Building explicit induction schemas for cyclic induction reasoning

Sorin Stratulat

Building explicit induction schemas for cyclic induction reasoning

Sorin Stratulat

Université de Lorraine
LITA, Department of Computer Science
Ile du Saulcy, Metz, F-57000, FRANCE
sorin.stratulat@univ-lorraine.fr

Abstract. In the setting of classical first-order logic with inductive predicates, two kinds of sequent-based induction reasoning are distinguished: cyclic and structural. Proving their equivalence is of great theoretical and practical interest for the automated reasoning community. In [3,4], it has been shown how to transform any structural proof developed with the LKID system into a cyclic proof using the CLKID$^\omega$ system. However, the inverse transformation was only conjectured.
We provide a simple procedure that performs the inverse transformation for an extension of LKID with explicit induction rules issued from the structural analysis of CLKID$^\omega$ proofs, then establish the equivalence of the two systems. This result is further refined for an extension of LKID with Noetherian induction rules. We show that Noetherian induction subsumes the two kinds of reasoning. This opens the perspective for building new effective induction proof methods and validation techniques supported by (higher-order) certification systems integrating the Noetherian induction principle, like Coq.

1 Introduction

(Mathematical) induction reasoning is known to be one of the most effective reasoning methods. Its distinctive feature is the use, during the proof process, of *induction hypotheses* representing 'not yet proved' formulas.

The induction hypotheses can be defined before their use, by *explicit* induction schemas that can be directly embedded in inference systems using explicit induction rules. On the other hand, the induction hypotheses can also be defined by need, at the moment of their use. In this case, the induction reasoning is schemata-free but should be performed at the proof level because the induction hypotheses can be (instances of) formulas positioned anywhere in the derivation.

The relation between the two kinds of induction reasoning was previously studied for Gentzen-style μ-calculus proof systems and proved *equivalent*, i.e., the proofs are convertible between systems. In [8], the explicit induction proofs are finite and the induction rules locally implement the Noetherian induction principle. The second kind of reasoning is captured by ω-regular proof trees that are finitely represented by stopping the development of the (sub-)derivations

for nodes labelled with formulas already encountered in the derivation. The soundness of the method is ensured if the *cyclic* induction reasoning associated to each strongly connected component of the proof graph is well-founded.

We are interested to extrapolate the equivalence result for sequent-based systems that extend the Gentzen's LK system [6] to reason on inductively defined predicates [1]. We focuss on two representative systems, proposed by Brotherston [3,4]: i) the LKID structural system that integrates induction rules generalizing Noetherian induction reasoning by the means of schemas issued from the recursion analysis of (mutually defined) inductive predicates, and ii) the CLKID$^\omega$ cyclic system, issued from LKID by replacing the induction rules with case-split rules. A main result of [3,4] is that any LKID proof is convertible into a CLKID$^\omega$ proof. However, the conversion in the other direction is only conjectured.

In this paper, we present a procedure that builds *dynamically* explicit induction schemas from the structural analysis of CLKID$^\omega$ proofs. Firstly, we show the equivalence between CLKID$^\omega$ and an *extension* of LKID, denoted by LKID$^\sim$, with induction rules that embed cyclic reasoning from existing CLKID$^\omega$ proofs. The procedure is further refined to build Noetherian induction schemas from the productions defining new inductive predicate symbols attached to strongly connected components. LKID and CLKID$^\omega$, extended to deal with the new symbols, are shown equivalent. We conclude that Noetherian induction is sufficient to represent the structural and cyclic induction reasoning in this setting. Aside the theoretical and technical interests, this result can help developing i) effective sequent-based induction methods that combine the advantages of the two kinds of reasoning, and ii) automated techniques for their validation, supported by (higher-order) certification environments integrating the Noetherian induction principle, like Coq [11].

The paper is structured in 6 sections. Section 2 gives the preliminaries for reasoning with sequent-based inference systems for classical first-order logic (FOL) with inductively defined predicates (FOL$_{\mathrm{ID}}$), then presents the CLKID$^\omega$ and LKID systems. The structural analysis of cyclic proofs is explained in Section 3. LKID$^\sim$ is introduced in Section 4. The extension of LKID with Noetherian induction rules, denoted by LKID$_e$, is presented in Section 5, as well as the procedure transforming CLKID$^\omega$ proofs into LKID$_e$ proofs. As a running example, we consider 'P and Q' [12]. The conclusions are presented in Section 6.

2 Syntax and sequent systems for FOL$_{\mathrm{ID}}$

Syntax. The logical framework is based on FOL$_{\mathrm{ID}}$ using a standard (countable) first-order language Σ. The predicate symbols are labelled either as *ordinary* or *inductive*. We assume that Σ has $P_1, \ldots, P_n$ as inductive predicate symbols. The terms are defined as usual. By $\bar{t}$, we denote a vector of terms $(t_1, \ldots, t_m)$ of length m, the value of m being usually deduced from the context. New terms and formulas are built from replacing (free) variables by terms via substitutions. A *substitution* is a mapping from variables to terms, of the form $\{x_1 \mapsto t_1; \ldots; x_p \mapsto t_p\}$, written in a more compact form as $\{\overline{x} \mapsto \overline{t}\}$, where $\overline{x}$ is the vector of variables

$(x_1, \ldots, x_p)$ and $\bar{t}$ is the vector of terms $(t_1, \ldots, t_p)$. An *identity* substitution consists only of mappings of the form $x \mapsto x$. $\equiv$ denotes the syntactical equality. A term t is an *instance* of t' if there is a substitution σ such that $t \equiv t'\sigma$. Let $FV(\phi)$ denote the free variables of a given formula or set of formulas ϕ. We denote the instance of a formula ψ with σ by $\psi[\sigma]$, where $dom(\sigma) \subseteq FV(\psi)$. Also, given a multiset of formulas Ψ, $\Psi[\sigma]$ denotes the set $\{\psi\sigma | \psi \in \Psi\}$, where $dom(\sigma) \subseteq FV(\Psi)$.

Following [3,4], a specification is the union of the finite inductive definition sets Φ_{P_i} defining each P_i ($i \in [1..n]$) and consisting of productions of the form

$$Q_1(\overline{u}_1), \ldots Q_h(\overline{u}_h), P_{j_1}(\bar{t}_1), \ldots, P_{j_m}(\bar{t}_m) \rightarrow P_i(\bar{t}) \tag{1}$$

where $i, j_1, \ldots j_m \in [1..n]$ and $Q_1, \ldots, Q_h$ are ordinary predicate symbols.

The sequent-based inference system. The Gentzen's LK system [6], presented in Fig. 1, is extended with equality and unfold rules using the definitions of inductive predicates. The proof derivations are built from sequents of the form $\Gamma \vdash \Delta$, where Γ and Δ are finite multisets of formulas, separated by commas. The comma is an associative and commutative operator which can be classically interpreted as a conjunction in the *antecedent* Γ, and as a disjunction in the *succedent* Δ; $\Gamma \vdash \Delta$ can also be interpreted in FOL as the clause $\mathcal{C}(\Gamma \vdash \Delta) \equiv (\bigvee_{F \in \Gamma}(\neg F)) \vee (\bigvee_{F \in \Delta} F)$. By abuse of notation, $FV(\Gamma \vdash \Delta)$ denotes the set of free variables of $\Gamma \vdash \Delta$. The instance of the sequent $\Gamma \vdash \Delta$ with a substitution δ, denoted by $(\Gamma \vdash \Delta)[\delta]$, is $\Gamma[\delta] \vdash \Delta[\delta]$. The inference system is built from inference rules. A rule is represented by a simple horizontal line that separates a sequent, called *conclusion* and written beneath it, from a (potentially empty) multiset of sequents, called *premises* and written above it. Side annotations may specify the name of the rule and its application conditions. *Derivations* are built by successive applications of inference rules and have a tree shape. When it is clear from the context, we use a double line instead of a simple line to mean that formulas may be permuted in the antecedent of the conclusion and one or several occurrences of the rules specified aside are successively applied.

