The failure of cut-elimination in cyclic proof for first-order logic with inductive definitions

Yukihiro Oda∗ James Brotherston† Makoto Tatsuta‡

February 15, 2024

A cyclic proof system is a proof system whose proof figure is a tree with cycles. The cut-elimination in a proof system is fundamental. It is conjectured that the cut-elimination in the cyclic proof system for first-order logic with inductive definitions does not hold. This paper shows that the conjecture is correct by giving a sequent not provable without the cut rule but provable in the cyclic proof system.

1 Introduction

A cyclic proof system, or a circular proof system, is a proof system whose proof figure is a tree with cycles [4]. Such proof systems have been used to formalize several logics and theories, such as modal μ-calculus [25, 24, 1], linear time μ-calculus [10, 15, 17], linear logic with fixed points [14, 18], Gödel-Löb provability logic [21], first-order μ-calculus [23], first-order logic with inductive definitions [6, 5, 2], arithmetic [22, 3], bunched logic [7], separation logic [8, 16, 19, 27], and Kleene algebra [13]. Cyclic proofs are also useful for software verification, including verifying properties of concurrent processes [20], termination of pointer programs [8], and decision procedures for symbolic heaps [9, 12, 26, 27].

The cut-elimination property is a fundamental property of a proof system. For example, the cut-elimination theorem for first-order logic immediately implies consistency, the subformula property and Craig’s interpolation theorem (see [11]).

Despite its importance, it was an open problem whether the cut-elimination property in the cyclic proof system for first-order logic with inductive definitions holds. In Conjecture 5.2.4. of [4], Brotherston has conjectured that the cut-elimination property in the system does not hold.

This paper presents a counterexample to cut-elimination in the cyclic proof system for first-order logic with inductive definitions. In other words, we show that the conjecture is correct. Our counterexample is a sequent that says an addition predicate implies another addition predicate with a different definition. In order to show it is not cut-free provable, under the assumption that it is cut-free provable, we construct an infinite sequence of sequents in a finite cyclic proof figure, which leads to a contradiction. For this purpose, we use cycle-normalization [4], and we also give a simpler proof of it.

There exist cut-free and complete cyclic proof systems for some logics and theories, including modal μ-calculus [24, 1], linear time μ-calculus [10], Gödel-Löb provability logic [21] and Kleene algebra [13].

Kimura et al. [16] give a counterexample to cut-elimination in cyclic proofs for separation logic. They also suggest their proof technique cannot be applied to show a counterexample to cut-elimination in a cyclic proof system with contraction and weakening on antecedents [16]. Their proof technique is to give a path in a cut-free proof that contradicts a soundness condition. In the counterexample we give in this article, constructing such a path seems complicated because the system we consider is a system with contraction and weakening of antecedents.

Section 2 describes the language for first-order logic with inductive definitions. Section 3 introduces Brotherston’s cyclic proof system CLKID⊙. Section 4 gives a sequent that is not cut-free provable in CLKID⊙ but provable in CLKID⊙.*

*Department of Informatics, The Graduate University for Advanced Studies, yukihiro_m [at] nii.ac.jp
†Department of Computer Science, University College London
‡National Institute of Informatics / Sokendai
2 Language

In this section, we give the syntax of a language for first-order logic with inductive definitions. The language
is the same as that given in [5].

Terms are defined by

\[ t ::= x \mid f(t_1, \ldots, t_n) \]

where \( x \) is a variable symbol and \( f \) is an \( n \)-ary function symbol. We write \( x \) for a sequence of variables and \( u(x) \) for a sequence of terms in which the variables \( x \) occur.

Predicate symbols consist of ordinary predicate symbols, denoted by \( Q_1, Q_2, \ldots \), and inductive predicate
symbols, denoted by \( P_1, \ldots, P_n \). Inductive predicate symbols are given with an inductive definition set, which
we define later. We assume that inductive predicate symbols are finite.

An atomic formula is defined as \( t_1 = t_2 \) or \( R(t_1, \ldots, t_n) \) where \( t_1, t_2, \ldots, t_n \) are terms and \( R \) is an \( n \)-ary
predicate symbol. \( Q_1u \) denotes \( Q_1(u) \), where \( u \) is a sequence of terms. Formulas are defined by

\[ \varphi ::= A \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \exists x \varphi \mid \forall x \varphi, \]

where \( A \) is an atomic formula and \( x \) is a variable. We define free variables as usual, and \( \text{FV}(\varphi) \) is defined
as the set of free variables in a formula \( \varphi \). We write \( \varphi[x_0 := t_0, \ldots, x_r := t_r] \) for a formula obtained from a
formula \( \varphi \) by simultaneously substituting terms \( t_0, \ldots, t_r \) for variables \( x_0, \ldots, x_r \), respectively. We sometimes
write \( \theta \) for \( x_0 := t_0, \ldots, x_r := t_r \).

Definition 1 (Inductive definition set). A production is defined as

\[ Q_1u_1 \cdots Q_hu_h \quad P_{j_1}t_1 \cdots P_{j_m}t_m, \]

where \( Q_1u_1, \ldots, Q_hu_h \) are atomic formulas with ordinal predicate symbols and \( P_{j_1}t_1, \ldots, P_{j_m}t_m \) and \( P_1t \) are
atomic formulas with inductive predicate symbols.

The formulas above the line of a production are called the assumptions of the production. The formula
under the line of a production is called the conclusion of the production. An inductive definition set is a finite
set of productions.

Definition 2 (Sequent). A sequent is a pair of finite sets of formulas, denoted by \( \Gamma \vdash \Delta \), where \( \Gamma \), \( \Delta \) are finite
sets of formulas. \( \Gamma \) is called the antecedent of \( \Gamma \vdash \Delta \) and \( \Delta \) is called the consequent of \( \Gamma \vdash \Delta \).

For a set of formulas \( \Gamma \), we define \( \text{FV}(\Gamma) \) as the set of free variables of formulas in \( \Gamma \).

The semantics of inductive predicates is given by the least fixed point of a monotone operator constructed
from the inductive definition set [5]. Since we do not use semantics in this paper, we do not discuss it in detail.

We write \( s^n.x \) for \( s \cdots s.x \).

3 Cyclic proof system CLKID\( \omega \) for first-order logic with inductive
definitions

In this section, we define a cyclic proof system CLKID\( \omega \) for first-order logic with inductive definitions. To define
it, we also define an infinitary proof system LKID\( \omega \) with the same language. Then, CLKID\( \omega \) is understood as
the subsystem LKID\( \omega \). These systems are the same as CLKID\( \omega \) and LKID\( \omega \) defined in [4, 5].

3.1 Inference rules of CLKID\( \omega \)

This section gives the inference rules of CLKID\( \omega \). The inference rules except for rules of inductive predicates are given in Figure 1. The principal formula of a rule is the distinguished formula introduced by the rule in its conclusion. We use the commas in sequents for a set union. The contraction rule is implicit.

We present the two inference rules for inductive predicates. First, for each production

\[ Q_1u_1(x) \cdots Q_hu_h(x) \quad P_{j_1}t_1(x) \cdots P_{j_m}t_m(x) \quad P_1t(x), \]

where \( x \) is a variable symbol and \( f \) is an \( n \)-ary function symbol. We write \( x \) for a sequence of variables and \( u(x) \) for a sequence of terms in which the variables \( x \) occur.

Predicate symbols consist of ordinary predicate symbols, denoted by \( Q_1, Q_2, \ldots \), and inductive predicate
symbols, denoted by \( P_1, \ldots, P_n \). Inductive predicate symbols are given with an inductive definition set, which
we define later. We assume that inductive predicate symbols are finite.

An atomic formula is defined as \( t_1 = t_2 \) or \( R(t_1, \ldots, t_n) \) where \( t_1, t_2, \ldots, t_n \) are terms and \( R \) is an \( n \)-ary
predicate symbol. \( Q_1u \) denotes \( Q_1(u) \), where \( u \) is a sequence of terms. Formulas are defined by

\[ \varphi ::= A \mid \neg \varphi \mid \varphi \land \varphi \mid \varphi \lor \varphi \mid \exists x \varphi \mid \forall x \varphi, \]

where \( A \) is an atomic formula and \( x \) is a variable. We define free variables as usual, and \( \text{FV}(\varphi) \) is defined
as the set of free variables in a formula \( \varphi \). We write \( \varphi[x_0 := t_0, \ldots, x_r := t_r] \) for a formula obtained from a
formula \( \varphi \) by simultaneously substituting terms \( t_0, \ldots, t_r \) for variables \( x_0, \ldots, x_r \), respectively. We sometimes
write \( \theta \) for \( x_0 := t_0, \ldots, x_r := t_r \).

Definition 1 (Inductive definition set). A production is defined as

\[ Q_1u_1 \cdots Q_hu_h \quad P_{j_1}t_1 \cdots P_{j_m}t_m, \]

where \( Q_1u_1, \ldots, Q_hu_h \) are atomic formulas with ordinal predicate symbols and \( P_{j_1}t_1, \ldots, P_{j_m}t_m \) and \( P_1t \) are
atomic formulas with inductive predicate symbols.

The formulas above the line of a production are called the assumptions of the production. The formula
under the line of a production is called the conclusion of the production. An inductive definition set is a finite
set of productions.

Definition 2 (Sequent). A sequent is a pair of finite sets of formulas, denoted by \( \Gamma \vdash \Delta \), where \( \Gamma \), \( \Delta \) are finite
sets of formulas. \( \Gamma \) is called the antecedent of \( \Gamma \vdash \Delta \) and \( \Delta \) is called the consequent of \( \Gamma \vdash \Delta \).

For a set of formulas \( \Gamma \), we define \( \text{FV}(\Gamma) \) as the set of free variables of formulas in \( \Gamma \).

The semantics of inductive predicates is given by the least fixed point of a monotone operator constructed
from the inductive definition set [5]. Since we do not use semantics in this paper, we do not discuss it in detail.

We write \( s^n.x \) for \( s \cdots s.x \).

3 Cyclic proof system CLKID\( \omega \) for first-order logic with inductive
definitions

In this section, we define a cyclic proof system CLKID\( \omega \) for first-order logic with inductive definitions. To define
it, we also define an infinitary proof system LKID\( \omega \) with the same language. Then, CLKID\( \omega \) is understood as
the subsystem LKID\( \omega \). These systems are the same as CLKID\( \omega \) and LKID\( \omega \) defined in [4, 5].

3.1 Inference rules of CLKID\( \omega \)

This section gives the inference rules of CLKID\( \omega \). The inference rules except for rules of inductive predicates are given in Figure 1. The principal formula of a rule is the distinguished formula introduced by the rule in its conclusion. We use the commas in sequents for a set union. The contraction rule is implicit.

We present the two inference rules for inductive predicates. First, for each production

\[ Q_1u_1(x) \cdots Q_hu_h(x) \quad P_{j_1}t_1(x) \cdots P_{j_m}t_m(x) \quad P_1t(x), \]
Structural rules:

\[
\frac{\Gamma \vdash \Delta}{(Axiom) (\Gamma \cap \Delta \neq \emptyset)}
\]

\[
\frac{\Gamma \vdash \varphi, \Delta, \Gamma, \varphi \vdash \Delta}{(Cut)}
\]

Logical rules:

\[
\frac{\Gamma \vdash \varphi, \Delta}{\Gamma, \neg \varphi \vdash \Delta} (\neg L)
\]

\[
\frac{\Gamma, \psi \vdash \Delta}{\Gamma, \psi \vee \psi \vdash \Delta} (\vee L)
\]

\[
\frac{\Gamma, \varphi, \psi \vdash \Delta}{\Gamma, \varphi \wedge \psi \vdash \Delta} (\wedge L)
\]

\[
\frac{\Gamma \vdash \varphi, \Delta}{\Gamma, \varphi \rightarrow \psi \vdash \Delta} (\rightarrow L)
\]

\[
\frac{\Gamma, \varphi \vdash \Delta}{\Gamma, \forall x \varphi \vdash \Delta} (\forall L)
\]

\[
\frac{\Gamma, \varphi \vdash \Delta}{\Gamma, \exists x \varphi \vdash \Delta} (\exists L) (x \notin \text{FV}(\Gamma \cup \Delta))
\]

\[
\frac{\Gamma [x := t, y := t] \vdash \Delta [x := t, y := t]}{\Gamma \vdash \Delta} (= L)
\]

\[
\frac{\Gamma [x := u, y := t] \vdash \Delta [x := u, y := t]}{\Gamma \vdash \Delta} (= R)
\]

\[
\frac{\Gamma \vdash \varphi \cdot \Delta}{\Gamma \vdash \varphi' \cdot \Delta'} (\text{Weak}) (\Gamma' \subseteq \Gamma, \Delta' \subseteq \Delta)
\]

\[
\frac{\Gamma \vdash \Delta}{\Gamma \vdash \Delta [\theta]} (\text{Subst})
\]

\[
\frac{\Gamma \vdash \varphi, \Delta}{\Gamma \vdash \varphi, \psi, \Delta} (\neg R)
\]

\[
\frac{\Gamma \vdash \varphi, \psi, \Delta}{\Gamma \vdash \varphi \vee \psi, \Delta} (\vee R)
\]

\[
\frac{\Gamma \vdash \varphi \cdot \Delta}{\Gamma \vdash \varphi \wedge \psi, \Delta} (\wedge R)
\]

\[
\frac{\Gamma \vdash \varphi \cdot \psi, \Delta}{\Gamma \vdash \varphi \rightarrow \psi, \Delta} (\rightarrow R)
\]

\[
\frac{\Gamma \vdash \varphi \cdot \Delta}{\Gamma \vdash \forall x \varphi, \Delta} (\forall R) (x \notin \text{FV}(\Gamma \cup \Delta))
\]

\[
\frac{\Gamma \vdash \varphi \cdot \Delta}{\Gamma \vdash \exists x \varphi, \Delta} (\exists R)
\]

\[
\frac{\Gamma [x := u, y := t] \vdash \Delta [x := u, y := t]}{\Gamma \vdash \Delta} (= L)
\]

\[
\frac{\Gamma \vdash \Delta [x := t, y := u]}{\Gamma \vdash t = u, \Delta} (= R)
\]

