Sequent calculus proof systems for inductive definitions

James Brotherston

Doctor of Philosophy
Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh
2006

Abstract

Inductive definitions are the most natural means by which to represent many families of structures occurring in mathematics and computer science, and their corresponding induction / recursion principles provide the fundamental proof techniques by which to reason about such families. This thesis studies formal proof systems for inductive definitions, as needed, e.g., for inductive proof support in automated theorem proving tools. The systems are formulated as sequent calculi for classical first-order logic extended with a framework for (mutual) inductive definitions.

The default approach to reasoning with inductive definitions is to formulate the induction principles of the inductively defined relations as suitable inference rules or axioms, which are incorporated into the reasoning framework of choice. Our first system LKID adopts this direct approach to inductive proof, with the induction rules formulated as rules for introducing atomic formulas involving inductively defined predicates on the left of sequents. We show this system to be sound and cut-free complete with respect to a natural class of Henkin models. As a corollary, we obtain cut-admissibility for LKID.

The well-known method of *infinite descent* à la Fermat, which exploits the fact that there are no infinite descending chains of elements of well-ordered sets, provides an alternative approach to reasoning with inductively defined relations. Our second proof system LKID^ω formalises this approach. In this system, the left-introduction rules for formulas involving inductively defined predicates are not induction rules but simple case distinction rules, and an infinitary, global *soundness condition* on proof trees — formulated in terms of "traces" on infinite paths in the tree — is required to ensure soundness. This condition essentially ensures that, for every infinite branch in the proof, there is an inductive definition that is unfolded infinitely often along the branch. By an infinite descent argument based upon the well-foundedness of inductive definitions, the infinite branches of the proof can thus be disregarded, whence the remaining portion of proof is well-founded and hence sound. We show this system to be cut-free complete with respect to standard models, and again infer the admissibility of cut.

The infinitary system LKID^ω is unsuitable for formal reasoning. However, it has a natural restriction to proofs given by regular trees, i.e. to those proofs representable by finite graphs. This restricted "cyclic" proof system, CLKID^ω, *is* suitable for formal reasoning since proofs have finite representations and the soundness condition on proofs is thus decidable.

We show how the formulation of our systems LKID^ω and CLKID^ω can be generalised to obtain soundness conditions for a general class of infinite proof systems and their corresponding cyclic restrictions. We provide machinery for manipulating and analysing the structure of proofs in these essentially arbitrary cyclic systems, based primarily on viewing them as generating regular infinite trees, and we show that any proof can be converted into an equivalent proof with a restricted cycle structure. For proofs in this "cycle normal form", a finitary, lo-

calised soundness condition exists that is strictly stronger than the general, infinitary soundness condition, but provides more explicit information about the proof.

Finally, returning to the specific setting of our systems for inductive definitions, we show that any LKID proof can be transformed into a CLKID$^{\omega}$ proof (that, in fact, satisfies the finitary soundness condition). We conjecture that the two systems are in fact equivalent, i.e. that proof by induction is equivalent to regular proof by infinite descent.

Acknowledgements

I owe a debt of thanks — and in some cases, money — to several people and organisations without whom this thesis would not exist.

First, and foremost, I should like to extend my sincerest gratitude and respect to my supervisor, Alex Simpson. Alex contributed technical advice, ideas, references, culture, detailed and constructive feedback, and personal support, and generally taught me an awful lot about how to be a researcher. It's a real privilege to have been his student, and I can only hope that some of his insight and critical ability turns out to have rubbed off on me.

I should also like to thank my second supervisor Alan Smaill, and Alberto Momigliano who acted as my second supervisor while Alan was on sabbatical for a year. I am also grateful to my PhD advisory panel, which consisted of the two aforementioned gentlemen and also Alan Bundy, for providing useful annual feedback on my overall progress.

I would like to especially thank René Vestergaard for introducing me to academic research in the first place, as well as hosting me on a visit to JAIST in October 2004, and offering many useful pieces of advice during our acquaintance. I'm also indebted to colleagues and those in the academic community who offered helpful discussions, references and advice including (but probably not limited to) Lucas Dixon, Ross Duncan, Peter Dybjer, Roy Dyckhoff, Jeremy Gow, Geoff Hamilton, Conor McBride, Dale Miller, Sara Negri, Claus Peter-Wirth, Frank Pfenning, and Christoph Sprenger, as well as those mentioned previously.

This PhD was made possible in the first place by the LFCS in the School of Informatics — who hosted me and provided a first-class research environment, as well as office space and logistical support — and by EPSRC, who funded the first three years of the work through an EPSRC PhD studentship. For the period thereafter, I would like to extend my gratitude to the the University of Edinburgh Hardship Fund (a charitable organisation), who were able to provide help when it was most needed, and to the generous people who fed me and bought me drinks.

The figures appearing in this thesis were produced with Paul Gastin's `gastex` package (for LaTeX), and the sequent calculus proofs were produced using Paul Taylor's `prooftree` package. Both are excellent, as is the `WinEdt` program I used for editing the thesis.

I would like to sincerely thank my friends and family for all their support, particularly during the long and fraught writing-up period. By writing a long list I run the very real risk of accidentally leaving someone out, so I will simply say that you all know who you are: thank you and I'll buy you a beer.

Having said that, there is nevertheless one very special person who deserves a proper mention. Jennie Fraser has been doing her utmost to keep me sane and solvent for the last couple of years, and I can't thank her enough for all her love and support, except perhaps by offering all mine in return. Thank you, Jennie.

Declaration

I declare that this thesis was composed by myself, that the work contained herein is my own except where explicitly stated otherwise in the text, and that this work has not been submitted for any other degree or professional qualification except as specified.

(*James Brotherston*)

Table of Contents

Chapter 1

Introduction

1.1 Overview

Inductive definitions are frequently encountered throughout mathematics and computer science, and the corresponding use of inductive proof methods to reason about inductively defined structures constitutes a fundamental part of mathematical reasoning. In all cases the aim of inductive reasoning is to exploit recursion in the definition of the inductively defined structures under consideration in order to prove general properties about those structures. The canonical example of an inductively defined structure occurring in mathematics is the set $\mathbb{N}$ of natural numbers, which can be given by the following inductive definition:

- 0 is a natural number;
- if n is a natural number, then so is $s(n)$ (the *successor* of n, i.e. $n+1$).

Mathematical reasoning about inductively defined structures is most often formulated in one of two main styles. The default approach, commonly termed *mathematical induction*, explicitly applies an induction principle derived from the inductive definition. For example, the induction principle for the natural numbers states that in order to prove that a general proposition P is true of every natural number, it suffices to demonstrate, firstly, that P is true of the number 0 and, secondly, that if P is true of an arbitrary natural number n then it is also true of $s(n)$. The second approach, called *infinite descent*, appeals directly to the well-foundedness of inductively defined structures. For example, one way of stating the infinite descent principle for the natural numbers is that in order to prove that P is true for every natural number, it suffices to demonstrate that if P is *not* true of a particular natural number n, then there exists an infinite strictly decreasing chain of natural numbers, which is impossible. (Typically, one shows that there exists a strictly smaller natural number $m < n$ such that P is not true of m, which implies the existence of such a chain.)

In computer science the use of inductive definitions is, if anything, perhaps even more widespread than in mathematics taken as a whole; many of the data structures ubiquitous throughout the subject, such as lists and trees, are most naturally given by inductive definitions. There has also been considerable effort devoted to the area of mechanised inductive theorem proving, that is to say, the formalisation and automation of inductive mathematical proofs using theorem proving tools. Despite considerable progress in this area, the automation of inductive reasoning still poses significant problems for the theorem-proving community. In particular, the simulation of the creativity required in inductive proofs in the selection of an appropriate induction schema and in the choice of appropriate induction hypotheses and intermediate lemmas causes huge difficulties for mechanical proof search, even when the search procedure is guided by heuristics derived from human proof attempts. For a survey article on inductive theorem proving see e.g. [12].

While there remain formidable obstacles in the way of successful automated theorem proving with inductive definitions, we believe that it is potentially useful as well as interesting to gain a clearer picture of the main relevant proof principles, i.e. mathematical induction and infinite descent, from a proof-theoretic standpoint. Accordingly, the goal of this thesis is to develop proof-theoretic foundations for reasoning with inductive definitions, in particular the methods of mathematical induction and infinite descent, and to thereby undertake a formal analysis and comparison of these methods. Although such a study has the benefit of being free from mechanical constraints, we hope that our theoretical analysis will nevertheless impact upon (some of) the practical considerations driving inductive theorem proving.

1.2 Mathematical induction

Although the method of induction has been employed in mathematical proofs for (at least) several centuries, according to the *Encyclopaedia Britannica* [2] the method was not formally identified under the term "mathematical induction" until 1838 by Augustus de Morgan [20]:

> The method of induction, in the sense in which the word is used in natural philosophy, is not known in pure mathematics. . .
>
> There is however one particular method of proceeding which is extremely common in mathematical investigation, and to which we propose to give the name "successive induction". It has the main character of induction in physics, because it is really the collection of a general truth from a demonstration which implies the examination of every particular case; but it differs from the process of physics inasmuch as each case depends on one which precedes . . .
>
> An instance of mathematical induction occurs in every equation of differences, in every recurring series, &c.
>
> *Augustus de Morgan, "Induction (Mathematics)", 1838*

The usual schema for mathematical induction over the natural numbers ($\mathbb{N}$) is the following:

$$\frac{P(0) \quad \forall x \in \mathbb{N}.\ (P(x) \to P(sx))}{P(t)}$$

where $P(x)$ is a statement of the underlying language (e.g. set theory) containing a free variable x, and t is an arbitrary term of the language. In an application of the induction schema above, the induction is said to take place on the variable x, the premise $P(0)$ is usually called the *base case* of the induction and the premise $\forall x \in \mathbb{N}\ (P(x) \to P(sx))$ is called the *step case*. In the step case, the formula $P(x)$ is called the *induction hypothesis* and normally plays an essential role in the deduction of $P(sx)$.

Despite the near-certainty that the reader is already familiar with proof by induction, we include an easy example here for the purposes of illustration and comparison with proof by infinite descent, which we survey in the next section. The following proof of the commutativity of addition over the natural numbers is a standard example whose mechanisation can be found in the libraries of most (if not all) inductive theorem proving tools:

Example 1.2.1. Define the operation $+$ on pairs of natural numbers by the following:

$$\begin{aligned} 0+y &= y && (1)\\ sx+y &= s(x+y) && (2) \end{aligned}$$

We claim that $+$ is commutative, i.e. $x+y=y+x$.

Proof. We first prove by induction on y that:

$$y = y+0 \ \ (3)$$

In the base case, we have to prove that $0=0+0$ which follows immediately from (1). In the step case, we have $y=sy'$ and require to prove $sy'=sy'+0$, i.e. $sy'=s(y'+0)$ by (2). Now by the induction hypothesis, we have $y'=y'+0$ and are thus done.

Next, we prove by induction on x that:

$$x+sy = s(x+y) \ \ (4)$$

In the base case, we have to prove that $0+sy=s(0+y)$ which follows by applying (1) to both sides of the equation. In the step case, we have $x=sx'$ and require to prove $sx'+sy=s(sx'+y)$, i.e. $s(x'+sy)=s(s(x'+y))$ by (2) applied to each side of the equation. Now by the induction hypothesis, we have $x'+sy=s(x'+y)$ and are thus done.

Now we can prove the main conjecture $x+y=y+x$ by induction on x. In the base case, we have to prove that $0+x=x+0$, which follows from (1) and (3). In the step case, we have $x=sx'$ and have to prove that $sx'+y=y+sx'$, i.e. $s(x'+y)=s(y+x')$ by (2) and (4). Now by the induction hypothesis, we have $x'+y=y+x'$ and are thus finished. □

Of course, mathematical induction is not limited to the natural numbers. Any inductively defined relation gives rise to a related induction proof principle which can be extracted from the definition in a wholly mechanical way. Similarly, from mutually inductively defined relations one obtains corresponding mutual induction proof principles, whose complexities increase in proportion with the complexity of the mutual definitions. For the purposes of carrying out proofs by mathematical induction, these induction principles are typically codified either as axioms or as inference rules and incorporated into the proof framework of choice. Because the use of an inductive proof principle in such proofs is always thus made explicit, we will sometimes refer to this approach as the *explicit induction* approach.

Most of the inductive theorem proving tools available at the present time employ the explicit induction approach; the major contemporary systems include ACL2 [37], INKA [3], the Isabelle-based proof planner IsaPlanner [21], Oyster-Clam [14] and to an extent also the logical framework-based system Twelf [61]. It is difficult to directly compare the relative success of these systems because of significant areas of non-overlap in the problems they address. However, despite the relative success of approaches based on heuristics such as Bundy's "rippling" technique [13, 22], it is unequivocally the case that the difficult challenges posed to mechanised theorem proving by inductive reasoning have not yet been overcome. These challenges largely manifest themselves in the choices made by a mathematician when attempting to prove a conjecture by mathematical induction:

- One needs to select an appropriate schema for the induction, which need not be the "usual" schema for the considered inductively defined relation derived from its definition. For example, the following alternative induction principle for $\mathbb{N}$ can be derived from the usual one:

$$\frac{P(0) \qquad \forall x \in \mathbb{N}.\ (P(x) \to P(ssx)) \qquad \forall x \in \mathbb{N}.\ (P(sx) \to P(x))}{P(t)}$$

  so that the fact that P is true of all even numbers follows from the first two premises, whence the fact that it is also true of all odd numbers follows from the third premise. In general, it may be the case that particular theorems may be proved (much) more easily by using a suitably chosen induction principle.

  Gow [31] demonstrates a technique for mechanising inductive proof search using proof planning whereby a proof attempt by induction starts by applying a "blank schema" containing metavariables which are incrementally instantiated during the remainder of the proof attempt. The final step in the proof then consists in showing that the induction rule thus generated is sound. (Of course, this is not necessarily the case.)

- One also needs to select an appropriate induction hypothesis or, in the case of mutually defined relations, hypotheses, and a variable (or variables) in the hypothesis over which

to induce. This is a particularly difficult issue for automated proof since there are many natural examples of theorems whose inductive proof requires the use of a "strengthened" induction hypothesis, i.e., a generalisation of the theorem (see e.g. [12]). There have been attempts within the theorem proving community to synthesize the necessary induction hypotheses for a proof either by *recursion analysis* — a syntactic analysis of the conjecture and the inductive definitions [7, 64, 81] — or by lazy generation schemes which attempt to delay the choice of induction hypotheses until an exploration of the proof makes it evident which hypotheses are required for the proof [31, 54]. While the techniques used in the former approach are sometimes quite ingenious, they are known to have significant limitations in the type and complexity of the problems that they can solve [54]. The latter approach shares some similarities with the method of infinite descent, described more fully in Section 1.3.

- Finally, one also needs to identify appropriate intermediate lemmas, as in, e.g., our proof of the commutativity of addition (Example 1.2.1 above). The question of whether such lemmas are strictly necessary essentially amounts to the question of whether cut is eliminable in the native proof system. However, even if intermediate lemmas are not technically required in order to give an inductive proof, in practice one may still need to identify and prove them because, as is well-known, cut-free proofs may be *much* larger in general than proofs containing cuts.

Despite some fairly widespread impressions to the contrary, cut-elimination is known to be possible in the presence of induction, at least for intuitionistic systems; Martin-Löf demonstrated cut-elimination for an intuitionistic natural deduction system with iterated inductive definitions [44], and the proof method used there has more recently been applied to an intuitionistic sequent calculus system with definitions and natural number induction by McDowell and Miller [48], and subsequently to further extensions of this system by Miller, Momigliano and Tiu [73, 49, 74]. Jervell [35] also gives a normalisation result for Heyting arithmetic, the intuitionistic version of Peano arithmetic (which includes natural number induction). The key point is that the induction rules in these systems are formulated in such a way as to allow the generalisation of induction hypotheses which, as we remarked earlier, is known to be necessary in general. For example, one way of writing a sequent calculus natural number induction rule so as to allow generalisation is as follows:

$$\frac{\Gamma \vdash F0, \Delta \qquad \Gamma, Fx \vdash Fsx, \Delta \qquad \Gamma, Ft \vdash \Delta}{\Gamma, Nt \vdash \Delta}$$

where we write *sequents* of the form $\Gamma \vdash \Delta$, where Γ and Δ are finite multisets of formulas, F is an arbitrary formula and the predicate N is used to stand for the property of being a natural number. (The rule above is a sequent calculus version of Martin-Löf's elimination rule for

the N predicate in his natural deduction system [44].) The induction rule above can thus be thought of as containing a cut in the sense that it introduces into its premises a formula (F) that need not be a subformula of any formula appearing in the conclusion, but this does not render cut-elimination meaningless in systems containing such rules. Rather, cut-free proofs in these systems enjoy the property that "new" formulas are only introduced as the result of the need to generalise an induction hypothesis when performing an induction. Martin-Löf summarises the situation as follows:

> The opinion seems to have been generally accepted that there be no real cut elimination theorem for first order arithmetic and that such a theorem could only be obtained by eliminating the induction schema in favour of the ω-rule. However, when arithmetic is formulated as a theory of inductive definitions, it becomes possible to formulate and prove a cut elimination theorem which is just as natural and basic as the one for pure first order logic, although, like in second order logic, the subformula principle is necessarily lost.
>
> *Per Martin-Löf, "Hauptsatz for the Intuitionistic Theory of Iterated Inductive Definitions", 1971*

One of the contributions of this thesis will be to show that the eliminability of cut in the presence of suitably formulated induction rules extends to the classical case. However, the proof techniques used to establish cut-elimination in the intuitionistic case are seemingly not sufficient, and we rely instead on semantic methods. In Chapter 3 we formulate a classical sequent calculus for general inductive definitions, in which the induction rules support generalisation as above, and demonstrate cut-eliminability for the calculus through semantic methods.

1.3 Infinite descent

Infinite descent, expressed as a self-contained proof principle, is considerably older than mathematical induction, having been precisely formulated by Fermat in 1659 in correspondence exchanged with Pierre de Carcavi. However, usage of the technique appears to date back as far as the ancient Greeks; Euclid's original proof of the irrationality of $\sqrt{2}$ essentially relies upon an infinite descent argument, and Wirth [85] reports that the first known usage of an infinite descent argument occurs in a proof of the irrationality of the number $\frac{1}{2}(1+\sqrt{5})$ due to Hippasos in the 5th century BC. Nevertheless, Fermat is generally credited with (re)inventing and articulating the method of infinite descent[1], which he described — in the context of its application to a proof of the fact that the area of a right-angled triangle cannot be a square number — as follows (c.f. [43]):

> Because the ordinary methods now in the books were insufficient for demonstrating such difficult propositions, I finally found a totally unique route for arriving at them... which I called infinite descent ...

[1] For this reason, infinite descent is still often known by its French name *descente infinie*.

> If there were any integral right triangle that had an area equal to a square, there would be another triangle less than that one which would have the same property ... Now it is the case that, given a number, there are not infinitely many numbers less than that one in descending order (I mean always to speak of integers). Whence one concludes that it is therefore impossible that there be any right triangle of which the area is a square...
>
> *Pierre de Fermat, "Relation des nouvelles découvertes en la science des nombres", letter to Pierre de Carcavi, 1659*

One way of expressing the principle of infinite descent for the natural numbers as an inference rule is the following:

$$\frac{P \to \exists (x_i \in \mathbb{N})_{i \geq 0}. \forall i \geq 0.\ x_{i+1} < x_i}{\neg P}$$

where P is any proposition and the existential quantifier ranges over infinite sequences of natural numbers. This rule interprets Fermat's infinite descent principle for the natural numbers as follows: if in some case of a proof we construct an infinite descending sequence of natural numbers, that case may be disregarded.

Example 1.3.1. $\sqrt{2}$ is not a rational number.

Proof. To say that $\sqrt{2}$ is not rational is to say that there do not exist $x, y \in \mathbb{N}$ such that $\sqrt{2} = x/y$. We shall construct an argument of this fact by infinite descent by assuming the contrary and using the denominator variable y to construct the required infinite decreasing sequence of natural numbers. Suppose that there exist $x, y \in \mathbb{N}$ such that $\sqrt{2} = x/y$, i.e. $x^2 = 2y^2$. From this we obtain $x(x-y) = y(2y-x)$, so that:

$$\frac{2y-x}{x-y} = \frac{x}{y} = \sqrt{2}.$$

Now define $x' = 2y - x$ and $y' = x - y$ whence we have $x'/y' = \sqrt{2}$ by the above. Now observe that we have $1 < x^2/y^2 < 4$, from which it follows that $y < x < 2y$, and so $0 < y' < y$. It is then readily seen that we have $x', y' \in \mathbb{N}$ such that $\sqrt{2} = x'/y'$, and $y' < y$. So given an arbitrary counterexample we can construct another counterexample with a strictly smaller denominator. This implies the existence of an infinitely decreasing chain of natural numbers, and hence $\sqrt{2}$ cannot be rational by the infinite descent principle for $\mathbb{N}$. □

In the example above, and many other natural examples, the infinite descent takes a special "cyclic" form: to prove that the statement P is true of all natural numbers one demonstrates that if P is not true of an arbitrary natural number n then it also is not true of a smaller number $m < n$, from which the existence of an infinite decreasing sequence of natural numbers can be inferred. As an inference rule this can be written:

$$\frac{\forall x \in \mathbb{N}.\ (\neg P(x) \to (\exists y < x \in \mathbb{N}. \neg P(y)))}{P(t)}$$

where, as in the induction principle, $P(x)$ is a statement of the language under consideration with a free variable x, and t is an arbitrary term of the language. This is classically equivalent to the principle of *complete induction* on $\mathbb{N}$:

$$\frac{\forall x \in \mathbb{N}.\ ((\forall y < x \in \mathbb{N}.\ P(y)) \rightarrow P(x))}{P(t)}$$

which is interderivable with the ordinary mathematical induction scheme for $\mathbb{N}$ (i.e. each is derivable from the other) in Peano arithmetic. Hence infinite descent for $\mathbb{N}$ subsumes complete induction for $\mathbb{N}$ and thus also ordinary induction for $\mathbb{N}$.

Just as in the case of mathematical induction, the infinite descent principle can be applied not just to natural numbers but to arbitrary (mutually) inductively defined relations, as follows: if in some case of a proof we unfold a particular inductive definition infinitely many times, then that case may be disregarded. This is because unfolding a definition infinitely often induces an infinite descending sequence of ordinals (via the ordinal-indexed approximants of inductively defined relations), which contradicts the fact that the ordinals are well-ordered.