$$\frac{}{\Gamma \vdash \Delta}\ \Gamma \cap \Delta \neq \emptyset\ (\text{Ax}) \qquad \frac{\Gamma' \vdash \Delta'}{\Gamma \vdash \Delta}\ \Gamma' \subseteq \Gamma, \Delta' \subseteq \Delta\ (\text{Wk}) \qquad \frac{\Gamma \vdash F, \Delta \quad \Gamma, F \vdash \Delta}{\Gamma \vdash \Delta}\ (\text{Cut})$$

$$\frac{\Gamma \vdash F, \Delta}{\Gamma, \neg F \vdash \Delta}\ (\neg L) \qquad \frac{\Gamma, F \vdash \Delta}{\Gamma \vdash \neg F, \Delta}\ (\neg R) \qquad \frac{\Gamma, F \vdash \Delta \quad \Gamma, G \vdash \Delta}{\Gamma, F \vee G \vdash \Delta}\ (\vee L) \qquad \frac{\Gamma, F, F \vdash \Delta}{\Gamma, F \vdash \Delta}\ (\text{contrL})$$

$$\frac{\Gamma \vdash F, F\Delta}{\Gamma \vdash F, \Delta}\ (contrR) \qquad \frac{\Gamma \vdash F, G, \Delta}{\Gamma \vdash F \vee G, \Delta}\ (\vee R) \qquad \frac{\Gamma, F, G \vdash \Delta}{\Gamma, F \wedge G \vdash \Delta}\ (\wedge L) \qquad \frac{\Gamma \vdash F, \Delta \quad \Gamma \vdash G, \Delta}{\Gamma \vdash F \wedge G, \Delta}\ (\wedge R)$$

$$\frac{\Gamma \vdash F, \Delta \quad \Gamma, G \vdash \Delta}{\Gamma, F \Rightarrow G \vdash \Delta}\ (\Rightarrow L) \qquad \frac{\Gamma, F \vdash G, \Delta}{\Gamma \vdash F \Rightarrow G, \Delta}\ (\Rightarrow R) \qquad \frac{\Gamma, F \vdash \Delta}{\Gamma, \exists \overline{x} F \vdash \Delta}\ \overline{x} \cap FV(\Gamma \cup \Delta) = \emptyset (\exists L)$$

$$\frac{\Gamma \vdash F, \Delta}{\Gamma \vdash \forall \overline{x} F, \Delta}\ \overline{x} \cap FV(\Gamma \cup \Delta) = \emptyset (\forall R) \qquad \frac{\Gamma, F[\{\overline{x} \mapsto \bar{t}\}] \vdash \Delta}{\Gamma, \forall \overline{x} F \vdash \Delta}\ (\forall L) \qquad \frac{\Gamma \vdash F[\{\overline{x} \mapsto \bar{t}\}], \Delta}{\Gamma \vdash \exists \overline{x} F, \Delta}\ (\exists R)$$

Fig. 1. The LK rules.

The *equality* rules are given in Fig. 2. The (=L) rule replaces in Γ and Δ occurrences of t by u, as well as of u by t.

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

Fig. 2. The equality rules.

Assuming that (1) is the r-th production defining P_i, the *unfold* rule applies on any sequent with a succedent formula $P_i(\overline{t}')$ for which there is a substitution σ such that $P_i(\overline{t}') \equiv P_i(\overline{t})[\sigma]$:

$$\frac{\Gamma \vdash Q_1(\overline{u}_1)[\sigma], \Delta \ldots \Gamma \vdash Q_h(\overline{u}_h)[\sigma], \Delta \quad \Gamma \vdash P_{j_1}(\overline{t}_1)[\sigma], \Delta \ldots \Gamma \vdash P_{j_m}(\overline{t}_m)[\sigma], \Delta}{\Gamma \vdash P_i(\overline{t}'), \Delta}\ (P_i R_r)$$

The CLKID$^\omega$ system. It consists of the rules from Fig. 1 and Fig. 2, as well as the *case-split* rule representing a left-introduction operation for inductive predicate symbols:

$$\frac{\text{case distinctions}}{\Gamma, P_i(\overline{t}') \vdash \Delta}\ (\text{Case } P_i)$$

where, for each production of the form (1), we define the case distinction

$$\Gamma, \overline{t}' = \overline{t}, Q_1(\overline{u}_1), \ldots, Q_h(\overline{u}_h),\ P_{j_1}(\overline{t}_1), \ldots, P_{j_m}(\overline{t}_m) \vdash \Delta \tag{2}$$

such that, for any variable y from (1), we have $y \notin FV(\Gamma \cup \Delta \cup P_i(\overline{t}'))$ (the variable y can be renamed to a fresh one, otherwise). The formulas $P_{j_1}(\overline{t}_1), \ldots, P_{j_m}(\overline{t}_m)$ are said to be *case-descendants* of the *active* formula $P_i(\overline{t}')$.

CLKID$^\omega$ also includes the *substitution* rule:

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

The CLKID$^\omega$ proofs can be represented as finite graphs. Special nodes are the leaves and buds, representing terminal nodes, and some internal nodes referred to as companions.

Definition 2.1 (Leaf, Bud). *Given a derivation tree, by* leaf *we understand any sequent that is the conclusion of a 0-premise inference rule, for example (Ax) or (= R). A* bud *is a sequent that is not the conclusion of any rule.*

Definition 2.2 (Companion). *Any internal node C in a derivation tree is said to be a* companion *for a bud B if C and B have the same sequent labelling.*

Definition 2.3 (CLKID$^\omega$ pre-proof tree, Induction function). *A CLKID$^\omega$* pre-proof tree *of a sequent S is a pair $(\mathcal{D}, \mathcal{R})$, where $\mathcal{D}$ is a finite derivation constructed with CLKID$^\omega$-rules and whose root is S. $\mathcal{R}$ is a defined* induction *function assigning a companion to every bud node in $\mathcal{D}$.*

Definition 2.4 (Path in a pre-proof tree). *A (finite or infinite)* path *in a pre-proof tree $(\mathcal{D}, \mathcal{R})$ is a sequence $(S_i)_{0 \leq i < \alpha}$ of sequents from $\mathcal{D}$, for some $\alpha \in \mathbb{N} \cup \{\infty\}$, such that S_{i+1} is a premise of the rule applied on S_i (resp. $\mathcal{R}(S_i)$) if S_i is a non-bud (resp. bud) node.*

A pre-proof is a proof if any infinite path satisfies some trace condition.

Definition 2.5 (Trace, Progress point). *Let $(\mathcal{D}, \mathcal{R})$ be a CLKID$^\omega$ pre-proof tree and let $(\Gamma_i \vdash \Delta_i)_{i \geq 0}$ be an infinite path from its graph. A* trace *following $(\Gamma_i \vdash \Delta_i)_{i \geq 0}$ is a sequence $(\tau_i)_{i \geq 0}$ such that, for all i, we have:*

1. *$\tau_i = P_{j_i}(\overline{t_i}) \in \Gamma_i$;*
2. *if $\Gamma_i \vdash \Delta_i$ is the conclusion of (Subst) then $\tau_i = \tau_{i+1}[\delta]$, where δ is the substitution associated with the rule instance;*
3. *if $\Gamma_i, t = u \vdash \Delta_i$ is the conclusion of (=L), there is a formula F and variables x, y such that $\tau_i = F[\{x \mapsto t; y \mapsto u\}]$ and $\tau_{i+1} = F[\{x \mapsto u; y \mapsto t\}]$;*
4. *if $\Gamma_i \vdash \Delta_i$ is the conclusion of a case-split rule then either a) $\tau_{i+1} = \tau_i$ or b) τ_i is the active formula of the rule instance and τ_{i+1} is a case descendant of τ_i. In the latter case, i is said to be a* progress point *of the trace;*
5. *if $\Gamma_i \vdash \Delta_i$ is the conclusion of any other rule then $\tau_{i+1} = \tau_i$.*

An *infinitely progressing* trace is a trace with infinitely many progress points.

