSC-terminating.

Corollary 4.6 (correctness of $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$) *Suppose that Φ is conclusion-linear and consistent, and the TRS $\{A \to A_i \mid \frac{A_1 \ \cdots \ A_n}{A} \in \Phi,\ 1 \le i \le n\}$ is GSC-terminating. A sequent $\Gamma \vdash \Delta$ is valid w.r.t. Φ if and only if $\mathsf{seq}(\widetilde{\Gamma},\widehat{\Delta}) \approx \mathsf{true}$ is an inductive theorem of $\mathcal{R}_\Phi^{co} \cup \mathcal{R}_{PL}$.*

5 Conclusion

In this paper, we showed a transformation from a conclusion-linear consistent inductive definition set Φ into a TRS $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$. We also showed that if the TRS $\{A \to A_i \mid \frac{A_1 \ \cdots \ A_n}{A} \in \Phi,\ 1 \le i \le n\}$ is GSC-terminating, then $\mathcal{R}^{co}_{\Phi} \cup \mathcal{R}_{PL}$ is ground terminating and ground confluent. The results in this paper enable us to establish the common setting for RI and cyclic proof systems, and thus, we are ready for transformations between them.

As future work, we will relax the conclusion-linear and termination assumptions. We will extend the results in this paper to ordinary predicates, transforming inductive definition sets with ordinary and inductive predicates into LCTRSs. For our ultimate goal, we will transform cyclic proofs and RI proofs each other in order to compare cyclic proof systems and RI.

References

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

[2] James Brotherston (2005): *Cyclic Proofs for First-Order Logic with Inductive Definitions*. In: *Proc. TABLEAUX 2005*, *LNCS* 3702, Springer, pp. 78–92, doi:10.1007/11554554_8.

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

[4] Gilles Dowek (2011): *Proofs and Algorithms: An Introduction to Logic and Computability*. UTiCS, Springer, doi:10.1007/978-0-85729-121-9.

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

[6] Jürgen Giesl, Thomas Arts & Enno Ohlebusch (2002): *Modular Termination Proofs for Rewriting Using Dependency Pairs*. *J. Symb. Comput.* 34(1), pp. 21–58, doi:10.1006/jsco.2002.0541.

[7] Cynthia Kop & Naoki Nishida (2013): *Term Rewriting with Logical Constraints*. In: *Proc. FroCoS 2013*, *LNCS* 8152, pp. 343–358, doi:10.1007/978-3-642-40885-4_24.

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

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

[10] Uday S. Reddy (1990): *Term Rewriting Induction*. In: *Proc. CADE 1990*, *LNCS* 449, Springer, pp. 162–177, doi:10.1007/3-540-52885-7_86.

[11] John C. Reynolds (2002): *Separation Logic: A Logic for Shared Mutable Data Structures*. In: *Proc. LICS 2002*, IEEE Computer Society, pp. 55–74, doi:10.1109/LICS.2002.1029817.

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

[13] Akihisa Yamada, Christian Sternagel, René Thiemann & Keiichirou Kusakari (2016): *AC Dependency Pairs Revisited*. In: *Proc. CSL 2016*, *LIPIcs* 62, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 8:1–8:16, doi:10.4230/LIPIcs.CSL.2016.8.