At the present time, infinite descent — while it remains, for mathematicians, a standard technique for reasoning with inductively defined relations — is not generally employed in (mechanised) inductive theorem proving, presumably because it was considered impractical or less useful than explicit induction by the developers of the major theorem provers. The notable exception to this rule seems to be the QUODLIBET system of Wirth et al [4, 85], which does employ an approach based on infinite descent to inductive reasoning, and whose successful treatment of the technique indicates potential for its future development in theorem proving [86]. Additionally, although explicit induction is the default approach to inductive reasoning, various forms of cyclic (or circular), reasoning have been employed in: local model checking (see e.g. [65, 8]); theorem proving tools and frameworks including Hamilton's Poìtin [32] and, to a lesser extent, Schürmann's TWELF [61] as well as type-theoretic approaches [17, 27]; in Turchin's supercompilation [77]; and in program verification based on automata [79]. Typically, a proof containing repeating sections is presented as a graph containing loops and a "guardedness" condition is imposed on proofs to ensure their soundness, which is often of the form "the event x happens infinitely often along every infinite path in the graph". An example of such a condition is the so-called size change principle for program termination of Lee et al [42]. Such conditions can be seen essentially as encoding generalised infinite descent arguments. Recently, tableau-style proof systems for the μ-calculus employing cyclic proofs were developed first by Dam and Gurov [19] and then investigated further by Schöpp and Simpson [58] and by Sprenger and Dam [62, 63]. These systems embody an infinite descent principle for the μ-calculus, based upon an ordinal indexing of the approximants by which the least and greatest fixed points of a formula can be iteratively approached. They also are closely related to the μ-calculus "refutations" of Niwinski and Walukiewicz [52], which

are infinite tableaux for the μ-calculus satisfying a similar guardedness condition. (Niwinski and Walukiewicz showed that for any refutation there is an equivalent refutation that is regular and thus representable as a cyclic graph, so their refutations are really another manifestation of cyclic reasoning in the μ-calculus.)

It is sometimes claimed that infinite descent and explicit induction are equivalent as proof principles, i.e., that any mathematical proof by induction could equally well be expressed as a proof by infinite descent and vice versa. Unfortunately, it is far from clear what exactly is meant by equivalence in this situation and, in fact, one of the contributions of this thesis will be to give formal precision to this claim by framing it as a conjecture of the equivalence of the two formal proof systems.

1.4 Our approach

The aim of this thesis is to undertake a rigorous proof-theoretic investigation of the proof principles of mathematical induction and infinite descent, not just for the natural numbers but for general inductively defined relations. We think that such an analysis is of independent interest, and of special interest to the inductive theorem proving community since it has implications for the provision of proof support for reasoning with inductively defined relations.

In order to undertake a formal analysis of induction and infinite descent in proofs, our first task is to formulate a language in which inductively defined relations can be expressed. The obvious choice seems to be to extend the language of standard first-order logic with an inductive definition schema by which to define (some chosen subset of) the predicates of the language. Of the many possible frameworks for the latter, we choose to work with ordinary (mutual) inductive definitions as formulated by Martin-Löf [44]. This choice keeps the logic relatively simple, thus allowing us to bring out the main interesting proof-theoretic issues that arise with inductive definitions, while including many important examples. (However, we claim that our techniques should be quite straightforwardly extensible to richer systems of inductive and coinductive definitions.)

Having fixed a language in which to study inductive definitions, our next objective is to formulate formal proof systems that capture the notions of induction and infinite descent. We opt to formulate our systems in the *sequent calculus* style due to Gentzen [25], as it provides an elegant system that is widely acknowledged as both amenable to proof-theoretical analysis and well-suited to a natural expression of the goal-directed proof construction employed in most proof assistants. In Gentzen's original sequent calculus LK for classical first-order logic, each logical connective is specified by two basic *introduction rules* introducing the connective on the left and on the right of sequents respectively. His well-known cut-elimination theorem then states that any LK-derivable sequent is provable without detours, that is, using these rules only

and without the use of intermediate lemmas. This constrains the locally applicable rules in any proof search attempt in LK to those rules that introduce a logical connective appearing in the current goal sequent. (However, this does not imply that provability in first-order logic is decidable, because in general there are infinitely many choices for the instantiation of a formula involving a quantifier.)

In this thesis, we extend Gentzen's LK to obtain similarly canonical sequent calculus proof systems for induction and for infinite descent in our first-order logic with inductive definitions. This allows us to undertake a formal analysis of these systems, and thus of the proof principles of induction and infinite descent themselves, in a setting with minimal additional complication (in the shape of other axioms or proof principles). In formulating the proof system formalising proof by infinite descent, a major issue is the question of how to build in the capacity for constructing infinite descending sequences of unfoldings of inductively defined predicates. We address this by allowing proofs to be non-well-founded, i.e. to contain infinite branches, and imposing an additional soundness condition on proofs ensuring that some inductive definition is unfolded infinitely often along each infinite branch. Thus the infinite branches in a non-well-founded proof represent the cases that can be disregarded by the infinite descent principle for inductively defined relations.

Example 1.4.1. Let the sets E and O of even and odd numbers be given by the following inductive definition:

- 0 is an even number;
- if n is an even number, then $s(n)$ is an odd number;
- if n is an odd number, then $s(n)$ is an even number.

We give an infinite descent proof that every natural number is then either even or odd. (Of course, this proposition can also easily be proved using induction. We include this proof for illustrative purposes, and defer harder examples until later in the thesis.)

Proof. Informally, the justification of the result is as follows. Let n be a natural number, and consider the two possible cases: either n is 0 or it is the successor of some natural number m. If $n = 0$ then we are immediately finished as 0 is an even number, so let us consider the case where $n = s(m)$ for some natural number m. We can then repeat the argument by considering cases on m: if $m = 0$ then we are done as $n = s(0)$ is an odd number, so we need only consider the case where $m = s(m')$ for some natural number m'. By repeating this argument infinitely often, we are left only with the case in which we have an infinite descending sequence $n > m > m' > m'' > \ldots$ of natural numbers. But the natural numbers are well-ordered, i.e. there are no infinite descending sequences of natural numbers, and so by contradiction this case may be disregarded. Thus every natural number must indeed be either even or odd.

The corresponding formal infinite proof can be given as follows, writing Nx, Ex and Ox to mean respectively that x is a natural, even or odd number, and using $\vdash$ for the provability relation:

$$\dfrac{\dfrac{}{n=0 \vdash En, On}\,(=\mathrm{L})}{Nn \vdash En, On}\,(\text{Case } N) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash O0, E0}\,(E\mathrm{R}_1)}{m=0 \vdash Om, Em}\,(=\mathrm{L}) \qquad \dfrac{\vdots\ (\text{etc.})}{Nm' \vdash Os(m'), Es(m')}\,(=\mathrm{L})}{m=s(m'), Nm' \vdash Om, Em}\,(\text{Case } N)}{Nm \vdash Om, Em}\,(O\mathrm{R}_1)}{Nm \vdash Om, Os(m)}\,(E\mathrm{R}_2)}{n=s(m), Nm \vdash En, On}\,(=\mathrm{L})$$

The rule label (Case N) is used to indicate a division into cases based on the definition of the predicate N, the label (=L) denotes rewriting according to an equality in the conclusion and the rule label ($P\mathrm{R}_i$) indicates a right-unfolding of the definition of predicate P (the i is an index indicating which clause of the definition was used). Informally speaking, the infinite proof tree above is a proof in our infinitary system because the inductive predicate N is "unfolded infinitely often" (on the left) along the only infinite branch in the tree. □

The infinitary proof system for infinite descent is rather powerful, and there are infinite proofs with no useful finite representation. For practical purposes, it makes sense to consider a restriction of the system in which proofs do have a convenient representation. In our case, we consider the restriction of the full infinitary systen to those proofs given by *regular* trees, i.e. trees having only finitely many distinct subtrees. It is well known that such trees are representable by finite graphs, and thus the restricted system is suitable for practical formal reasoning.

Example 1.4.2. The argument given above in Example 1.4.1 that every natural number is either even or odd can be written as a "cyclic proof":

$$\dfrac{\dfrac{}{n=0 \vdash En, On}\,(=\mathrm{L}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash E0, O0}\,(E\mathrm{R}_1)}{Nn \vdash On, En\ \ (*)}\,(\text{Subst})}{Nm \vdash Om, Em}\,(O\mathrm{R}_1)}{Nm \vdash Om, Os(m)}\,(E\mathrm{R}_2)}{n=s(m), Nm \vdash En, On}\,(=\mathrm{L})}{Nn \vdash En, On\ \ (*)}\,(\text{Case } N)$$

where (Subst) denotes a use of a rule for substitution and $(*)$ indicates the "loop" in the graph. Except for the uses of substitution, the tree obtained by unfolding the loop in this graph is the infinite proof tree given in the previous example, and the justification for soundness is similar.

1.5 Synopsis

The remainder of this thesis is structured as follows:

Chapter 2: We define the syntax and semantics of the language $\mathrm{FOL_{ID}}$ obtained by extending standard first-order logic with a schema for (mutual) inductive definitions. As well as the standard semantics for the latter obtained by considering the least fixed point of a monotone operator constructed from the definitions, we also define a more general, non-standard semantics based on considering Henkin models for the inductive definitions.

Chapter 3: We define a sequent calculus, LKID, for proof by induction in $\mathrm{FOL_{ID}}$, which extends Gentzen's LK by adding new left- and right-introduction rules for the inductively defined predicates. The right-introduction rules for such a predicate P are just sequent versions of the productions defining P. The left-introduction rule for P embodies the natural principle of "rule induction" over the definition of P. This division between the roles of the left and right rules is closely related to Martin-Löf's natural deduction system for intuitionistic logic with (iterated) inductive definitions [44], in which induction rules were included as elimination rules for inductively defined predicates. (This observation subsequently became a cornerstone of the treatment of inductive types in Martin-Löf's type theory [45].) As is well known, elimination rules in natural deduction serve the same purpose as left-introduction rules in sequent calculus.

We establish that LKID is sound with respect to the non-standard Henkin semantics of $\mathrm{FOL_{ID}}$ (and thus, in particular, the standard semantics), and then show that the cut-free fragment of LKID is complete with respect to the Henkin semantics. The latter result is of interest in its own right, and we also obtain a semantic proof of the eliminability of cut in LKID by combining the soundness and cut-free completeness results. The consistency of Peano arithmetic is subsequently derived as a corollary of cut-eliminability in LKID. Readers familiar with [35, 44, 48, 73] will not be surprised that cut is eliminable in our system, since these papers contain analogous normalization/cut-elimination theorems for related intuitionistic systems. Their proofs, however, are based on Tait's "computability" method, and do not adapt to the classical setting, hence our reliance on semantic methods. As far as we are aware, our proof is the first demonstration of the eliminability of cut in a classical proof system for induction to appear in the literature.

Chapter 4: We define a second proof system, LKID^{ω}, for proof by infinite descent in $\mathrm{FOL_{ID}}$. In this system, the left-introduction rule for an inductively defined predicate is not an induction rule but rather a weaker "casesplit" rule. (The other rules of the system are the same as the rules for LKID.) However, we allow proofs in LKID^{ω} to be infinite (i.e. non-well-founded) so long as they satisfy the condition that some inductive predicate

is unfolded infinitely often along every infinite branch in the proof. By an infinite descent argument using the well-foundedness of the inductive definitions, any such branch can be disregarded. Thus the remaining portion of the proof is well-founded and hence sound. (However, the formal justification for the soundness of LKID^ω is somewhat more complicated than this.)

After defining the system and formulating the "infinite descent" soundness condition on proofs, we prove that LKID^ω is sound and cut-free complete, this time with respect to the standard semantics of $\mathrm{FOL}_{\mathrm{ID}}$. It follows that the infinitary proof system LKID^ω is strictly more powerful than the finitary proof system LKID. Both the soundness and completeness proofs have elements in common with their LKID counterparts, but the necessity of accounting for the global soundness condition in LKID^ω proofs entails some interesting twists in both proofs. A semantic proof of the eliminability of cut in LKID^ω is then obtained from the soundness and cut-free completeness theorems, just as for LKID in the preceding chapter.

We also show how the soundness condition on LKID^ω proofs, based on *traces* on paths in the proof, can be extended to other settings to yield a sound notion of infinite proof, in circumstances that permit the formulation of a similar infinite descent argument.

Chapter 5: As infinitary proof systems are generally unsuitable for practical formal reasoning, we consider the restriction of such systems to those proofs given by *regular trees*, i.e. trees representable by finite cyclic graphs. Infinitary systems restricted in this way *are* suitable for formal reasoning, because the global soundness condition on proofs is decidable in this restricted setting. In particular, we consider the cyclic proof system CLKID^ω obtained as the restriction of the system LKID^ω to regular trees. As CLKID^ω arises as a simple restriction of a complete infinitary system, it is a highly natural system in its own right, although we conjecture that the eliminability of cut no longer holds.

Chapter 6: We develop machinery for analysing the structure of proofs in cyclic proof systems of the type considered in chapter 5, and obtain a natural notion of equivalence between such proofs: two cyclic proofs are considered equivalent if they represent the same infinite tree (up to isomorphism). We then show the *cycle-normalisation* property for cyclic proofs: for every cyclic proof there is an equivalent proof with a restricted cycle structure, said to be in *cycle normal form*. We first give a proof based on unfolding the proof into an infinite tree, and then folding the infinite branches on this tree to obtain an equivalent proof in cycle normal form. We also give an alternative proof of cycle-normalisation which shows algorithmically how to iteratively "untangle" a proof into an equivalent one in cycle normal form, and also gives an improved complexity bound of the proof thereby obtained.

Chapter 7: We analyse the trace condition ensuring soundness of proofs in infinitary proof systems of the type considered in chapter 4 and, in particular, their cyclic restrictions as formulated in chapter 5. We formulate an alternative soundness condition for cyclic proofs in cycle normal form, which is more restrictive than the general trace condition but appears simpler from a combinatorial perspective, and provides more explicit information about the proof. Two alternative formulations of this *trace manifold* condition are considered, and demonstrated to be equivalent.

Finally, we return to the specific setting of our proof systems for inductive definitions, and show that an arbitrary LKID proof can be transformed into an CLKID$^\omega$ proof satisfying the trace manifold condition, thus demonstrating that CLKID$^\omega$ is at least as powerful as LKID. We conjecture that LKID and CLKID$^\omega$ are in fact equivalent, i.e., that proof by induction is equivalent to regular proof by infinite descent. However, the problem of establishing whether CLKID$^\omega$ proofs are subsumed by LKID proofs appears a difficult one, and we leave this as the main open question arising from the work in this thesis.

Chapter 8: We present our conclusions and outline what we consider to be the main directions for future work arising from the thesis.

The formulation of the systems LKID and CLKID$^\omega$, together with the material appearing in chapters 6 and 7, has previously been published in a TABLEAUX paper [9] by the author. The material in chapters 3-5 forms the basis of a second article by the author and Alex Simpson, currently in preparation [10].

Chapter 2

First-order logic with inductive definitions ($\mathbf{FOL}_{ID}$)

In this chapter we introduce the language $\mathrm{FOL}_{\mathrm{ID}}$ of (classical) first-order logic with inductive definitions. The syntax of $\mathrm{FOL}_{\mathrm{ID}}$ is essentially that of ordinary first-order logic with equality (see e.g. [6]), with the constants, functions and predicates of a language interpreted in a first-order structure and the variables interpreted in an environment mapping variables to elements of the domain of the structure. However, we designate a finite number of predicate symbols in the language as *inductive*, and interest ourselves only in those structures in which the inductive predicates have a special interpretation, determined by a given set of inductive definitions. The essentials of first-order logic with equality are reviewed in Section 2.1. In Section 2.2 we introduce our inductive definition schema, based on Martin-Löf's "ordinary productions" [44], and define *standard models* of $\mathrm{FOL}_{\mathrm{ID}}$. In standard models, the inductive predicates of a language are interpreted as (components of) the least prefixed point of a monotone operator constructed from their definitions (see e.g. [1]). This least prefixed point can be constructed in iterative stages called *approximants*. The approximant approach to least prefixed point construction will be essential to our consideration of infinitary proof systems for $\mathrm{FOL}_{\mathrm{ID}}$ in subsequent chapters.

However, it happens that the inductive predicates of a language also have a class of natural interpretations which generalises the standard interpretation. In these non-standard interpretations, the least prefixed point of the monotone operator for the inductive predicates is obtained inside a class of subsets of tuples over the domain of interpretation (as opposed to the powerset of tuples over the domain). This approach is based on an idea originally employed by Henkin [33] who obtained a completeness theorem for the second-order functional calculus by considering validity with respect to his more general notion of model, in which second order variables of arity n are interpreted as ranging over a selected class of subsets of n-tuples in the domain. Henkin extended this approach to obtain a completeness result for a simple type theory. We introduce a corresponding notion of a *Henkin model* for $\mathrm{FOL}_{\mathrm{ID}}$ in Section 2.3 (and go

on to obtain an analogous completeness theorem in Chapter 3). The material on Henkin models in this section is somewhat novel (though obviously influenced by Henkin's own work); the material in the earlier sections is completely standard.

Throughout this chapter and the remainder of the thesis, we use the following notational conventions and mathematical definitions:

- the powerset of a set X is denoted by $\mathrm{Pow}(X)$;
- vectors are set in bold type, e.g. $\mathbf{x}$. For convenience, we will often use vector notation to denote finite sequences, e.g. $\mathbf{x}$ for $(x_1,\ldots,x_n)$;
- for any set X, $n>0$ and $i\in\{1,\ldots,n\}$, we define the projection function $\pi^n_i : X^n \to X$ by $\pi^n_i(x_1,\ldots,x_n) = x_i$;
- we extend the usual set inclusion, union and intersection to tuples of sets $(X_1,\ldots,X_n)$ by:

$$\begin{aligned}(X_1,\ldots,X_n)\subseteq(Y_1,\ldots,Y_n) &= X_1\subseteq Y_1\wedge\ldots\wedge X_n\subseteq Y_n\\ (X_1,\ldots,X_n)\cup(Y_1,\ldots,Y_n) &= (X_1\cup Y_1,\ldots,X_n\cup Y_n)\\ (X_1,\ldots,X_n)\cap(Y_1,\ldots,Y_n) &= (X_1\cap Y_1,\ldots,X_n\cap Y_n)\end{aligned}$$

- a *graph* G is a pair (V,E) where V is the set of *vertices* of G and $E\subseteq V\times V$ is a set of *edges* of G.

2.1 First-order logic with equality

In this section we briefly review the syntax and semantics of classical first-order logic with equality. The only difference from the standard presentations is that we designate finitely many predicate symbols of our languages as special *inductive* symbols, in order to distinguish those predicates whose intended interpretation we wish to be given by an inductive definition. (The number of inductive predicate symbols is restricted to be finite for reasons of technical convenience, although this restriction is not strictly necessary.) Our use of two-sorted predicate symbols allows us to extend the language of (first-order) formulas without altering the underlying language of terms, which is the standard approach used in other extensions of first-order logic (such as second-order logic).

The remainder of this section can be safely skipped by the reader familiar with first-order logic; we review the fundamentals here for completeness and for the purpose of fixing notation.

Definition 2.1.1 (First-order language with inductive predicates)**.** A *(first-order) language with inductive predicates* Σ is a set of symbols including:

- denumerably many constant symbols $c_1, c_2, \ldots$;

- denumerably many function symbols $f_1, f_2, \ldots$, each with associated arity $k > 0$;
- denumerably many *ordinary* predicate symbols $Q_1, Q_2, \ldots$, each with associated arity $k \geq 0$;
- finitely many *inductive* predicate symbols $P_1, \ldots, P_n$, each with associated arity $k \geq 0$.

We also assume the existence of a denumerably infinite set $\mathcal{V}$ of variables $x_1, x_2, \ldots$, each of which is distinct from any symbol of Σ.

Throughout this thesis, when we refer to a "first-order language", we shall mean a first-order language with inductive predicates, in the sense of Definition 2.1.1 above.

Definition 2.1.2 (Terms)**.** The set of *terms* of a first-order language Σ, $Terms(\Sigma)$, is the smallest set of expressions of Σ closed under the following rules:

1. any variable $x \in \mathcal{V}$ is a term;
2. any constant symbol $c \in \Sigma$ is a term;
3. if $f \in \Sigma$ is a function symbol of arity k, and $t_1, \ldots, t_k$ are terms, then $f(t_1, \ldots, t_k)$ is a term.

We write $t[u/x]$, where $x \in \mathcal{V}$ is a variable and t, u are terms, to denote the term obtained by substituting u for all occurrences of x in t. We write $Var(t)$ for the set of variables appearing in the term t, and write $t(x_1, \ldots, x_n)$ for a term t such that $Var(t) \subseteq \{x_1, \ldots, x_n\}$, where $x_1, \ldots, x_n$ are distinct. In this case we may write $t(t_1, \ldots, t_n)$ to denote the term obtained by substituting $t_1, \ldots, t_n$ for $x_1, \ldots, x_n$ respectively.

The interpretations of the symbols in a first-order language are given by a first-order structure in the standard way:

Definition 2.1.3 (First-order structure)**.** Given a first-order language Σ, a *(first-order) structure for* Σ (also called a Σ-structure) is a tuple:

$$M = (D, c_1^M, c_2^M, \ldots, f_1^M, f_2^M, \ldots, Q_1^M, Q_2^M, \ldots, P_1^M, \ldots, P_n^M)$$

where D is any set of objects (called the *domain* of the structure) and:

- each $c_i^M \in D$;
- each $f_i^M : D^k \to D$, where f_i is a function symbol of arity k;
- each $Q_i^M \subseteq D^k$, where Q_i is an ordinary predicate symbol of arity k;
- each $P_i^M \subseteq D^k$, where P_i is an inductive predicate symbol of arity k.

Note that we consistently write $Q^M\mathbf{d}$, where Q is a predicate symbol (either ordinary or inductive), to mean $\mathbf{d} \in Q^M$. Also, if $t(x_1,\ldots,x_k)$ is a term of Σ, then $t^M(x_1,\ldots,x_k) : D^k \to D$ is obtained by replacing every constant symbol c by c^M and every function symbol f by f^M in $t(x_1,\ldots,x_n)$.

We interpret variables as elements of the domain of a structure using environments:

Definition 2.1.4 (Environment)**.** Given a Σ-structure $M = (D,\ldots)$, an *environment* for M is a function $\rho : \mathcal{V} \to D$. Where ρ is an environment for M, $x \in \mathcal{V}$ and $d \in D$, we write $\rho[x \mapsto d]$ for the "substituted" environment defined by:

$$\rho[x \mapsto d](y) = \begin{cases} d & \text{if } y = x \\ \rho(y) & \text{otherwise} \end{cases}$$

We extend the domain of any environment ρ to all terms of Σ by:

- $\rho(c_i) = c_i^M$
- $\rho(f_i(t_1,\ldots,t_k)) = f_i^M(\rho(t_1),\ldots,\rho(t_k))$

(We also extend ρ to vectors of terms in the obvious way: $\rho(t_1,\ldots,t_k) = (\rho(t_1),\ldots,\rho(t_k))$.)