Definition 2.6 (CLKID$^\omega$ proof tree). *A CLKID$^\omega$* proof *tree is any CLKID$^\omega$ pre-proof tree $(\mathcal{D}, \mathcal{R})$ that satisfies the following global trace condition: for every infinite path $(\Gamma_i \vdash \Delta_i)_{i \geq 0}$ in $\mathcal{D}$, there is an infinitely progressing trace following some tail of the path $(\Gamma_i \vdash \Delta_i)_{i \geq k}$, for some $k \geq 0$.*

The soundness of CLKID$^\omega$ is defined as in [5]. Let $\mathcal{I}$ be a non-empty set of interpretations of sequents. If $I \in \mathcal{I}$, we write $\mathcal{I} \models \mathcal{S}$ to mean that S is true under I. S is *valid* iff $I \models S$, for any $I \in \mathcal{I}$. An *ordinal trace* function can be defined for a CLKID$^\omega$ proof in order to interpret as ordinals the τ-values from the traces following paths from the CLKID$^\omega$ proof. If $I \not\models S_i$, there is $I' \in \mathcal{I}$ such that $I' \not\models S_{i+1}$ and $\alpha \leq \alpha'$, where α (resp. α') is the ordinal assigned to τ_i under I (resp. τ_{i+1} under I'). We have $\alpha < \alpha'$ if i is a progress point. The ordinals are issued from the usual semantics of inductive predicates, which can be generated by sequences of approximants $(P^\gamma)_{\gamma \geq 0}$. The interpretation of the τ-value for $\vdash P(\overline{t})$ under I is the minimal γ such that $\vdash P^\gamma(\overline{t})$ is true under I.

Theorem 2.7 (Soundness [5]). *Let us assume an ordinal trace function for a CLKID$^\omega$ proof tree and $\mathcal{I}$. If S is the root sequent, then S is valid.*

The LKID system. It consists of the rules from Fig. 1 and Fig. 2 as well as the (Subst) and induction rules. The induction rules are left-introduction rules for predicate symbols. Compared to case-split rules, a formula F_j is associated to some inductive symbol P_j, then is instantiated following the schema:

$$\frac{\text{minor premises} \qquad \Gamma, F_j(\overline{t}) \vdash \Delta}{\Gamma, P_j(\overline{t}) \vdash \Delta} \text{ (Ind } P_j\text{)}$$

A *minor premise* is built from each production defining a predicate P_i that is P_j or mutually dependent with P_j. The minor premise corresponding to (1) is

$$\Gamma, Q_1(\overline{u}_1), \ldots Q_h(\overline{u}_h), G_{j_1}(\overline{t}_1), \ldots, G_{j_m}(\overline{t}_m) \vdash F_i(\overline{t}), \Delta \tag{3}$$

if (1) and $\Gamma, P_j(\overline{t}) \vdash \Delta$ do not share variables (otherwise, the variables from (1) can be renamed accordingly), and $G_{j_1}, \ldots, G_{j_m}, F_i$ are predicates associated to $P_{j_1}, \ldots, P_{j_m}, P_i$, respectively. $\Gamma, F_j(\overline{t}) \vdash \Delta$ is called *major premise*.

Definition 2.8 (LKID-proof). *A finite LKID derivation tree is a* proof *if all branches end in an axiom.*

Theorem 2.9 ([3]). *Any LKID proof can be converted to a CLKID$^\omega$ proof.*

3 Structuring cyclic proofs

Any CLKID$^\omega$ proof can be structured by transforming it into a (Subst)-free forest, then defining a well-founded ordering on its strongly connected components.

Converting proof trees to forests. A proof forest can be built from any CLKID$^\omega$ proof tree by deleting its (Subst) steps. Before doing this, any bud S that is not the premise of a (Subst) step will become one by the following 'stuttering' transformation, where σ_{id} is the identity substitution that instantiates the free variables from S:

$$\frac{\ldots \quad S \quad \ldots}{S'}\,(rule) \quad \text{becomes} \quad \frac{\ldots \quad \dfrac{S}{S[\sigma_{id}]}\,(\text{Subst}) \quad \ldots}{S'}\,(rule)$$

Then, the subtree rooted by the premise S of any (Subst) rule that is not a bud node is detached from the proof tree to represent a new derivation tree; a copy of S is kept as a bud node and premise of the (Subst) rule, as below:

$$\frac{\begin{array}{c}\vdots\\ S\end{array}}{S[\delta]}\,(\text{Subst}) \quad \text{becomes} \quad \begin{array}{c}\begin{array}{c}\vdots\\ S\end{array}\\ (new\ tree)\end{array} \qquad \begin{array}{c}(bud)\\ \dfrac{S}{S[\delta]}\,(\text{Subst})\end{array}$$

The premise of any (Subst) rule can now be deleted and a new induction function $\mathcal{R}'$ is defined. The conclusion $S[\delta]$ of (Subst) becomes a bud node for which $\mathcal{R}'(S[\delta]) = (S', \delta)$, where S' is the companion referred to by $\mathcal{R}(S)$ in the CLKID$^\omega$ tree if S is a bud node; otherwise, S' is the root of the new CLKID$^\omega$ tree and labelled as S. S' is referred to as the (forest) *companion* of $S[\delta]$.

Definition 3.1 (CLKID$^\omega$ pre-proof forest). *A CLKID$^\omega$* pre-proof forest *of a sequent S is a pair ($\mathcal{F}$, $\mathcal{R}'$), where $\mathcal{R}'$ is the new induction function and $\mathcal{F}$ is a set of finite derivation trees resulted by deleting the (Subst) rules from the CLKID$^\omega$ proof tree of S.*

Example 3.2 (The 'P and Q' example [12]). Let us consider the productions for the inductive predicate N, as well as the mutually defined predicates P and Q:

$$\begin{array}{ccc} \to N(0) & \to P(0) & \to Q(x,0) \\ N(x) \to N(s(x)) & P(x), Q(x,s(x)) \to P(s(x)) & Q(x,y), P(x) \to Q(x,s(y)) \end{array}$$

The sequent $N(x) \vdash P(x)$ is the root of the annotated CLKID$^\omega$ pre-proof, where $\mathcal{R}((*1)) = \mathcal{R}((*2)) = (*)$ and $\mathcal{R}((\dagger 1)) = (\dagger)$.

$$\dfrac{\dfrac{\dfrac{}{\vdash P(0)}\,(P\mathrm{R}_1)}{x = 0 \vdash P(x)}\,(=L) \qquad \dfrac{\dfrac{N(x) \vdash P(x)\ (*1)}{N(x') \vdash P(x')}\,(\mathrm{Subst}) \qquad (**)}{\dfrac{N(s(x')), N(x') \vdash P(s(x'))}{N(x), x = s(x'), N(x') \vdash P(x)}\,(=L)}\,(P\mathrm{R}_2),(\mathrm{Wk})}{\dfrac{N(x), N(x) \vdash P(x)}{N(x) \vdash P(x)\ (*)}\,(\mathrm{ContrL})}\,(\mathrm{Case}\ N),(\mathrm{Wk})$$

For lack of horizontal space, the rightmost branch of the pre-proof tree, annotated by $(**)$, is developed separately below.

$$\dfrac{\dfrac{\dfrac{}{N(u) \vdash Q(u,0)}\,(Q\mathrm{R}_1)}{N(u), v = 0 \vdash Q(u,v)}\,(=L) \qquad \dfrac{\dfrac{N(x) \vdash P(x)\ (*2)}{N(u) \vdash P(u)}\,(\mathrm{Subst}) \qquad \dfrac{N(u), N(v) \vdash Q(u,v)\ (\dagger 1)}{N(u), N(v') \vdash Q(u,v')}\,(\mathrm{Subst})}{\dfrac{N(u), N(v') \vdash Q(x, s(v'))}{N(u), v = s(v'), N(v') \vdash Q(u,v)}\,(=L)}\,(Q\mathrm{R}_2),(\mathrm{Wk})}{\dfrac{N(u), N(v) \vdash Q(u,v)\ (\dagger)}{N(x'), N(s(x')) \vdash Q(x', s(x'))\ (**)}\,(\mathrm{Subst})}\,(\mathrm{Case}\ Q)$$