Figure 1: Inference rules except rules for inductive predicates
there is the inference rule

$$\frac{\Gamma \vdash Q_1 u_1(u), \Delta \quad \ldots \quad \Gamma \vdash Q_n u_n(u), \Delta \quad \Gamma \vdash P_j t_j(u), \Delta \quad \ldots \quad \Gamma \vdash P_m t_m(u), \Delta}{\Gamma \vdash P t(u), \Delta} \quad (P_i \text{ R})$$

Next, we define the left introduction rule for the inductive predicate. A *case distinctions* of $\Gamma, P u \vdash \Delta$ is defined as a sequent

$$\Gamma, u = t(y), Q_1 u_1(y), \ldots, Q_n u_n(y), P_j t_j(y), \ldots, P_m t_m(y) \vdash \Delta,$$

where $y$ is a sequence of distinct variables of the same length as $x$ and $y \not\in \text{FV}(\Gamma \cup \Delta \cup \{P u\})$ for all $y \in y$, and there is a production

$$\frac{Q_1 u_1(x) \quad \ldots \quad Q_n u_n(x) \quad P_j t_j(x) \quad \ldots \quad P_m t_m(x)}{P t(x)}$$

The inference rule (Case $P_i$) is

$$\frac{\text{All case distinctions of } \Gamma, P_i u \vdash \Delta}{\Gamma, P_i u \vdash \Delta} \quad \text{(Case } P_i \text{)}$$

The formulas $P_j t_j(y), \ldots, P_m t_m(y)$ in case distinctions are said to be *case-descendants* of the principal formula $P u$.

**Example 3.** Let $\text{Add}_1$, $\text{Add}_2$ be inductive predicates of arity three, $0$ be a constant symbol, and $s$ be a function symbol of arity one.

We define the productions of $\text{Add}_1$, $\text{Add}_2$ by

$$\begin{align*}
\text{Add}_1(0, y, y) \rightarrow, & \quad \text{Add}_1(x, y, z) \rightarrow, \\
\text{Add}_2(0, y, y) \rightarrow, & \quad \text{Add}_2(x, sy, z) \rightarrow.
\end{align*}$$

The inference rules for $\text{Add}_1$, $\text{Add}_2$ are

$$\frac{\Gamma \vdash \text{Add}_1(0, b, b), \Delta}{\Gamma \vdash \text{Add}_1(a, b, c), \Delta} \quad (\text{Add}_1 \text{ R}_1),$$

$$\frac{\Gamma \vdash \text{Add}_2(0, b, b), \Delta}{\Gamma \vdash \text{Add}_2(a, b, c), \Delta} \quad (\text{Add}_2 \text{ R}_1),$$

$$\frac{\Gamma, a = 0, b = y, c = y \vdash \Delta}{\Gamma, \text{Add}_1(a, b, c) \vdash \Delta} \quad \text{(Case } \text{Add}_1)$$

$$\frac{\Gamma, a = 0, b = y, c = y \vdash \Delta}{\Gamma, \text{Add}_2(a, b, c) \vdash \Delta} \quad \text{(Case } \text{Add}_2)$$

$(x, y, z \not\in \text{FV}(\Gamma \cup \Delta \cup \{\text{Add}_1(a, b, c)\})$ and $x, y, z$ are all distinct) and

$$\frac{\Gamma, a = 0, b = y, c = y \vdash \Delta}{\Gamma, \text{Add}_1(a, b, c) \vdash \Delta},$$

$$\frac{\Gamma, a = 0, b = y, c = y \vdash \Delta}{\Gamma, \text{Add}_2(a, b, c) \vdash \Delta} \quad \text{(Case } \text{Add}_2)$$

$(x, y, z \not\in \text{FV}(\Gamma \cup \Delta \cup \{\text{Add}_2(a, b, c)\})$ and $x, y, z$ are all distinct).

### 3.2 Infinitary proof system LKIDω

In this section, we define an infinitary proof system for first-order logic with inductive definitions LKIDω. The inference rules of LKIDω are the same as that of CLKIDω.

**Definition 4** (Derivation tree). Let Rule be the set of names for the inference rules of CLKIDω. Let Seq be the set of sequents. $\mathbb{N}^*$ denotes the set of finite sequences of natural numbers. We write $\langle n_1, \ldots, n_k \rangle$ for the sequence of the numbers $n_1, \ldots, n_k$. We write $\sigma_1 \sigma_2$ for the concatenation of $\sigma_1$ and $\sigma_2$ with $\sigma_1$, $\sigma_2 \in \mathbb{N}^*$. We write $\sigma n$ for $\sigma(n)$ for $\sigma \in \mathbb{N}^*$ and $n \in \mathbb{N}$. We define a *derivation tree* to be a partial function $\mathcal{D} : \mathbb{N}^* \rightarrow \text{Seq} \times \{\text{Rule} \cup \{\text{Bud}\}\}$ satisfying the following conditions:

1. $\text{dom}(\mathcal{D})$ is prefixed-closed, that is to say, if $\sigma_1 \sigma_2 \in \text{dom}(\mathcal{D})$ for $\sigma_1, \sigma_2 \in \mathbb{N}^*$, then $\sigma_1 \in \text{dom}(\mathcal{D})$.
2. If $\sigma n \in \text{dom}(\mathcal{D})$ for $\sigma \in \mathbb{N}^*$ and $n \in \mathbb{N}$, then $\sigma n \in \text{dom}(\mathcal{D})$ for any $m \leq n$.
3. Let $\mathcal{D}(\sigma) = (\Gamma_{\sigma} \vdash \Delta_{\sigma}, R_{\sigma})$.
   (a) If $R_{\sigma} = \text{Bud}$, then $\sigma 0 \notin \text{dom}(\mathcal{D})$.  

4
We sometimes write $(\Gamma_i \vdash \Delta_i)_{0 \leq i < \alpha}$ for the path $(\sigma_i)_{0 \leq i < \alpha}$ in a derivation tree $D$ if $D(\sigma_i) = (\Gamma_i \vdash \Delta_i, R_i)$.

**Definition 6** (Trace). For a path $(\Gamma_i \vdash \Delta_i)_{0 \leq i < \alpha}$ in a derivation tree $D$, we define a trace following $(\Gamma_i \vdash \Delta_i)_{0 \leq i < \alpha}$ to be a sequence of formulas $(\tau_i)_{0 \leq i < \alpha}$ such that the following hold:

1. $\tau_i$ is an atomic formula with an inductive predicate in $\Gamma_i$.
2. If $\Gamma_i \vdash \Delta_i$ is the conclusion of (Subst) with $\theta$, then $\tau_i \equiv \tau_{i+1}[\theta]$.
3. If $\Gamma_i \vdash \Delta_i$ is the conclusion of $(= L)$ with the principal formula $t = u$ and $\tau_i \equiv F[x := t, y := u]$, then $\tau_{i+1} \equiv F[x := u, y := t]$.
4. If $\Gamma_i \vdash \Delta_i$ is the conclusion of (Case $P_i$), then either
   - $\tau_i$ is the principal formula of the rule and $\tau_{i+1}$ is a case-descendant of $\tau_i$, or
   - $\tau_{i+1}$ is the same as $\tau_i$.

In the former case, $\tau_i$ is said to be a progress point of the trace.

5. If $\Gamma_i \vdash \Delta_i$ is the conclusion of any other rules, then $\tau_{i+1} \equiv \tau_i$.

**Definition 7** (Global trace condition). If a trace has infinitely many progress points, we call the trace an infinitely progressing trace. If there exists an infinitely progressing trace following a tail of the path $(\Gamma_i \vdash \Delta_i)_{i \geq k}$ with some $k \geq 0$ for every infinite path $(\Gamma_i \vdash \Delta_i)_{i \geq 0}$ in a derivation tree, we say the derivation tree satisfies the global trace condition.

**Definition 8** (LKID$^\omega$ pre-proof). We define an LKID$^\omega$ pre-proof to be a (possibly infinite) derivation tree $D$ without buds. When the root is $\Gamma \vdash \Delta$, we call $\Gamma \vdash \Delta$ the conclusion of the proof.

**Definition 9** (LKID$^\omega$ proof). We define an LKID$^\omega$ proof to be an LKID$^\omega$ pre-proof that satisfies the global trace condition.

Because of the global trace condition, the soundness of LKID$^\omega$ for the standard models hold [4, 5]. In other words, if there exists an LKID$^\omega$ proof of a sequent $\Gamma \vdash \Delta$, then $\Gamma \vdash \Delta$ is valid in any standard models. Moreover, cut-free completeness of LKID$^\omega$ for the standard models hold. In other words, if $\Gamma \vdash \Delta$ is valid in any standard models, there exists an LKID$^\omega$ cut-free proof of $\Gamma \vdash \Delta$ [4, 5].

### 3.3 Cyclic proof system CLKID$^\omega$

In this section, we introduce a cyclic proof system CLKID$^\omega$.

**Definition 10** (Companion). For a finite derivation tree $D$, we define the companion for a bud $b$ as an inner node $\sigma$ in $D$ with $(D(\sigma))_0 = (D(b))_0$.
Definition 11 (CLKID∞ pre-proof). We define a CLKID∞ pre-proof to be a pair \((D, \mathcal{C})\) such that \(D\) is a finite derivation tree and \(\mathcal{C}\) is a function mapping each bud to its companion. When the root is \(\Gamma \vdash \Delta\), we call \(\Gamma \vdash \Delta\) the conclusion of the proof.

Definition 12 (Tree-unfolding). For a CLKID∞ pre-proof \((D, \mathcal{C})\), a tree-unfolding \(T(D, \mathcal{C})\) of \((D, \mathcal{C})\) is recursively defined by

\[
T(D, \mathcal{C})(\sigma) = \begin{cases} 
D(\sigma), & \text{if } \sigma \in \text{dom}(D) \setminus \text{Bud}(D), \\
T(D, \mathcal{C})(\sigma_2), & \text{if } \sigma \notin \text{dom}(D) \setminus \text{Bud}(D) \text{ with } \sigma = \sigma_1 \sigma_2, \sigma_1 \in \text{Bud}(D) \text{ and } \sigma_3 = \mathcal{C}(\sigma_1),
\end{cases}
\]

where Bud\((D)\) is the set of buds in \(D\).

Note that a tree-unfolding is an LKID∞ pre-proof.

Definition 13 (CLKID∞ proof). We define a CLKID∞ proof of a sequent \(\Gamma \vdash \Delta\) to be a CLKID∞ pre-proof of \(\Gamma \vdash \Delta\) whose tree-unfolding satisfies the global trace condition. If a CLKID∞ proof of \(\Gamma \vdash \Delta\) exists, we say \(\Gamma \vdash \Delta\) is provable in CLKID∞. A CLKID∞ proof in which (Cut) does not occur is called cut-free. If a cut-free CLKID∞ proof of \(\Gamma \vdash \Delta\) exists, we say \(\Gamma \vdash \Delta\) is cut-free provable in CLKID∞.