Lemma 2.1.5 (Environment substitution sanity)**.** *Let ρ be an environment for $M = (D,\ldots)$. Then for any term t and for any variable $x \in \mathcal{V}$:*

1. *for all $d \in D$, if $x \notin Var(t)$ then $\rho[x \mapsto d](t) = \rho(t)$;*
2. *for all terms u, $\rho[x \mapsto \rho(u)](t) = \rho(t[u/x])$.*

Proof. Both parts of the lemma follow by straightforward structural inductions on t.

□

The formulas of FOL_{ID} are just the formulas of first-order logic with equality:

Definition 2.1.6 (Formulas)**.** Given a first-order language Σ, the set of Σ-formulas of FOL_{ID} is the smallest set of expressions closed under the following rules:

1. if $t_1,\ldots,t_k$ are terms of Σ, and Q is a predicate symbol in Σ of arity k, then $Q(t_1,\ldots,t_k)$ is a formula;
2. if t and u are terms of Σ then $t = u$ is a formula;
3. if F is a formula then so is $\neg F$;
4. if F_1 and F_2 are formulas then so are $F_1 \wedge F_2$, $F_1 \vee F_2$ and $F_1 \to F_2$;
5. if F is a formula and $x \in \mathcal{V}$ is a variable, then $\exists x F$ and $\forall x F$ are formulas.

We use the standard precedences on the logical connectives, and use parentheses to disambiguate where necessary. Any formula of the form $Q(t_1,\ldots,t_k)$ or $t = u$ is called an *atomic* formula, and formulas not of this form are called *non-atomic* or *compound* formulas. We write $F \leftrightarrow G$, where F and G are formulas, to abbreviate the formula $(F \rightarrow G) \wedge (G \rightarrow F)$.

Definition 2.1.7 (Free variables)**.** The set of *free variables* occurring in a formula F of $\mathrm{FOL_{ID}}$, $FV(F)$, is defined by recursion on the structure of F as follows:

1. $FV(Q(t_1,\ldots,t_k)) = \bigcup_{1\leq i\leq k} Var(t_i)$
2. $FV(t = u) = Var(t) \cup Var(u)$
3. $FV(\neg F) = FV(F)$
4. $FV(F_1 \wedge F_2) = FV(F_1 \vee F_2) = FV(F_1 \rightarrow F_2) = FV(F_1) \cup FV(F_2)$
5. $FV(\exists x F) = FV(\forall x F) = FV(F) \setminus \{x\}$

(Informally, a variable in $\mathcal{V}$ is in $FV(F)$ if it has an occurrence in F that is outside the scope of every quantifier in F with the same name.)

Definition 2.1.8 (Satisfaction relation for $\mathrm{FOL_{ID}}$)**.** Let M be a Σ-structure and let ρ be an environment for M. We define the satisfaction relation $M \models_\rho F$ on formulas by:

$$\begin{array}{lcl}
M \models_\rho Q\mathbf{t} & \Leftrightarrow & Q^M(\rho(\mathbf{t})) \quad (Q \text{ ordinary or inductive}) \\
M \models_\rho t = u & \Leftrightarrow & \rho(t) = \rho(u) \\
M \models_\rho \neg F & \Leftrightarrow & M \not\models_\rho F \\
M \models_\rho F \wedge G & \Leftrightarrow & M \models_\rho F \text{ and } M \models_\rho G \\
M \models_\rho F \vee G & \Leftrightarrow & M \models_\rho F \text{ or } M \models_\rho G \\
M \models_\rho F \rightarrow G & \Leftrightarrow & M \not\models_\rho F \text{ or } M \models_\rho G \\
M \models_\rho \forall x F & \Leftrightarrow & M \models_{\rho[x\mapsto d]} F \text{ for all } d \in D \\
M \models_\rho \exists x F & \Leftrightarrow & M \models_{\rho[x\mapsto d]} F \text{ for some } d \in D
\end{array}$$

(Informally, $M \models_\rho F$ means: "the formula F is true in M under the environment ρ".)

Lemma 2.1.9 (Formula substitution sanity)**.** *For any $x \in \mathcal{V}$, for any term t of Σ, and for any formula F, the following hold:*

1. *for all $d \in D$, if $x \notin FV(F)$ then $M \models_\rho F$ if and only if $M \models_{\rho[x\mapsto d]} F$;*
2. *$M \models_\rho F[t/x]$ if and only if $M \models_{\rho[x\mapsto\rho(t)]} F$.*

Proof. Both results follow by a straightforward structural induction on F, using the appropriate part of Lemma 2.1.5 for the cases where F is an atomic formula or an equality formula. □

We introduce our inductive definition schema and the standard interpretation of inductive predicates in the following section, and give a natural class of non-standard interpretations in Section 2.3.

2.2 Inductive definitions and standard semantics for $\mathbf{FOL}_{ID}$

As mentioned previously, we shall only be interested in those structures in which the inductive predicates of the language are given a specific interpretation obtained from their definitions. In this section we introduce our schema for inductive definitions, which is based upon Martin-Löf's "ordinary production" schema [44], and define the corresponding notion of a *standard model* in $\mathrm{FOL_{ID}}$. For this section, and indeed the remainder of the chapter, we consider a fixed first-order language Σ with (exactly) n inductive predicates $P_1, \ldots, P_n$.

Definition 2.2.1 (Inductive definition set)**.** An *inductive definition set* Φ for a language Σ is a finite set of *productions*, which are rules of the form:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1, \ldots, j_m, i \in \{1, \ldots, n\}$$

i.e., $Q_1, \ldots, Q_h$ are ordinary predicates and $P_{j_1}, \ldots, P_{j_m}, P_i$ are inductive predicates of Σ.

It is possible to generalise the schema for "ordinary inductive definitions" in Definition 2.2.1 to more expressive schemas by allowing non-atomic formulas to appear in the premises of a production. However, appropriate restrictions must then be placed upon the productions to ensure that the inductive definitions are well-founded. For example, Martin-Löf obtains "iterated inductive definitions" [44] by associating to each inductive predicate symbol P a *level*, $l(P) \in \mathbb{N}$, setting the level of a non-atomic formula to be the maximum of the levels of the inductive predicates occurring within it, and then allowing productions of the form:

$$\frac{F(\mathbf{x}) \rightarrow P_j\mathbf{u}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad \begin{array}{l} j, i \in \{1, \ldots, n\}, F \text{ an arbitrary formula}, \\ l(P_i) \geq l(P_j), l(P_i) > l(F) \end{array}$$

$$\frac{\forall\mathbf{y}(P_j\mathbf{u}(\mathbf{x}, \mathbf{y}))}{P_i\mathbf{t}(\mathbf{x})} \qquad j, i \in \{1, \ldots, n\}, l(P_i) \geq l(P_j)$$

and ordinary productions of the form of the production in Definition 2.2.1 above also receive the restriction that $l(P_i) \geq max(l(P_{j_1}), \ldots, l(P_{j_m}))$.

In this thesis, however, we shall not consider such schemas, and instead confine our attention to the ordinary (mutual) inductive definitions given by the schema in Definition 2.2.1. This schema is already powerful enough to admit the definition of many familiar inductively defined structures (we give some examples below). Furthermore, when we consider the incorporation of inductive definitions into formal proof systems, the main interesting issues arise even in the case of ordinary inductive definitions, and the relative simplicity of the induction schema makes the analysis of such issues somewhat more manageable.

We now define the standard interpretation of the inductively defined predicates in a first-order structure M for Σ. Following [1], we take the usual approach of constructing a monotone operator from a given inductive definition set Φ:

Definition 2.2.2 (Monotone operator)**.** Let A be a set, let $f : A \to A$ be a function and let $\leq$ be a partial ordering on A. Then f is said to be a *monotone operator* if $x \leq y$ implies $f(x) \leq f(y)$. x is said to be a *prefixed point* of a monotone operator f if $f(x) \leq x$.

Definition 2.2.3 (Definition set operator)**.** Let M be a first-order structure for Σ, let Φ be an inductive definition set for Σ, and for $i \in \{1,\ldots,n\}$ (where n is the number of inductive predicate symbols of Σ), let k_i be the arity of the inductive predicate symbol P_i. Then partition Φ into disjoint subsets $\Phi_1,\ldots,\Phi_n \subseteq \Phi$ by:

$$\Phi_i = \{\frac{u}{v} \in \Phi \,|\, P_i \text{ appears in } v\}$$

Now let each definition set Φ_i be indexed by r with $1 \leq r \leq |\Phi_i|$, and let $\Phi_{i,r} \in \Phi$ be an arbitrary production in Φ_i, say:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1,\ldots,j_m,i \in \{1,\ldots,n\}$$

Now we define a corresponding function $\varphi_{i,r} : (\mathrm{Pow}(D^{k_1}) \times \ldots \times \mathrm{Pow}(D^{k_n})) \to \mathrm{Pow}(D^{k_i})$ by:

$$\varphi_{i,r}(X_1,\ldots,X_n) = \{\mathbf{t}^M(\mathbf{x}) \mid Q_1^M\mathbf{u_1}^M(\mathbf{x}),\ldots,Q_h^M\mathbf{u_h}^M(\mathbf{x}),\mathbf{t_1}^M(\mathbf{x}) \in X_{j_1},\ldots,\mathbf{t_m}^M(\mathbf{x}) \in X_{j_m}\}$$

(Note that any variables occurring in the right hand side but not the left hand side of the set expression above are, implicitly, existentially quantified over the entire right hand side of the expression.) Then the *definition set operator for* Φ is the operator φ_Φ, with domain and codomain $\mathrm{Pow}(D^{k_1}) \times \ldots \times \mathrm{Pow}(D^{k_n})$, defined by:

$$\varphi_\Phi(X_1,\ldots,X_n) \;=\; (\varphi_1(X_1,\ldots,X_n),\ldots,\varphi_n(X_1,\ldots,X_n))$$

$$\text{where } \varphi_i(X_1,\ldots,X_n) \;=\; \textstyle\bigcup_r \varphi_{i,r}(X_1,\ldots,X_n) \text{ for each } i \in \{1,\ldots,n\}$$

Proposition 2.2.4. *For any inductive definition set Φ, the operator φ_Φ is monotone (with respect to the subset ordering $\subseteq$ on its domain).*

Proof. To show that φ_Φ is monotone, one needs to prove that $(X_1,\ldots,X_n) \subseteq (Y_1,\ldots,Y_n)$ implies $\varphi_\Phi(X_1,\ldots,X_n) \subseteq \varphi_\Phi(Y_1,\ldots,Y_n)$, i.e. that $X_k \subseteq Y_k$ for all $k \in \{1,\ldots,n\}$ implies $\varphi_i(X_1,\ldots,X_n) \subseteq \varphi_i(Y_1,\ldots,Y_n)$ for all $i \in \{1,\ldots,n\}$.

Suppose $y \in \varphi_i(X_1,\ldots,X_n) = \bigcup_r \varphi_{i,r}(X_1,\ldots,X_n)$ for some $i \in \{1,\ldots,n\}$. So there is a production $\Phi_{i,r} \in \Phi$ such that $y \in \varphi_{i,r}(X_1,\ldots,X_n)$, say:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1,\ldots,j_m,i \in \{1,\ldots,n\}$$

So we have:

$$y \in \{\mathbf{t}^M(\mathbf{x}) \mid Q_1^M\mathbf{u_1}^M(\mathbf{x}),\ldots,Q_h^M\mathbf{u_h}^M(\mathbf{x}),\mathbf{t_1}^M(\mathbf{x}) \in X_{j_1},\ldots,\mathbf{t_m}^M(\mathbf{x}) \in X_{j_m}\}$$

and since $X_i \subseteq Y_i$ for all $i \in \{1,\ldots,n\}$ by assumption, we have:

$$y \in \{\mathbf{t}^M(\mathbf{x}) \mid Q_1^M\mathbf{u}_1^M(\mathbf{x}),\ldots,Q_h^M\mathbf{u}_\mathbf{h}^M(\mathbf{x}),\mathbf{t}_1^M(\mathbf{x}) \in Y_{j_1},\ldots,\mathbf{t}_\mathbf{m}^M(\mathbf{x}) \in Y_{j_m}\}$$

i.e. $y \in \varphi_{i,r}(Y_1,\ldots,Y_n)$ and thus $y \in \varphi_i(Y_1,\ldots,Y_n)$. This shows that $\varphi_i(X_1,\ldots,X_n) \subseteq \varphi_i(Y_1,\ldots,Y_n)$ and thus $\varphi_\Phi(X_1,\ldots,X_n) \subseteq \varphi_\Phi(Y_1,\ldots,Y_n)$ as required. □

Example 2.2.5. Let N be an inductive predicate symbol of arity 1 and let Φ_N be the definition set consisting of the productions:

$$\frac{}{N0} \qquad \frac{Nx}{Nsx}$$

The definition set operator for Φ_N is then:

$$\varphi_{\Phi_N}(X) = \{0^M\} \cup \{s^M x \mid x \in X\}$$

Example 2.2.6. Let E and O be inductive predicate symbols of arity 1 and let Φ_{EO} be the definition set consisting of the productions:

$$\frac{}{E0} \qquad \frac{Ex}{Osx} \qquad \frac{Ox}{Esx}$$

The definition set operator for Φ_{EO} is then:

$$\varphi_{\Phi_{EO}}(X,Y) = (\{0^M\} \cup \{s^M y \mid y \in Y\}, \{s^M x \mid x \in X\})$$

Example 2.2.7. Let R be an ordinary predicate symbol of arity 2, let R^+ be an inductive predicate symbol also of arity 2, and let Φ_{R^+} be the definition set consisting of the productions:

$$\frac{R(x,y)}{R^+(x,y)} \qquad \frac{R^+(x,y) \quad R^+(y,z)}{R^+(x,z)}$$

The definition set operator for Φ_{R^+} is then:

$$\varphi_{\Phi_{R^+}}(X) = \{(x,y) \mid R^M(x,y)\} \cup \{(x,z) \mid \exists y.(x,y) \in X, (y,z) \in X\}$$

It is a standard result for inductive definitions that the least n-tuple of sets closed under the productions in Φ is the least prefixed point of the operator φ_Φ (see e.g. [1, 50]), and that this least prefixed point can be approached in iterative *approximant* stages. Since such approximants are essential to understanding the proof systems we present in Chapters 4 and 5, we include here full details of their construction and fundamental properties.

Definition 2.2.8 (Approximants)**.** Let $M = (D,\ldots)$ be a first-order structure for Σ, let Φ be an inductive definition set and for each $i \in \{1,\ldots,n\}$, let k_i be the arity of the inductive predicate P_i. Define a chain of ordinal-indexed sets $(\varphi_\Phi^\alpha \subseteq \mathrm{Pow}(D^{k_1}) \times \ldots \times \mathrm{Pow}(D^{k_n}))_{\alpha \geq 0}$ by transfinite induction: $\varphi_\Phi^\alpha = \bigcup_{\beta<\alpha} \varphi_\Phi(\varphi_\Phi^\beta)$ (note that this implies $\varphi_\Phi^0 = (\emptyset,\ldots,\emptyset)$). Then for $i \in \{1,\ldots,n\}$, the set $\pi_i^n(\varphi_\Phi^\alpha)$ is called the α^{th} *approximant of* P_i, written as P_i^α.

Proposition 2.2.9. *For any inductive definition set Φ, and for all ordinals α, we have $\varphi^\alpha_\Phi \subseteq \varphi_\Phi(\varphi^\alpha_\Phi)$.*

Proof. By transfinite induction on α. We then have:

	$\forall \beta < \alpha. \varphi^\beta_\Phi \subseteq \varphi_\Phi(\varphi^\beta_\Phi)$	by induction hypothesis
i.e.	$\forall \beta < \alpha. \varphi^\beta_\Phi \subseteq \bigcup_{\gamma<\alpha} \varphi_\Phi(\varphi^\gamma_\Phi)$	since $\varphi_\Phi(\varphi^\beta_\Phi) \subseteq \bigcup_{\gamma<\alpha} \varphi_\Phi(\varphi^\gamma_\Phi)$
i.e.	$\forall \beta < \alpha. \varphi_\Phi(\varphi^\beta_\Phi) \subseteq \varphi_\Phi(\bigcup_{\gamma<\alpha} \varphi_\Phi(\varphi^\gamma_\Phi))$	by Proposition 2.2.4
i.e.	$\bigcup_{\beta<\alpha} \varphi_\Phi(\varphi^\beta_\Phi) \subseteq \varphi_\Phi(\bigcup_{\gamma<\alpha} \varphi_\Phi(\varphi^\gamma_\Phi))$	
i.e.	$\varphi^\alpha_\Phi \subseteq \varphi_\Phi(\varphi^\alpha_\Phi)$	

□

Corollary 2.2.10. *For any inductive definition set Φ and for all ordinals α, $\varphi^{\alpha+1}_\Phi = \varphi_\Phi(\varphi^\alpha_\Phi)$.*

Proof. By definition, $\varphi^{\alpha+1}_\Phi = \bigcup_{\beta<\alpha+1} \varphi_\Phi(\varphi^\beta_\Phi) = \bigcup_{\beta<\alpha} \varphi_\Phi(\varphi^\beta_\Phi) \cup \varphi_\Phi(\varphi^\alpha_\Phi) = \varphi^\alpha_\Phi \cup \varphi_\Phi(\varphi^\alpha_\Phi)$. However, since $\varphi^\alpha_\Phi \subseteq \varphi_\Phi(\varphi^\alpha_\Phi)$ by Proposition 2.2.9, we can deduce $\varphi^{\alpha+1}_\Phi = \varphi_\Phi(\varphi^\alpha_\Phi)$ as required. □

Corollary 2.2.10 implies that for any α, we have $\varphi^{\alpha+1}_\Phi = \varphi^\alpha_\Phi$, i.e. α is a stationary point of the sequence[1] $(\varphi^\alpha_\Phi)_{\alpha\geq 0}$, if and only if $\varphi_\Phi(\varphi^\alpha_\Phi) \subseteq \varphi^\alpha_\Phi$, i.e. iff φ^α_Φ is a prefixed point of the monotone operator φ_Φ.

Note that the corollary also implies that instead of Definition 2.2.8, we could equivalently define the approximant sequence $(\varphi^\alpha_\Phi)_{\alpha\geq 0}$ by the following, as is sometimes used:

$$\begin{array}{rcl} \varphi^0_\Phi & = & \emptyset \\ \varphi^{\alpha+1}_\Phi & = & \varphi_\Phi(\varphi^\alpha_\Phi) \\ \varphi^\lambda_\Phi & = & \bigcup_{\beta<\lambda} \varphi^\beta_\Phi \quad (\lambda \text{ a limit ordinal}) \end{array}$$

Our next result, which is again standard, tells us that all our inductive definitions "close" at the first limit ordinal ω. (This is emphatically not the case for more complex definition schemas such as iterated inductive definitions.)

Lemma 2.2.11. *For any inductive definition set Φ, the least prefixed point of φ_Φ is φ^ω_Φ.*

Proof. We make use of the equality $\varphi^{\alpha+1}_\Phi = \varphi_\Phi(\varphi^\alpha_\Phi)$ given by Corollary 2.2.10 throughout this proof. We first show that φ^ω_Φ is *a* prefixed point of φ_Φ, i.e. that $\varphi_\Phi(\varphi^\omega_\Phi) \subseteq \varphi^\omega_\Phi$. We require to prove that $\varphi_i(\varphi^\omega_\Phi) \subseteq \pi^n_i(\varphi^\omega_\Phi)$ for all $i \in \{1,\ldots,n\}$ (where n is the number of inductive predicates, i.e. the arity of φ_Φ). Suppose $y \in \varphi_i(\varphi^\omega_\Phi)$ for some $i \in \{1,\ldots,n\}$. By construction of φ_i, there is a rule $\Phi_{i,r}$ such that $y \in \varphi_{i,r}(\varphi^\omega_\Phi)$:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1,\ldots,j_m,i \in \{1,\ldots,n\}$$

[1] Technically, $(\varphi^\alpha_\Phi)_{\alpha\geq 0}$ is not a sequence but a chain of ordinal-indexed sets. However, we shall show that it is sufficient to consider the subchain $(\varphi^\alpha_\Phi)_{0\leq\alpha\leq\omega}$, which can certainly be considered a sequence.