The premise of the (Subst) rule applied on $(**)$ is not a bud and a new tree is generated. Finally, the pre-proof forest $\mathcal{F}$ is made of two trees:

$$\dfrac{\dfrac{\dfrac{}{\vdash P(0)}\,(P\mathrm{R}_1)}{x = 0 \vdash P(x)}\,(=L) \qquad \dfrac{\overset{((*),\ \{x \mapsto x'\})}{N(x') \vdash P(x')} \qquad \overset{((\dagger),\ \{u \mapsto x';\, v \mapsto s(x')\})}{N(v'), N(s(x')) \vdash Q(x', s(x'))}\,(P\mathrm{R}_2),(\mathrm{Wk})}{\dfrac{N(s(x')), N(x') \vdash P(s(x'))}{N(x), x = s(x'), N(x') \vdash P(x)}\,(=L)}\,(\mathrm{Case}\ N),(\mathrm{Wk})}{\dfrac{N(x), N(x) \vdash P(x)}{N(x) \vdash P(x)\ (*)}\,(\mathrm{ContrL})}$$

$$\dfrac{\dfrac{\dfrac{}{N(u) \vdash Q(u,0)}\,(Q\mathrm{R}_1)}{N(u), v = 0 \vdash Q(u,v)}\,(=L) \qquad \dfrac{\overset{((*),\ \{x \mapsto u\})}{N(u) \vdash P(u)} \qquad \overset{((\dagger),\{v \mapsto v'\})}{N(u), N(v') \vdash Q(u,v')}\,(Q\mathrm{R}_2),(\mathrm{Wk})}{\dfrac{N(u), N(v') \vdash Q(u, s(v'))}{N(u), v = s(v'), N(v') \vdash Q(u,v)}\,(=L)}\,(\mathrm{Case}\ Q)}{N(u), N(v) \vdash Q(u,v)\ (\dagger)}$$

For each bud, the pair value computed by $\mathcal{R}'$ is displayed above the bud. $\triangle$

Definition 3.3 (Path in a pre-proof forest). *A (finite or infinite)* path *in the pre-proof forest $(\mathcal{F}, \mathcal{R}')$ is a sequence $(S_i)_{0 \le i < \alpha}$ of sequents from $\mathcal{F}$, for some $\alpha \in \mathbb{N} \cup \{\infty\}$, such that S_{i+1} is a premise of the inference rule applied on S_i (resp. S satisfying $\mathcal{R}'(S_i) = (S, \delta)$) in a non-bud (resp. bud) node.*

A finite path $(S_i)_{0 \le i \le n}$ is *bud-free* (resp. *companion-free*) if no S_j with $j \in [0..n-1]$ is a bud (resp. companion).

The traces in pre-proof forests are defined similarly as for pre-proof trees. Let $(\mathcal{F}, \mathcal{R}')$ be a CLKID$^\omega$ pre-proof forest and $(\Gamma_i \vdash \Delta_i)_{i \ge 0}$ a path in it. A *(forest) trace* following $(\Gamma_i \vdash \Delta_i)_{i \ge 0}$ is a sequence $(\tau_i)_{i \ge 0}$ such that, for all i, the conditions from Definition 2.5 are satisfied. In addition, when $\Gamma_i \vdash \Delta_i$ is a bud node for which $\mathcal{R}'(\Gamma_i \vdash \Delta_i) = (S, \delta)$, we have $\tau_i = \tau_{i+1}[\delta]$.

Lemma 3.4. *$\mathcal{G}$ is a path (resp. trace, progress point) in a pre-proof tree iff it is also a path (resp. trace, progress point) in the corresponding pre-proof forest.*

Proof. By the construction of pre-proof forests. □

Definition 3.5 (CLKID$^\omega$ proof forest). *A CLKID$^\omega$* proof *forest is any CLKID$^\omega$ pre-proof forest $(\mathcal{F}, \mathcal{R}')$ that satisfies the following global trace condition: for every infinite path $(\Gamma_i \vdash \Delta_i)_{i\geq 0}$ in $\mathcal{F}$, there is an infinitely progressing trace following some tail of the path $(\Gamma_i \vdash \Delta_i)_{i\geq k}$, for some $k \geq 0$.*

Theorem 3.6. *Let us assume a CLKID$^\omega$ proof tree based on an ordinal trace function and $\mathcal{I}$. The corresponding CLKID$^\omega$ pre-proof forest is a proof.*

Proof. By Lemma 3.4 and the fact that the CLKID$^\omega$ proof tree satisfies the global trace condition. □

Building the partition with ordered parts. Given a pre-proof forest $\mathcal{F}$ built from a proof tree $\mathcal{T}$, we say that a node A is *connected* to a node B if there is a path from A to B. The companions found in each strongly connected component of the graph represented by $\mathcal{F}$ and identifying maximal *cycles* in the graph, will build a part for the partition $\mathcal{P}$. $\mathcal{P}$ will also integrate the singleton containing the root sequent of $\mathcal{T}$ if it is not a companion, as well as the singletons built from the companions that are not included in any strongly connected component.

Example 3.7. The partition generated for the pre-proof forest built by Example 3.2 consists of only one part made of the two companions $N(x) \vdash P(x)$ and $N(u), N(v) \vdash Q(u, v)$. △

Definition 3.8 (dependency relation). *Let π_1 and π_2 be two parts from a partition $\mathcal{P}$. We say that π_2* depends *on π_1 and write $\pi_1 <_\mathcal{P} \pi_2$ if, for any $S_1 \in \pi_1$ and $S_2 \in \pi_2$, there is a path from S_2 to S_1 but no path from S_1 to S_2.*

Lemma 3.9. *$<_\mathcal{P}$ is a terminating ordering relation.*

Proof. We show that $<_\mathcal{P}$ is irreflexive, asymmetric and transitive.

- irreflexivity. For any part $\pi \in \mathcal{P}$ and any two distinct sequents S_1, $S_2 \in \pi$, there is a path from S_1 to S_2 but also a path from S_2 to S_1. So, $\pi \not<_\mathcal{P} \pi$.
- asymmetry. If $\pi_1 <_\mathcal{P} \pi_2$ then $\pi_2 \not<_\mathcal{P} \pi_1$, for any distinct $\pi_1, \pi_2 \in \mathcal{P}$. Otherwise, the elements of π_1 and π_2 are in the same strongly connected component.
- transitivity. Assume that $\pi_1, \pi_2, \pi_3 \in \mathcal{P}$ such that $\pi_1 <_\mathcal{P} \pi_2$ and $\pi_2 <_\mathcal{P} \pi_3$. Then there exist $S_1 \in \pi_1$, $S_2 \in \pi_2$ and $S_3 \in \pi_3$ such that there are two paths leading S_1 to S_2, and S_2 to S_3, respectively. Their concatenation is a path leading S_1 to S_3. On the other hand, there is no sequence followed by a trace leading S_3 to S_1. Otherwise, S_1, S_2 and S_3 would be in the same strongly connected component, hence in the same part. We conclude that $\pi_1 <_\mathcal{P} \pi_3$.

$<_\mathcal{P}$ is also terminating (well-founded) since $\mathcal{P}$ has a finite number of parts. □

4 The LKID$^\sim$ system

We define LKID$^\sim$ as the extension of LKID with the (PInd)-rule:

$$\frac{\text{companions} \qquad \text{axioms}}{S}\ S \in \pi\ (\text{PInd}\ \pi)$$

The rule requires that S be a sequent from the part π included in a partition $\mathcal{P}$ computed for some CLKID$^\omega$ forest proof. For each terminal node or companion N for which there exists a bud- and companion-free path leading a π-sequent to N, we will consider as premise for (PInd):

- N if it is an axiom, or
- the axiom $\mathcal{C}(N) \vdash \mathcal{C}(N)$ if N is a companion $\in \pi$, or
- the axiom $\mathcal{C}(N') \vdash \mathcal{C}(N')$ if N is a bud and $\mathcal{R}'(N) = (N', \delta)$, $N' \in \pi$, or
- N if it is a companion $\notin \pi$, or
- N' if N is a bud, $\mathcal{R}'(N) = (N', \delta)$ and $N' \notin \pi$.

Any LKID$^\sim$ derivation is a proof if all branches end in an axiom.