Example 14. The derivation tree given in Figure 2 is the proof of Add\(_1\)(\(x_1, sy_1, z_1\)) \(\vdash\) Add\(_1\)(\(sx_1, y_1, z_1\)) in CLKID∞, where \((\ast)\) indicates the pairing of a companion with a bud and the underlined formulas are the infinitely progressing trace for the infinite path (some applying rules and some labels are omitted for limited space).

\[
\begin{align*}
0, & \vdash \text{Add}_1(0, y_1, y_1) \quad \text{(Add}_1 \text{ R}_1) \\
0, & \vdash \text{Add}_1(s0, y_1, sy_1) \quad \text{(Add}_1 \text{ R}_2) \\
0, & \vdash \text{Add}_1(s0, y_1, sy_1) \\
sy_1 &= y_2, \vdash \text{Add}_1(s0, y_1, sy_1) \\
x_1 &= 0, \\
sy_1 &= y_2, \vdash \text{Add}_1(sx_1, y_1, z_1) \\
z_1 &= y_2 \\
\end{align*}
\]

\[
\begin{align*}
(sy_1 = y_2) & \vdash \text{Add}_1(s0, y_1, sy_1) \\
\end{align*}
\]

\[
\begin{align*}
0, & \vdash \text{Add}_1(sx_1, y_1, z_1) \quad \text{(Case Add}_1) \\
0, & \vdash \text{Add}_1(sx_1, y_1, z_1) \\
\end{align*}
\]

Figure 2: A CLKID∞ proof

3.4 Cycle-normalization

This section proves cycle-normalization for CLKID∞. It is proved in [4], but we will give a much shorter proof.

A CLKID∞ pre-proof in which each companion is an ancestor of the corresponding bud is called cycle-normal. The following proposition states the cycle-normalization holds for CLKID∞.

Proposition 15. For a CLKID∞ pre-proof \((D, \mathcal{C})\), we have a CLKID∞ cycle-normal pre-proof \((D', \mathcal{C}')\) such that the tree-unfolding of \((D, \mathcal{C})\) is that of \((D', \mathcal{C}')\).

Proof. We write \(\sigma \subseteq \sigma'\) when \(\sigma\) is an initial segment of \(\sigma'\). We write \(|\sigma|\) for the length of a sequence \(\sigma\). We define \(D(\sigma)\) by \(D(\sigma)(\sigma) = D(\sigma_1), \overline{\sigma}\) as \(\{\sigma' \mid \sigma' \subseteq \sigma \in S\}\), and \(S^0\) as \(\{\sigma' \mid \sigma' \nsubseteq \sigma \in S\}\).

Let \(D_1\) be the tree unfolding of \((D, \mathcal{C})\).

Define

\[
S_1 = \{\sigma \in \text{dom}(D_1) \mid \exists \sigma' \subseteq \sigma(D(\sigma_1) = D(\sigma'), \forall \sigma_1 \subseteq \sigma \forall \sigma_2 \not\subseteq \sigma(D(\sigma_1) \neq D(\sigma_2)), \forall n \exists \sigma_1 \geq \sigma(\sigma_1 \in \text{dom}(D_1), |\sigma_1| \geq n)\},
\]

\[
S_2 = \{\sigma \in \text{dom}(D_1) \mid \sigma \notin \text{dom}(D_1), \forall \sigma \not\subseteq \sigma(\text{~}\sigma \notin S_1)\}.
\]

\(S_1\) is the set of nodes such that the node is on some infinite path and the node is of the smallest height on the path among nodes, each of which has some inner node of the same subtree. \(S_2\) is the set of leaf nodes of finite paths which are not cut by \(S_1\).
Define $D'$ by

$$D'(σ) = D_1(σ) \text{ if } σ \in (S_1)' \cup \overline{S_2},$$

$$D'(σ) = (Γ \vdash Δ, Bud) \text{ if } σ \in S_1, D_1(σ) = (Γ \vdash Δ, R).$$

Define $C'$ by $C'(σ) = σ'$ for a bud $σ$ of $D'$ where $σ' \not\subseteq σ, D(σ) = D(σ').$

We can show that $\text{dom}(D')$ is finite as follows. Since $\text{dom}(D') = S_1 \cup \overline{S_2}$, we have $\text{dom}(D') \subseteq \text{dom}(D_1).$ Since $D_1$ is finite-branching, $D'$ is so. Assume $\text{dom}(D')$ is infinite to show contradiction. By König’s lemma, there is some infinite path $(σ_i)$, such that $σ_i \in \text{dom}(D')$. Since $D_1$ is regular, the set $\{D(σ_i)\}_{i}$ is finite. Hence there are $j < k$ such that $D(σ_j) = D(σ_k)$. Take the smallest $k$ among such $k$’s. Then $σ_k \in S_1$. Hence $σ_{k+1} \not\subseteq S_1$. Hence $σ_{k+1} \not\subseteq \text{dom}(D')$, which contradicts.

Then $(D', C')$ is a CLKIDω cycle-normal pre-proof.

Define $D'_1$ as the tree-unfolding of $(D', C')$. We can show $D_1 = D'_1$ on $\text{dom}(D'_1)$ as follows.

Case 1 where for any $σ' \subseteq σ, σ' \not\subseteq S_1$. $D'_1(σ) = D'(σ) = D_1(σ)$.

Case 2 where there is some $σ_1 \subseteq σ$ such that $σ_1 \in S_1$. Let $σ_1σ_2$ be $σ$ and $σ_3$ be $C'(σ_1)$. Then $D_1(σ) = D_1(σ_1)σ_2 = D'(σ_1)σ_2 = D'_1(σ_1σ_2) = D'_1(σ_1σ_2)$ by the induction hypothesis, it is $D'_1(σ_1σ_2)$ by definition of $D'_1$, and it is $D'_1(σ)$.

We can show $\text{dom}(D_1) \subseteq \text{dom}(D'_1)$ as follows. By induction on $|σ|$, we will show $σ \in \text{dom}(D_1)$ implies $σ \in \text{dom}(D'_1)$. If $σ \in S_1 \cup \overline{S_2}$, then $σ \in \text{dom}(D'_1)$. Hence $σ \in \text{dom}(D'_1)$. If there is some $σ_1 \subseteq σ$ such that $σ_1 \in S_1$, then by letting $σ = σ_1σ_2$ and $σ_3 = C'(σ_1), D_1(σ) = D_1(σ_1σ_2)$ by definition of $C'$, by the induction hypothesis for $σ_3σ_2$ it is $D'_1(σ_1σ_2)$, and it is $D'_1(σ)$ by definition of $D'_1$. Hence we have shown $\text{dom}(D_1) \subseteq \text{dom}(D'_1)$.

Hence $D_1 = D'_1$.

4 A counterexample to cut-elimination in CLKIDω

In this section, we prove the following theorem, which is the main theorem.

Theorem 16. Let $0$ be a constant symbol, $s$ be a function symbol of arity one, and $\text{Add}_1$ and $\text{Add}_2$ be inductive predicates of arity three with the following productions:

| $\text{Add}_1(0, y, y)$ | $\text{Add}_2(x, y, z)$ | $\text{Add}_1(sx, y, syz)$ | $\text{Add}_2(0, y, y)$ | $\text{Add}_2(sx, y, z)$ |
|------------------------|------------------------|------------------------|------------------------|------------------------|
| (1) $\text{Add}_2(x, y, z) \vdash \text{Add}_1(x, y, z)$ is provable in CLKIDω. |
| (2) $\text{Add}_2(x, y, z) \vdash \text{Add}_1(x, y, z)$ is not cut-free provable in CLKIDω. |

This theorem means that $\text{Add}_2(x, y, z) \vdash \text{Add}_1(x, y, z)$ is a counterexample to cut-elimination in CLKIDω. Note that $\text{Add}_1$ and $\text{Add}_2$ in the theorem are the same predicates in Example 3.

4.1 The outline of the proof

Before proving the theorem, we outline our proof for (2) of the theorem.

Assume there exists a cut-free CLKIDω proof of $\text{Add}_2(x, y, z) \vdash \text{Add}_1(x, y, z)$. Because of a technical issue, we use CLKIDω, a cyclic proof system with the same provability as CLKIDω whose inference rules are the same as CLKIDω except for the rule $= L$ (Definition 17). We show that there exists a cut-free cycle-normal CLKIDω proof of the sequent (Proposition 18). Let $(D_{cl}, C_{cl})$ be the CLKIDω proof.

Next, we define the relation $\equiv_{Γ}$ for a sequent $Γ \vdash Δ$ to be the smallest congruence relation on terms containing $t_1 = t_2 \in Γ$ (Definition 19). Then, we define the index of $\text{Add}_2(a, b, c)$ in a sequent $Γ \vdash Δ$ (Definition 27). If there uniquely exists $n - m$ such that $n, m \in \mathbb{N}$ and $σ^n b \equiv_{Γ} σ^m b'$ for some $\text{Add}_1(a', b', c') \in Δ$, then the index of $\text{Add}_2(a, b, c)$ is defined as $m - n$. If $σ^n b \not\equiv_{Γ} σ^m b'$ for any $\text{Add}_1(a', b', c') \in Δ$ and any $n, m \in \mathbb{N}$, the index of $\text{Add}_2(a, b, c)$ is defined as $\bot$. The index of $\text{Add}_2(a, b, c)$ may be undefined, but the index is always defined in a special sequent, called an index sequent (Definition 28). A switching point is defined as a node that is the conclusion of (Case Add2) with the principal formula whose index is $\bot$ (Definition 30). An index path is defined as a path $⟨Γ_i \vdash Δ_i⟩_{0 < i < α}$ of $T((D_{cl}, C_{cl})$ such that $Γ_0 \vdash Δ_0$ is an index sequent and $Γ_i \vdash Δ_i$ is a switching point if $Γ_{i+1} \vdash Δ_{i+1}$ is the left assumption of $Γ_i \vdash Δ_i$ (Definition 31). Then, we have (1) the root...
is an index sequent. (2) Every sequent in an index path is an index sequent (Lemma 32). (3) There exists a switching point on an infinite index path (Lemma 34). (4) The rightmost path from an index sequent is infinite (Lemma 36).

At last, we show there exist infinite nodes in the derivation tree $D_{cf}$. Because of (1) and (4), the rightmost path from the root is an infinite index path. By (3), there exists a switching point on the path. Let $c_0$ be the node of the smallest height among such switching points. Let $a_0$ be the left assumption of $c_0$. By (2), the sequent of $a_0$ is an index sequent. By (4), the rightmost path from $a_0$ is infinite. Therefore, there exists a bud $b_0$ in the rightmost path from $a_0$. By (3) and the definition of $c_0$, there exists a switching point between $a_0$ and $b_0$. Let $c_1$ be the node of the smallest height among such switching points. The nodes $c_0$ and $c_1$ are distinct by their definitions. We repeat this process as in Figure 3. Finally, we get a set of infinite nodes $\{c_i \mid i \in \mathbb{N}\}$. This is a contradiction since the set of nodes of $D_{cf}$ is finite.

At last, we show there exist infinite nodes in the derivatio

![Figure 3: Construction of $(c_i)_{i \in \mathbb{N}}$](image)

### 4.2 Another cyclic proof system $CLKID^\omega_a$

We give some definitions and lemmas for proving (2) of Theorem 16. We consider a cyclic proof system $CLKID^\omega_a$, which is obtained by changing the left introduction rule for “=” slightly.

**Definition 17** ($CLKID^\omega_a$). $CLKID^\omega_a$ is the cyclic proof system obtained by replacing $(= L)$ with

$$
\Gamma[x := u, y := t] : t = u \vdash \Delta[x := u, y := t]
$$

(= L$_a$).

The principal formula of the rule of $(= L_a)$ is defined as $t = u$.

Definitions of derivation trees, companions, pre-proofs, proofs for $CLKID^\omega_a$ are similar to $CLKID^\omega$.

The provability of $CLKID^\omega_a$ is the same as that of $CLKID^\omega$, since $(= L)$ is derivable in $CLKID^\omega_a$ by

$$
\Gamma[x := u, y := t] : t = u \vdash \Delta[x := u, y := t]
$$

(Weak) $\Gamma[x := u, y := t] : t = u \vdash \Delta[x := u, y := t]$ (Weak) $\Gamma[x := u, y := t] : t = u \vdash \Delta[x := u, y := t]$ (Weak) $\Gamma[x := u, y := t] : t = u \vdash \Delta[x := u, y := t]$.  

$CLKID^\omega_a$ is necessary because of Lemma 33 (3). For $CLKID^\omega$, Lemma 33 (3) does not hold.

**Proposition 18.** If there exists a cut-free $CLKID^\omega_a$ proof of $\Gamma \vdash \Delta$, then there exists a cut-free cycle-normal $CLKID^\omega_a$ proof of $\Gamma \vdash \Delta$.

**Proof.** Let $P_0$ be a cut-free $CLKID^\omega_a$ proof of $\Gamma \vdash \Delta$. By Proposition 15, there exists a cycle-normal $CLKID^\omega_a$ pre-proof $P_1$ whose tree-unfolding is the same as that of $P_0$. Since the tree-unfolding of $P_1$ satisfies the global trace condition, $P_1$ is a cycle-normal cut-free $CLKID^\omega_a$ proof of $\Gamma \vdash \Delta$.  