So we have:

$$\begin{aligned} & y \in \{\mathbf{t}^M(\mathbf{x}) \mid Q_1^M\mathbf{u}_\mathbf{1}^M(\mathbf{x}),\ldots,Q_h^M\mathbf{u}_\mathbf{h}^M(\mathbf{x}),\mathbf{t}_\mathbf{1}^M(\mathbf{x}) \in \pi^n_{j_1}(\varphi^\omega_\Phi),\ldots,\mathbf{t}_\mathbf{m}^M(\mathbf{x}) \in \pi^n_{j_m}(\varphi^\omega_\Phi)\} \\ \text{i.e.}\quad & y \in \{\mathbf{t}^M(\mathbf{x}) \mid Q_1^M\mathbf{u}_\mathbf{1}^M(\mathbf{x}),\ldots,Q_h^M\mathbf{u}_\mathbf{h}^M(\mathbf{x}), \\ & \qquad \mathbf{t}_\mathbf{1}^M(\mathbf{x}) \in \pi^n_{j_1}(\textstyle\bigcup_{k<\omega}\varphi_\Phi(\varphi^k_\Phi)),\ldots,\mathbf{t}_\mathbf{m}^M(\mathbf{x}) \in \pi^n_{j_m}(\bigcup_{k<\omega}\varphi_\Phi(\varphi^k_\Phi))\} \end{aligned}$$

Then there exist $k_1,\ldots,k_m \in \mathbb{N}$ such that:

$$y \in \{\mathbf{t}^M(\mathbf{x}) \mid Q_1^M\mathbf{u}_\mathbf{1}^M(\mathbf{x}),\ldots,Q_h^M\mathbf{u}_\mathbf{h}^M(\mathbf{x}),\mathbf{t}_\mathbf{1}^M(\mathbf{x}) \in \pi^n_{j_1}(\varphi_\Phi(\varphi^{k_1}_\Phi)),\ldots,\mathbf{t}_\mathbf{m}^M(\mathbf{x}) \in \pi^n_{j_m}(\varphi_\Phi(\varphi^{k_m}_\Phi))\}$$

Now let k be the maximum of $k_1,\ldots,k_m$. Note by that Proposition 2.2.9 and Corollary 2.2.10 we thus have $\varphi^{k_i}_\Phi \subseteq \varphi^k_\Phi$ for all $i \in \{1,\ldots,m\}$ and hence $\varphi_\Phi(\varphi^{k_i}_\Phi) \subseteq \varphi_\Phi(\varphi^k_\Phi)$ for all $i \in \{1,\ldots,m\}$ by monotonicity of φ_Φ (Proposition 2.2.4). Thus we have:

$$\begin{aligned} & y \in \{\mathbf{t}^M(\mathbf{x}) \mid Q_1^M\mathbf{u}_\mathbf{1}^M(\mathbf{x}),\ldots,Q_h^M\mathbf{u}_\mathbf{h}^M(\mathbf{x}),\mathbf{t}_\mathbf{1}^M(\mathbf{x}) \in \pi^n_{j_1}(\varphi_\Phi(\varphi^k_\Phi)),\ldots,\mathbf{t}_\mathbf{m}^M(\mathbf{x}) \in \pi^n_{j_m}(\varphi_\Phi(\varphi^k_\Phi))\} \\ \text{i.e.}\quad & y \in \{\mathbf{t}^M(\mathbf{x}) \mid Q_1^M\mathbf{u}_\mathbf{1}^M(\mathbf{x}),\ldots,Q_h^M\mathbf{u}_\mathbf{h}^M(\mathbf{x}),\mathbf{t}_\mathbf{1}^M(\mathbf{x}) \in \pi^n_{j_1}(\varphi^{k+1}_\Phi),\ldots,\mathbf{t}_\mathbf{m}^M(\mathbf{x}) \in \pi^n_{j_m}(\varphi^{k+1}_\Phi)\} \end{aligned}$$

So $y \in \varphi_{i,r}(\varphi^{k+1}_\Phi)$, i.e. $y \in \pi^n_i(\varphi_\Phi(\varphi^{k+1}_\Phi)) = \pi^n_i(\varphi^{k+2}_\Phi)$. It is obvious that $\varphi^{k+2}_\Phi \subseteq \varphi^\omega_\Phi$ and so $y \in \pi^n_i(\varphi^\omega_\Phi)$ as required. So φ^ω_Φ is a prefixed point of φ_Φ.

To see that φ^ω_Φ is in fact the least prefixed point of φ_Φ, let $(X_1,\ldots,X_n)$ be an arbitrary prefixed point of φ_Φ. We show that $\varphi^\omega_\Phi \subseteq (X_1,\ldots,X_n)$, i.e. that $\bigcup_{m<\omega}\varphi_\Phi(\varphi^m_\Phi) \subseteq (X_1,\ldots,X_n)$. As $\varphi_\Phi(X_1,\ldots,X_n) \subseteq (X_1,\ldots,X_n)$ by assumption, it suffices to show $\varphi_\Phi(\varphi^m_\Phi) \subseteq \varphi_\Phi(X_1,\ldots,X_n)$ for each $m < \omega$, i.e. for all $m \in \mathbb{N}$. Since φ_Φ is monotone (Proposition 2.2.4), we then just need to show $\varphi^m_\Phi \subseteq (X_1,\ldots,X_n)$ for all $m \in \mathbb{N}$. We proceed by induction on m:

Case $m = 0$**:** We trivially have $\varphi^0_\Phi = (\emptyset,\ldots,\emptyset) \subseteq (X_1,\ldots,X_n)$.

Case $m = k+1$**:** By induction hypothesis $\varphi^k_\Phi \subseteq (X_1,\ldots,X_n)$. By monotonicity of φ_Φ (Proposition 2.2.4 again) we have $\varphi_\Phi(\varphi^k_\Phi) \subseteq \varphi_\Phi(X_1,\ldots,X_n)$ and, using Corollary 2.2.10 and the fact that $(X_1,\ldots,X_n)$ is a prefixed point of φ_Φ, we obtain $\varphi^{k+1}_\Phi \subseteq (X_1,\ldots,X_n)$ as required. This completes the induction and thus the proof. □

Definition 2.2.12 (Standard model)**.** Where Φ is an inductive definition set for Σ, a structure M for Σ is said to be a *standard model* for (Σ,Φ) if for all $i \in \{1,\ldots,n\}$, $P_i^M = \bigcup_\alpha P_i^\alpha$.

Definition 2.2.12 thus fixes within a structure a standard interpretation of the inductive predicates of Σ that is uniquely determined by the other components of the structure. Actually, by Lemma 2.2.11, we know that $\bigcup_\alpha P_i^\alpha$ is just P_i^ω, but we keep the more general notation for modularity with more complex forms of inductive definition which may not close at ω.

Example 2.2.13. Let Φ_N be the definition set containing only the productions for N given in Example 2.2.5). We obtain the sequence of N-approximants:

$$\begin{array}{rcl} N^0 & = & \emptyset \\ N^1 & = & \{0^M\} \\ N^2 & = & \{0^M, s^M 0^M\} \\ & \vdots & \\ N^k & = & \{0^M, s^M 0^M, \ldots, (s^M)^k 0^M\} \\ & \vdots & \end{array}$$

If all "numerals" $(s^M)^k 0^M$ for $k \geq 0$ are interpreted in M as distinct elements, the predicate N is thus interpreted as the property of being a natural number.

Example 2.2.14. Let Φ_{EO} be the definition set given in Example 2.2.6. We obtain the following sequence of approximants of E and O:

$$\begin{array}{rcl} (E^0, O^0) & = & (\emptyset, \emptyset) \\ (E^1, O^1) & = & (\{0^M\}, \emptyset) \\ (E^2, O^2) & = & (\{0^M\}, \{s^M 0^M\}) \\ (E^3, O^3) & = & (\{0^M, s^M s^M 0^M\}, \{s^M 0^M\}) \\ & \vdots & \\ (E^{2k}, O^{2k}) & = & (\{0^M, s^M s^M 0^M, \ldots, (s^M s^M)^k 0^M\}, \{s^M 0^M, \ldots, s^M (s^M s^M)^{k-1} 0^M\}) \\ & \vdots & \end{array}$$

If all "numerals" $(s^M)^k 0^M$ for $k \geq 0$ are interpreted in M as distinct elements, the predicates E and O are interpreted as the property of being an even and odd natural number respectively.

Example 2.2.15. Let Φ_{R^+} be the definition set given in Example 2.2.7. We obtain the sequence of approximants of R^+:

$$\begin{array}{rcl} R^{+0} & = & \emptyset \\ R^{+1} & = & \{(x,y) \mid R^M(x,y)\} \\ R^{+2} & = & \{(x,y) \mid R^M(x,y)\} \cup \{(x,z) \mid \exists y. R^M(x,y), R^M(y,z)\} \\ R^{+3} & = & \{(x,y) \mid R^M(x,y)\} \cup \{(x,z) \mid \exists y. R^M(x,y), R^M(y,z)\} \cup \\ & & \{(x,z) \mid \exists y \exists y'. R^M(x,y), R^M(y,y'), R^M(y',z)\} \cup \\ & & \{(x,z) \mid \exists y \exists y' \exists y''. R^M(x,y), R^M(y,y'), R^M(y',y''), R^M(y'',z)\} \\ & \vdots & \end{array}$$

R^+ is thus interpreted in M as the property of being in the transitive closure of the interpretation of R in M.

Here one sees that it might be useful to allow non-atomic formulas in the premises of inductive productions, in order to allow the definition of the transitive closure of an arbitrary

formula in two variables. However, allowing arbitrary formulas in the premises of productions leads to possible non-monotonicity of definitions, as can be immediately seen by considering the "production" $\dfrac{\neg R^+xy}{R^+xy}$. To prevent this, we would need to impose conditions on the formulas appearing in the premises of productions, such as those for iterated inductive definitions discussed above. (Doing so would be perfectly feasible, but would unnecessarily complicate the general framework we consider.)

2.3 Henkin semantics for $\mathbf{FOL}_{ID}$

As well as the standard interpretation of inductive predicates within a structure defined in the previous section, we shall also be interested in certain non-standard interpretations. As seen previously, the standard way of interpreting inductive predicates defined by an inductive definition set Φ is as the least fixed point of the definition set operator φ_Φ in its domain $\mathrm{Pow}(D^{k_1}) \times \ldots \times \mathrm{Pow}(D^{k_n})$ (where k_i is the arity of the inductive predicate P_i for each $i \in \{1,\ldots,n\}$). However, one can also consider constructing this least fixed point in certain restricted classes of subsets of $\mathrm{Pow}(D^{k_1}) \times \ldots \times \mathrm{Pow}(D^{k_n})$, known as *Henkin classes*:

Definition 2.3.1 (Henkin class)**.** A *Henkin class* for a Σ-structure $M = (D,\ldots)$ is a family of sets $\mathcal{H} = \{H_k \mid k \in \mathbb{N}\}$, where $H_k \subseteq \mathrm{Pow}(D^k)$ for each $k \in \mathbb{N}$ and:

(H1) $\{(d,d) \mid d \in D\} \in H_2$;

(H2) if Q is a predicate symbol (either ordinary or inductive) of arity k then $\{(d_1,\ldots,d_k) \mid Q^M(d_1,\ldots,d_k)\} \in H_k$;

(H3) if $R \in H_{k+1}$ and $d \in D$ then $\{(d_1,\ldots,d_k) \mid (d_1,\ldots,d_k,d) \in R\} \in H_k$;

(H4) if $R \in H_k$ and $t_1(x_1,\ldots,x_m),\ldots,t_k(x_1,\ldots,x_m)$ are terms (containing only variables in $\{x_1,\ldots,x_m\}$) then $\{(d_1,\ldots,d_m) \mid (t_1^M(d_1,\ldots,d_m),\ldots,t_k^M(d_1,\ldots,d_m)) \in R\} \in H_m$;

(H5) if $R \in H_k$ then $\overline{R} = D^k \setminus R \in H_k$;

(H6) if $R_1, R_2 \in H_k$ then $R_1 \cap R_2 \in H_k$;

(H7) if $R \in H_{k+1}$ then $\{(d_1,\ldots,d_k) \mid \exists d.(d_1,\ldots,d_k,d) \in R\} \in H_k$.

Our next result states, essentially, that Henkin classes contain sufficiently many relations to interpret any formula of $\mathrm{FOL}_{\mathrm{ID}}$. We obtain the result first with a restriction on the free variables of the formula, and subsequently without this restriction:

Lemma 2.3.2. *If $\mathcal{H} = \{H_k \mid k \in \mathbb{N}\}$ is a Henkin class for a structure M, ρ is an environment for M, F is a Σ-formula of FOL_{ID} and $x_1,\ldots,x_k \in \mathcal{V}$ are distinct variables, then:*

$$FV(F) \subseteq \{x_1,\ldots,x_k\} \text{ implies } \{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F\} \in H_k$$

Proof. We proceed by structural induction on the formula F:

Case $F = Q(t_1,\ldots,t_m)$, where Q is either an ordinary or an inductive predicate symbol . By assumption $FV(F) = \bigcup_{1\leq i\leq m} Var(t_m) \subseteq \{x_1,\ldots,x_k\}$, so we can write each t_i (for $i \in \{1,\ldots,m\}$) as $t_i(x_1,\ldots,x_k)$. Now by (H2) we have $\{(d_1,\ldots,d_m) \mid Q^M(d_1,\ldots,d_m)\} \in H_m$ and, as $t_1,\ldots,t_m$ are terms whose variables are contained in $\{x_1,\ldots,x_k\}$, we obtain by applying (H4):

$$\{(d_1,\ldots,d_k) \mid Q^M(t_1^M(d_1,\ldots,d_k),\ldots,t_m^M(d_1,\ldots,d_k))\} \in H_k$$

and since $\rho(t_i) = t_i^M(\rho(x_1),\ldots,\rho(x_j))$ for each $i \in \{1,\ldots,m\}$, we thus have as required:

$$\{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} Q(t_1,\ldots,t_m)\} \in H_k$$

Case $F = (t_1 = t_2)$. As in the previous two cases, we write each t_i as $t_i(x_1,\ldots,x_k)$ for $i \in \{1,2\}$. By (H1) we have $\{(d,d) \mid d \in D\} \in H_2$, i.e. $\{(d_1,d_2) \mid d_1 = d_2\} \in H_2$. Since t_1,t_2 are terms whose variables are contained in $\{x_1,\ldots,x_k\}$, we obtain by applying (H4):

$$\{(d_1,\ldots,d_k) \mid t_1^M(d_1,\ldots,d_k) = t_2^M(d_1,\ldots,d_k)\} \in H_k$$

and since $\rho(t_i) = t_i^M(\rho(x_1),\ldots,\rho(x_k))$ for $i \in \{1,2\}$, we thus have as required:

$$\{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k,]} t_1 = t_2\} \in H_k$$

Case $F = \neg F'$. Note $FV(F') = FV(F) \subseteq \{x_1,\ldots,x_k\}$ and so by induction hypothesis we have:

$$\{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F'\} \in H_k$$

Since H_k is closed under complement by (H5) we thus have:

$$\begin{aligned} &\{(d_1,\ldots,d_k) \mid M \not\models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F'\} \in H_k \\ \text{i.e.}\quad &\{(d_1,\ldots,d_j) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_j\mapsto d_j]} \neg F'\} \in H_k \end{aligned}$$

which completes the case.

Case $F = F_1 \wedge F_2$. Since $FV(F_1) \cup FV(F_2) = FV(F_1 \wedge F_2) \subseteq \{x_1,\ldots,x_k\}$, we have $FV(F_1) \subseteq \{x_1,\ldots,x_k\}$ and $FV(F_2) \subseteq \{x_1,\ldots,x_k\}$ and so by induction hypothesis we have:

$$\{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F_1\} \in H_k \text{ and } \{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F_2\} \in H_k$$

Since H_k is closed under intersection by (H6), we then have as required:

$$\begin{aligned} &\{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F_1 \text{ and } M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F_2\} \in H_k \\ \text{i.e.}\quad &\{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F_1 \wedge F_2\} \in H_k \end{aligned}$$

Case $F = F_1 \vee F_2$. Notice that H_k is closed under union since it is closed under complement (H5) and intersection (H6), by the identity $A \cup B = \overline{\overline{A} \cap \overline{B}}$. This case then follows by the induction hypothesis in a similar manner to the previous case.

Case $F = F_1 \rightarrow F_2$. This case follows from the induction hypothesis in a similar manner to the previous two cases.

Case $F = \exists x F'$. Since $FV(F) = FV(F') \setminus \{x\} \subseteq \{x_1, \ldots, x_k\}$, we have $FV(F') \subseteq \{x_1, \ldots, x_k, x\}$, where x is distinct from all of $x_1, \ldots, x_k$. Thus by induction hypothesis:

$$\{(d_1, \ldots, d_k, d) \mid M \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k, x \mapsto d]} F'\} \in H_{k+1}$$

and by applying (H7) we obtain:

$$\begin{aligned} & \{(d_1, \ldots, d_k) \mid \exists d \in D.\ M \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k, x \mapsto d]} F'\} \in H_k \\ \text{i.e.} \quad & \{(d_1, \ldots, d_k) \mid M \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k]} \exists x F'\} \in H_k \end{aligned}$$

as required.

Case $F = \forall x F'$. Note that for any environment ρ, we have $M \models_\rho \forall x F' \Leftrightarrow M \models_\rho \neg \exists x \neg F'$. This case then follows from the cases $F \equiv \neg F'$ and $F \equiv \exists x F'$ above. □

Proposition 2.3.3. *If $\mathcal{H} = \{H_k \mid k \in \mathbb{N}\}$ is a Henkin class for a structure M, ρ is an environment for M, F is a formula of FOL_{ID} and $x_1, \ldots, x_k \in \mathcal{V}$ are distinct variables, then:*

$$\{(d_1, \ldots, d_k) \mid M \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k]} F\} \in H_k$$

Proof. First notice that if $FV(F) \subseteq \{x_1, \ldots, x_k\}$ then we are immediately done by Lemma 2.3.2. So suppose that $FV(F) \not\subseteq \{x_1, \ldots, x_k\}$ and let $FV(F) \setminus \{x_1, \ldots, x_k\} = \{y_1, \ldots, y_j\}$, where $y_1, \ldots, y_j$ are all distinct (and necessarily distinct from all of $x_1, \ldots, x_k$). Then $FV(F) \subseteq \{x_1, \ldots, x_k, y_1, \ldots, y_j\}$, so by applying Lemma 2.3.2 we obtain:

$$\{(d_1, \ldots, d_k, e_1, \ldots, e_j) \mid (M, \mathcal{H}) \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k, y_1 \mapsto e_1, \ldots, y_j \mapsto e_j]} F\} \in H_{k+j}$$

Obviously $\rho(y_1), \ldots, \rho(y_j) \in D$, so by applying (H3) j times it holds that:

$$\{(d_1, \ldots, d_k) \mid (M, \mathcal{H}) \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k, y_1 \mapsto \rho(y_1), \ldots, y_j \mapsto \rho(y_j)]} F\} \in H_k$$

and by applying part 2 of Lemma 2.1.9 (again j times), we then have:

$$\begin{aligned} & \{(d_1, \ldots, d_k) \mid (M, \mathcal{H}) \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k]} F[y_1/y_1, \ldots, y_j/y_j]\} \in H_k \\ \text{i.e.} \quad & \{(d_1, \ldots, d_k) \mid (M, \mathcal{H}) \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k]} F\} \in H_k \end{aligned}$$

as required. □

Proposition 2.3.3 will be important when establishing soundness of proof systems with respect to our Henkin models in the next chapter. The next result shows essentially the converse of the proposition: that the interpretations of formulas in $\mathrm{FOL_{ID}}$ in a particular structure and interpretation actually form a Henkin class.

Proposition 2.3.4. *Let $M = (D, \ldots)$ be a Σ-structure and let ρ be an environment for M. Then the class $\mathcal{H} = \{H_k \mid k \in \mathbb{N}\}$ defined by:*

$$H_k = \{\{(d_1, \ldots, d_k) \mid M \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k]} F\} \mid F \text{ a formula, } x_1, \ldots, x_k \text{ distinct variables}\}$$

for each $k \geq 0$ is a Henkin class for M.

Proof. We need to verify that the 7 closure conditions on Henkin classes (c.f. Definition 2.3.1) hold for $\mathcal{H}$:

(H1) For any distinct variables x_1, x_2 of Σ, $x_1 = x_2$ is a formula of Σ, so we have:

$$\begin{aligned} & \{(d_1, d_2) \mid M \models_{\rho[x_1 \mapsto d_1, x_2 \mapsto d_2]} x_1 = x_2\} \in H_2 \\ \text{i.e.} \quad & \{(d_1, d_2) \mid \rho[x_1 \mapsto d_1, x_2 \mapsto d_2](x_1) = \rho[x_1 \mapsto d_1, x_2 \mapsto d_2](x_2)\} \in H_2 \\ \text{i.e.} \quad & \{(d_1, d_2) \mid d_1 = d_2\} \in H_2 \\ \text{i.e.} \quad & \{(d, d) \mid d \in D\} \end{aligned}$$

as required.

(H2) Let Q be a predicate symbol (either ordinary or inductive) of arity k, and let $x_1, \ldots, x_k$ be distinct variables. Then $Q(x_1, \ldots, x_k)$ is a formula of Σ, so by definition of H_k we have:

$$\begin{aligned} & \{(d_1, \ldots, d_k) \mid M \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k]} Q(x_1, \ldots, x_k)\} \in H_k \\ \text{i.e.} \quad & \{(d_1, \ldots, d_k) \mid Q^M(d_1, \ldots, d_k)\} \in H_k \end{aligned}$$

as required.

(H3) Let $R \in H_{k+1}$ and let $d \in D$. By definition of H_{k+1}, there is a formula F such that:

$$R = \{(d_1, \ldots, d_{k+1}) \mid M \models_{\rho[x_1 \mapsto d_1, \ldots, x_{k+1} \mapsto d_{k+1}]} F\}$$

where $x_1, \ldots, x_{k+1}$ are distinct variables. Now let z be a variable distinct from any of $x_1, \ldots, x_k$ and not occurring in F. Without loss of generality, we may set $\rho(z) = d$. Since $F[z/x_{k+1}]$ is a formula of Σ, we have by definition of H_k:

$$\begin{aligned} & \{(d_1, \ldots, d_k) \mid M \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k]} F[z/x_{k+1}]\} \in H_k \\ \text{i.e.} \quad & \{(d_1, \ldots, d_k) \mid M \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k, x_{k+1} \mapsto \rho(z)]} F\} \in H_k \quad \text{by Lemma 2.1.9} \\ \text{i.e.} \quad & \{(d_1, \ldots, d_k) \mid (d_1, \ldots, d_k, d) \in R\} \in H_k \end{aligned}$$

as required (note that we can combine the substitutions in the environment in the second step since x_{k+1} is distinct from each of $x_1, \ldots, x_k$).

(H4) Let $R \in H_k$ so that, by definition of H_k, there is a formula F and distinct variables $x_1,\ldots,x_k$ such that:

$$R = \{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F\}$$

Now let $t_1(y_1,\ldots,y_m),\ldots,t_k(y_1,\ldots,y_m)$ be Σ-terms whose variables are all contained in $\{y_1,\ldots,y_m\}$. Without loss of generality, we may assume that $y_1,\ldots,y_m$ do not occur in F and are distinct from all the x_i. Since $F[t_1(y_1,\ldots,y_m)/x_1,\ldots,t_k(y_1,\ldots,y_m)/x_k]$ is a formula, we have by definition of H_m:

$$\{(d_1,\ldots,d_m) \mid M \models_{\rho[y_1\mapsto d_1,\ldots,y_m\mapsto d_m]} F[t_1(y_1,\ldots,y_m)/x_1,\ldots,t_k(y_1,\ldots,y_m)/x_k]\} \in H_m$$

Now, writing ρ' to abbreviate $\rho[y_1 \mapsto d_1,\ldots,y_m \mapsto d_m]$, we have by Lemma 2.1.9:

$$\begin{aligned} &\{(d_1,\ldots,d_m) \mid M \models_{\rho'[x_1\mapsto \rho'(t_1(y_1,\ldots,y_m)),\ldots,x_k\mapsto \rho'(t_k(y_1,\ldots,y_m))]} F\} \in H_m \\ \text{i.e.}\quad &\{(d_1,\ldots,d_m) \mid M \models_{\rho'[x_1\mapsto t_1^M(d_1,\ldots,d_m),\ldots,x_k\mapsto t_k^M(d_1,\ldots,d_m)]} F\} \in H_m \end{aligned}$$

and since $y_1,\ldots,y_m$ do not occur in F, we have again by Lemma 2.1.9:

$$\begin{aligned} &\{(d_1,\ldots,d_m) \mid M \models_{\rho'[x_1\mapsto t_1^M(d_1,\ldots,d_m),\ldots,x_k\mapsto t_k^M(d_1,\ldots,d_m)]} F\} \in H_m \\ \text{i.e.}\quad &\{(d_1,\ldots,d_m) \mid (t_1^M(d_1,\ldots,d_m),\ldots,t_k^M(d_1,\ldots,d_m)) \in R\} \in H_m \end{aligned}$$

as required.