Example 4.1. The LKID$^\sim$ proof of $N(x) \vdash P(x)$ starts with the application of (PInd), issued from the structural analysis of the proof forest $\mathcal{F}$ from Example 3.2. According to Example 3.7, there is only one partition for $\mathcal{F}$. The premises of (PInd) are built only from the axioms $\vdash P(0)$, $N(u) \vdash Q(u, 0)$, and $\mathcal{C} \vdash \mathcal{C}$, where $\mathcal{C}$ is the clause interpreting each of the sequents $N(x') \vdash P(x')$, $N(v'), N(s(x')) \vdash Q(x', s(x'))$, $N(u) \vdash P(u)$, and $N(u), N(v') \vdash Q(u, v')$. △

Theorem 4.2. *LKID$^\sim$ and CLKID$^\omega$ are equivalent.*

Proof. **The '⇒' part.** We follow similar arguments as in [4]. The transformation of the (Ind) rule is described by Lemma 7.3.1. The (PInd π) rule is translated into the strongly connected component corresponding to the part π computed for a previous CLKID$^\omega$ forest. As in the proof of Theorem 7.3.2, the set of strongly connected components of the resulted CLKID$^\omega$ forest $\mathcal{F}$ is the disjoint union of the strongly connected components introduced by the (Ind) and (PInd) rules. Hence, Proposition 7.2.3 can be applied for $\mathcal{F}$ and conclude that it is a proof.

The '⇐' part. Given a CLKID$^\omega$ proof $\mathcal{D}$, it is firstly transformed into a proof forest $\mathcal{F}$, then into a successive applications of (PInd) rules using parts from $\mathcal{F}$ and starting with the root sequent of $\mathcal{D}$. The resulted LKID$^\sim$ derivation, denoted by $\mathcal{E}$, is finite because i) for any application of a (PInd π) rule, the part π' of any companion from its premises is smaller (w.r.t. $<_\mathcal{P}$) than π, and ii) $<_\mathcal{P}$ is a terminating ordering, by Lemma 3.9. □

5 The LKID$_e$ system

Given the partition $\mathcal{P}$ for a CLKID$^\omega$ proof, the LKID$_e$ system extends LKID to deal with a new predicate symbol P_π defined for each part $\pi \in \mathcal{P}$. The set Φ_{P_π} has a unique production of the form $F(\overline{x}) \to P_\pi(\overline{x})$, where $F(\overline{x})$ is the

conjunction formula $\bigwedge_{S\in\pi}\mathcal{C}(S)$ and $\overline{x}$ is its vector of free variables. The other productions of Φ_{P_π} are of the form $P_\pi(\overline{s}_j) \to P_\pi(\overline{t}_i)$ or $\to P_\pi(\overline{t}_k)$. Productions using non-atomic premises have already been proposed by Martin-Löf [7] for iterated inductive definitions.

The conclusion of the (Ind) rule extended for P_π is the sequent $P_\pi(\overline{x}) \vdash F(\overline{x})$ and the two rightmost premises are axioms (they are further omitted), the first derived from the production $F(\overline{x}) \to P_\pi(\overline{x})$ and the second as the major premise:

$$\frac{F(\overline{s}_j) \vdash F(\overline{t}_i) \;\; ... \qquad \vdash F(\overline{t}_k) \;\; ... \qquad F(\overline{x}) \vdash F(\overline{x}) \qquad F(\overline{x}) \vdash F(\overline{x})}{P_\pi(\overline{x}) \vdash F(\overline{x})}\ (\text{Ind } P_\pi)$$

The LKID$_e$ system also integrates an axiom rule for P_π:

$$\frac{}{\vdash P_\pi(\overline{x})}\ (\text{Ax } P_\pi)$$

The (NInd P_π) rule encodes the Noetherian induction. Defined below on the left of $\equiv$, it is the abbreviation of the LKID$_e$ derivation on the right of $\equiv$:

$$\frac{F(\overline{s}_j) \vdash F(\overline{t}_i) ... \vdash F(\overline{t}_k) ...}{\vdash F(\overline{x})}\ (\text{NInd } P_\pi) \equiv \frac{\dfrac{F(\overline{s}_j) \vdash F(\overline{t}_i) ... \;\vdash F(\overline{t}_k)) ...}{P_\pi(\overline{x}) \vdash F(\overline{x})}\ (\text{Ind } P_\pi) \qquad \dfrac{}{\vdash P_\pi(\overline{x})}\ (\text{Ax } P_\pi)}{\vdash F(\overline{x})}\ (\text{Cut})$$

Definition 5.1 (LKID$_e$-proof). *A finite LKID$_e$ derivation tree is a* proof *if all branches end in an axiom.*

Generating the productions for P_π. Each production for P_π corresponds to an *induction case* of the induction schema built for $\vdash F(\overline{x})$, which is a *composition* of *individual* induction schemas built for the sequents $\vdash \mathcal{C}(S)$, $S \in \pi$.

Building the individual induction schemas. Each node from the pre-proof forest will be decorated by a substitution. Any premise from a case-split rule, of the form (2), will be decorated by $\{x \mapsto t | (x \mapsto t) \in \mu$ and x is a variable from $\overline{t}'\}$, where μ is the mgu of $\overline{t}'$ and $\overline{t}$ such that $\overline{t}'\mu = \overline{t}\mu$. In the following, we assume that $\overline{t}$ and $\overline{t}'$ are constructor terms built only from variables and *free* constructor symbols, i.e. there is no equality relation between two distinct constructor symbols. In this case, the mgu relation can be computed using *syntactic unification* algorithms [2]. The other nodes will be decorated with identity substitutions that instantiate their free variables.

Given the sequents S from π and B from a bud-free path starting with S, called *cumulative* path, we can build a unique *cumulative* substitution.

Definition 5.2 (Cumulative substitution). *Given a cumulative path* $(\Gamma_i \vdash \Delta_i)_{i\in[0..n]}$*, its* cumulative substitution θ *is*

$\{\sigma_0 \dots \sigma_{n-1} \mid \sigma_i$ *is the substitution decorating* $(\Gamma_i \vdash \Delta_i), \forall i \in [0..n-1]\}$

Example 5.3. 0 and s are the free constructor symbols for naturals. The substitutions associated to the left and right premises of the (Case N) rule are $\{x \mapsto 0\}$ and $\{x \mapsto s(x')\}$, and for (Case Q) are $\{v \mapsto 0\}$ and $\{v \mapsto s(v')\}$, respectively.

The cumulative substitutions computed for the cumulative paths leading the root sequents to the terminal nodes of the pre-proof forest built in Example 3.2 are: i) $\{x \mapsto 0\}$ for $\vdash P(0)$, ii) $\{x \mapsto s(x')\}$ for $N(x') \vdash P(x')$ and $N(x'), N(s(x')) \vdash Q(x', s(x'))$, iii) $\{v \mapsto 0\}$ for $N(u) \vdash Q(u, 0)$, and iv) $\{v \mapsto s(v')\}$ for $N(u) \vdash P(u)$ and $N(u), N(v') \vdash Q(u, v')$. △

Lemma 5.4. *If θ_B is the cumulative substitution of the cumulative path $(S_i)_{0\leq i\leq n}$ leading S to B (i.e., $S_0 \equiv S$ and $S_n \equiv B$) i) the rules applied at each step $i \in [0..n-1]$ can be reapplied to build a new path $(S'_i)_{0\leq i\leq n}$ such that $S'_0 \equiv S_0[\theta_B]$, $S'_n \equiv B$, ii) for any $i \in [1..n-1]$, there is a substitution θ_i such that $S'_i[\theta_i] \equiv S_i$, and iii) the (case-split) rules do not instantiate free variables.*

Proof. We will perform by induction on the length of the path leading S_0 to B. If $n = 0$, then $B \equiv S_0$ and θ_B is the identity substitution.

If $n > 0$, let us assume that the path p of length n and leading S_0 to B has the form $S_0, \ldots, S_{n-1}, B$. Also, let $\theta_{S_{n-1}}$ be the cumulative substitution for $S_0, \ldots, S_{n-1}$. By induction hypothesis, we assume that any path of length $n-1$ satisfies the property, in particular $S_0, \ldots, S_{n-1}$. So, the steps from $S_0, \ldots, S_{n-1}$ can be reapplied to build the path $S''_0, \ldots, S''_{n-1}$ such that $S''_0 \equiv S_0[\theta_{S_{n-1}}]$, $S''_{n-1} \equiv S_{n-1}$ and for $i \in [1..n-2]$, there is a substitution θ_i such that $S''_i[\theta'_i] \equiv S_i$.