8
A cut-free cycle-normal CLKID proof of $\Gamma \vdash \Delta$ is transformed into the CLKID proof of $\Gamma \vdash \Delta$ by replacing all applications from $(= L)$ to $(= L_3)$ and weakening. Since this replacement does not change the rules except $(= L)$ in the CLKID proof and the sequents of buns and companions, the obtained CLKID proof is cut-free and cycle-normal.

4.3 Assuming cut-free proof

In Sections 4.3, 4.4, 4.5 and 4.6, we assume there exists a cut-free CLKID proof of $Add_2(x, y, z) \vdash Add_1(x, y, z)$ for contradiction. By Proposition 18, there exists a cut-free cycle-normal CLKID proof of $Add_2(x, y, z) \vdash Add_1(x, y, z)$. We write $(\mathcal{D}_{cf}, \mathcal{C}_{cf})$ for a cut-free cycle-normal CLKID proof of $Add_2(x, y, z) \vdash Add_1(x, y, z)$.

Remark. Let $\Gamma \vdash \Delta$ be a sequent in $(\mathcal{D}_{cf}, \mathcal{C}_{cf})$. By induction on the height of sequents in $\mathcal{D}_{cf}$, we can easily show the following statements:

1. $\Gamma$ consists of only atomic formulas with $=$, Add.
2. $\Delta$ consists of only atomic formulas with Add.
3. A term in $\Gamma$ and $\Delta$ is of the form $s^n0$ or $s^n x$ with some variable $x$.
4. The possible rules in $(\mathcal{D}_{cf}, \mathcal{C}_{cf})$ are (Weak), (Subst), $(= L_3)$, (Case Add), (Add, R1) and (Add, R2).

By (3), without loss of generality, we can assume terms in Sections 4.4, 4.5 and 4.6 are of the form $s^n0$ or $s^n x$ with some variable $x$.

4.4 Equality in a sequent

In this section, we define the equality $\equiv_{\Gamma}$ in a sequent $\Gamma \vdash \Delta$ and show some properties.

Definition 19 ($\equiv_{\Gamma}$). For a set of formulas $\Gamma$, we define the relation $\equiv_{\Gamma}$ to be the smallest congruence relation on terms which satisfies the condition that $t_1 = t_2$ implies $t_1 \equiv_{\Gamma} t_2$.

Definition 20 ($\sim_{\Gamma}$). For a set of formulas $\Gamma$ and terms $t_1, t_2$, we define $t_1 \sim_{\Gamma} t_2$ by $s^n t_1 \equiv_{\Gamma} s^m t_2$ for some $n, m \in \mathbb{N}$.

For a term $t$, we define VC($t$) as a variable or a constant in $t$. Note that $\sim_{\Gamma}$ is a congruence relation and also note that $t \sim_{\Gamma} u$ if VC($t$) = VC($u$).

Lemma 21. Let $\Gamma$ be a set of formulas and $\theta$ be a substitution.

1. For any terms $t_1$ and $t_2$, $t_1[\theta] \equiv_{\Gamma[\theta]} t_2[\theta]$ if $t_1 \equiv_{\Gamma} t_2$.
2. For any terms $t_1$ and $t_2$, $t_1 \not\sim_{\Gamma} t_2$ if $t_1[\theta] \not\sim_{\Gamma[\theta]} t_2[\theta]$.

Proof. (1) We prove the statement by induction on the definition of $\equiv_{\Gamma}$. We only show the base case. Assume $t_1 = t_2 \in \Gamma$. Then, $t_1[\theta] = t_2[\theta] \in \Gamma[\theta]$. Thus, $t_1[\theta] \equiv_{\Gamma[\theta]} t_2[\theta]$.

(2) By Definition 20 and (1), we have the statement.

Lemma 22. Let $\Gamma$ be a set of formulas, $u_1, v_2$ be terms, $v_1, v_2$ be variables, $\Gamma_1 \equiv (\Gamma[v_1 := u_1, v_2 := u_2], u_1 = u_2)$, and $\Gamma_2 \equiv (\Gamma[v_1 := u_2, v_2 := u_1], u_1 = u_2)$.

1. For any terms $t_1$ and $t_2$, $t_1[v_1 := u_1, v_2 := u_2] \equiv_{\Gamma_1} t_2[v_1 := u_1, v_2 := u_2]$ if $t_1[v_1 := u_2, v_2 := u_1] \equiv_{\Gamma_2} t_2[v_1 := u_2, v_2 := u_1]$.
2. For any terms $t_1$ and $t_2$, $t_1[v_1 := u_1, v_2 := u_1] \not\equiv_{\Gamma_1} t_2[v_1 := u_2, v_2 := u_1]$ if $t_1[v_1 := u_1, v_2 := u_2] \not\equiv_{\Gamma_1} t_2[v_1 := u_1, v_2 := u_2]$.

Proof. (1) We prove the statement by induction on the definition of $\equiv_{\Gamma_2}$. We only show the base case. Assume $t_1[v_1 := u_2, v_2 := u_1] = t_2[v_1 := u_2, v_2 := u_1] \in \Gamma_2$ to show $t_1[v_1 := u_1, v_2 := u_2] \equiv_{\Gamma_1} t_2[v_1 := u_1, v_2 := u_2]$.

If $t_1[v_1 := u_2, v_2 := u_1] = t_2[v_1 := u_2, v_2 := u_1]$ is $u_1 = u_2$, then $t_1 = t_2$ is $v_1 = v_2$, $v_2 = u_1$, or $u_1 = u_2$. Therefore, $t_1[v_1 := u_1, v_2 := u_2] \equiv_{\Gamma_1} t_2[v_1 := u_1, v_2 := u_2]$.

Assume $t_1[v_1 := u_2, v_2 := u_1] = t_2[v_1 := u_2, v_2 := u_1]$ is not $u_1 = u_2$. By case analysis, we have $t_1 = t_2 \in \Gamma$. Hence, $t_1[v_1 := u_1, v_2 := u_2] = t_2[v_1 := u_1, v_2 := u_2] \in \Gamma_1$. Therefore, we have $t_1[v_1 := u_1, v_2 := u_2] \equiv_{\Gamma_1} t_2[v_1 := u_1, v_2 := u_2]$.

(2) By Definition 20 and (1), we have the statement.
Lemma 23. For a set of formulas $\Gamma$, the following statements are equivalent:

(1) $u_1 \equiv_{\Gamma} u_2$.

(2) There exists a finite sequence of terms $(t_i)_{0 \leq i \leq n}$ with $n \geq 0$ such that $t_0 \equiv u_1$, $t_n \equiv u_2$ and $t_i = t_{i+1} \in [\Gamma]$ for $0 \leq i < n$, where

$$[\Gamma] = \{s^nt_1 = s^nt_2 \mid n \in \mathbb{N} \text{ and either } t_1 = t_2 \in \Gamma \text{ or } t_2 = t_1 \in \Gamma\}.$$ 

Proof. (1) $\Rightarrow$ (2): Assume $u_1 \equiv_{\Gamma} u_2$ to prove (2) by induction on the definition of $\equiv_{\Gamma}$. We consider cases according to the clauses of the definition.

Case 1. If $u_1 = u_2 \in \Gamma$, then we have $u_1 \equiv u_2 \in \Gamma$. Thus, we have (2).

Case 2. If $u_1 \equiv u_2$, then we have (2).

Case 3. We consider the case where $u_1 \equiv_{\Gamma} u_1$. By the induction hypothesis, there exists a finite sequence of terms $(t_i)_{0 \leq i \leq n}$ such that $t_0 \equiv u_2$, $t_n \equiv u_1$ and $t_i = t_{i+1} \in [\Gamma]$ with $0 \leq i < n$. Let $t'_i \equiv t_{n-i}$. The finite sequence of terms $(t'_i)_{0 \leq i \leq n}$ satisfies $t'_0 \equiv u_1$, $t'_n \equiv u_2$ and $t'_i = t'_{i+1} \in [\Gamma]$. Thus, we have (2).

Case 4. We consider the case where $u_1 \equiv_{\Gamma} u_3$, $u_3 \equiv_{\Gamma} u_2$. By the induction hypothesis, there exist two finite sequences of terms $(t_i')_{0 \leq i \leq n}$, $(t'_j)_{0 \leq j \leq m}$ such that $t_0 \equiv u_1$, $t_n \equiv u_3$, $t'_m \equiv u_2$, $t_i = t_{i+1} \in [\Gamma]$ and $t'_i = t'_{i+1} \in [\Gamma]$ with $0 \leq i < n$, $0 \leq j < m$. Define $t_k$ as $t_k$ if $0 \leq k < n$ and $t_k$ if $n \leq k \leq n + m$. The finite sequence of terms $(t_k)$ satisfies $t_0 \equiv u_1$, $t_n \equiv u_2$ and $t_k = t_{k+1} \in [\Gamma]$. Thus, we have (2).

Case 5. We consider the case where $u_1 \equiv_{\Gamma} \hat{u}_2$, $u_1 \equiv u[v := \hat{u}_1]$ and $u_2 \equiv u[v := \hat{u}_2]$. By the induction hypothesis, there exists a finite sequence of terms $(t_i)_{0 \leq i \leq n}$ with $n \in \mathbb{N}$ such that $t_0 \equiv \hat{u}_1$, $t_n \equiv \hat{u}_2$, $t_i = t_{i+1} \in [\Gamma]$ with $0 \leq i < n$. Assume $v$ does not occur in $u$. In this case, we have $u_1 \equiv u[v := \hat{u}_1] \equiv u[v := \hat{u}_2] \equiv u_2$. Hence, (2) holds.

Assume $v$ occurs in $u$. In this case, we have $v \equiv s^m v$ for some natural numbers $m$. Let $t'_i = s^m t_i$ for $0 \leq i \leq n$. The finite sequence of terms $(t'_i)_{0 \leq i \leq n}$ satisfies $t'_0 \equiv u_1$, $t'_n \equiv u_2$ and $t'_i = t'_{i+1} \in [\Gamma]$.

(2) $\Rightarrow$ (1): Assume (2) to show (1). By the assumption, there exists a finite sequence of terms $(t_i)_{0 \leq i \leq n}$ with $n \in \mathbb{N}$ such that $t_0 \equiv u_1$, $t_n \equiv u_2$ and $t_i = t_{i+1} \in [\Gamma]$ with $0 \leq i < n$. If $t_i = t_{i+1} \in [\Gamma]$, then $t_i = t_{i+1}$ is $s^m t_1 = s^m t_2$, where $t_1 = t_2 \in \Gamma$ or $t_2 = t_1 \in \Gamma$. Therefore, $t_i \equiv_{\Gamma} t_{i+1}$. Because of the transitivity of $\equiv_{\Gamma}$, we have $u_1 \equiv_{\Gamma} u_2$.

Lemma 24. For a set of formulas $\Gamma_1$ and $\Gamma_2 \equiv (\Gamma_1, u_1 = u'_1, u_2 = u'_2, u_3 = u'_3)$, if $\text{VC}(u'_i)$ ($i = 1, 2, 3$) do not occur in $\Gamma_1, u_1, u_2, u_3$, $t, t'$ and are all distinct variables, then $t \equiv_{\Gamma_2} t'$ implies $t \equiv_{\Gamma_1} t'$.

Proof. Let $\text{VC}(u'_i) = v_i$ for each $i = 1, 2, 3$. Assume $t \equiv_{\Gamma_2} t'$, $t \not\equiv_{\Gamma_1} v_i$ for all $i = 1, 2, 3$. By Lemma 23, there exists a sequence $(t_j)_{0 \leq j \leq n}$ with $n \in \mathbb{N}$ such that $t_0 \equiv t$, $t_n \equiv t'$ and $t_j = t_{j+1} \in [\Gamma_2]$ with $0 \leq j < n$. We show $t \equiv_{\Gamma_1} t'$ by induction on $n$.

For $n = 0$, we have $t \equiv_{\Gamma_1} t'$ immediately.

We consider the case where $n > 0$.

If $t_j \neq s^m u'_i$ for all $i = 1, 2, 3$, $0 \leq j \leq n$ and $m \in \mathbb{N}$, then $t_j = t_{j+1} \in [\Gamma_1]$ with $0 \leq i < n$. By Lemma 23, we have $t \equiv_{\Gamma_1} t'$.

Assume that there exists $j_0$ with $0 \leq j_0 \leq n$, such that $t_{j_0} \equiv s^m u'_i$ for some $i = 1, 2, 3$ and $m \in \mathbb{N}$. Since any formula of $[\Gamma_2]$ in which $u'_i$ occurs is either $s^l u_i = s^l u'_i$ or $s^l u_i = s^l s^m u_i$ with $l \in \mathbb{N}$ and $\text{VC}(u'_i)$ ($i = 1, 2, 3$) do not occur in $t, t'$, we have $t_{j_0-1} \equiv t_{j_0+1} \equiv s^m u_i$. Define $t_k$ as $t_k$ if $0 \leq k < j_0$ and $t_{k+1}$ if $j_0 \leq k \leq n - 1$. Then, $t_0 \equiv t$, $t_{n-1} \equiv t'$ and $t_k = t_{k+1} \in [\Gamma_1]$ with $0 \leq k < n - 1$. By the induction hypothesis, we have $t \equiv_{\Gamma_1} t'$.