(H5) Let $R \in H_k$ so that, by definition of H_k, there is a formula F and distinct variables $x_1,\ldots,x_k$ such that:

$$R = \{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F\}$$

Then as $\neg F$ is a formula, we have:

$$\begin{aligned} &\{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} \neg F\} \in H_k \\ \text{i.e.}\quad &\{(d_1,\ldots,d_k) \mid M \not\models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F\} \in H_k \\ \text{i.e.}\quad &\{(d_1,\ldots,d_k) \mid (d_1,\ldots,d_k) \notin R\} \in H_k \end{aligned}$$

i.e. $\overline{R} \in H_k$ as required.

(H6) Let $R_1,R_2 \in H_k$, so that there are formulas F_1,F_2 such that:

$$\begin{aligned} R_1 &= \{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F_1\} \\ R_2 &= \{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F_2\}) \end{aligned}$$

where $x_1,\ldots,x_k$ are distinct variables. Note that we can choose the same variables $x_1,\ldots,x_k$ for R_1 and R_2 without loss of generality. Since $F_1 \wedge F_2$ is a formula, we have:

$$\begin{aligned} &\{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F_1 \wedge F_2\} \in H_k \\ \text{i.e.}\quad &\{(d_1,\ldots,d_k) \mid M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F_1 \text{ and } M \models_{\rho[x_1\mapsto d_1,\ldots,x_k\mapsto d_k]} F_2\} \in H_k \\ \text{i.e.}\quad &\{(d_1,\ldots,d_k) \mid (d_1,\ldots,d_k) \in R_1 \text{ and } (d_1,\ldots,d_k) \in R_2\} \in H_k \end{aligned}$$

i.e. $R_1 \cap R_2 \in H_k$ as required.

(H7) Let $R \in H_{k+1}$ so that there is a formula F and distinct variables $x_1, \ldots, x_{k+1}$ such that:

$$R = \{(d_1, \ldots, d_{k+1}) \mid M \models_{\rho[x_1 \mapsto d_1, \ldots, x_{k+1} \mapsto d_{k+1}]} F\}$$

Then since $\exists x_{k+1} F$ is a formula, we have:

$$\begin{aligned} & \{(d_1, \ldots, d_k) \mid M \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k]} \exists x_{k+1} F\} \in H_k \\ \text{i.e.} \quad & \{(d_1, \ldots, d_k) \mid \text{for some } d \in D,\ M \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k, x_{k+1} \mapsto d]} F\} \in H_k \\ \text{i.e.} \quad & \{(d_1, \ldots, d_k) \mid \exists d \in D.(d_1, \ldots, d_k, d) \in R\} \in H_k \end{aligned}$$

as required.

□

We now move on to defining Henkin models, which are simply Henkin classes inside which a least prefixed point of the inductive operator φ_Φ exists:

Definition 2.3.5 ($\mathcal{H}$-point)**.** Let M be a structure for Σ, let Φ be an inductive definition set for Σ and let $\mathcal{H}$ be a Henkin class for M. Also let k_i be the arity of the inductive predicate symbol P_i for each $i \in \{1, \ldots, n\}$. Then $(X_1, \ldots, X_n)$ is said to be an *$\mathcal{H}$-point* (of φ_Φ) if $X_i \in H_{k_i}$ for each $i \in \{1, \ldots, n\}$.

Lemma 2.3.6. *Let $\mathcal{H}$ be a Henkin class for a Σ-structure $M = (D, \ldots)$ and let Φ be an inductive definition set. Then if $(X_1, \ldots, X_n)$ is an $\mathcal{H}$-point of φ_Φ then so is $\varphi_\Phi(X_1, \ldots, X_n)$.*

Proof. By definition of φ_Φ, we are required to show $\bigcup_r \varphi_{i,r}(X_1, \ldots, X_n) \in H_{k_i}$ for an arbitrary $i \in \{1, \ldots, n\}$, where k_i is the arity of the inductive predicate symbol P_i. Note that H_{k_i} is closed under complement and intersection by clauses (H5) and (H6) of Definition 2.3.1. H_{k_i} is thus closed under union since $A \cup B = \overline{\overline{A} \cap \overline{B}}$. It therefore suffices to prove that $\varphi_{i,r}(X_1, \ldots, X_n) \in H_{k_i}$, where $\Phi_{i,r} \in \Phi$ is a production with P_i in its conclusion:

$$\frac{Q_1 \mathbf{u_1}(\mathbf{x}) \ldots Q_h \mathbf{u_h}(\mathbf{x}) \quad P_{j_1} \mathbf{t_1}(\mathbf{x}) \ldots P_{j_m} \mathbf{t_m}(\mathbf{x})}{P_i \mathbf{t}(\mathbf{x})} \qquad j_1, \ldots, j_m, i \in \{1, \ldots, n\}$$

i.e. we require to show:

$$\{\mathbf{t}^M(\mathbf{x}) \mid Q_1^M \mathbf{u_1}^M(\mathbf{x}), \ldots, Q_h^M \mathbf{u_h}^M(\mathbf{x}), \mathbf{t_1}^M(\mathbf{x}) \in X_{j_1}, \ldots, \mathbf{t_m}^M(\mathbf{x}) \in X_{j_m}\} \in H_{k_i}$$

First write $t_1(x_1, \ldots, x_l), \ldots, t_{k_i}(x_1, \ldots, x_l)$ for $\mathbf{t}(\mathbf{x})$ with $x_1, \ldots, x_l$ all distinct. Let $y_1, \ldots, y_{k_i}$ be variables of Σ distinct from each other and from $x_1, \ldots, x_l$, let $j \in \{1, \ldots, k_i\}$ and note that y_j and $t_j(x_1, \ldots, x_l)$ are terms whose variables are contained in $\{y_1, \ldots, y_{k_i}, x_1, \ldots, x_l\}$. Since $\{(d,d) \mid d \in D\} = \{(d_1, d_2) \mid d_1 = d_2\} \in H_2$ by (H1), we can apply (H4) to obtain:

$$\{(d_1, \ldots, d_{k_i}, e_1, \ldots, e_l) \mid d_j = t_j^M(e_1, \ldots, e_l)\} \in H_{k_i + l} \text{ for all } j \in \{1, \ldots, k_i\} \qquad (1)$$

Now let $j \in \{1,\ldots,h\}$, and note that by (H2) we have $\{\mathbf{d} \mid Q_j^M\mathbf{d}\} \in H_k$ where $k = |\mathbf{d}|$ is the arity of Q_j. Since the variables of $\mathbf{u_j}(\mathbf{x})$ are again contained in $\{y_1,\ldots,y_{k_i},x_1,\ldots,x_l\}$, we have by (H4):

$$\{(d_1,\ldots,d_{k_i},e_1,\ldots,e_l) \mid Q_j^M\mathbf{u_j}(e_1,\ldots,e_l)\} \in H_{k_i+l} \text{ for all } j \in \{1,\ldots,h\} \tag{2}$$

Next, let $z \in \{1,\ldots,m\}$ and note that by assumption we have $X_{j_z} \in H_{k_{j_z}}$, i.e. $\{\mathbf{d} \mid \mathbf{d} \in X_{j_z}\} \in H_{k_{j_z}}$. Since the variables of $\mathbf{t_z}(\mathbf{x})$ are again contained in $\{y_1,\ldots,y_{k_i},x_1,\ldots,x_l\}$, we have by (H4):

$$\{(d_1,\ldots,d_{k_i},e_1,\ldots,e_l) \mid \mathbf{t_z}(e_1,\ldots,e_l) \in X_{j_z}\} \in H_{k_i+l} \text{ for all } z \in \{1,\ldots,m\} \tag{3}$$

Now since H_{k_i+l} is closed under intersection, we can combine (1), (2) and (3) above to obtain:

$$\begin{aligned}\{(d_1,\ldots,d_{k_i},e_1,\ldots,e_l) \mid\ & d_1 = t_1^M(e_1,\ldots,e_l),\ldots,d_{k_i} = t_{k_i}^M(e_1,\ldots,e_l),\\ & Q_1^M\mathbf{u_1}(e_1,\ldots,e_l),\ldots,Q_h^M\mathbf{u_h}(e_1,\ldots,e_l),\\ & \mathbf{t_1}(e_1,\ldots,e_l) \in X_{j_1},\ldots,\mathbf{t_m}(e_1,\ldots,e_l) \in X_{j_m}\} \in H_{k_i+l}\end{aligned}$$

By applying (H7) l times, we thus obtain:

$$\begin{aligned}\{(d_1,\ldots,d_{k_i}) \mid\ & \exists e_1.\ldots.\exists e_l.\, d_1 = t_1^M(e_1,\ldots,e_l),\ldots,d_{k_i} = t_{k_i}^M(e_1,\ldots,e_l),\\ & Q_1^M\mathbf{u_1}(e_1,\ldots,e_l),\ldots,Q_h^M\mathbf{u_h}(e_1,\ldots,e_l),\\ & \mathbf{t_1}(e_1,\ldots,e_l) \in X_{j_1},\ldots,\mathbf{t_m}(e_1,\ldots,e_l) \in X_{j_m}\} \in H_{k_i}\end{aligned}$$

which can be rewritten to:

$$\{(\mathbf{t}^M(\mathbf{e}) \mid Q_1^M\mathbf{u_1}(\mathbf{e}),\ldots,Q_h^M\mathbf{u_h}(\mathbf{e}),\mathbf{t_1}(\mathbf{e}) \in X_{j_1},\ldots,\mathbf{t_m}(\mathbf{e}) \in X_{j_m}\} \in H_{k_i}$$

as required. □

Given a Henkin class $\mathcal{H}$ and a structure, a non-standard interpretation of the inductive predicates is then obtained by considering the least prefixed $\mathcal{H}$-point of φ_Φ. Note that, by Lemma 2.3.6, $\varphi_\Phi(X_1,\ldots,X_n)$ is an $\mathcal{H}$-point if $(X_1,\ldots,X_n)$ is, so this least prefixed point, if it exists, is constructed entirely inside $\mathcal{H}$.

Definition 2.3.7 (Henkin model)**.** Let M be a structure and Φ be an inductive definition set for Σ, and let $\mathcal{H}$ be a Henkin class for M. Then $(M,\mathcal{H})$ is said to be a *Henkin model* for (Σ,Φ) if the operator φ_Φ has a least prefixed $\mathcal{H}$-point, which we write as $\mu_{\mathcal{H}}.\varphi_\Phi$, and for each $i \in \{1,\ldots,n\}$, $P_i^M = \pi_i^n(\mu_{\mathcal{H}}.\varphi_\Phi)$.

We observe that any standard model M for (Σ,Φ) can be viewed as a special case of a Henkin model for (Σ,Φ), by taking as the Henkin class for the structure the powerset of tuples over the domain of M:

Proposition 2.3.8. *If $M = (D, \ldots)$ is a standard model for (Σ, Φ) then $(M, \{Pow(D^k) \mid k \in \mathbb{N}\})$ is a Henkin model for Φ.*

Proof. We write k_i for the arity of the inductive predicate P_i for each $i \in \{1, \ldots, n\}$. First, we observe that $\{\mathrm{Pow}(D^k) \mid k \in \mathbb{N}\}$ trivially satisfies the conditions necessary for it to be a Henkin class (c.f. Definition 2.3.1). Secondly, we observe that the results of Section 2.2 show that φ_Φ has a least fixed point in $\mathrm{Pow}(D^{k_1}) \times \ldots \times \mathrm{Pow}(D^{k_n}) \subseteq (\{\mathrm{Pow}(D^k) \mid k \in \mathbb{N}\})^n$, as required by Definition 2.3.7. □

One sees by Proposition 2.3.8 that there are at least as many of our Henkin models as there are standard models for a given language and set of inductive definitions. However, we can also give a direct construction of a non-standard Henkin model, thus demonstrating that the class of Henkin models is more general than the class of standard models. We start by recalling the standard definition of Peano arithmetic (PA):

Definition 2.3.9 (Peano arithmetic)**.** Let Σ_{PA} be the first-order language consisting of the constant symbol 0, unary function symbol s, and binary function symbols $\cdot$ and $+$. Then *Peano arithmetic* (PA) is the theory in the language Σ_{PA} axiomatized by the following:

(PA1) $\forall x.\ \neg(sx = 0)$

(PA2) $\forall x \forall y.\ sx = sy \rightarrow x = y$

(PA3) $\forall x.\ 0 + y = y$

(PA4) $\forall x \forall y.\ sx + y = s(x + y)$

(PA5) $\forall x.\ 0 \cdot y = 0$

(PA6) $\forall x \forall y.\ sx \cdot y = x \cdot y + y$

(PA7) $\forall z_1 \ldots \forall z_n \forall z.\ F[0/z] \wedge \forall y(F[y/z] \rightarrow F[sy/z]) \rightarrow \forall x F[x/z]$
where F is a formula and $FV(F) \subseteq \{z_1, \ldots, z_n, z\}$

Proposition 2.3.10. *Let Σ'_{PA} be the language obtained by extending Σ_{PA} with a unary inductive predicate symbol N, and let Φ_N be the inductive definition set consisting of the "natural number" productions for N defined in Example 2.2.5. Then from every Σ_{PA}-structure in which all of the axioms of PA are true, one can construct a Henkin model of (Σ'_{PA}, Φ_N).*

Proof. Let $M = (D, \ldots)$ be a structure in which all the axioms of PA are true and extend M to Σ'_{PA} by defining $N^M = D$. Now define a Henkin class $\mathcal{H} = \{H_k \mid k \in \mathbb{N}\}$ by:

$$H_k = \{\{(d_1, \ldots, d_k) \mid M \models_{\rho[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k]} F\} \mid F \text{ a formula, } x_1, \ldots, x_k \text{ distinct variables}\}$$

By Proposition 2.3.4, $\mathcal{H}$ is a Henkin class for M. We claim $N^M = D$ is the least prefixed $\mathcal{H}$-point of the definition set operator φ_{Φ_N} (given in Example 2.2.5). By clause (H2) of the definition of a Henkin class, $\{d \mid N^M d\} = D \in H_1$, and trivially we have $\varphi_{\Phi_N}(D) \subseteq D$, so N^M is a prefixed $\mathcal{H}$-point of φ_{Φ_N}. Now let X be a prefixed $\mathcal{H}$-point of φ_{Φ_N}. Since X is a $\mathcal{H}$-point of φ_{Φ_N}, i.e. $X \in H_1$, there exists a formula F and a variable $z \in \mathcal{V}$ such that $X = \{d \mid M \models_{\rho[z \mapsto d]} F\}$. Also, we have $\varphi_{\Phi_N}(X) \subseteq X$, i.e. $0^M \in X$ and $d \in X$ implies $s^M d \in X$ for all $d \in D$ by definition of φ_{Φ_N}. By definition of X, we then have:

$$M \models_{\rho[z \mapsto 0^M]} F \text{ and } M \models_{\rho[z \mapsto d]} F \text{ implies } M \models_{\rho[z \mapsto s^M d]} F \text{ for all } d \in D$$

By Lemma 2.1.9, part 2, it follows that:

$$M \models_\rho F[0/z] \text{ and } M \models_\rho \forall y(F[y/z] \to F[sy/z])$$

Now, since the induction axiom (PA7) is true in M by assumption, we have:

$$\begin{aligned}
& M \models_\rho \forall x F[x/z] \\
\text{i.e.} \quad & M \models_{\rho[x \mapsto d]} F[x/z] \text{ for all } d \in D \\
\text{i.e.} \quad & M \models_{\rho[x \mapsto d][z \mapsto \rho[x \mapsto d](x)]} F \text{ for all } d \in D, \text{ by part 2 of Lemma 2.1.9} \\
\text{i.e.} \quad & M \models_{\rho[x \mapsto d, z \mapsto d]} F \text{ for all } d \in D
\end{aligned}$$

Without loss of generality, we can choose x not to occur free in F, whence we have $M \models_{\rho[z \mapsto d]} F$ for all $d \in D$ by part 1 of Lemma 2.1.9. Thus $d \in X$ for all $d \in D$, so $X \subseteq D$. So $D = N^M$ is the least prefixed $\mathcal{H}$-point of φ_{Φ_N}, whence $(M, \mathcal{H})$ is indeed a Henkin model for $(\Sigma'_{\mathrm{PA}}, \Phi_N)$. □

Corollary 2.3.11. *There are Henkin models that are not standard models.*

Proof. It is well-known that there are first-order structures for Σ_{PA} in which all of the axioms of PA are true, but which are not isomorphic to $\mathbb{N}$ (for examples of the construction of such structures, which are often called "nonstandard models of PA", see e.g. [38]). Let $M = (D, \ldots)$ be such a structure and let $(M, \mathcal{H})$ be the corresponding Henkin model of $(\Sigma'_{\mathrm{PA}}, \Phi_N)$ constructed by Proposition 2.3.10.

Now if $(M, \mathcal{H})$ were a standard model for $(\Sigma'_{\mathrm{PA}}, \Phi_N)$, we would have $N^M = D = \varphi^\omega_{\Phi_N}$ (using Lemma 2.2.11). But then, by inspection of the N-approximants as given in Example 2.2.13, D could clearly contain no element not of the form $(s^M)^n 0^M$ for some $n \in \mathbb{N}$. The lack of such "nonstandard elements" implies M is isomorphic to $\mathbb{N}$, contrary to assumption (the simple argument that structures non-isomorphic to $\mathbb{N}$ must contain a "nonstandard element" appears in [38]). So $(M, \mathcal{H})$ is a Henkin model of $(\Sigma'_{\mathrm{PA}}, \Phi_N)$, but not a standard model. □

In general, a sufficiently powerful (finitary) proof system for $\mathrm{FOL}_{\mathrm{ID}}$ cannot be complete with respect to standard models because PA can be formalised within it. We will demonstrate this in detail in Section 3.4.2. The completeness of such a system with respect to standard

models would then imply that the true statements of arithmetic are recursively enumerable via an enumeration of all proofs, which is known not to be the case. However, the fact that the class of Henkin models is strictly larger than the class of standard models raises the possibility of finding completeness results for such a system with respect to Henkin models, as Henkin himself did for second-order logic. Indeed, we obtain just such a result in the next chapter.

Chapter 3

LKID: a proof system for explicit induction in $\mathbf{FOL}_{ID}$

In this chapter we formulate a proof system, LKID, for the logic $\mathrm{FOL}_{\mathrm{ID}}$ in the sequent calculus style originally developed by Gentzen in 1935 [25]. Our system can be seen essentially as a classical sequent calculus adaptation of Martin-Löf's intuitionistic natural deduction proof system in [44]. McDowell, Miller, Momigliano and Tiu have considered similar intuitionistic sequent calculi, albeit with somewhat different rules for induction and definitional unfolding [47, 48, 49, 73, 74]. In Section 3.1 we give the proof rules of LKID, which are essentially the rules of Gentzen's original sequent calculus for first-order logic LK, augmented with rules for equality and rules (adapted from [44]) for introducing atomic formulas involving inductive predicates on the left and right of sequents. The right-introduction rules for inductive predicates correspond to definitional unfolding and the left-introduction rules correspond to induction over a definition. In Section 3.2 we show that our proof rules are sound with respect to the Henkin semantics for $\mathrm{FOL}_{\mathrm{ID}}$ (and hence in particular sound with respect to the standard semantics) given in the previous chapter. We also introduce our technical apparatus for formally defining proofs in an arbitrary system — although proofs in LKID are just the usual finite trees of sequents — for use in the development and analysis of infinite and cyclic proof systems in subsequent chapters.

In Section 3.3, we show that LKID is complete with respect to Henkin model validity, i.e. that any sequent true in every Henkin model has a proof in LKID. Moreover, this result is obtained for the cut-free part of LKID, and so from the soundness and completeness results we obtain a semantic proof of the eliminability of the cut rule for LKID. This result is perhaps not entirely surprising, as proofs of cut-elimination have been given previously for the analogous intuitionistic natural deduction and sequent calculus proof systems mentioned above [44, 48, 73], but the proof technique used for these systems does not straightforwardly adapt to the classical setting. We believe our result to be the first of its kind for a classical system. Finally,

in Section 3.4, we give a wider context to this result by showing, firstly, that LKID can be embedded into Takeuti's sequent calculus for second-order logic [69] and, secondly, that the eliminability of cut in LKID implies the consistency of Peano arithmetic. It follows from the latter that there can be no elementary proof of cut-eliminability for LKID.

Throughout this chapter, we shall assume a fixed language Σ with exactly n inductive predicate symbols $P_1, \ldots, P_n$, and a fixed inductive definition set Φ for Σ.

3.1 Sequent calculus proof rules for $\mathbf{FOL}_{ID}$

In this section we shall give the rules of a proof system for $\mathrm{FOL_{ID}}$ presented in the *sequent calculus* style invented by Gentzen [25]. We use sequent calculus because it is by now well-established as a convenient formalism for proof-theoretic reasoning, which is our focus. It also serves as an elegant framework in which to write formal proofs; in fact, similar formalisms typically underlie automated theorem-proving tools.

A *sequent* of $\mathrm{FOL_{ID}}$ is written in the form $\Gamma \vdash \Delta$, where Γ and Δ are finite multisets of formulas of $\mathrm{FOL_{ID}}$. A sequent calculus *proof rule* (R) is written in the form:

$$\frac{\Gamma_1 \vdash \Delta_1 \quad \ldots \quad \Gamma_n \vdash \Delta_n}{\Gamma \vdash \Delta}\,(\mathrm{R})$$

where $n \in \mathbb{N}$; the sequents above the line are called the *premises* of the rule and the sequent below the line is called the *conclusion* of the rule. If a rule has no premises then it is said to be an *axiom*.

In Figure 3.1 we give a version of the proof rules of Gentzen's original sequent calculus LK for classical first-order logic [25]. Note that in an instance of any of the non-structural rules, the distinguished formula that is introduced in the conclusion, i.e., that does not appear in any of the premises, is called the *active formula* of the rule instance. (We shall maintain this convention in our rules for equality and for inductive predicates below.) Also, in instances of the rules ($\forall$R) and ($\exists$L), the "fresh" variable x used for the introduced quantifier is called the *eigenvariable* of the rule instance. For convenience, our version of LK exhibits some minor deviations from Gentzen's original formulation, specifically:

- Gentzen's original "exchange" rules, governing the ordering of formulas within a sequent, are rendered redundant by our use of multisets in sequents;
- Gentzen's original logical axioms $A \vdash A$, for A an atomic formula, have been generalised so that a logical axiom is just one in which a (possibly non-atomic) formula occurs on both the left and right of a sequent;
- Gentzen's original left- and right-weakening rules have been combined into a single weakening rule (Wk) allowing the weakening of a sequent by an arbitrary number of formulas simultaneously on the left and/or right;

- we have included an explicit substitution rule (Subst), which is easily proven admissible in LK. (However, this rule will be important for our cyclic proof system in Chapter 5, and it is convenient to include it here.)