Let σ be the substitution decorating S_n and θ_B the cumulative substitution for p. σ can be either an identity substitution, or an mgu substitution instantiating free variables from S_{n-1}. For the first case, θ_B equals $\theta_{S_{n-1}}$ and the re-execution of the rule applied at the step $n-1$ of p on S''_{n-1} generates B. In the second case, θ_B is the substitution composition $\theta_{S_{n-1}}\sigma$. The path $(S'_i)_{0\leq i\leq n}$ is defined as $S'_i \equiv S''_i[\sigma]$, for any $i \in [0..n-1]$, and $S'_n \equiv B$. No instantiation of free variables is performed when the case-split rule is applied on S''_{n-1}. □

Example 5.5. Let us assume the cumulative path pre-proof forest given in Example 3.2, for which the sequents are decorated with substitutions (excepting the last sequent): $(N(x) \vdash P(x), \{x \mapsto x\}), (N(x), N(x) \vdash P(x), \{x \mapsto s(x')\}), (N(x), x = s(x'), N(x) \vdash P(x), \{x \mapsto x; x' \mapsto x'\}), (N(s(x')), N(x') \vdash P(s(x')), \{x' \mapsto x'\}), (N(x') \vdash P(x'))$.

The cumulative substitution is $\{x \mapsto s(x')\}$ and the new path is:
$(N(s(x')) \vdash P(s(x'))), (N(s(x')), N(s(x')) \vdash P(s(x'))), (N(s(x')), s(x') = s(x'), N(s(x')) \vdash P(s(x'))), (N(s(x')), N(x') \vdash P(s(x'))), (N(x') \vdash P(x'))$. △

An *induction schema* for a sequent S is a collection of *induction cases* that attach a (potentially empty) set of sequents, called *induction hypotheses*, to an instance of S, called *induction conclusion*.

Definition 5.6 (individual induction schema). *Let $S \in \pi$ ($\pi \in \mathcal{P}$). For each companion-free cumulative path leading S to (a terminal node or companion) N, we build an induction case of the* individual induction schema *for $\vdash \mathcal{C}(S)$ having:*

- *$\vdash \mathcal{C}(S)[\theta_N]$ as induction conclusion built with the cumulative substitution θ_N,*
- *$\vdash \mathcal{C}(S_N)[\delta]$ as induction hypothesis if N is a bud node for which $\mathcal{R}'(N) = (S_N, \delta)$ and $S_N \in \pi$, or $\vdash \mathcal{C}(N)$ if N is a companion from π.*

Example 5.7. Considering the part from Example 3.7, let us denote by $C_1(x)$ (resp. $C_2(u,v)$) the clause $\mathcal{C}(N(x) \vdash P(x))$ (resp. $\mathcal{C}(N(u), N(v) \vdash Q(u,v))$). By using the cumulative substitutions from Example 5.3, the induction cases of the individual induction schema for

- $\vdash C_1(x)$ are: i) $\vdash C_1(0)$, and ii) $\vdash C_1(s(x'))$ with the induction hypotheses $\vdash C_1(x')$ and $\vdash C_2(x', s(x'))$;
- $\vdash C_2(u,v)$ are: i) $\vdash C_2(u,0)$, and ii) $\vdash C_2(u, s(v'))$ with the induction hypotheses $\vdash C_1(u)$ and $\vdash C_2(u,v')$. $\triangle$

Generating productions for new predicate symbols. Let $\pi_{i_1 \ldots i_k}$ be a part built on $k(>0)$ sequents $S_{i_1}, \ldots, S_{i_k}$, where $i_1, \ldots, i_k \in [1..k]$ is a permutation of the values from $[1..k]$. Let also $C_{i_p}(\overline{x}_{i_p})$ be $\mathcal{C}(S_{i_p})$ ($p \in [1..k]$) with $\overline{x}_{i_p}$ the vector of free variables from $\mathcal{C}(S_{i_p})$ and $D^1_{i_p} \times \ldots \times D^{j_{i_p}}_{i_p}$ the domain of C_{i_p}. The new inductive predicate symbol $P_{\pi_{i_1 \ldots i_k}}$ with the domain $D^1_{i_1} \times \ldots \times D^{j_{i_1}}_{i_1} \times \ldots \times D^1_{i_k} \times \ldots \times D^{j_{i_k}}_{i_k}$ is defined by the smallest set of productions of the form

$$\textbf{premises} \rightarrow P_{\pi_{i_1 \ldots i_k}}(t^1_{i_1}, \ldots, t^{j_{i_1}}_{i_1}, \ldots, t^1_{i_k}, \ldots, t^{j_{i_k}}_{i_k}) \tag{4}$$

Each of the sequents $\vdash C_{i_p}(t^1_{i_1}, \ldots, t^{j_{i_1}}_{i_1}, \ldots, t^1_{i_p}, \ldots, t^{j_{i_p}}_{i_p})$ ($p \in [1..k]$) defined on every (sub)vector given as argument to the conclusion of (4) is the conclusion of an individual induction case defined for $\vdash C_{i_p}(\overline{x}_{i_p})$. For each of its attached induction hypotheses, there is a premise of (4) defined as follows. If the induction hypothesis is the sequent $\vdash C_{i_r}(\overline{t}_{i_r})$ ($r \in [1..k]$), the premise is the induction conclusion of (4) for which the rth (sub)vector is replaced by $\overline{t}_{i_r}$. The premises that replace distinct (sub)vectors are factorized into a single premise.

Example 5.8. $P_{\pi_{12}}$ is defined by the productions issued from the individual induction schemas for $\vdash C_1(x)$ and $\vdash C_2(u,v)$, computed at Example 5.7: $\rightarrow P_{\pi_{12}}(0,u,0)$, $P_{\pi_{12}}(x', x', s(x')) \rightarrow P_{\pi_{12}}(s(x'), u, 0)$, $P_{\pi_{12}}(u,u,v') \rightarrow P_{\pi_{12}}(0,u,s(v'))$, and $P_{\pi_{12}}(x', x', s(x')), P_{\pi_{12}}(u,u,v') \rightarrow P_{\pi_{12}}(s(x'), u, s(v'))$. $\triangle$

Applying the (NInd) rule on a part sequent. The proof of $\Gamma \vdash \Delta \in \pi_{i_1 \ldots i_k}$ starts by applying a (Cut) using the conjunction formula of $\pi_{i_1 \ldots i_k}$:

$$\dfrac{\dfrac{\dfrac{}{\Gamma, \mathcal{C}(\Gamma \vdash \Delta) \vdash \Delta}\,(\vee L),(\neg L),(\text{Ax})}{\bigwedge_{S \in \pi_{i_1 \ldots i_k}} \mathcal{C}(S), \Gamma \vdash \Delta}\,(\wedge L),(\text{Wk}) \qquad \dfrac{\dfrac{(\textit{the induction cases})}{\vdash \bigwedge_{S \in \pi_{i_1 \ldots i_k}} \mathcal{C}(S)}\,(\text{NInd } P_{\pi_{i_1 \ldots i_k}})}{\Gamma \vdash \bigwedge_{S \in \pi_{i_1 \ldots i_k}} \mathcal{C}(S), \Delta}\,(\text{Wk})}{\Gamma \vdash \Delta}\,(\text{Cut})$$

Each induction case is further transformed into the individual induction cases that built it, by firstly deleting the conjunction symbols from the antecedent using successive applications of $(\wedge L)$ rules, then the conjunction symbols from the succedent using the $(\wedge R)$ rule. The antecedent formulas from each new sequent, that are not induction hypotheses attached to the induction conclusion from the succedent, are deleted by the (Wk) rule. The induction step is summarized as:

$$\dfrac{\{C_1 \dots C_p \vdash C \mid \text{`}\vdash C \text{ attaches } \vdash C_1 \dots \vdash C_p\text{' is an individual case}\}}{\vdash C_{i_1}(x^1_{i_1}, \dots, x^{j_{i_1}}_{i_1}) \wedge \dots \wedge C_{i_k}(x^1_{i_k}, \dots, x^{j_{i_k}}_{i_k})}\ (\text{NInd}\quad P_{\pi_{i_1 \dots i_k}}),\ (\wedge L),(\wedge R),(\text{Wk})$$