Lemma 25. For a set of formulas $\Gamma_1$ and $\Gamma_2 \equiv (\Gamma_1, u_1 = u'_1, \ldots, u_n = u'_n)$ with a natural number $n$, if $t \not\equiv_{\Gamma_1} u_i$ and $t \not\equiv_{\Gamma_1} u'_i$, with $i = 1, \ldots, n$, then $t \equiv_{\Gamma_2} t'$ implies $t \equiv_{\Gamma_1} t'$.

Proof. Assume $t \not\equiv_{\Gamma_1} u_i$, $t \not\equiv_{\Gamma_1} u'_i$ for $i = 1, \ldots, n$, and $t \equiv_{\Gamma_2} t'$. By Lemma 23, there exists a sequence $(t_j)_{0 \leq j \leq m}$ with $m \in \mathbb{N}$ such that $t_0 \equiv t$, $t_m \equiv t'$ and $t_j = t_{j+1} \in [\Gamma_2]$ with $0 \leq j < m$.

If $t_j \neq s^l u_i$ and $t_j \neq s^l u'_i$ for all $0 \leq j \leq n$, $i = 1, \ldots, n$, and any $l \in \mathbb{N}$, then $t_j = t_{j+1} \in [\Gamma_1]$ with all $0 \leq j < m$. By Lemma 23, we have $t \equiv_{\Gamma_1} t'$.

Assume that there exists $j$ with $0 \leq j \leq n$, such that $t_j \equiv s^l u_i$ or $t_j \equiv s^l u'_i$ for $i = 1, \ldots, n$, and some $l \in \mathbb{N}$. Let $j_0$ be the least number among such $j$'s. Since $j_0$ is the least, we have $t_j = t_{j+1} \in [\Gamma_1]$ for all $0 \leq j < j_0$. By Lemma 23, we have $t \equiv_{\Gamma_1} s^l u_i$ or $t \equiv_{\Gamma_1} s^l u'_i$. This contradicts $t \not\equiv_{\Gamma_1} u_i$ and $t \not\equiv_{\Gamma_1} u'_i$. \□
We call the assumption of (Case \text{Add}_2) whose form

$$\Gamma, a = sx, b = y, c = z, \text{Add}_2(x, sy, z) \vdash \Delta$$

the right assumption of the rule. The other assumption is called the left assumption of the rule.

**Lemma 26.** Let $\Gamma \vdash \Delta$ be in $\mathcal{D}_{cf}$ and

$$\Lambda(\Gamma \vdash \Delta) = \{ a \mid \text{Add}_2(a, b, c) \in \Gamma, \text{Add}_1(a, b, c) \in \Delta, \text{ or } a \equiv 0 \}$$

and

$$\text{BC}(\Gamma \vdash \Delta) = \{ b \mid \text{Add}_2(a, b, c) \in \Gamma \text{ or } \text{Add}_1(a, b, c) \in \Delta \} \cup \{ c \mid \text{Add}_2(a, b, c) \in \Gamma \text{ or } \text{Add}_1(a, b, c) \in \Delta \}.$$

If $t \in \Lambda(\Gamma \vdash \Delta)$ and $u \in \text{BC}(\Gamma \vdash \Delta)$, then $t \not\sim_{\Gamma} u$.

**Proof.** We prove the statement by induction on the height of the node $\Gamma \vdash \Delta$ in $\mathcal{D}_{cf}$.

The root of $\mathcal{D}_{cf}$ satisfies the statement.

Assume $\Gamma \vdash \Delta$ is not the root. Let $\Gamma' \vdash \Delta'$ be the parent of $\Gamma \vdash \Delta$. We consider cases according to the rule with the conclusion $\Gamma' \vdash \Delta'$.

- **Case 1.** In the case (Weak), we have the statement by $\Gamma \subseteq \Gamma'$.
- **Case 2.** In the case (Subst), we have the statement by Lemma 21 (2).
- **Case 3.** In the case (\textit{L}_a), we have the statement by Lemma 22 (2).
- **Case 4.** We consider the case where the rule is (Case \text{Add}_2) and $\Gamma \vdash \Delta$ is the right assumption of the rule. Let $\text{Add}_2(a, b, c)$ be the principal formula of the rule. There exists $\Pi$ such that $\Gamma' \equiv (\Pi, \text{Add}_2(a, b, c))$ and $\Gamma \equiv (\Pi, a = sx, b = y, c = z, \text{Add}_2(x, sy, z))$ for fresh variables $x, y, z$.

Assume $t \in \Lambda(\Gamma \vdash \Delta)$ and $u \in \text{BC}(\Gamma \vdash \Delta)$ and $t \sim_{\Gamma} u$ for contradiction.

Define $t$ as $a$ if $t \equiv y$ and $t$ otherwise. We also define $u$ as $b$ if $u \equiv sy$, $c$ if $u \equiv z$ and $u$ otherwise. Since $t \sim_{\Gamma} u$ holds, we have $t \sim_{\Gamma} u$. By Lemma 24, we have $t \sim_{\Gamma'} u$. Since $t \in \Lambda(\Gamma' \vdash \Delta')$ and $u \in \text{BC}(\Gamma' \vdash \Delta')$ hold, this contradicts the induction hypothesis.

**Case 5.** We consider the case where the rule is (Case \text{Add}_2) and $\Gamma \vdash \Delta$ is the left assumption of the rule. Let $\text{Add}_2(a, b, c)$ be the principal formula of the rule. There exists $\Pi$ such that $\Gamma' \equiv (\Pi, \text{Add}_2(a, b, c))$ and $\Gamma \equiv (\Pi, a = 0, b = y, c = y)$.

Let $t \in \Lambda(\Gamma \vdash \Delta)$ and $u \in \text{BC}(\Gamma \vdash \Delta)$. Since $\Lambda(\Gamma \vdash \Delta) \subseteq \Lambda(\Gamma' \vdash \Delta')$ holds, we have $t \in \Lambda(\Gamma' \vdash \Delta')$. By $\text{BC}(\Gamma \vdash \Delta) \subseteq \text{BC}(\Gamma' \vdash \Delta')$, we have $u \in \text{BC}(\Gamma' \vdash \Delta')$. By the induction hypothesis, $t \not\sim_{\Gamma'} u$, $t \not\sim_{\Gamma'} b$ and $t \not\sim_{\Gamma'} c$. Since the set of formulas with $= \in \Pi$ is the same as the set of formulas with $=$ in $\Gamma'$, we have $t \not\sim_{\Pi'} b$ and $t \not\sim_{\Pi'} c$. By Lemma 25, $t \not\sim_{\Pi'} u$.

By the induction hypothesis, $u \not\sim_{\Gamma'} a$, $a \not\sim_{\Gamma'} b$ and $a \not\sim_{\Gamma'} c$. Since the set of formulas with $= \in \Pi$ is the same as the set of formulas with $=$ in $\Gamma'$, we have $u \not\sim_{\Pi'} a$, $a \not\sim_{\Pi'} b$ and $a \not\sim_{\Pi'} c$. By Lemma 25, $a \not\sim_{\Pi'} a$.

By the induction hypothesis, $u \not\sim_{\Gamma'} 0$, $0 \not\sim_{\Gamma'} b$ and $0 \not\sim_{\Gamma'} c$. Since the set of formulas with $= \in \Pi$ is the same as the set of formulas with $=$ in $\Gamma'$, we have $0 \not\sim_{\Pi'} b$ and $0 \not\sim_{\Pi'} c$. By Lemma 25, $u \not\sim_{\Pi'} 0$.

By Lemma 25 and these three facts, $t \not\sim_{\Gamma} u$.

**Case 6.** In the case (Add1 \text{R}_2), $\Gamma \equiv \Gamma'$ implies the statement by the induction hypothesis.

\[ \square \]

**4.5 Index**

In this section, we define a key concept, called an index, to prove Theorem 16.

**Definition 27 (Index).** For a sequent $\Gamma \vdash \Delta$ and $\text{Add}_2(a, b, c) \in \Gamma$, we define the index of $\text{Add}_2(a, b, c)$ in $\Gamma \vdash \Delta$ as follows:

1. If $b \not\sim_{\Gamma} b'$ for any $\text{Add}_1(a', b', c') \in \Delta$, then the index of $\text{Add}_2(a, b, c)$ in $\Gamma \vdash \Delta$ is $\bot$, and
2. If there exists uniquely $m - n$ such that $m, n \in \mathbb{N}$, $s^m b \equiv_{\Gamma} s^n b'$ and $\text{Add}_1(a', b', c') \in \Delta$, then the index of $\text{Add}_2(a, b, c)$ in $\Gamma \vdash \Delta$ is $m - n$ (namely the uniqueness means that $s^m b \equiv_{\Gamma} s^n b'$ for $m', n' \in \mathbb{N}$ and $\text{Add}_1(a', b', c') \in \Delta$ imply $m - n = m' - n'$).

Note that if there exists $n, m \in \mathbb{N}$ such that $s^m b \equiv_{\Gamma} s^n b'$ for some $\text{Add}_1(a', b', c')$ and $m - n$ is not unique, then the index of $\text{Add}_2(a, b, c)$ in $\Gamma \vdash \Delta$ is undefined.

**Definition 28 (Index sequent).** The sequent $\Gamma \vdash \Delta$ is said to be an index sequent if the following conditions hold:

1. If $t \in B_1(\Gamma \vdash \Delta)$ and $u \in C(\Gamma \vdash \Delta)$, then $t \not\sim_{\Gamma} u$, and
(2) if $s^nb \equiv_\Gamma s^mb'$ with $b, b' \in B_1(\Gamma \vdash \Delta)$, then $n = m$, where

$$B_1(\Gamma \vdash \Delta) = \{ b \mid \text{Add}_1(a, b, c) \in \Delta \}, \quad \text{and}$$

$$C(\Gamma \vdash \Delta) = \{ c \mid \text{Add}_2(a, b, c) \in \Gamma \ or \ \text{Add}_1(a, b, c) \in \Delta \}.$$ 

This condition (2) guarantees the existence of an index, as shown in the following lemma. We will use (1) to calculate an index in Lemma 33 (1) and an infinite sequence in Lemma 36.

**Lemma 29.** If $\Gamma \vdash \Delta$ is an index sequent, the index of any Add$_2(a, b, c) \in \Gamma$ in $\Gamma \vdash \Delta$ is defined.

**Proof.** If $b \not\in \Gamma b'$ for any Add$_1(a', b', c') \in \Delta$, then the index is $\bot$.

Assume $b \sim b_0'$ for some Add$_1(a_0', b_0', c_0') \in \Delta$. By Definition 20, there exist $n_0$ and $m_0$ such that $s^{n_0}b \equiv_\Gamma s^{m_0}b_0'$. To show the uniqueness, we fix Add$_1(a_1', b_1', c_1') \in \Delta$ and assume $s^{n_1}b \equiv s^{m_1}b_1'$. Since $s^{n_0+n_1}b \equiv s^{m_0+m_1}b_0'$ and $s^{n_0+n_1}b \equiv s^{m_0+m_1}b_0'$, we have $s^{n_0+n_1}b_0' \equiv s^{m_0+m_1}b_1'$. From (2) of Definition 28, $m_0 + n_1 = m_1 + n_0$. Thus, $m_0 + n_0 = m_1 - n_1$. □

**Definition 30** (Switching point). A node $\sigma$ in a derivation tree is called a switching point if the rule with the conclusion $\sigma$ is (Case Add$_2$) and the index of the principal formula for the rule in the conclusion is $\bot$.

**Definition 31** (Index path). A path $(\Gamma_i \vdash \Delta_i)_{0 \leq i < \alpha}$ in $T((\mathcal{D}_c, \mathcal{C}_c))$ with some $\alpha \in \mathbb{N} \cup \{ \omega \}$ is said to be an index path if the following conditions hold:

1. $(\Gamma_0 \vdash \Delta_0)$ is an index sequent, and
2. if the rule for $\Gamma_i \vdash \Delta_i$ is (Case Add$_2$) and $\Gamma_{i+1} \vdash \Delta_{i+1}$ is the left assumption of the rule, then $\Gamma_i \vdash \Delta_i$ is a switching point.

**Lemma 32.** Every sequent in an index path is an index sequent.

**Proof.** Let $(\Gamma_i \vdash \Delta_i)_{0 \leq i < \alpha}$ be an index path. We use $B_1(\Gamma \vdash \Delta)$ and $C(\Gamma \vdash \Delta)$ in Definition 28. We prove the statement by the induction on $i$.