These differences are inessential, and are commonly adopted for the purposes of convenience. It is easily seen that a sequent is provable in Gentzen's original formulation of LK just in case it is provable in our variant.

In $\mathrm{FOL_{ID}}$ we treat equality $=$ as a primitive logical symbol. Accordingly, we give rules for introducing equality formulas on the left and right of sequents in Figure 3.2, and write $\mathrm{LK_e}$ for the sequent calculus for first-order logic with equality obtained by adding these rules to LK. The first such rules seem to have been formulated by Wang [83] in 1960, and were later investigated more fully by Kanger [36]. It is also quite usual to formulate versions of $\mathrm{LK_e}$ by adding axioms governing equality to LK (see e.g. [15, 70]). However, cut is not eliminable in $\mathrm{LK_e}$ formulated this way whereas cut *is* known to be eliminable when equality is treated using inference rules similar to those in Figure 3.2 [6, 75, 51]. (We shall incidentally provide our own demonstration of this fact later.)

Our right-introduction rule for equality merely states the axiom that equality is reflexive (in an arbitrary context). The left-introduction rule embodies the principle that if two terms t and u are equal then one can replace any occurrence of t in a sequent by u and vice versa. From these rules the usual symmetry and transitivity properties of equality are immediately derivable:

$$\dfrac{\dfrac{}{\vdash t = t}\,(=\mathrm{R})}{t = u \vdash u = t}\,(=\mathrm{L}) \qquad \dfrac{\dfrac{}{t_1 = t_3 \vdash t_1 = t_3}\,(\mathrm{Ax})}{t_1 = t_2, t_2 = t_3 \vdash t_1 = t_3}\,(=\mathrm{L})$$

However, it should be noted that our formulation of (=L) is not well-suited to proof search, because of the many different ways in which it can be applied to a given sequent. Given a sequent containing m occurrences of t and n occurrences of u as well as the active formula $t = u$, there are 2^{m+n} possible premises that can be obtained by applying (=L), working bottom-up (i.e. from conclusion to premises). On the other hand, our rule permits a compact treatment of equality which is of advantage in writing shorter proofs here and, as we have already observed, it also admits cut-elimination.

Our proof system, which we call LKID (the "LK" part being derived from Gentzen's original system and the "ID" part standing for "inductive definitions"), is then obtained from $\mathrm{LK_e}$ by adding proof rules for introducing inductively defined predicates on the right and left of sequents. First, for each production $\Phi_{i,r} \in \Phi$, say:

$$\dfrac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1, \ldots, j_m, i \in \{1, \ldots, n\}$$

there is a sequent calculus right introduction rule for P_i:

$$\frac{\Gamma \vdash Q_1\mathbf{u_1}(\mathbf{u}),\Delta \quad \ldots \quad \Gamma \vdash Q_h\mathbf{u_m}(\mathbf{u}),\Delta \quad \Gamma \vdash P_{j_1}\mathbf{t_1}(\mathbf{u}),\Delta \quad \ldots \quad \Gamma \vdash P_{j_m}\mathbf{t_m}(\mathbf{u}),\Delta}{\Gamma \vdash P_i\mathbf{t}(\mathbf{u}),\Delta}\,(P_i\mathrm{R}_r)$$

Example 3.1.1. The right introduction rules for the "natural number" predicate N defined in Example 2.2.5 are:

$$\frac{}{\Gamma \vdash N0,\Delta}\,(N\mathrm{R}_1) \qquad \frac{\Gamma \vdash Nt,\Delta}{\Gamma \vdash Nst,\Delta}\,(N\mathrm{R}_2)$$

Example 3.1.2. The right introduction rules for the "even/odd number" predicates E and O defined in Example 2.2.6 are:

$$\frac{}{\Gamma \vdash E0,\Delta}\,(E\mathrm{R}_1) \qquad \frac{\Gamma \vdash Ot,\Delta}{\Gamma \vdash Est,\Delta}\,(E\mathrm{R}_2) \qquad \frac{\Gamma \vdash Et,\Delta}{\Gamma \vdash Ost,\Delta}\,(O\mathrm{R}_1)$$

Example 3.1.3. The right introduction rules for the predicate R^+ (the transitive closure of a binary predicate R) defined in Example 2.2.7 are:

$$\frac{\Gamma \vdash Rtu,\Delta}{\Gamma \vdash R^+tu,\Delta}\,(R^+\mathrm{R}_1) \qquad \frac{\Gamma \vdash R^+tt',\Delta \quad \Gamma \vdash R^+t'u,\Delta}{\Gamma \vdash R^+tu,\Delta}\,(R^+\mathrm{R}_2)$$

Before giving the rules for introducing inductive predicates on the left of sequents, we first give a formal definition of what it means for two inductive predicates to have a mutual definition in Φ:

Definition 3.1.4 (Mutual dependency)**.** Define the binary relation *Prem* on the inductive predicate symbols $\{P_1,\ldots,P_n\}$ of Σ as the least relation satisfying: whenever P_i occurs in the conclusion of some production $\Phi_{i,r} \in \Phi$, and P_j occurs amongst the premises of that production, then $Prem(P_i,P_j)$ holds. Also define $Prem^*$ to be the reflexive-transitive closure of *Prem*. Then we say two predicate symbols P and Q are *mutually dependent* if both $Prem^*(P,Q)$ and $Prem^*(Q,P)$ hold.

We remark that by the definition above, mutual dependency between inductive predicate symbols is immediately reflexive, symmetric and transitive, and thus gives an equivalence relation on inductive predicate symbols.

Now to obtain an instance of the induction rule for any inductive predicate P_j, we first associate with every inductive predicate P_i a tuple $\mathbf{z_i}$ of k_i distinct variables (called *induction variables*), where k_i is the arity of P_i. Furthermore, we associate to every predicate P_i that is mutually dependent with P_j a formula (called an *induction hypothesis*) F_i, possibly containing some of the induction variables. Next, define the formula G_i for each $i \in \{1,\ldots,n\}$ by:

$$G_i = \begin{cases} F_i & \text{if } P_i \text{ and } P_j \text{ are mutually dependent} \\ P_i\mathbf{z_i} & \text{otherwise} \end{cases}$$

Structural rules:

$$\frac{}{\Gamma \vdash \Delta}\ \Gamma \cap \Delta \neq \emptyset\ \text{(Axiom)} \qquad \frac{\Gamma' \vdash \Delta'}{\Gamma \vdash \Delta}\ \Gamma' \subseteq \Gamma, \Delta' \subseteq \Delta\ \text{(Wk)}$$

$$\frac{\Gamma, F, F \vdash \Delta}{\Gamma, F \vdash \Delta}\ \text{(ContrL)} \qquad \frac{\Gamma \vdash F, F, \Delta}{\Gamma \vdash F, \Delta}\ \text{(ContrR)}$$

$$\frac{\Gamma \vdash F, \Delta \quad \Gamma, F \vdash \Delta}{\Gamma \vdash \Delta}\ \text{(Cut)} \qquad \frac{\Gamma \vdash \Delta}{\Gamma[\theta] \vdash \Delta[\theta]}\ \text{(Subst)}$$

Propositional rules:

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

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

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

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

Quantifier rules:

$$\frac{\Gamma, F[t/x] \vdash \Delta}{\Gamma, \forall x F \vdash \Delta}\ (\forall\text{L}) \qquad \frac{\Gamma \vdash F, \Delta}{\Gamma \vdash \forall x F, \Delta}\ x \notin FV(\Gamma \cup \Delta)\ (\forall\text{R})$$

$$\frac{\Gamma, F \vdash \Delta}{\Gamma, \exists x F \vdash \Delta}\ x \notin FV(\Gamma \cup \Delta)\ (\exists\text{L}) \qquad \frac{\Gamma \vdash F[t/x], \Delta}{\Gamma \vdash \exists x F, \Delta}\ (\exists\text{R})$$

Figure 3.1: Proof rules for the sequent calculus LK for classical first-order logic. The formulation here is equivalent to Gentzen's original formulation.

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

Figure 3.2: Sequent calculus proof rules for equality.

For convenience, we shall write $G_i\mathbf{t}$ for $G_i[\mathbf{t}/\mathbf{z_i}]$, where $\mathbf{t}$ is a tuple of k_i terms. Then an instance of the induction rule for P_j has the following schema:

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

where the premise $\Gamma, F_j\mathbf{t} \vdash \Delta$ is called the *major premise* of the rule, and for each production of Φ having in its conclusion a predicate P_i that is mutually dependent with P_j, say:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1, \ldots, j_m, i \in \{1, \ldots, n\}$$

there is a corresponding *minor premise*:

$$\Gamma, Q_1\mathbf{u_1}(\mathbf{x}), \ldots, Q_h\mathbf{u_h}(\mathbf{x}), G_{j_1}\mathbf{t_1}(\mathbf{x}), \ldots, G_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash F_i\mathbf{t}(\mathbf{x}), \Delta$$

where $x \notin FV(\Gamma \cup \Delta \cup \{P_j\mathbf{t}\})$ for all $x \in \mathbf{x}$.

Example 3.1.5. The induction rule for the natural number predicate N defined in Example 2.2.5 is:

$$\frac{\Gamma \vdash F0, \Delta \qquad \Gamma, Fx \vdash Fsx, \Delta \qquad \Gamma, Ft \vdash \Delta}{\Gamma, Nt \vdash \Delta}\ (\text{Ind } N)$$

where F is the induction hypothesis associated with N, which is one way of writing the familiar induction scheme for N in sequent calculus style.

(One might intuitively expect our induction rule for N to be the following:

$$\frac{\Gamma \vdash F0, \Delta \qquad \Gamma, Fx \vdash Fsx, \Delta}{\Gamma, Nt \vdash Ft, \Delta}$$

However, this formulation does not give cut-elimination, which we obtain for LKID later in the chapter, essentially because it does not allow an arbitrary induction hypothesis to be introduced in the premises, and thus forces generalisation to be performed separately from induction. Note that our induction rule (Ind N) can be seen as combining the rule above with a cut on the formula Ft.)

Example 3.1.6. The induction rule for the "even number" predicate E defined mutually with the "odd number" predicate O in Example 2.2.6 is the following:

$$\frac{\Gamma \vdash F_E0, \Delta \qquad \Gamma, F_Ex \vdash F_Osx, \Delta \qquad \Gamma, F_Ox \vdash F_Esx, \Delta \qquad \Gamma, F_Et \vdash \Delta}{\Gamma, Et \vdash \Delta}\ (\text{Ind } E)$$

where F_E and F_O are the induction hypotheses associated with E and O respectively. (Note that E,O are mutually dependent, c.f. Definition 3.1.4.)

Example 3.1.7. The induction rule for the predicate R^+, the transitive closure of the binary predicate R, defined in Example 2.2.7 is the following:

$$\frac{\Gamma, Rxy \vdash Fxy, \Delta \qquad \Gamma, Fxy, Fyz \vdash Fxz, \Delta \qquad \Gamma, Ftu \vdash \Delta}{\Gamma, R^+tu \vdash \Delta}\ (\text{Ind } R^+)$$

where F is the induction hypothesis associated with R^+. Note that, even if R is an inductive predicate, R and R^+ are not mutually dependent (because $Prem^*(R,R^+)$ does not hold, although $Prem^*(R^+,R)$ does).

In [44], Martin-Löf uses the following auxiliary definition of a predicate symbol being "linked" with another in deciding which productions have corresponding minor premises in the induction rule for a predicate:

> "First, every predicate is linked with itself. Second, if P occurs in the conclusion of an ordinary production
>
> $$\frac{Qq(x)\ldots Rr(x)}{Pp(x)}$$
>
> then P is linked with every predicate symbol which is linked with one of $Q,\ldots,R$."

The induction rule for P then has a minor premise for every production which in its conclusion has a predicate symbol which is linked with P. However, this definition can lead to redundant minor premises appearing in induction rules. For example, consider the usual productions for the natural number predicate N together with a single production for a second predicate Q:

$$\frac{}{N0} \qquad \frac{Nx}{Nsx} \qquad \frac{Nx}{Qx}$$

According to the definition above, Q is linked with N by virtue of the third production, and the induction rule for N then has a minor premise for every production with Q in the conclusion:

$$\frac{\Gamma \vdash F0,\Delta \quad \Gamma,Fx \vdash Fsx,\Delta \quad \Gamma,Fx \vdash Gx,\Delta \quad \Gamma,Ft \vdash \Delta}{\Gamma,Nt \vdash \Delta}\ (\text{Ind } N)$$

The third minor premise corresponding to the production for Q is clearly redundant in the rule above. Our motivation for using Definition 3.1.4 to generate minor premises, as opposed to Martin-Löf's definition, is to avoid having redundant minor premises such as the one above appear in our induction rules.

As we remarked previously, we treat equality as a primitive logical symbol in FOL_{ID}, with corresponding left- and right-introduction LKID proof rules (given in Figure 3.2). However, as demonstrated in [44], it is also possible to formulate equality as an inductively defined predicate. Assuming that $=$ is a binary inductive predicate symbol in our language and writing as usual $x=y$ for $=(x,y)$, consider the following production for $=$:

$$\frac{}{x=x}$$

If we stipulate that $=$ may not appear in the conclusion of any other productions of our inductive definition set, then the inductive definition of $=$ gives the following introduction rules in LKID:

$$\frac{\Gamma \vdash Fxx,\Delta \quad \Gamma,Ftu \vdash \Delta}{\Gamma,t=u \vdash \Delta}\ (=\text{L}) \qquad \frac{}{\Gamma \vdash t=t,\Delta}\ (=\text{R})$$

where $x \notin FV(\Gamma \cup \Delta \cup \{t = u\})$. So we obtain an identical right-introduction rule, but a different left-introduction rule for $=$ involving an arbitrary induction hypothesis F in the premises. As this has obvious implications for proof search control in the system, we prefer to take $=$ as primitive.

An LKID proof of a sequent $\Gamma \vdash \Delta$ is, as usual, a finite tree labelled with sequents and constructed according to the proof rules of LKID. For now, we just give some examples by way of illustration, and defer formal definitions and considerations of soundness until the next section. Because it is used so frequently, and to make our written proofs smaller, we use a double line between premises and conclusion to indicate (possibly multiple) uses of the weakening rule (Wk). For example, the following would indicate a use of the rule ($\wedge$R) and some weakenings:

$$\dfrac{\Gamma' \vdash F \quad \Gamma \vdash G, \Delta}{\Gamma, \Gamma' \vdash F \wedge G, \Delta}\ (\wedge\mathrm{R})$$

Example 3.1.8. Let N, E and O be the "natural number", "even" and "odd" predicates given by the productions in Examples 2.2.5 and 2.2.6 (and whose corresponding rules in LKID are given in the examples above). Then the following is an LKID proof of $\vdash \forall x(Nx \to Ex \vee Ox)$:

$$\dfrac{\dfrac{\dfrac{}{\vdash E0, O0}\ (E\mathrm{R}_1)}{\vdash E0 \vee O0}\ (\vee\mathrm{R}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{}{Ey \vdash Ey}\ (\mathrm{Ax}) \quad \dfrac{}{Oy \vdash Oy}\ (\mathrm{Ax})}{Ey \vee Oy \vdash Oy, Ey}\ (\vee\mathrm{L})}{Ey \vee Oy \vdash Oy, Osy}\ (O\mathrm{R}_1)}{Ey \vee Oy \vdash Esy, Osy}\ (E\mathrm{R}_2)}{Ey \vee Oy \vdash Esy \vee Osy}\ (\vee\mathrm{R}) \qquad \dfrac{}{Ex \vee Ox \vdash Ex \vee Ox}\ (\mathrm{Ax})}{\dfrac{\dfrac{Nx \vdash Ex \vee Ox}{\vdash Nx \to Ex \vee Ox}\ (\to\mathrm{R})}{\vdash \forall x(Nx \to Ex \vee Ox)}\ (\forall\mathrm{R})}\ (\mathrm{Ind}\ N)$$

Note that at the application of induction on Nx, we associate with N the induction variable z (say) and the induction hypothesis $Ez \vee Oz$.

Example 3.1.9. Let R be an ordinary binary predicate and let R^+ be the "transitive closure" predicate given by the productions in Example 2.2.7 (and whose corresponding rules in LKID are given in examples in Section 3.1 above). Then the following is an LKID proof of

$\vdash (\forall x \forall y(Rxy \to Ryx)) \to (\forall x \forall y(R^+xy \to R^+yx))$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{Rxy \vdash Rxy}(\text{Ax}) \quad \dfrac{\dfrac{}{Ryx \vdash Ryx}(\text{Ax})}{Ryx \vdash R^+yx}(R^+\text{R}_1)}{Rxy \to Ryx, Rxy \vdash R^+yx}(\to\text{L})}{\forall y(Rxy \to Ryx), Rxy \vdash R^+yx}(\forall\text{L})}{\forall x \forall y(Rxy \to Ryx), Rxy \vdash R^+yx}(\forall\text{L}) \quad \dfrac{\dfrac{}{R^+zy \vdash R^+zy}(\text{Ax}) \quad \dfrac{}{R^+yx \vdash R^+yx}(\text{Ax})}{R^+yx, R^+zy \vdash R^+zx}(R^+\text{R}_2) \quad \dfrac{}{R^+yx \vdash R^+yx}(\text{Ax})}{\forall x \forall y(Rxy \to Ryx), R^+xy \vdash R^+yx}(\text{Ind } R^+)$$

$$\dfrac{\dfrac{\dfrac{\dfrac{\forall x \forall y(Rxy \to Ryx), R^+xy \vdash R^+yx}{\forall x \forall y(Rxy \to Ryx) \vdash R^+xy \to R^+yx}(\to\text{R})}{\forall x \forall y(Rxy \to Ryx) \vdash \forall y(R^+xy \to R^+yx)}(\forall\text{R})}{\forall x \forall y(Rxy \to Ryx) \vdash \forall x \forall y(R^+xy \to R^+yx)}(\forall\text{R})}{\vdash (\forall x \forall y(Rxy \to Ryx)) \to (\forall x \forall y(R^+xy \to R^+yx))}(\to\text{R})$$

Note that at the application of induction on R^+xy, we associate with R^+ the induction variables z_1, z_2 and the induction hypothesis $R^+z_2z_1$.

3.2 Henkin soundness of LKID

Definition 3.2.1 (Henkin validity of sequents)**.** Let $(M, \mathcal{H})$ be a Henkin model for (Σ, Φ). Then a sequent $\Gamma \vdash \Delta$ is said to be *true in* $(M, \mathcal{H})$ if for all environments ρ, we have $M \models_\rho J$ for all $J \in \Gamma$ implies $M \models_\rho K$ for some $K \in \Delta$. $\Gamma \vdash \Delta$ is said to be *Henkin valid* if it is true in all Henkin models for (Σ, Φ).

We now prove that the proof rules of LKID are *locally sound* with respect to the Henkin semantics of FOL_{ID}, i.e., they preserve Henkin validity from premises to conclusion:

Lemma 3.2.2 (Local soundness of LKID)**.** *Let $(M, \mathcal{H})$ be a Henkin model for (Σ, Φ). If all of the premises of a rule of LKID are true in $(M, \mathcal{H})$, then the conclusion of the rule is also true in $(M, \mathcal{H})$.*

Proof. Of course, we just need to check that each rule of LKID has the desired property. The main interesting cases are the rules for inductive predicates. Although the proof is in the remaining cases fairly standard, we nevertheless treat them comprehensively here for the sake of completeness.

Case (Ax):

$$\dfrac{}{\Gamma \vdash \Delta}\ \Gamma \cap \Delta \neq \emptyset\ (\text{Axiom})$$

Let ρ be an environment and suppose $M \models_\rho J$ for all $J \in \Gamma$. Since $\Gamma \cap \Delta \neq \emptyset$, there is a $K \in \Delta$ such that $K \in \Gamma$, i.e. $M \models_\rho K$ as required.

Case (Wk):

$$\frac{\Gamma' \vdash \Delta'}{\Gamma \vdash \Delta}\ \Gamma' \subseteq \Gamma, \Delta' \subseteq \Delta\ \text{(Wk)}$$

Let ρ be an environment and suppose $M \models_\rho J$ for all $J \in \Gamma$. Since $\Gamma' \subseteq \Gamma$, $M \models_\rho J$ for all $J \in \Gamma'$, and since $\Gamma' \vdash \Delta'$ is true in $(M, \mathcal{H})$ by assumption, there is a $K \in \Delta'$ such that $M \models_\rho K$. Then, since $\Delta' \subseteq \Delta$, there is a $K \in \Delta$ such that $M \models_\rho K$ as required.

Cases (ContrL), (ContrR): Immediate.

Case (Cut):

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

Let ρ be an environment and suppose $M \models_\rho J$ for all $J \in \Gamma$. Since the first premise $\Gamma \vdash F, \Delta$ is true in $(M, \mathcal{H})$, there is a $K \in \{F\} \cup \Delta$ such that $M \models_\rho K$. If $K \in \Delta$, then we are done; otherwise $K = F$ and since the second premise $\Gamma, F \vdash \Delta$ is true in $(M, \mathcal{H})$, there is a $K' \in \Delta$ such that $M \models_\rho K'$, as required.

Case (Subst):

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

Let ρ be an environment and suppose $M \models_\rho J$ for all $J \in \Gamma[\theta]$, i.e. $M \models_\rho J[\theta]$ for all $J \in \Gamma$. By Lemma 2.1.9, part 2, it therefore holds that $M \models_{\rho\circ\theta} J$ for all $J \in \Gamma$. Since the premise $\Gamma \vdash \Delta$ is true in $(M, \mathcal{H})$, we have $M \models_{\rho\circ\theta} K$ for some $K \in \Delta$, and applying part 2 of Lemma 2.1.9 again, we thus have $M \models_\rho K[\theta]$ for some $K \in \Delta$. Hence $M \models_\rho K$ for some $K \in \Delta[\theta]$ as required.