Example 5.9. The LKID derivation of $N(x) \vdash P(x)$ from Example 3.2 starts as

$$\dfrac{\dfrac{\dfrac{\dfrac{}{N(x), P(x) \vdash P(x)}\,(\text{Ax}) \qquad \dfrac{\dfrac{}{N(x) \vdash N(x), P(x)}\,(\text{Ax})}{N(x), \neg N(x) \vdash P(x)}\,(\neg L)}{N(x), C_1(x)(\equiv \neg N(x) \vee P(x)) \vdash P(x)}\,(\vee L)}{N(x), C_1(x), C_2(u,v) \vdash P(x)}\,(\text{Wk})}{N(x), C_1(x) \wedge C_2(u,v) \vdash P(x)}\,(\wedge L) \qquad \dfrac{(\dagger)}{N(x) \vdash C_1(x) \wedge C_2(u,v), P(x)}\,(\text{Wk})}{N(x) \vdash P(x)}\,(\text{Cut})$$

(NInd $P_{\pi_{12}}$) is applied on (†) using the productions from Example 5.8:

$$\dfrac{\vdash C_1(0) \wedge C_2(u,0) \quad C_1(x') \wedge C_2(x', s(x')) \vdash C_1(s(x')) \wedge C_2(u,0) \quad (*) \quad (**)}{\vdash C_1(x) \wedge C_2(u,v)\ (\dagger)}\,(\text{NInd } P_{\pi_{12}})$$

The derivations of its premises are

$$\dfrac{\vdash C_1(0) \qquad \vdash C_2(u,0)}{\vdash C_1(0) \wedge C_2(u,0)}\,(\wedge R)$$

$$\dfrac{\dfrac{C_1(x'), C_2(x', s(x')) \vdash C_1(s(x')) \qquad \dfrac{\vdash C_2(u,0)}{C_1(x'), C_2(x', s(x')) \vdash C_2(u,0)}\,(\text{Wk})}{C_1(x'), C_2(x', s(x')) \vdash C_1(s(x')) \wedge C_2(u,0)}\,(\wedge R)}{C_1(x') \wedge C_2(x', s(x')) \vdash C_1(s(x')) \wedge C_2(u,0)}\,(\wedge L)$$

$$\dfrac{\dfrac{\dfrac{\vdash C_1(0)}{C_1(u), C_2(u,v') \vdash C_1(0)}\,(\text{Wk}) \qquad C_1(u), C_2(u,v') \vdash C_2(u, s(v'))}{C_1(u), C_2(u,v') \vdash C_1(0) \wedge C_2(u, s(v'))}\,(\wedge R)}{C_1(u) \wedge C_2(u,v') \vdash C_1(0) \wedge C_2(u, s(v'))\ (*)}\,(\wedge L)$$

$$\dfrac{\dfrac{C_1(x'), C_2(x', s(x')) \vdash C_1(s(x')) \quad C_1(u), C_2(u,v') \vdash C_2(u, s(v'))}{C_1(x'), C_2(x', s(x')), C_1(u), C_2(u,v') \vdash C_1(s(x')) \wedge C_2(u, s(v'))}\,(\wedge R),(\text{Wk})}{C_1(x') \wedge C_2(x', s(x')), C_1(u) \wedge C_2(u,v') \vdash C_1(s(x')) \wedge C_2(u, s(v'))\ (**)}\,(\wedge L)$$

Proving the individual induction cases. The derivation of the individual case $C_1, \dots, C_p \vdash \mathcal{C}(\Gamma \vdash \Delta)[\theta]$ for $\Gamma \vdash \Delta$ using the cumulative substitution θ is

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\Gamma_j, C_j[\delta] \vdash \Delta_j}\,(\vee L),(\neg L),(\text{Ax})}{\Gamma_j, C_j \vdash \Delta_j}\,(\forall L)}{C_1, \dots, C_j, \dots, C_p, \Gamma_j \vdash \Delta_j}\,(\text{Wk}) \qquad \dfrac{\dfrac{\textit{proof of comp.}}{\Gamma_j \vdash \Delta_j}}{\dfrac{\Gamma_j[\delta] \vdash \Delta_j[\delta]}{\textit{other bud}}\,(\text{Wk})}\,(\text{Subst}) \qquad \dfrac{\textit{proof of comp.}}{\dfrac{\Gamma_j \vdash \Delta_j}{\textit{other comp.}}\,(\text{Wk})}}{\textit{comp./bud with comp. in } \pi_{i_1 \dots i_k}} \qquad \dfrac{}{\textit{leaf}}\,(\text{Ax})$$

$$\vdots$$

(*following the companion (comp.)-free cumulative paths with θ as cumulative substitution*)

$$\dfrac{C_1, \dots, C_p, \Gamma[\theta] \vdash \Delta[\theta]}{C_1, \dots, C_p \vdash \mathcal{C}(\Gamma \vdash \Delta)[\theta]}\,(\vee R),(\neg R)$$

The derivation of $C_1, \ldots, C_p, \Gamma[\theta] \vdash \Delta[\theta]$ starts by executing the rules from the companion-free cumulative paths of the CLKID$^\omega$ pre-proof forest leading $\Gamma \vdash \Delta$ to the terminal nodes and companions for which the cumulative substitution is θ, as shown in the proof of Lemma 5.4. We perform a case analysis on them.

The LKID$_e$ derivation for a leaf is (Ax). For any companion $\Gamma_j \vdash \Delta_j$ or bud whose companion $\Gamma_j \vdash \Delta_j$ is not in $\pi_{i_1 \ldots i_k}$, the derivation is developed as for $\Gamma \vdash \Delta$. The sequent corresponding to any other node has in the antecedent a clause C that interprets the node; it is proved by decomposing C, as previously.

Example 5.10. The derivations of the individual induction cases $\vdash C_1(0)$ and $\vdash C_2(u, 0)$ from Example 5.9, after expansion to their clausal definition, are:

$$\dfrac{\dfrac{}{\vdash P(0)}\,(\text{Ax})}{\vdash \neg N(0) \vee P(0)}\,(\vee R),(\text{Wk}) \qquad \dfrac{\dfrac{}{\vdash Q(u,0)}\,(\text{Ax})}{\vdash \neg N(u) \vee \neg N(0) \vee Q(u,0)}\,(\vee R),(\text{Wk})$$

The derivation of the case $C_1(u), C_2(u, v') \vdash C_2(u, s(v'))$ is:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{P(u) \vdash \neg N(u), P(u)}\,(\text{Ax}) \quad \dfrac{}{\neg N(u) \vdash \neg N(u), P(u)}\,(\text{Ax})}{\neg N(u) \vee P(u) \vdash \neg N(u), P(u)}\,(\vee L)}{\neg N(u) \vee P(u), \neg N(u) \vee \neg N(v') \vee Q(u,v') \vdash \neg N(u), \neg N(v'), Q(u, s(v'))}\,(\text{QR}_2),(\text{Wk})}{\neg N(u) \vee P(u), \neg N(u) \vee \neg N(v') \vee Q(u,v') \vdash \neg N(u) \vee \neg N(v') \vee Q(u, s(v'))}\,(\vee R)$$

(**)

where the derivation of the leftmost sequent, denoted by (**), is

$$\dfrac{}{\neg N(u) \vee \neg N(v') \vee Q(u,v') \vdash \neg N(u), \neg N(v'), Q(u,v')\ (**)}\,(\vee L),\ (\text{Ax})$$

The derivation of $C_1(x'), C_2(x', s(x')), N(x') \vdash C_1(s(x'))$ is done similarly. $\triangle$