For $i = 0$, $\Gamma_0 \vdash \Delta_0$ is an index sequent by Definition 31.

For $i > 0$, we consider cases according to the rule with the conclusion $\Gamma_{i-1} \vdash \Delta_{i-1}$.

**Case 1.** The case (Weak).

1. Assume that $\Gamma \in B_1(\Gamma_i \vdash \Delta_i)$ and $u \in C(\Gamma_i \vdash \Delta_i)$. Since $B_1(\Gamma_i \vdash \Delta_i) \subseteq B_1(\Gamma_{i-1} \vdash \Delta_{i-1})$ holds, we have $t \in B_1(\Gamma_{i-1} \vdash \Delta_{i-1})$. By $C(\Gamma_i \vdash \Delta_i) \subseteq C(\Gamma_{i-1} \vdash \Delta_{i-1})$, we have $u \in C(\Gamma_{i-1} \vdash \Delta_{i-1})$. By the induction hypothesis (1), we have $t \not\in \Gamma_{i-1} u$. By $\Gamma_i \subseteq \Gamma_{i-1}$, we have $t \not\in \Gamma_i u$.

2. Assume that $s^nb \equiv_\Gamma s^mb'$ with $b, b' \in B_1(\Gamma_i \vdash \Delta_i)$ for $n, m \in \mathbb{N}$. By $\Gamma_i \subseteq \Gamma_{i-1}$, we have $s^nb \equiv_\Gamma s^mb'$.

Since $B_1(\Gamma_i \vdash \Delta_i) \subseteq B_1(\Gamma_{i-1} \vdash \Delta_{i-1})$, we have $b, b' \in B_1(\Gamma_{i-1} \vdash \Delta_{i-1})$. By the induction hypothesis (2), we have $n = m$.

**Case 2.** The case (Subst) with a substitution $\theta$.

1. Assume that $\Gamma \in B_1(\Gamma_i \vdash \Delta_i)$ and $u \in C(\Gamma_i \vdash \Delta_i)$. Since $\Gamma_{i-1} \equiv \Gamma_i[\theta]$ and $\Delta_{i-1} \equiv \Delta_i[\theta]$ hold, we have $t[\theta] \in B_1(\Gamma_{i-1} \vdash \Delta_{i-1})$ and $u[\theta] \in C(\Gamma_{i-1} \vdash \Delta_{i-1})$. By the induction hypothesis (1), we have $t[\theta] \not\in \Gamma_{i-1} u[\theta]$. By Lemma 21 (2), we have $t \not\in \Gamma_i u$.

2. Assume that $s^nb \equiv_\Gamma s^mb'$ with $b, b' \in B_1(\Gamma_i \vdash \Delta_i)$ for $n, m \in \mathbb{N}$. By Lemma 21 (1), $s^nb[\theta] \equiv_\Gamma s^mb'[\theta]$.

Since $\Delta_{i-1} \equiv \Delta_i[\theta]$ holds, we have $b[\theta], b'[\theta] \in B_1(\Gamma_{i-1} \vdash \Delta_{i-1})$. By the induction hypothesis (2), we have $n = m$.

**Case 3.** The case (= $I_\alpha$).

Let $u_1 = u_2$ be the principal formula of the rule. There exist $\Gamma$ and $\Delta$ such that

$$\Gamma_{i-1} \equiv (\Gamma[v_1 := u_1, v_2 := u_2], u_1 = u_2),$$

$$\Delta_{i-1} \equiv (\Delta[v_1 := u_1, v_2 := u_2], u_1 = u_2),$$

$$\Gamma \equiv (\Gamma[v_1 := u_2, v_2 := u_1], u_1 = u_2),$$

$$\Delta \equiv (\Delta[v_1 := u_2, v_2 := u_1], u_1 = u_2).$$

(1) Assume that $t \in B_1(\Gamma_i \vdash \Delta_i)$ and $u \in C(\Gamma_i \vdash \Delta_i)$. From the definition of $\Gamma_i$ and $\Delta_i$, there exist terms $\bar{t}$, $\bar{u}$ such that $t \equiv \bar{t}[v_1 := u_2, v_2 := u_1]$ and $u \equiv \bar{u}[v_1 := u_2, v_2 := u_1]$. Then, $\bar{t}[v_1 := u_1, v_2 := u_2] \in B_1(\Gamma_{i-1} \vdash \Delta_{i-1})$ and $\bar{u}[v_1 := u_1, v_2 := u_2] \in C(\Gamma_{i-1} \vdash \Delta_{i-1})$. By the induction hypothesis (1), we have
\[ \tilde{t}[v_1 := u_1, v_2 := u_2] \not\in_{\Gamma_{i-1}} \tilde{u}[v_1 := u_1, v_2 := u_2]. \] By Lemma 22 (2), we have \[ \tilde{t}[v_1 := u_2, v_2 := u_1] \not\in_{\Gamma}, \tilde{u}[v_1 := u_2, v_2 := u_1]. \] Thus, \( t \not\in_{\Gamma}. \)

(2) Assume that \( s^0b \equiv_{\Gamma_i} s^0b' \) with \( b, b' \in B_1(\Gamma_i \vdash \Delta_i) \) for \( n, m \in \mathbb{N} \). From the definition of \( \Gamma_i \) and \( \Delta_i \), there exist terms \( b, b' \in \Delta \) such that \( b \equiv s^0b[v_1 := u_2, v_2 := u_1] \) and \( b' \equiv s^0b'[v_1 := u_2, v_2 := u_2] \). By Lemma 22 (1), \( s^0b[v_1 := u_1, v_2 := u_2] \equiv_{\Gamma_{i-1}} s^0b'[v_1 := u_1, v_2 := u_2] \). From the definition of \( \Gamma_{i-1} \) and \( \Delta_{i-1} \), \( b[v_1 := u_1, v_2 := u_2], b'[v_1 := u_1, v_2 := u_2] \in B_1(\Gamma_{i-1} \vdash \Delta_{i-1}) \). By the induction hypothesis (2), we have \( n = m \).

Case 4. The case (Case Add2) with the right assumption \( \Gamma_i \vdash \Delta_i \).

Let Add2 \( (a, b, c) \) be the principal formula of the rule. There exists II such that \( \Gamma_{i-1} \equiv (\Pi, \text{Add2} (a, b, c)) \) and \( \Gamma_i \equiv (\Pi, a = sz, b = y, c = z, \text{Add2}(x, y, z)) \) for fresh variables \( x, y, z \).

(1) Assume that \( t \in B_1(\Gamma_i \vdash \Delta_i) \) and \( u \in C(\Gamma_i \vdash \Delta_i) \). Assume that \( t \not\sim_{\Gamma_i} u \) for contradiction. Define \( \hat{u} \) as \( c \) if \( u \equiv z \) and \( u \) otherwise. Since \( t \sim_{\Gamma_i} \hat{u} \) holds, we have \( t \sim_{\Gamma_i} \hat{u} \). By Lemma 24, we have \( t \sim_{\Gamma_{i-1}} \hat{u} \). Since \( t \in B_1(\Gamma_{i-1} \vdash \Delta_{i-1}) \) and \( \hat{u} \in C(\Gamma_{i-1} \vdash \Delta_{i-1}) \) hold, this contradicts the induction hypothesis (1).

(2) Assume that \( s^0b \equiv_{\Gamma_i} s^0b' \) with \( b, b' \in B_1(\Gamma_i \vdash \Delta_i) \) for \( n, m \in \mathbb{N} \). By Lemma 24, \( s^0b \equiv_{\Gamma_{i-1}} s^0b' \). By the induction hypothesis (2), we have \( n = m \).

Case 5. The case (Case Add2) with the left assumption \( \Gamma_i \vdash \Delta_i \). In this case, \( \Gamma_{i-1} \vdash \Delta_{i-1} \) is a switching point.

Let Add2 \( (a, b, c) \) be the principal formula of the rule. There exists II such that \( \Gamma_{i-1} \equiv (\Pi, \text{Add2} (a, b, c)) \) and \( \Gamma_i \equiv (\Pi, a = 0, b = y, c = y) \) with a fresh variable \( y \).

(1) Assume that \( t \in B_1(\Gamma_i \vdash \Delta_i) \) and \( u \in C(\Gamma_i \vdash \Delta_i) \). Since \( B_1(\Gamma_i \vdash \Delta_i) = B_1(\Gamma_{i-1} \vdash \Delta_{i-1}) \) holds, we have \( t \in B_1(\Gamma_{i-1} \vdash \Delta_{i-1}) \). By \( C(\Gamma_i \vdash \Delta_i) \subseteq C(\Gamma_{i-1} \vdash \Delta_{i-1}) \), we have \( u \in C(\Gamma_{i-1} \vdash \Delta_{i-1}) \). By the induction hypothesis (1), \( t \not\sim_{\Gamma_{i-1}} u \) and \( t \not\sim_{\Gamma_{i-1}} c \). By Lemma 26, \( t \not\sim_{\Gamma_{i-1}} a \) and \( t \not\sim_{\Gamma_{i-1}} 0 \). Since \( y \) is fresh, we have \( t \not\sim_{\Gamma_{i-1}} y \). Since \( \Gamma_{i-1} \vdash \Delta_{i-1} \) is a switching point, we have \( t \not\sim_{\Gamma_{i-1}} b \). By Lemma 25, \( t \not\sim_{\Gamma_i} u \).

(2) Assume that \( s^0b \equiv_{\Gamma_i} s^0b' \) with \( b, b' \in B_1(\Gamma_i \vdash \Delta_i) \) for \( n, m \in \mathbb{N} \) to show \( n = m \). By Lemma 26, \( s^0b \not\equiv_{\Gamma_i} a \) and \( s^0b \not\equiv_{\Gamma_i} 0 \). Since \( \Gamma_{i-1} \vdash \Delta_{i-1} \) is a switching point, we have \( s^0b \not\equiv_{\Gamma_{i-1}} b \). By the induction hypothesis (1), \( s^0b \not\equiv_{\Gamma_{i-1}} c \). Since \( y \) is fresh, we have \( s^0b \not\equiv_{\Gamma_{i-1}} y \). By Lemma 25, we have \( s^0b \equiv_{\Gamma_{i-1}} s^0b' \). Because \( B_1(\Gamma_i \vdash \Delta_i) = B_1(\Gamma_{i-1} \vdash \Delta_{i-1}) \), we have \( b, b' \in B_1(\Gamma_{i-1} \vdash \Delta_{i-1}) \). By the induction hypothesis (2), we have \( n = m \).

Case 6. The case (Add1 R2). Let Add1 \( (sa, b, sc) \) be the principal formula of the rule.

(1) Assume that \( t \in B_1(\Gamma_i \vdash \Delta_i) \) and \( u \in C(\Gamma_i \vdash \Delta_i) \) and \( t \not\sim_{\Gamma_i} u \) for contradiction. Define \( \hat{u} \) as \( sc \) if \( u \equiv c \) and \( u \) otherwise. Since \( t \sim_{\Gamma_i} \hat{u} \) holds, we have \( t \sim_{\Gamma_i} \hat{u} \). Since \( \Gamma_{i-1} = \Gamma_i \) holds, we have \( t \sim_{\Gamma_{i-1}} \hat{u} \). Since \( t \in B_1(\Gamma_{i-1} \vdash \Delta_{i-1}) \) and \( \hat{u} \in C(\Gamma_{i-1} \vdash \Delta_{i-1}) \) hold, this contradicts the induction hypothesis (1).

(2) Assume that \( s^0b \equiv_{\Gamma_i} s^0b' \) with \( b, b' \in B_1(\Gamma_i \vdash \Delta_i) \) for \( n, m \in \mathbb{N} \). Because \( \Gamma_{i-1} = \Gamma_i \), we have \( s^0b \equiv_{\Gamma_{i-1}} s^0b' \). Since the second argument of a formula with Add1 in \( \Delta_i \) is that in \( \Delta_{i-1} \), we have \( b, b' \in B_1(\Gamma_{i-1} \vdash \Delta_{i-1}) \). By the induction hypothesis (2), we have \( n = m \).

Lemma 33. For an index path \( (\Gamma_i \vdash \Delta_i)_{0 \leq i < \alpha} \) and a trace \( (\tau_k)_{k \geq 0} \) following \( (\Gamma_i \vdash \Delta_i)_{i \geq \rho} \), if \( d_k \) is the index of \( \tau_k \), the following statements holds:

(1) If \( d_k = \bot \), then \( d_{k+1} = \bot \).

(2) If the rule with the conclusion \( \Gamma_{p+k} \vdash \Delta_{p+k} \) is (Weak) or (Subst), then \( d_{k+1} = d_k \) or \( d_{k+1} = \bot \).

(3) If the rule with the conclusion \( \Gamma_{p+k} \vdash \Delta_{p+k} \) is (= L\( a \)) or (Add1 R2), then \( d_{k+1} = d_k \).