Case (=L):

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

Suppose for contradiction that the premise is true and the conclusion false in $(M, \mathcal{H})$. Since the conclusion is false in $(M, \mathcal{H})$, there is an environment ρ such that $M \models_\rho J$ for all $J \in \Gamma[t/x, u/y] \cup \{t = u\}$ and $M \not\models_\rho K$ for all $K \in \Delta[t/x, u/y]$. That is, we have $\rho(t) = \rho(u)$, and $M \models_\rho J[t/x, u/y]$ for all $J \in \Gamma$ and $M \not\models_\rho K[t/x, u/y]$ for all $K \in \Delta$. By Lemma 2.1.9, part 2, it therefore holds that $M \models_{\rho[x \mapsto \rho(t), y \mapsto \rho(u)]} J$ for all $J \in \Gamma$ and $M \not\models_{\rho[x \mapsto \rho(t), y \mapsto \rho(u)]} K$ for all $K \in \Delta$. But since $\rho(t) = \rho(u)$, we then have $M \models_{\rho[x \mapsto \rho(u), y \mapsto \rho(t)]} J$ for all $J \in \Gamma$ and $M \not\models_{\rho[x \mapsto \rho(u), y \mapsto \rho(t)]} K$ for all $K \in \Delta$. So for all $J \in \Gamma$ and for all $K \in \Delta$, $M \models_\rho J[u/x, t/y]$ and $M \not\models_\rho K[u/x, t/y]$, which contradicts our assumption that the premise $\Gamma[u/x, t/y] \vdash \Delta[u/x, t/y]$ is true in $(M, \mathcal{H})$. So the conclusion of the rule is true in $(M, \mathcal{H})$ as required.

Case (=R):

$$\frac{}{\Gamma \vdash t = t, \Delta}\ (=\mathrm{R})$$

Let ρ be an environment. Since trivially $\rho(t) = \rho(t)$, we have $M \models_\rho t = t$, which shows that the conclusion of the rule is true in $(M, \mathcal{H})$.

Case (∧L):

$$\frac{\Gamma, F, G \vdash \Delta}{\Gamma, F \wedge G \vdash \Delta}\ (\wedge\mathrm{L})$$

Let ρ be an environment and suppose $M \models_\rho J$ for all $J \in \Gamma \cup \{F \wedge G\}$. Then $M \models_\rho J$ for all $J \in \Gamma \cup \{F, G\}$, and since the premise $\Gamma, F, G \vdash \Delta$ is true in $(M, \mathcal{H})$ by assumption, there is a $K \in \Delta$ such that $M \models_\rho K$, which completes this case.

Case (∧R):

$$\frac{\Gamma \vdash F, \Delta \qquad \Gamma \vdash G, \Delta}{\Gamma \vdash F \wedge G, \Delta}\ (\wedge\mathrm{R})$$

Let ρ be an environment and suppose $M \models_\rho J$ for all $J \in \Gamma$. As the premises $\Gamma \vdash F, \Delta$ and $\Gamma \vdash G, \Delta$ are true in $(M, \mathcal{H})$ by assumption, there exist $K \in \{F\} \cup \Delta$ and $K' \in \{G\} \cup \Delta$ such that $M \models_\rho K$ and $M \models_\rho K'$. If $K \in \Delta$ or $K' \in \Delta$ then we are done; otherwise $K = F$ and $K' = G$, whence we have $M \models_\rho F \wedge G$ and we are likewise done.

Cases (¬L), (¬R), (∨R), (∨L), (→L), (→R): Similar to the cases (∧L) and (∧R) above.

Case (∃L):

$$\frac{\Gamma, F \vdash \Delta}{\Gamma, \exists x F \vdash \Delta}\ x \notin FV(\Gamma \cup \Delta)\ (\exists\mathrm{L})$$

Let ρ be an environment and suppose $M \models_\rho J$ for all $J \in \Gamma \cup \{\exists x F\}$, where $x \notin FV(\Gamma \cup \Delta)$. So $M \models_\rho J$ for all $J \in \Gamma$ and for some $d \in D$ (where D is the domain of M), $M \models_{\rho[x \mapsto d]} F$. Since $x \notin FV(\Gamma)$, by part 1 of Lemma 2.1.9, it also holds that $M \models_{\rho[x \mapsto d]} J$ for all $J \in \Gamma$. Now as the premise $\Gamma, F \vdash \Delta$ is true in $(M, \mathcal{H})$, there is a $K \in \Delta$ such that $M \models_{\rho[x \mapsto d]} K$, and since $x \notin FV(\Delta)$ we have, again by Lemma 2.1.9, that $M \models_\rho K$ as required.

Case (∃R):

$$\frac{\Gamma \vdash F[t/x], \Delta}{\Gamma \vdash \exists x F, \Delta}\ (\exists\mathrm{R})$$

Let ρ be an environment and suppose $M \models_\rho J$ for all $J \in \Gamma$. Since the premise $\Gamma \vdash F[t/x], \Delta$ is true in $(M, \mathcal{H})$, there is a $K \in \{F[t/x]\} \cup \Delta$ such that $M \models_\rho K$. If $K \in \Delta$ we are immediately finished; otherwise $M \models_\rho F[t/x]$ and by part 2 of Lemma 2.1.9, we then have $M \models_{\rho[x \mapsto \rho(t)]} F$, i.e. $M \models \exists x F$, which completes this case.

Cases ($\forall$L), ($\forall$R): Similar to the cases ($\exists$R) and ($\exists$L) above.

Case (P_iR$_r$):

$$\frac{\Gamma \vdash Q_1\mathbf{u_1}(\mathbf{u}),\Delta \quad \ldots \quad \Gamma \vdash Q_h\mathbf{u_m}(\mathbf{u}),\Delta \quad \Gamma \vdash P_{j_1}\mathbf{t_1}(\mathbf{u}),\Delta \quad \ldots \quad \Gamma \vdash P_{j_m}\mathbf{t_m}(\mathbf{u}),\Delta}{\Gamma \vdash P_i\mathbf{t}(\mathbf{u}),\Delta}\,(P_i\mathrm{R}_r)$$

where there is a production $\Phi_{i,r} \in \Phi$:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1,\ldots,j_m,i \in \{1,\ldots,n\}$$

Let ρ be an environment and suppose $M \models_\rho J$ for all $J \in \Gamma$. Since all of the premises are true in $(M,\mathcal{H})$, it follows that either there is a $K \in \Delta$ such that $M \models_\rho K$, in which case we are done, or:

$$\begin{array}{ll} & M \models_\rho Q_1\mathbf{u_1}(\mathbf{u}),\ldots,M \models_\rho Q_h\mathbf{u_m}(\mathbf{u}),M \models_\rho P_{j_1}\mathbf{t_1}(\mathbf{u}),\ldots,M \models_\rho P_{j_m}\mathbf{t_m}(\mathbf{u}) \\ \text{i.e.} & Q_1^M(\rho(\mathbf{u_1}(\mathbf{u}))),\ldots,Q_h^M(\rho(\mathbf{u_h}(\mathbf{u}))),\rho(\mathbf{t_1}(\mathbf{u})) \in \pi^n_{j_1}(\mu_{\mathcal{H}}.\varphi_\Phi),\ldots,\rho(\mathbf{t_m}(\mathbf{u})) \in \pi^n_{j_m}(\mu_{\mathcal{H}}.\varphi_\Phi) \\ \text{i.e.} & Q_1^M\mathbf{u_1^M}(\rho(\mathbf{u})),\ldots,Q_h^M\mathbf{u_h^M}(\rho(\mathbf{u})),\mathbf{t_1^M}(\rho(\mathbf{u})) \in \pi^n_{j_1}(\mu_{\mathcal{H}}.\varphi_\Phi),\ldots,\mathbf{t_m^M}(\rho(\mathbf{u})) \in \pi^n_{j_m}(\mu_{\mathcal{H}}.\varphi_\Phi) \end{array}$$

Now we have by the construction of φ_Φ (c.f. Definition 2.2.3):

$$\varphi_{i,r}(X_1,\ldots,X_n) = \{\mathbf{t}^M(\mathbf{x}) \mid Q_1^M\mathbf{u_1}^M(\mathbf{x}),\ldots,Q_h^M\mathbf{u_h}^M(\mathbf{x}),\mathbf{t_1}^M(\mathbf{x}) \in X_{j_1},\ldots,\mathbf{t_m}^M(\mathbf{x}) \in X_{j_m}\}$$

It follows by the above that $\mathbf{t}^M(\rho(\mathbf{u})) \in \varphi_{i,r}(\mu_{\mathcal{H}}.\varphi_\Phi) \subseteq \varphi_i(\mu_{\mathcal{H}}.\varphi_\Phi)$, i.e. $\mathbf{t}^M(\rho(\mathbf{u})) \in \pi^n_i(\varphi_\Phi(\mu_{\mathcal{H}}.\varphi_\Phi))$. Since $\mu_{\mathcal{H}}.\varphi_\Phi$ is a fixed point of φ_Φ, it then holds that $\mathbf{t}^M(\rho(\mathbf{u})) = \rho(\mathbf{t}(\mathbf{u})) \in \pi^n_i(\mu_{\mathcal{H}}.\varphi_\Phi)$, i.e. $M \models_\rho P_i\mathbf{t}(\mathbf{u})$, which shows the conclusion of (P_iR$_r$) to be true in $(M,\mathcal{H})$.

Case (Ind P_j):

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

First, let ρ be an environment and suppose $M \models_\rho J$ for all $J \in \Gamma$ and $M \models_\rho P_j\mathbf{t}$, i.e. $\rho(\mathbf{t}) \in \pi^n_j(\mu_{\mathcal{H}}.\varphi_\Phi)$. Suppose for contradiction that $M \not\models_\rho K$ for all $K \in \Delta$.

Now, for each $i \in \{1,\ldots,n\}$, let $\mathbf{z_i}$ and G_i be, respectively, the induction variables and the formula associated with P_i in the description of the induction rule (Ind P_j) in section 3.1 above. We write k_i for the arity of P_i. Next, let $\mathbf{x}$ be the fresh variables appearing in the minor premises and let $\mathbf{e}$ be a tuple of arbitrary elements of D such that $|\mathbf{x}| = |\mathbf{e}|$. Define an environment ρ' by $\rho' = \rho[\mathbf{x} \mapsto \mathbf{e}]$, and note that since $x \notin FV(\Gamma \cup \Delta)$ for all $x \in \mathbf{x}$ by the rule side condition, it holds by part 1 of Lemma 2.1.9 that $M \models_{\rho'} J$ for all $J \in \Gamma$ and $M \not\models_{\rho'} K$ for all $K \in \Delta$.

Now define an n-tuple of sets $(Y_1,\ldots,Y_n) \in \mathrm{Pow}(D^{k_1}) \times \ldots \times \mathrm{Pow}(D^{k_n})$ by:

$$Y_i = \begin{cases} \{(d_1,\ldots,d_{k_i}) \mid M \models_{\rho'[z_1 \mapsto d_1,\ldots,z_{k_i} \mapsto d_{k_i}]} G_i\} & \text{if } \mathit{Prem}^*(P_j,P_i) \\ D^{k_i} & \text{otherwise} \end{cases}$$

for each $i \in \{1,\ldots,n\}$, where $\mathbf{z_i} = (z_1,\ldots,z_{k_i})$ are the induction variables for P_i. We assert the following:

CLAIM: $(Y_1,\ldots,Y_n)$ is a prefixed $\mathcal{H}$-point of φ_Φ.

Then, since $\mu_{\mathcal{H}}.\varphi_\Phi$ is the least prefixed $\mathcal{H}$-point of φ_Φ, it holds that $\pi^n_j(\mu_{\mathcal{H}}.\varphi_\Phi) \subseteq Y_j$. As the major premise $\Gamma, F_j\mathbf{t} \vdash \Delta$ is true in $(M,\mathcal{H})$, and as $M \models_{\rho'} J$ for all $J \in \Gamma$ but $M \not\models_{\rho'} K$ for all $K \in \Delta$, we must have $M \not\models_{\rho'} F_j\mathbf{t}$. By part 2 of Lemma 2.1.9 we then have $M \not\models_{\rho'[\mathbf{z_j}\mapsto\rho'(\mathbf{t})]} F_j$, i.e. $\rho'(\mathbf{t}) \notin Y_j$, so also $\rho'(\mathbf{t}) \notin \pi^n_j(\mu_{\mathcal{H}}.\varphi_\Phi)$. As $x \notin FV(P_j\mathbf{t})$, i.e. $x \notin Var(\mathbf{t})$ for all $x \in \mathbf{x}$ by the rule side condition, we must have $\rho'(\mathbf{t}) = \rho(\mathbf{t})$ by part 1 of Lemma 2.1.5. But then we have $\rho(\mathbf{t}) \notin \pi^n_j(\mu_{\mathcal{H}}.\varphi_\Phi)$, which contradicts our original assumption, so $M \models_\rho K$ for some $K \in \Delta$ as required.

To finish the proof, it suffices to prove the claim above that $(Y_1,\ldots,Y_n)$ is a prefixed $\mathcal{H}$-point of φ_Φ. First, writing $\mathcal{H} = \{H_k \mid k \in \mathbb{N}\}$, observe that by Proposition 2.3.3, $Y_i \in H_{k_i}$ for each $i \in \{1,\ldots,n\}$, i.e. $(Y_1,\ldots,Y_n)$ is a $\mathcal{H}$-point. (Note that the proposition does indeed show that $D^{k_i} \in H_{k_i}$, because D^{k_i} can be expressed as e.g. $\{(d_1,\ldots,d_{k_i}) \mid M \models_{\rho'[z_1\mapsto d_1,\ldots,z_{k_i}\mapsto d_{k_i}]} t = t\}$, where t is any term.) It remains to show that $(Y_1,\ldots,Y_n)$ is a prefixed point of φ_Φ, i.e. that $\varphi_i(Y_1,\ldots,Y_n) \subseteq Y_i$ for each $i \in \{1,\ldots,n\}$. We argue by cases on i as follows:

1. $\neg Prem^*(P_j,P_i)$ holds. It is then trivial that $\varphi_i(Y_1,\ldots,Y_n) \subseteq D^{k_i} = Y_i$.

2. $Prem^*(P_j,P_i)$ and $\neg Prem^*(P_i,P_j)$ hold. As P_j and P_i are thus not mutually dependent, $G_i = P_i\mathbf{z_i}$ and we have:

$$\begin{aligned} Y_i &= \{(d_1,\ldots,d_{k_i}) \mid M \models_{\rho'[z_1\mapsto d_1,\ldots,z_{k_i}\mapsto d_{k_i}]} P_i\mathbf{z_i}\} \\ &= \{(d_1,\ldots,d_{k_i}) \mid M \models_{\rho'[z_1\mapsto d_1,\ldots,z_{k_i}\mapsto d_{k_i}]} P_i(z_1,\ldots,z_{k_i})\} \\ &= \{(d_1,\ldots,d_{k_i}) \mid \rho'[z_1 \mapsto d_1,\ldots,z_{k_i} \mapsto d_{k_i}](z_1,\ldots,z_{k_i}) \in \pi^n_i(\mu_{\mathcal{H}}.\varphi_\Phi)\} \\ &= \{(d_1,\ldots,d_{k_i}) \mid (d_1,\ldots,d_{k_i}) \in \pi^n_i(\mu_{\mathcal{H}}.\varphi_\Phi)\} \\ &= \pi^n_i(\mu_{\mathcal{H}}.\varphi_\Phi) \end{aligned}$$

It suffices to show that $\varphi_{i,r}(Y_1,\ldots,Y_n) \subseteq Y_i = \pi^n_i(\mu_{\mathcal{H}}.\varphi_\Phi)$ for an arbitrary production $\Phi_{i,r}$:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1,\ldots,j_m,i \in \{1,\ldots,n\}$$

We are thus required to show:

$$\{\mathbf{t}^M(\mathbf{d}) \mid Q^M_1\mathbf{u}^M_1(\mathbf{d}),\ldots,Q^M_h\mathbf{u}^M_\mathbf{h}(\mathbf{d}),\mathbf{t}^M_\mathbf{1}(\mathbf{d}) \in Y_{j_1},\ldots,\mathbf{t}^M_\mathbf{m}(\mathbf{d}) \in Y_{j_m}\} \subseteq \pi^n_i(\mu_{\mathcal{H}}.\varphi_\Phi)$$

Note that for each of the inductive predicates P_{j_k} appearing in the premises of the production $\Phi_{i,r}$, $Prem^*(P_j,P_{j_k})$ holds (because $Prem^*(P_j,P_i)$ and $Prem(P_i,P_{j_k})$ hold), and $\neg Prem^*(P_{j_k},P_j)$ holds (because otherwise $Prem^*(P_i,P_j)$ would hold, which contradicts

the case assumption), and we therefore have $Y_{j_k} = \pi^n_{j_k}(\mu_{\mathcal{H}}.\varphi_\Phi)$ by the argument above. We can therefore rewrite the statement we need to prove as:

$$\{\mathbf{t}^M(\mathbf{d}) \mid Q_1^M\mathbf{u_1}^M(\mathbf{d}),\ldots,Q_h^M\mathbf{u_h}^M(\mathbf{d}),\\ \mathbf{t_1}^M(\mathbf{d}) \in \pi^n_{j_1}(\mu_{\mathcal{H}}.\varphi_\Phi),\ldots,\mathbf{t_m}^M(\mathbf{d}) \in \pi^n_{j_m}(\mu_{\mathcal{H}}.\varphi_\Phi)\} \subseteq \pi^n_i(\mu_{\mathcal{H}}.\varphi_\Phi)$$

i.e., $\varphi_{i,r}(\mu_{\mathcal{H}}.\varphi_\Phi) \subseteq \pi^n_i(\mu_{\mathcal{H}}.\varphi_\Phi)$, which is true since $\mu_{\mathcal{H}}.\varphi_\Phi$ is a prefixed $\mathcal{H}$-point of φ_Φ. This completes the case.

3. $Prem^*(P_j,P_i)$ and $Prem^*(P_i,P_j)$ both hold, i.e. P_i and P_j are mutually dependent. As in the previous case, we require to show $\varphi_{i,r}(Y_1,\ldots,Y_n) \subseteq Y_i$, i.e.

$$\{\mathbf{t}^M(\mathbf{d}) \mid Q_1^M\mathbf{u_1}^M(\mathbf{d}),\ldots,Q_h^M\mathbf{u_h}^M(\mathbf{d}),\mathbf{t_1}^M(\mathbf{d}) \in Y_{j_1},\ldots,\mathbf{t_m}^M(\mathbf{d}) \in Y_{j_m}\} \subseteq Y_i$$

As P_i and P_j are mutually dependent, there is a minor premise of the instance of (Ind P_j):

$$\Gamma, Q_1\mathbf{u_1}(\mathbf{x}),\ldots,Q_h\mathbf{u_h}(\mathbf{x}),G_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,G_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash F_i\mathbf{t}(\mathbf{x}),\Delta \quad (j_1,\ldots,j_m \in \{1,\ldots,n\})$$

As each minor premise is true in $(M,\mathcal{H})$ by assumption and we have $M \models_{\rho'} J$ for all $J \in \Gamma$ and $M \not\models_{\rho'} K$ for all $K \in \Delta$, it holds that:

$$M \models_{\rho'} Q_1\mathbf{u_1}(\mathbf{x}),\ldots,M \models_{\rho'} Q_h\mathbf{u_h}(\mathbf{x}),M \models_{\rho'} G_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,M \models_{\rho'} G_{j_m}\mathbf{t_1}(\mathbf{x})\\ \text{implies } M \models_{\rho'} F_i\mathbf{t}(\mathbf{x})$$

and by applying the semantic definitions and part 2 of Lemma 2.1.9 we obtain:

$$Q_1\mathbf{u_1}^M(\rho'(\mathbf{x})),\ldots,Q_h\mathbf{u_h}^M(\rho'(\mathbf{x})),M \models_{\rho'[\mathbf{z_{j_1}}\mapsto\mathbf{t_1^M}(\rho'(\mathbf{x}))]} G_{j_1},\ldots,M \models_{\rho'[\mathbf{z_{jm}}\mapsto\mathbf{t_m^M}(\rho'(\mathbf{x}))]} G_{j_m}\\ \text{implies } M \models_{\rho'[\mathbf{z_i}\mapsto\mathbf{t^M}(\rho'(\mathbf{x}))]} F_i$$

Note that for each inductive predicate P_{j_k} appearing in the premises of the production in question, $Prem^*(P_j,P_{j_k})$ holds (since $Prem^*(P_j,P_i)$ and $Prem(P_i,P_{j_k})$ hold). Recalling that $\rho'(\mathbf{x}) = \mathbf{e}$, we thus have:

$$Q_1\mathbf{u_1}^M(\mathbf{e}),\ldots,Q_h\mathbf{u_h}^M(\mathbf{e}),\mathbf{t_1}^M(\mathbf{e}) \in Y_{j_1},\ldots,\mathbf{t_m}^M(\mathbf{e}) \in Y_{j_m} \text{ implies } \mathbf{t}^M(\mathbf{e}) \in Y_i$$

which, as $\mathbf{e}$ was arbitrarily chosen, completes the case and thus the proof.

□

We now proceed to give a fully formal definition of an LKID *proof* in line with our previous informal definition, i.e., as finite trees labelled with sequents and constructed according to the proof rules of LKID. We define proofs here with more formality than is usual in order to facilitate a precise comparison first with *infinite proofs* in Chapter 4 and then with *cyclic proofs* in Chapter 5. Our formal definition of proof will also be useful when analysing proof structure in detail in Chapters 6 and 7. We first define *derivation graphs*, which capture formally what it means to be a graph constructed according to the rules of some proof system:

Definition 3.2.3 (Derivation graph)**.** Let *Seqs* denote the set of all well-formed sequents and *Rules* denote the set of rules of some proof system S. Then an *S derivation graph* is given by (V,s,r,p), where:

- V is a set, $s : V \to Seqs$ is a total function, and $r : V \to Rules$ and $p : \mathbb{N} \times V \to V$ are partial functions (we write $p_j(v)$ for $p(j,v)$);
- for all $v \in V$, $p_j(v)$ is defined just in case $r(v)$ is a rule with m premises, $1 \leq j \leq m$ and:

$$\frac{s(p_1(v)) \quad \ldots \quad s(p_m(v))}{s(v)}\,(\mathrm{r(v)})$$

  is an instance of rule $r(v)$.

A derivation graph G can be seen as a conventional graph whose vertex set is V and whose edge set is $E = \{(v, p_j(v)) \mid v \in V \text{ and } p_j(v) \text{ is defined}\}$.