Lemma 5.11. *(NInd) is locally sound and implements Noetherian induction.*

Theorem 5.12. *LKID$_e$ and CLKID$^\omega$ are equivalent.*

Proof. **The ‘ $\Rightarrow'$ part.** (Ind) using new predicate symbols is an instance of LKID's (Ind), hence translatable into CLKID$^\omega$. The application of (Ax P_π) on $\vdash P_\pi(\overline{x})$ is translated to the application of $(P_\pi R)$ using the production $\bigwedge_{S \in \pi} \mathcal{C}(S) \rightarrow P_\pi(\overline{x})$ which yields the sequent $\vdash \bigwedge_{S \in \pi} \mathcal{C}(S)$. By decomposing it using conveniently the $(\wedge R)$, $(\vee R)$ and $(\neg R)$ rules, we get the sequents $\vdash S$, $S \in \pi$. Each of them is proved by the CLKID$^\omega$ (forest) proof of π.
The ‘ $\Leftarrow'$ part. By generating the partition $\mathcal{P}$ from the CLKID$^\omega$ proof $\mathcal{T}$, then applying recursively the procedure for developing the derivation of a part sequent starting with the root of $\mathcal{T}$. The process terminates as $<_{\mathcal{P}}$ is terminating. $\square$

Corollary 5.13. *Noetherian induction can represent structural and cyclic induction reasoning performed by LK extended with new inductive predicates.*

Proof. By Theorems 4.2 and 5.12 since our procedure converts any CLKID$^\omega$ proof to an LKID$_e$ proof using only (NInd)-based induction reasoning. $\square$

6 Conclusions and perspectives

We have shown that structural and cyclic induction reasoning in FOL_{ID} are convertible and are subsumed by Noetherian induction. Compared to LKID, LKID_e can generate proofs that combine the advantages of structural and cyclic reasoning by the dynamic definition of new induction schemas. Our procedure transforms effectively cyclic to structural proofs and does not require unfolding operations as in [8] or as suggested in [3,4].

These results open the perspective for validating CLKID^ω-like proofs by (higher-order) certification environments integrating the Noetherian induction principle, like Coq [11], by using similar certification methodologies as for implicit induction [10]. Also, it would be interesting to extend CLKID^ω for integrating reductive-free cycles [9].

References

1. P. Aczel. An introduction to inductive definitions. In J. Barwise, editor, *Handbook of Mathematical Logic*, pages 739–782. North Holland, 1977.
2. F. Baader and T. Nipkow. *Term Rewriting and All That.* Cambridge University Press, 1998.
3. J. Brotherston. Cyclic proofs for first-order logic with inductive definitions. In *Proceedings of TABLEAUX-14*, volume 3702 of *LNAI*, pages 78–92. Springer-Verlag, 2005.
4. J. Brotherston. *Sequent Calculus Proof Systems for Inductive Definitions.* PhD thesis, University of Edinburgh, November 2006.
5. J. Brotherston, N. Gorogiannis, and R. L. Petersen. A generic cyclic theorem prover. In R. Jhala and A. Igarashi, editors, *Proceedings of Programming Languages and Systems - 10th Asian Symposium (APLAS-10)*, volume 7705 of *LNCS*, pages 350–367. Springer, 2012.
6. G. Gentzen. Untersuchungen über das logische Schließen. I. *Mathematische Zeitschrift*, 39:176–210, 1935.
7. P. Martin-Löf. Haupstatz for the intuitionistic theory of iterated inductive definitions. In *Proceedings of the Second Scandinavian Logic Symposium*, pages 179–216. North-Holland, 1971.
8. C. Sprenger and M. Dam. On the structure of inductive reasoning: Circular and tree-shaped proofs in the μ calculus. In A. Gordon, editor, *Foundations of Software Science and Computation Structures*, volume 2620 of *Lecture Notes in Computer Science*, pages 425–440. Springer Berlin / Heidelberg, 2003.
9. S. Stratulat. A unified view of induction reasoning for first-order logic. In A. Voronkov, editor, *Turing-100 (The Alan Turing Centenary Conference)*, volume 10 of *EPiC Series*, pages 326–352. EasyChair, 2012.
10. S. Stratulat and V. Demange. Automated certification of implicit induction proofs. In *CPP'2011 (First International Conference on Certified Programs and Proofs)*, volume 7086 of *Lecture Notes Computer Science*, pages 37–53. Springer Verlag, 2011.
11. The Coq Development Team. *The Coq Reference Manual - version 8.4.* INRIA, 2013.
12. C.-P. Wirth. Descente infinie + Deduction. *Logic Journal of the IGPL*, 12(1):1–96, 2004.

A The proof of Lemma 5.11

Let π be a part of the partition $\mathcal{P}$ from a proof forest $\mathcal{F}$ built from a CLKID$^\omega$ proof tree $\mathcal{T}$, based on the ordinal trace function f and the non-empty set of interpretations $\mathcal{I}$.

By absurd, assume that (NInd P_π) is applied in some CLKID$_e$ derivation $\mathcal{D}$ with valid premises but false conclusion. The conclusion is $\vdash F(\overline{x})(\equiv \bigwedge_{S\in\pi}\mathcal{C}(S))$, hence there is a sequent $S \in \pi$ that is false. By Definition 3.5, the definition of ordinal trace functions and Lemma 3.4, there is an infinite path $S_{i\geq 0}$ of false sequents in $\mathcal{F}$ starting with S, and an infinitely progressing trace following some tail of $S_{i\geq k}$, for some $k \geq 0$. Since $\mathcal{F}$ has a finite number of nodes, there is a companion S' that is infinitely often visited in the path. Since any companion from $\mathcal{F}$ is in a part of $\mathcal{P}$, we consider $\pi' \in \mathcal{P}$ such that $S' \in \pi'$.

The rule (NInd $P_{\pi'}$) should be also applied in $\mathcal{D}$, otherwise S' cannot occur infinitely in $S_{i\geq k}$. Its conclusion $\vdash F'(\overline{x'})(\equiv \bigwedge_{S\in\pi'}\mathcal{C}(S))$ is false. Let S_{k_1} be the first occurence of S' in $S_{i\geq k}$ and S_{k_2}, with $k_1 < k_2$, the occurence of the next visited companion $S'' \in \pi'$ in $S_{i\geq k}$. Let also assume that the τ-value of S_{k_1} and S_{k_2} in the trace is α_{k_1} and α_{k_2}, respectively. There should be a premise of (NInd $P_{\pi'}$), built from a companion-free cumulative path $\in \mathcal{F}$ leading a false cumulative instance of S' to a false instance of S'' or some bud node whose companion is S'', by Lemma 5.4. We denote by $F'(\bar{t}_{k_2}) \vdash F'(\bar{t}_{k_1})$ the instance of the premise leading S_{k_1} to S_{k_2} and notice that both $F'(\bar{t}_{k_2})$ and $F'(\bar{t}_{k_1})$ are false. We associate to the term vectors $\bar{t}_{k_1}$ and $\bar{t}_{k_2}$ the weights α_{k_1} and α_{k_2}, respectively. We proceed similarly for S_{k_2} as for S_{k_1}, as well as for any future occurrence of companions from π' in $S_{i\geq k}$.

Let us define $\mathcal{E}' = \{\bar{t}_p |\ S_p$ is an occurrence of S' in $S_{i\geq k_1}$ and $F'(\bar{t}_p)$ has $\mathcal{C}(S_p)$ as one of its conjuncts$\}$. We have that

$$(\forall \bar{t}_p \in \mathcal{E}', \neg F'(\bar{t}_p) \Rightarrow (\exists \bar{t}_j \in \mathcal{E}', \bar{t}_j < \bar{t}_p \wedge \neg F'(\bar{t}_j))) \tag{5}$$

i) by taking, for an arbitrary $\bar{t}_p$, the vector $\bar{t}_j$ such that S_j is the next occurrence of S' in $S_{i\geq k_1}$ that follows S_p for which there is a progress point between S_j and S_p, ii) by successively considering the companions from π' in $S_{j\leq i\leq p}$, and iii) by using the definition of ordinal trace functions and the transitivity property of $<$. The Noetherian induction principle

$$\forall\phi, (\forall m \in \mathcal{E}, (\forall k \in \mathcal{E}, k < m \Rightarrow \phi(k)) \Rightarrow \phi(m)) \Rightarrow \forall p \in \mathcal{E}, \phi(p)$$

can be applied on the contrapositive version of (5) when instantiating $\mathcal{E}$ by $\mathcal{E}'$ and ϕ by F', to conclude that $\forall p \in \mathcal{E}', F'(p)$. Contradiction since $F'(p)$ is false, for any $p \in \mathcal{E}'$. □