(4) Assume the rule with the conclusion \( \Gamma_{p+k} \vdash \Delta_{p+k} \) is (Case Add2).

(a) If \( \Gamma_{p+k+1} \vdash \Delta_{p+k+1} \) is the left assumption of the rule, then \( d_{k+1} = d_k \).

(b) If \( \Gamma_{p+k+1} \vdash \Delta_{p+k+1} \) is the right assumption of the rule and \( \tau_k \) is not a progress point of the trace, then \( d_{k+1} = d_k \).

(c) If \( \Gamma_{p+k+1} \vdash \Delta_{p+k+1} \) is the right assumption of the rule and \( \tau_k \) is a progress point of the trace, then \( d_{k+1} = d_k + 1 \).

Proof. Let \( \tau_k \equiv \text{Add2}(a_k, b_k, c_k) \).

(1) It suffices to show that \( b_{k+1} \not\sim_{\Gamma_{p+k+1}} b' \) holds for any \( \text{Add1}(a', b', c') \in \Delta_{p+k+1} \) if \( b_k \not\sim_{\Gamma_{p+k}} b \) holds for any \( \text{Add1}(a, b, c) \in \Delta_{p+k} \). We consider cases according to the rule with the conclusion \( \Gamma_{p+k} \vdash \Delta_{p+k} \).
The case where $\Gamma \vdash_\mathit{p+b}\ b$, assume $b_{k+1} \sim_{\Gamma_{p+b}} b'$ for some $\mathit{Add}(a',b',c') \in \Delta_{p+b+1}$. Define $t$ as $b$ if $b_{k+1} \equiv sy$ and $b_{k+1}$ otherwise. Since $b_{k+1} \sim_{\Gamma_{p+b+1}} b'$ holds, we have $t \sim_{\Gamma_{p+b+1}} b'$. By Lemma 24, $t \sim_{\Gamma_{p+b}} b'$. By $b_k \equiv t$, we have $b_{k} \sim_{\Gamma_{p+b}} b'$.

Case 5. The case ($\mathit{Add}$) with the left assumption $\Gamma_{p+b+1} \vdash \Delta_{p+b+1}$. In this case, $\Gamma_{p+b} \vdash \Delta_{p+b}$ is a switching point. Let $\mathit{Add}(a,b,c)$ be the principal formula of the rule. There exists $\Pi$ such that $\Gamma_{p+b} \equiv (\Pi, \mathit{Add}(a,b,c))$ and $\Gamma_{p+b} \equiv (\Pi, a = sz, b = y, c = z, \mathit{Add}(x, sy, z))$ for fresh variables $x, y, z$.

We prove this case by contrapositive. To show $b_{k+1} \sim_{\Gamma_{p+b}} b'$, assume $b_{k+1} \sim_{\Gamma_{p+b}} b'$ for some $\mathit{Add}(a',b',c') \in \Delta_{p+b+1}$. Define $t$ as $b$ if $b_{k+1} \equiv sy$ and $b_{k+1}$ otherwise. Since $b_{k+1} \sim_{\Gamma_{p+b+1}} b'$ holds, we have $t \sim_{\Gamma_{p+b+1}} b'$. By Lemma 24, $t \sim_{\Gamma_{p+b}} b'$. By $b_k \equiv t$, we have $b_{k} \sim_{\Gamma_{p+b}} b'$.

Case 6. The case ($\mathit{Add}_2$) \(R_2\).

In this case, $\Gamma_{p+b} \equiv \Pi$, the same as $\Gamma_{p+b+1}$ and the second argument of a formula with $\mathit{Add}_2$ or $\mathit{Add}_1$ in $\Gamma_{p+b} \vdash \Delta_{p+b+1}$ is the same as that in $\Gamma_{p+b+1} \vdash \Delta_{p+b+1}$. We thus have the statement.

Let $d_k = n$.

Case 1. The case ($\mathit{Weak}$).

If $b_{k+1} \sim_{\Gamma_{p+b}} b$ for any $\mathit{Add}(a,b,c) \in \Delta_{p+b+1}$, then $d_k = 1$.

Assume $b_{k+1} \sim_{\Gamma_{p+b}} b$, for some $\mathit{Add}(a,b,c) \in \Delta_{p+b+1}$. By Definition 20, there exist $m, l \in \mathbb{N}$ such that $s^m b = \Gamma_{p+b+1} \vdash s^m b$. By $\Gamma_{p+b+1} \subseteq \Gamma_{p+b}$, we have $s^m b = \Gamma_{p+b} \vdash s^m b$. Since $b_k = b_{k+1}$, we have $s^m b_k = \Gamma_{p+b} \vdash s^m b$. Since $\Delta_{p+b+1} \subseteq \Delta_{p+b}$, we have $\mathit{Add}(a,b,c) \in \Delta_{p+b}$. By $d_k = n$, we have $l = m$. Thus, $d_k = 1$.

Case 2. The case ($\mathit{Subst}$) with a substitution $\theta$. Note that $b_k \equiv b_{k+1}[\theta]$.

If $b_{k+1} \sim_{\Gamma_{p+b}} b$ for any $\mathit{Add}(a,b,c) \in \Delta_{p+b+1}$, then $d_k = 1$.

Assume $b_{k+1} \sim_{\Gamma_{p+b}} b$, for some $\mathit{Add}(a,b,c) \in \Delta_{p+b+1}$. By Definition 20, there exist $m, l \in \mathbb{N}$ such that $s^m b_k \equiv \Gamma_{p+b+1} \vdash s^m b[\theta]$. By $\Gamma_{p+b+1} \subseteq \Gamma_{p+b}$, we have $s^m b_k \equiv \Gamma_{p+b} \vdash s^m b[\theta]$. Since $b_k = b_{k+1}[\theta]$ holds, we have $s^m b_k \equiv \Gamma_{p+b} \vdash s^m b[\theta]$. Since $\Delta_{p+b} \subseteq \Gamma_{p+b+1}[\theta]$ holds, we have $\mathit{Add}(a[\theta],b[\theta],c[\theta]) \in \Delta_{p+b}$. By $d_k = n$, we have $l = m$. Thus, $d_k = 1$.

Let $d_k = n$.

Case 1. The case ($\mathit{Weak}$) with the principal formula $u_1 = u_2$.

Let $b_k \equiv b[v_1 := u_1, v_2 := u_2]$ and $b_{k+1} \equiv b[v_1 := u_2, v_2 := u_1]$ for variables $v_1, v_2$.

By $d_k = n$, there exist $m, l \in \mathbb{N}$ such that $s^m b[v_1 := u_1, v_2 := u_2] \equiv \Gamma_{p+b} \vdash s^m b[v_1 := u_1, v_2 := u_2]$ for some $\mathit{Add}(a[v_1 := u_1, v_2 := u_2], b[v_1 := u_1, v_2 := u_2], c[v_1 := u_1, v_2 := u_2]) \in \Delta_{p+b}$ and $l = m$. From Lemma 22 (1), $s^m b[v_1 := u_1, v_2 := u_2] \equiv \Gamma_{p+b} \vdash s^m b'[v_1 := u_1, v_2 := u_1]$. Moreover, $\mathit{Add}(a[v_1 := u_1, v_2 := u_2], b[v_1 := u_2, v_2 := u_1], c[v_1 := u_1, v_2 := u_1]) \in \Delta_{p+b+1}$. Thus, $d_k = 1$.

Case 2. The case ($\mathit{Add}_1$) \(R_2\).

Since $\tau_{p+b+1} \equiv \tau_{p+b}$ holds, $\Gamma_{p+b}$ is the same as $\Gamma_{p+b+1}$ and the second argument of a formula with $\mathit{Add}_1$ in $\Delta_{p+b}$ is the same as that in $\Delta_{p+b+1}$. We thus have $d_k = 1$.

(4) Let $d_k = n$. Let $\mathit{Add}(a,b,c)$ be the principal formula of the rule ($\mathit{Add}_2$) with the conclusion $\Gamma_{p+b} \vdash \Delta_{p+b}$.

(a) The case where $\Gamma_{p+b+1} \vdash \Delta_{p+b+1}$ is the left assumption of the rule. In this case, $\Gamma_{p+b} \vdash \Delta_{p+b}$ is a switching point.

There exists $\Pi$ such that $\Gamma_{p+b} \equiv (\Pi, \mathit{Add}(a,b,c))$ and $\Gamma_{p+b+1} \equiv (\Pi, a = 0, b = y, c = y)$ with a fresh variable $y$. By $d_k = n$, there exist $m, l \in \mathbb{N}$ such that $s^m b_k \equiv \Gamma_{p+b} \vdash s^m b'$ for some $\mathit{Add}(a',b',c') \in \Delta_{p+b}$ and $l = m$. Since the set of formulas with $= \in \Gamma_{p+b+1}$ includes the set of formulas with $= \in \Gamma_{p+b}$, we have $s^m b_k \equiv \Gamma_{p+b} \vdash s^m b'$. By $\tau_{k+1} \equiv \tau_k$, we have $s^m b_{k+1} \equiv \Gamma_{p+b+1} \vdash s^m b'$. Since $\Delta_{p+b} \subseteq \Gamma_{p+b+1}$, $d_k = n$.

(b) The case where $\Gamma_{p+b+1} \vdash \Delta_{p+b+1}$ is the right assumption of the rule and $\tau_k$ is not a progress point of the trace.

Since $\tau_k$ is not a progress point of the trace, we have $\tau_{k+1} \equiv \tau_k$. By $d_k = n$, there exist $m, l \in \mathbb{N}$ such that $s^m b_k \equiv \Gamma_{p+b} \vdash s^m b'$ for some $\mathit{Add}(a',b',c') \in \Delta_{p+b}$ and $l = m$. Since the set of formulas with $= \in \Gamma_{p+b+1}$ includes the set of formulas with $= \in \Gamma_{p+b}$, we have $s^m b_{k+1} \equiv \Gamma_{p+b+1} \vdash s^m b'$. By $\tau_{k+1} \equiv \tau_k$, we have $s^m b_{k+1} \equiv \Gamma_{p+b+1} \vdash s^m b'$. Since $\Delta_{p+b} \subseteq \Gamma_{p+b+1}$ holds, we have $\mathit{Add}(a',b',c') \in \Delta_{p+b+1}$. Thus, $d_k = 1$. 

14
The case where $\Gamma_{p+k+1} \vdash \Delta_{p+k+1}$ is the right assumption of the rule and $\tau_k$ is a progress point of the trace.

There exists $\Pi$ such that $\Gamma_{p+k} \equiv (\Pi, \text{Add}_2(a, b, c))$ and $\Gamma_{p+k+1} \equiv (\Pi, a = sx, b = y, c = z, \text{Add}_2(x, sy, z))$ for fresh variables $x, y, z$. Since $\tau_k$ is a progress point of the trace, we have $\tau_k \equiv \text{Add}_2(a, b, c)$ and $\tau_{k+1} \equiv \text{Add}_2(x, sy, z)$. Therefore, $b_k \equiv b$ and $b_{k+1} \equiv sy$. By $d_k = n$, there exist $m, l \in \mathbb{N}$ such that $s^m b \preceq \Gamma_{p+k} s^l b'$ for some $\text{Add}_1(a', b', c') \in \Delta_{p+k}$ and $l - m = n$. Since the set of formulas with $= \in \Gamma_{p+k+1}$ includes the set of formulas with $= \in \Gamma_{p+k}$, we have $s^m b \preceq \Gamma_{p+k+1} s^l b'$. By $b \preceq \Gamma_{p+k+1} y$, we have $s^m y \preceq \Gamma_{p+k+1} s^l b'$. Hence, $s^m b_{k+1} \preceq \Gamma_{p+k+1} s^{l+1} b'$. Thus, $d_{k+1} = l + 1 - m = n + 1$.

**Lemma 34.** For an infinite index path $(\Gamma_i \vdash \Delta_i)_{i \geq 0}$ in $T((D_{cf}, C_{cf}))$, there exists $l \in \mathbb{N}$ such that the following conditions hold:

1. $\Gamma_l \vdash \Delta_l$ is a switching point in $T((D_{cf}, C_{cf}))$, and
2. $\Gamma_{l+1} \vdash \Delta_{l+1}$ is the right assumption of the rule with the conclusion $\Gamma_1 \vdash \Delta_l$.

**Proof.** Since $(\Gamma_i \vdash \Delta_i)_{i \geq 0}$ is an infinite path and $T((D_{cf}, C_{cf}))$ satisfies the global trace condition, there exists an infinitely progressing trace following a tail of the path. Let $(\tau_k)_{k \geq 0}$ be an infinitely progressing trace following $(\Gamma_i \vdash \Delta_i)_{i \geq p}$. Let $d_k$ be the index of $\tau_k$ in $\Gamma_{p+k} \vdash \Delta_{p+k}$.

We show that there exists $l \in \mathbb{N}$ such that $d_l = \bot$. The set $\{d_k \mid k \geq 0\}$ is finite since the set of sequents in $(\Gamma_i \vdash \Delta_i)_{i \geq 0}$ is finite and we have a unique index of an atomic formula with $\text{Add}_2$ in $\Gamma_{i+1} \vdash \Delta_i$. Since $(\tau_k)_{k \geq 0}$ is an infinitely progressing trace following $(\Gamma_i \vdash \Delta_i)_{i \geq p}$, if there does not exist $k' \in \mathbb{N}$ such that $d_{k'} = \bot$, Lemma 33 implies that $\{d_k \mid k \geq 0\}$ is infinite. Thus, there exists $k' \in \mathbb{N}$ such that $d_{k'} = \bot$.

Since $(\tau_k)_{k \geq 0}$ is an infinitely progressing trace following $(\Gamma_i \vdash \Delta_i)_{i \geq p}$, there exists a progress point $\tau_l$ with $l > k'$. By Lemma 33, $d_l = \bot$. Since $\tau_k$ is a progress point, $\Gamma_{p+k} \vdash \Delta_{p+k}$ is a switching point and $\Gamma_{p+k+1} \vdash \Delta_{p+k+1}$ is the right assumption of the rule.

**Definition 35** (Rightmost path). For a derivation tree $D$ and a node $\sigma$ in $D$, we define the **rightmost path** from the node $\sigma$ as the path $(\sigma_i)_{0 \leq i < \alpha}$ satisfying the following conditions:

1. The node $\sigma_0$ is $\sigma$.
2. If $\sigma_i$ is the conclusion of (Case $\text{Add}_2$), the node $\sigma_{i+1}$ is the right assumption of the rule.
3. If $\sigma_i$ is the conclusion of the rules (Weak), (Subst), ($= \text{L}_a$), or (Add$_1$ R$_2$), the node $\sigma_{i+1}$ is the assumption of the rule.

**Lemma 36.** The rightmost path from an index sequent in $T((D_{cf}, C_{cf}))$ is infinite.

**Proof.** By Definition 31, the rightmost path from an index sequent in $T((D_{cf}, C_{cf}))$ is an index path. By Lemma 32, every sequent on the path is an index sequent. By Definition 28, (Add$_1$ R$_1$) does not occur in the path. Thus, the path is infinite.

**Remark.** For an infinite path in $T((D_{cf}, C_{cf}))$, the corresponding path in $D_{cf}$ has a bud.

### 4.6 Proof of main theorem

We prove Theorem 16.

**Proof of Theorem 16.** (1) The derivation tree given in Figure 4 is the $\text{CLKID}^\omega$ proof of $\text{Add}_2(x, y, z) \vdash \text{Add}_1(x, y, z)$, where $(\dagger)$ indicates the pairing of the companion with the bud, $D_1$ is the derivation tree in Figure 2 (some applying rules and some labels of rules are omitted for limited space). We use the underlined formulas to denote the infinitely progressing trace for the tails of the infinite path.

(2) We show that there exists a sequence $(c_i)_{i \in \mathbb{N}}$ of switching points in $D_{cf}$ which satisfies the following conditions:

1. The height of $c_i$ is greater than the height of $c_{i-1}$ in $D_{cf}$ for $i > 0$.
2. For any node $\sigma$ on the path from the root to $c_i$ in $D_{cf}$ excluding $c_i$, $\sigma$ is a switching point if and only if the child of $\sigma$ on the path is the left assumption of the rule (Case $\text{Add}_2$).
by induction on $i$ and $j$.

By the definition of $(2)$, we have a companion and an infinitely progressing trace following a tail of the path that might use contraction and weakening on antecedents. Therefore, there is a bud on the rightmost path in $T((\mathcal{D}_cf, \mathcal{C}_cf))$ such that its child on the path is the right assumption of the rule. Define $c_i$ as the switching point of the smallest height among such switching points.

We consider the case $i > 0$.

Let $a$ be the left assumption of the rule with the conclusion $c_{i-1}$. Because of $(2)$, the path from the root to $c_{i-1}$ is an infinite path. Hence, there exists at least one switching point on the path. By Lemma 34, there exists a switching point on the path. Hence, there exists a switching point on the rightmost path from the root in $\mathcal{D}_cf$. Let $c_0$ be the switching point with the smallest height among such switching points. $(1)$ and $(2)$ follow immediately for $c_0$.

We consider the case $i = 0$.

Let $a$ be the left assumption of the rule with $c_0$ as the conclusion. Because of $(2)$, the path from the root to $c_0$ is also an infinite path. Hence, there is at least one switching point on the path. By Lemma 32, there exists a switching point on the path. By Lemma 36, there is a switching point on the rightmost path from the root in $\mathcal{D}_cf$.

We complete the construction of the properties. Because of $(1)$, $c_0, c_1, \ldots$ are all distinct in $\mathcal{D}_cf$. Thus, $\{c_i \mid i \in \mathbb{N}\}$ is infinite. This is a contradiction since the set of nodes in $\mathcal{D}_cf$ is finite.

Now, we discuss why we cannot apply the proof technique of a counterexample to cut-elimination in cyclic proofs for separation logic given in [16]. In order to show their counterexample is not provable without a cut rule, assuming that there exists a cut-free proof of the counterexample, they prove that the rightmost path from the root has no infinitely progressing trace following a tail of the path if the path has a companion. On the other hand, the rightmost path from the root in a cut-free CLKIDω pre-proof of $\text{Add}_2(x, y, z) \vdash \text{Add}_1(x, y, z)$ might have a companion and an infinitely progressing trace following a tail of the path that might cause contraction and weakening on antecedents. For example, the rightmost path from the root in Figure 5 has both a companion and an infinitely progressing trace following a tail of the path. Thus, we cannot use their proof technique.

5 Conclusion

We have shown that $\text{Add}_2(x, y, z) \vdash \text{Add}_1(x, y, z)$ is not cut-free provable in CLKIDω but provable in CLKIDω. Consequently, we have shown that $(\text{Cut})$ cannot be eliminated in CLKIDω.
Future work would be (1) to restrict principal formulas of (Cut) with the same provability, (2) to study whether (Cut) can be eliminated in CLKID$\omega$ by restricting the language such as unary predicates, (3) to study a subsystem of LKID$\omega$ including CLKID$\omega$ which satisfies the cut-elimination property.

**Acknowledgments**

We are grateful to Dr. Daisuke Kimura for giving us helpful suggestions from the early stage of this work. We would also like to thank Dr. Taro Sekiyama for giving us valuable comments.

**References**

[1] B. Afshari and G. E. Leigh. Cut-free completeness for modal mu-calculus. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12, June 2017.

[2] Stefano Berardi and Makoto Tatsuta. Classical system of martin-lof’s inductive definitions is not equivalent to cyclic proofs. 12 2017.

[3] Stefano Berardi and Makoto Tatsuta. Equivalence of inductive definitions and cyclic proofs under arithmetic. In *2017 32nd Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–12. IEEE, 2017.

[4] J. Brotherston. *Sequent Calculus Proof Systems for Inductive Definitions*. PhD thesis, University of Edinburgh, 2006.

[5] J. Brotherston and A. Simpson. Sequent calculi for induction and infinite descent. *Journal of Logic and Computation*, 21(6):1177–1216, 2011.

[6] James Brotherston. Cyclic Proofs for First-Order Logic with Inductive Definitions. In Bernhard Beckert, editor, *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 78–92, Berlin, Heidelberg, 2005. Springer Berlin Heidelberg.

[7] James Brotherston. Formalised Inductive Reasoning in the Logic of Bunched Implications. In Hanne Riis Nielson and Gilberto Filé, editors, *Static Analysis*, volume 4634, pages 87–103. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.

[8] James Brotherston, Richard Bornat, and Cristiano Calcagno. Cyclic proofs of program termination in separation logic. *SIGPLAN Not.*, 43(1):101–112, January 2008.

[9] James Brotherston, Nikos Gorogiannis, and Rasmus L. Petersen. A Generic Cyclic Theorem Prover. In David Hutchison, Takeo Kanade, Josef Kittler, Jon M. Kleinberg, Friedemann Mattern, John C. Mitchell, Moni Naor, Oscar Nierstrasz, C. Pandu Rangan, Bernhard Steffen, Madhu Sudan, Demetri Terzopoulos, Doug Tygar, Moshe Y. Vardi, Gerhard Weikum, Ranjit Jhala, and Atsushi Igarashi, editors, *Programming Languages and Systems*, volume 7705, pages 350–367. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[10] Kai Brümmler and Martin Lange. Cut-free sequent systems for temporal logic. *The Journal of Logic and Algebraic Programming*, 76(2):216 – 225, 2008.
[11] Samuel R Buss. An introduction to proof theory. *Handbook of proof theory*, 137:1–78, 1998.

[12] Duc-Hiep Chu, Joxan Jaffar, and Minh-Thai Trinh. Automatic induction proofs of data-structures in imperative programs. In *Proceedings of the 36th ACM SIGPLAN Conference on Programming Language Design and Implementation*, pages 457–466, 2015.

[13] Anupam Das and Damien Pous. A Cut-Free Cyclic Proof System for Kleene Algebra. In Renate A. Schmidt and Cláudia Nalon, editors, *Automated Reasoning with Analytic Tableaux and Related Methods*, pages 261–277, Cham, 2017. Springer International Publishing.

[14] Amina Doumane. *On the infinitary proof theory of logics with fixed points*. PhD thesis, 06 2017.

[15] Amina Doumane, David Baelde, Lucca Hirschi, and Alexis Saurin. Towards completeness via proof search in the linear time µ-calculus: The case of büchi inclusions. In *2016 31st Annual ACM/IEEE Symposium on Logic in Computer Science (LICS)*, pages 1–10. IEEE, 2016.

[16] Daïsuke KIMURA, Koji NAKAZAWA, Tachio TERAUCHI, and Hiroshi UNNO. Failure of cut-elimination in cyclic proofs of separation logic. *Computer Software*, 37(1):39–52, 2020.

[17] Ioannis Kokkinis and Thomas Studer. *Cyclic Proofs for Linear Temporal Logic*, pages 171 – 192. De Gruyter, Berlin, Boston, 25 Jul. 2016.

[18] Rémi Nollet, Alexis Saurin, and Christine Tasson. Local validity for circular proofs in linear logic with fixed points. In *27th EACSL Annual Conference on Computer Science Logic (CSL 2018)*. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2018.

[19] Kenji Saotome, Koji Nakazawa, and Daisuke Kimura. Restriction on Cut in Cyclic Proof System for Symbolic Heaps. In Keisuke Nakano and Konstantinos Sagonas, editors, *Functional and Logic Programming*, pages 88–105, Cham, 2020. Springer International Publishing.

[20] Ulrich Schöpp and Alex Simpson. Verifying Temporal Properties Using Explicit Approximants: Completeness for Context-free Processes. In Mogens Nielsen and Uffe Engberg, editors, *Foundations of Software Science and Computation Structures*, pages 372–386, Berlin, Heidelberg, 2002. Springer Berlin Heidelberg.

[21] Daniyar Salkarbekovich Shamkanov. Circular proofs for the gödel-löb provability logic. *Mathematical Notes*, 96(3-4):575–585, 2014.

[22] Alex Simpson. Cyclic Arithmetic Is Equivalent to Peano Arithmetic. In Javier Esparza and Andrzej S. Murawski, editors, *Foundations of Software Science and Computation Structures*, pages 283–300, Berlin, Heidelberg, 2017. Springer Berlin Heidelberg.

[23] Christoph Sprenger and Mads Dam. On the structure of inductive reasoning: Circular and tree-shaped proofs in the µ-calculus. In *International Conference on Foundations of Software Science and Computation Structures*, pages 425–440. Springer, 2003.

[24] Colin Stirling. A tableau proof system with names for modal mu-calculus. In Andrei Voronkov and Margarita Korovina, editors, *HOWARD-60. A Festschrift on the Occasion of Howard Barringer’s 60th Birthday*, volume 42 of *EPiC Series in Computing*, pages 306–318. EasyChair, 2014.

[25] Colin Stirling and David Walker. Local model checking in the modal mu-calculus. *Theoretical Computer Science*, 89(1):161–177, 1991.

[26] Quang-Trung Ta, Ton Chanh Le, Sian-Cheng Khoo, and Wei-Ngan Chin. Automated lemma synthesis in symbolic-heap separation logic. *Proceedings of the ACM on Programming Languages*, 2(POPL):1–29, 2017.

[27] Makoto Tatsuta, Koji Nakazawa, and Daisuke Kinura. Completeness of Cyclic Proofs for Symbolic Heaps with Inductive Definitions. In Anthony Widjaja Lin, editor, *Programming Languages and Systems*, pages 367–387, Cham, 2019. Springer International Publishing.