A *path* in a derivation graph is a (possibly infinite) sequence $v_0 j_0 v_1 j_1 v_2 j_2 \ldots$ such that for each $i \geq 0$, $v_{i+1} = p_{j_i}(v_i)$. (We often write paths simply as $v_0 v_1 v_2 \ldots$)

Next, we define *derivation trees*, which bear the same relationship to derivation graphs as ordinary trees do to ordinary graphs:

Definition 3.2.4 (Derivation tree)**.** An S derivation graph $\mathcal{D} = (V,s,r,p)$ is an *S derivation tree* if there is a distinguished node $root(\mathcal{D}) \in V$ such that for all $v \in V$, there is a unique path in $\mathcal{D}$ from $root(\mathcal{D})$ to v. The sequent $s(root(\mathcal{D}))$ is called the *endsequent* of the tree. A path in $\mathcal{D}$ is said to be *rooted* iff its first vertex is $root(\mathcal{D})$. $\mathcal{D}$ is said to be an *infinite tree* iff V is infinite.

For $v, v' \in V$, v' is an *ancestor* of v if v' is on the unique path from $root(\mathcal{D})$ to v in $\mathcal{D}$, and is a *strict ancestor* of v if v' is an ancestor of v and $v' \neq v$. v is said to be a *descendant* of v' just in case v' is a strict ancestor of v.

We shall sometimes drop the prefix S from the term "S derivation graph/tree" when the proof system S is clear from the context. It shall also be useful to have an easy way to distinguish between those derivation trees that are "finished" in the sense that every path in the tree ends in an axiom instance, and those that are "unfinished", and for this purpose we introduce the notion of *bud nodes* in a derivation tree[1]:

Definition 3.2.5 (Bud nodes)**.** Let $\mathcal{D} = (V,s,r,p)$ be a derivation tree. A *bud node* of $\mathcal{D}$ is a vertex $B \in V$ such that $r(B)$ is undefined, i.e. B is not the conclusion of any proof rule instance in $\mathcal{D}$. (Notice that if B is a bud node then $p_j(B)$ must be undefined for all j.) We write $\mathrm{Bud}(\mathcal{D})$ to denote the set of all bud nodes occurring in $\mathcal{D}$.

[1] Thanks to René Vestergaard for suggesting the terminology "bud node", which is intended to be suggestive of a part of the tree not yet expanded.

We remark that buds differ from the usual notion of a *leaf* of a derivation tree; leaves are generally considered to be the conclusions of axiom instances.

We are now in a position to define formally the meaning of a *proof* in our system LKID (which coincides exactly with the usual meaning of proof in a sequent calculus proof system):

Definition 3.2.6 (LKID proof)**.** An *LKID proof* of a sequent $\Gamma \vdash \Delta$ is a finite LKID derivation tree $\mathcal{D}$ whose endsequent is $\Gamma \vdash \Delta$ and such that $\mathrm{Bud}(\mathcal{D}) = \emptyset$ (i.e. every sequent in the proof tree is the conclusion of some proof rule application, so the derivation is "finished").

When we write actual proofs in LKID, we use the standard representation for sequent calculus proofs as per our earlier examples, from which suitable definitions of (V, s, r, p) can obviously be recovered if required.

It follows from the local soundness of our proof rules that the system LKID is sound with respect to the Henkin semantics of $\mathrm{FOL_{ID}}$, i.e. that any sequent which is provable in LKID is Henkin valid. The proof is fairly trivial, but we spell out the details anyway for the sake of completeness:

Definition 3.2.7 (Height)**.** If $\mathcal{D} = (V, s, r, p)$ is a derivation tree, then the *height (in $\mathcal{D}$)* of a node $v \in V$ is defined to be the length of the unique path in $\mathcal{D}$ from $root(\mathcal{D})$ to v. The height of a derivation tree is defined to be the maximum of the heights of the nodes occurring within it.

Proposition 3.2.8 (Soundness of LKID)**.** *If there is an LKID proof of $\Gamma \vdash \Delta$ then $\Gamma \vdash \Delta$ is Henkin valid (i.e. true in all Henkin models).*

Proof. By induction on the height h of the LKID proof of $\Gamma \vdash \Delta$ (which is a derivation tree). Consider the last rule used to derive $\Gamma \vdash \Delta$, say:

$$\frac{\begin{array}{ccc}\vdots & & \vdots \\ \Gamma_1 \vdash \Delta_1 & \ldots & \Gamma_n \vdash \Delta_n\end{array}}{\Gamma \vdash \Delta}\,(\mathrm{R})$$

where (R) is a rule of LKID. The derivations of $\Gamma_1 \vdash \Delta_1, \ldots, \Gamma_n \vdash \Delta_n$ each have height $< h$, so by the induction hypothesis $\Gamma_1 \vdash \Delta_1, \ldots, \Gamma_n \vdash \Delta_n$ are all Henkin valid. Thus by the local soundness of rule (R) (Lemma 3.2.2), $\Gamma \vdash \Delta$ is Henkin valid as required. □

3.3 Cut-free Henkin completeness of LKID

We now present a proof of the *Henkin completeness* property of LKID, namely that the converse of Proposition 3.2.8 holds: if $\Gamma \vdash \Delta$ is Henkin valid, then there exists an LKID proof of $\Gamma \vdash \Delta$. Moreover, this proof does not use the rules (Cut), (Subst) or (Wk); hence we obtain via our soundness and completeness theorems a proof of the eliminability of these rules.

A cut-eliminability result for a classical proof system with induction rules does not seem to have appeared previously in the literature, but is not entirely surprising, as proper cut-elimination theorems[2] have been already been established for related intuitionistic proof systems. The first such was Martin-Löf's proof of normalisation for a natural deduction system for intuitionistic logic with iterated inductive definitions [44], adapted from a method used by Tait to prove normal form theorems for primitive recursive functionals of finite type [68]. Around the same time, Jervell proved normalisation for Heyting arithmetic, the intuitionistic version of Peano arithmetic [35]. The Tait/Martin-Löf proof method has been adapted to prove cut-elimination for an intuitionistic sequent calculus $FO\lambda^{\triangle\mathbb{N}}$ with natural number induction and definitions by McDowell and Miller [47, 48], and subsequently for Linc, an extension of $FO\lambda^{\triangle\mathbb{N}}$ including both induction and coinduction, by Tiu and Momigliano [49, 73]. However, this proof method cannot easily be adapted to a classical setting because the definition of "computability" of a derivation, on which the proof method depends, crucially relies on the derivation having a single end formula (which in sequent calculus corresponds to at most one formula appearing on the right hand side of sequents). Urban and Bierman have given a strongly normalising cut-elimination procedure for sequent calculus for classical first-order logic [78] using an adaptation of Barbanera and Berardi's symmetric reducibility candidates for λ-calculus [5]. This technique could possibly be adapted to give cut-elimination in LKID, but is already rather complicated in the setting of first-order logic, and would presumably become considerably more so in the presence of inductive definitions. In any case, our semantic approach has some advantages: it gives quite a clean proof of cut-eliminability, and also establishes completeness with respect to validity in Henkin models, which is of independent interest.

We now turn to our completeness proof, which employs an extension of the direct style of completeness proof for calculi for standard first-order logic as given in e.g. [15, 80]. When we say a sequent $\Gamma \vdash \Delta$ is *cut-free provable*, we mean that there is an LKID proof of $\Gamma \vdash \Delta$ that does not contain any instances of the rules (Cut), (Subst) or (Wk), and we show that every sequent that is Henkin valid is cut-free provable in this sense. (We thus establish via soundness and cut-free completeness a very slight sharpening of the usual notion of cut-eliminability. However, the rules (Wk) and (Subst) are easily shown eliminable in any case.) The structure of our proof is then roughly as follows:

1. Assume that $\Gamma \vdash \Delta$ is not cut-free provable, and, using a *schedule* on which every formula of $\mathrm{FOL_{ID}}$ appears infinitely often, construct from $\Gamma \vdash \Delta$ a *limit sequent* $\Gamma_\omega \vdash \Delta_\omega$, where Γ_ω and Δ_ω are infinite sets, such that no finite subsequent of $\Gamma_\omega \vdash \Delta_\omega$ is cut-free provable;

2. Define an equivalence relation $\sim$ on the terms of Σ that essentially "factors out" the

[2] The term "cut-elimination" usually refers to the syntactic transformation of a proof containing cuts to a proof without cuts; therefore, what we deduce from our soundness and cut-free completeness theorems is not cut-elimination but, more weakly, the *eliminability* of cut (and of the weakening and substitution rules).

equality formulas appearing in Γ_ω and use $\Gamma_\omega \vdash \Delta_\omega$ to construct a first-order structure M_ω on the terms of Σ modulo $\sim$ and an environment ρ_ω interpreting the terms of Σ modulo $\sim$ as elements of M_ω;

3. Prove that $\Gamma_\omega \vdash \Delta_\omega$ is false in M_ω under the environment ρ_ω;

4. Use M_ω and ρ_ω to define a Henkin class $\mathcal{H}_\omega$ for M_ω, and prove $(M_\omega, \mathcal{H}_\omega)$ is a Henkin model for (Σ, Φ);

5. It now follows from steps 3 and 4 that every finite subsequent of $\Gamma_\omega \vdash \Delta_\omega$ is false in the Henkin model $(M_\omega, \mathcal{H}_\omega)$, including $\Gamma \vdash \Delta$, so $\Gamma \vdash \Delta$ is not Henkin valid.

Steps 1,2,3 and the analogous version of step 5 for standard models also appear in the standard completeness proof for first-order logic (although the equivalence relation $\sim$ constructed in step 2 need only be defined if one considers first-order logic with equality). However, in our setting, the construction of the limit sequent in step 1 must take account of the induction rules. Applications of right-introduction rules for inductive predicates must also be accounted for in step 3. That said, the bulk of the new work in our proof goes into establishing that $(M_\omega, \mathcal{H}_\omega)$ defined in step 4 is indeed a Henkin model.

Definition 3.3.1 (Schedule)**.** For each $i \in \{1, \ldots, n\}$, let k_i be the arity of the inductive predicate symbol P_i. Then an *LKID-schedule element* of Σ is defined as follows:

- any formula of the form $\neg F$, $F_1 \wedge F_2$, $F_1 \vee F_2$, or $F_1 \to F_2$ is a LKID-schedule element;
- for any term t of Σ, any variable $x \in \mathcal{V}$, and any formula F, the pairs $\langle \forall x F, t \rangle$ and $\langle \exists x F, t \rangle$ are LKID-schedule elements;
- if P_i is an inductive predicate symbol, $\mathbf{t}$ is a sequence of k_i terms of Σ and, for each $j \in \{1, \ldots, n\}$, $\mathbf{z_j}$ is a sequence of k_j distinct variables and F_j is a formula, then the tuple $\langle P_i \mathbf{t}, \mathbf{z_1}, F_1, \ldots, \mathbf{z_n}, F_n \rangle$ is an LKID-schedule element.

An *LKID-schedule* for Σ is then an enumeration $(E_i)_{i \geq 0}$ of schedule elements of Σ such that every schedule element of Σ appears infinitely often in the enumeration.

Note that since we assume that our languages are countable (c.f. Definition 2.1.1), it follows via an argument à la Cantor (i.e. a diagonal traversal of a two-dimensional grid of all possible term constructions) that the terms of Σ are enumerable. By the same type of argument, the Σ-formulas of $\mathrm{FOL_{ID}}$ are enumerable, and hence the LKID-schedule elements of Σ are enumerable, so an LKID-schedule indeed exists for Σ.

Definition 3.3.2 (Limit sequent)**.** Suppose that $\Gamma \vdash \Delta$ is not cut-free provable. We define an infinite sequence $(\Gamma_i \vdash \Delta_i)_{i \geq 0}$ of sequents such that for each $i \geq 0$, $\Gamma_i \vdash \Delta_i$ is not cut-free provable and $\Gamma_i \vdash \Delta_i$ is a subsequent of $\Gamma_{i+1} \vdash \Delta_{i+1}$. We set $\Gamma_0 \vdash \Delta_0 = \Gamma \vdash \Delta$, so this is trivially true

for $i = 0$. We inductively assume we have constructed $(\Gamma_j \vdash \Delta_j)_{0 \leq i \leq j}$ with $\Gamma_i \vdash \Delta_i$ not cut-free provable for all $i \in \{0, \ldots, j\}$, and show how to construct the next sequent $S = (\Gamma_{j+1}, \Delta_{j+1})$.

We first remark that no formula can be in both Γ_j and Δ_j, otherwise $\Gamma_j \vdash \Delta_j$ would be cut-free provable via (Ax). Let E be the $(j+1)$th element in the schedule. We proceed by case distinction on E:

- $E = \neg F$. If $\neg F \notin \Gamma_j \cup \Delta_j$ then just define $S = \Gamma_j \vdash \Delta_j$.

  If $\neg F \in \Gamma_j$ then consider the rule applications:

  $$\dfrac{\dfrac{\Gamma_j \vdash F, \Delta_j}{\Gamma_j, \neg F \vdash \Delta_j}\,(\neg\text{L})}{\Gamma_j \vdash \Delta_j}\,(\text{ContrL})$$

  It is clear that $\Gamma_j \vdash F, \Delta_j$ is not cut-free provable since otherwise $\Gamma_j \vdash \Delta_j$ would be cut-free provable. We define $S = \Gamma_j \vdash F, \Delta_j$.

  If $\neg F \in \Delta_j$ then consider:

  $$\dfrac{\dfrac{\Gamma_j, F \vdash \Delta_j}{\Gamma_j \vdash \neg F, \Delta_j}\,(\neg\text{L})}{\Gamma_j \vdash \Delta_j}\,(\text{ContrR})$$

  Again, it is clear that $\Gamma_j, F \vdash \Delta_j$ cannot be cut-free provable, and we define $S = \Gamma_j, F \vdash \Delta_j$.

- $E = F_1 \wedge F_2$. If $F_1 \wedge F_2 \notin \Gamma_j \cup \Delta_j$ then just define $S = \Gamma_j \vdash \Delta_j$.

  If $F_1 \wedge F_2 \in \Gamma_j$ then consider the rule applications:

  $$\dfrac{\dfrac{\Gamma_j, F_1, F_2 \vdash \Delta_j}{\Gamma_j, F_1 \wedge F_2 \vdash \Delta_j}\,(\wedge\text{L})}{\Gamma_j \vdash \Delta_j}\,(\text{ContrL})$$

  It is clear that $\Gamma_j, F_1, F_2 \vdash \Delta_j$ is not cut-free provable, since otherwise $\Gamma_j \vdash \Delta_j$ would be cut-free provable via the rule applications above. We thus define $S = \Gamma_j, F_1, F_2 \vdash \Delta_j$.

  On the other hand, if $F_1 \wedge F_2 \in \Delta_j$ then consider:

  $$\dfrac{\dfrac{\Gamma_j \vdash F_1, \Delta_j \quad \Gamma_j \vdash F_2, \Delta_j}{\Gamma_j \vdash F_1 \wedge F_2, \Delta_j}\,(\wedge\text{R})}{\Gamma_j \vdash \Delta_j}\,(\text{ContrR})$$

  From the above it is clear that one of $\Gamma_j \vdash \Delta_j, F_1$ or $\Gamma_j \vdash \Delta_j, F_2$, is not cut-free provable. We define S to be $\Gamma_j \vdash \Delta_j, F_1$ if $\Gamma_j \vdash \Delta_j, F_1$ is not cut-free provable and $\Gamma_j \vdash \Delta_j, F_2$ otherwise.

- $E = F_1 \vee F_2$. If $F_1 \vee F_2 \notin \Gamma_j \cup \Delta_j$ then just define $S = \Gamma_j \vdash \Delta_j$.

  If $F_1 \vee F_2 \in \Gamma_j$ then, by a similar argument to the previous case, one of $\Gamma_j, F_1 \vdash \Delta_j$, $\Gamma_j, F_2 \vdash \Delta_j$ is not cut-free provable. We define S to be $\Gamma_j, F_1 \vdash \Delta_j$ if this is not cut-free provable and $\Gamma_j, F_2 \vdash \Delta_j$ otherwise.

  Similarly, if $F_1 \vee F_2 \in \Delta_j$ then define $S = \Gamma_j, F_1, F_2 \vdash \Delta_j$, which can similarly be seen to be not cut-free provable.

- $F = F_1 \rightarrow F_2$. If $F_1 \rightarrow F_2 \notin \Gamma_j \cup \Delta_j$ then just define $S = \Gamma_j \vdash \Delta_j$.

  If $F_1 \rightarrow F_2 \in \Gamma_j$ then, by a similar argument to the previous two cases, either $\Gamma_j, F_2 \vdash \Delta_j$ or $\Gamma_j \vdash F_1, \Delta_j$ is not cut-free provable. We define S to be $\Gamma_j, F_2 \vdash \Delta_j$ if this is not cut-free provable and $\Gamma_j \vdash F_1, \Delta_j$ otherwise.

  On the other hand, if $F_1 \rightarrow F_2 \in \Delta_j$ then define $S = \Gamma_j, F_1 \vdash F_2, \Delta_j$, which can similarly be seen to be not cut-free provable.

- $E = \langle \exists x F, t \rangle$. If $\exists x F \notin \Gamma_j \cup \Delta_j$ then just define $S = \Gamma_j \vdash \Delta_j$.

  If $\exists x F \in \Gamma_j$ then consider:

  $$\dfrac{\dfrac{\Gamma_j, F[z/x] \vdash \Delta_j}{\Gamma_j, \exists x F \vdash \Delta_j}\,(\exists \text{L})}{\Gamma_j \vdash \Delta_j}\,(\text{ContrL})$$

  where $z \notin FV(\Gamma_j \cup \Delta_j)$. It is clear that $\Gamma_j, F[z/x] \vdash \Delta_j$ cannot be cut-free provable and we thus define $S = \Gamma_j, F[z/x] \vdash \Delta_j$.

  On the other hand, if $\exists x F \in \Delta_j$ then consider:

  $$\dfrac{\dfrac{\Gamma_j \vdash F[t/x], \Delta_j}{\Gamma_j \vdash \exists x F, \Delta_j}\,(\exists \text{R})}{\Gamma_j \vdash \Delta_j}\,(\text{ContrR})$$

  It is clear that $\Gamma_j \vdash F[t/x], \Delta_j$ cannot be cut-free provable and we thus define $S = \Gamma_j \vdash F[t/x], \Delta_j$.

- $E = \langle \forall x F, t \rangle$. If $\forall x F \notin \Gamma_j \cup \Delta_j$ then just define $S = \Gamma_j \vdash \Delta_j$.

  If $\forall x F \in \Gamma_j$ then define $S = \Gamma_j, F[t/x] \vdash \Delta_j$. If $\forall x F \in \Delta_j$ then define $S = \Gamma_j \vdash F[z/x], \Delta_j$, where $z \notin FV(\Gamma_j \cup \Delta_j)$. The justification that S is not cut-free provable is similar to the previous case $E = \langle \exists x F, t \rangle$.

- $E = \langle P_i \mathbf{t}, \mathbf{z_1}, F_1, \ldots, \mathbf{z_n}, F_n \rangle$, where $i \in \{1, \ldots, n\}$. If $P_i \mathbf{t} \notin \Gamma_j$ then just define $S = \Gamma_j \vdash \Delta_j$.

  If $P_i \mathbf{t} \in \Gamma_j$ then consider:

  $$\dfrac{\dfrac{\text{minor premises} \quad \Gamma_j, F_i \mathbf{t} \vdash \Delta_j}{\Gamma_j, P_i \mathbf{t} \vdash \Delta_j}\,(\text{Ind } P_i)}{\Gamma_j \vdash \Delta_j}\,(\text{ContrL})$$

where for each rule with P_k in the conclusion such that P_i and P_k are mutually dependent, say:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_k\mathbf{t}(\mathbf{x})} \qquad j_1,\ldots,j_m,k \in \{1,\ldots,n\}$$

there is a minor premise:

$$\Gamma_j, Q_1\mathbf{u_1}(\mathbf{x}),\ldots,Q_h\mathbf{u_h}(\mathbf{x}), G_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,G_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash F_k\mathbf{t}(\mathbf{x}),\Delta_j$$

where $x \notin FV(\Gamma_j \cup \Delta_j)$ for all $x \in \mathbf{x}$, G_j is defined for each $j \in \{1,\ldots,n\}$ by:

$$G_j = \begin{cases} F_j & \text{if } P_j \text{ and } P_i \text{ are mutually dependent} \\ P_j\mathbf{z_j} & \text{otherwise} \end{cases}$$

and for all $i \in \{1,\ldots,m\}$, $G_{j_i}\mathbf{t_i}(\mathbf{x})$ is obtained by substituting $\mathbf{t_i}(\mathbf{x})$ for $\mathbf{z_i}$ in the obvious way. (In other words, we use (some of) the variables $\mathbf{z_1},\ldots,\mathbf{z_n}$ as the induction variables and (some of) the formulas $F_1,\ldots,F_n$ as induction hypotheses in the instance of (Ind P_i).)

Since $\Gamma_j \vdash \Delta_j$ is not cut-free provable, it follows that either $\Gamma_j, F_i\mathbf{t} \vdash \Delta_j$ is not cut-free provable, or else some minor premise is not cut-free provable. If the former holds, we set $S = \Gamma_j, F_i\mathbf{t} \vdash \Delta_j$, and otherwise we define S to be a minor premise which is not cut-free provable (any will do).

Observe that by construction, we have $\Gamma_j \subseteq \Gamma_{j+1}$ and $\Delta_j \subseteq \Delta_{j+1}$ for all $j \geq 0$. Let $\Gamma_\omega = \bigcup_{j\geq 0}\Gamma_j$ and $\Delta_\omega = \bigcup_{j\geq 0}\Delta_j$. Then the *limit sequent for* $\Gamma \vdash \Delta$ is defined to be $\Gamma_\omega \vdash \Delta_\omega$. Of course, the limit sequent is not strictly speaking a sequent, since in general Γ_ω and Δ_ω will be infinite sets. When we say that e.g. $\Gamma_\omega \vdash \Delta_\omega$ is provable, we mean that $\Gamma' \vdash \Delta'$ is provable for some finite $\Gamma' \subseteq \Gamma_\omega$ and $\Delta' \subseteq \Delta_\omega$.

We remark that the rules for equality and the right-unfolding rules for inductive predicates are not used when constructing the limit sequent in the definition above. The roles of equality and of right-unfolding of inductive predicates are instead accounted for by separate lemmas to appear below.

For the remainder of this section, until the statement of our cut-free completeness theorem, we shall assume that some fixed sequent $\Gamma \vdash \Delta$ is not cut-free provable, and that $\Gamma_\omega \vdash \Delta_\omega$ is the limit sequent for $\Gamma \vdash \Delta$ constructed according to Definition 3.3.2.

Definition 3.3.3. Define the relation $\sim$ on terms of Σ by:

$$\frac{t_1 = t_2 \in \Gamma_\omega}{t_1 \sim t_2}\ (\sim \text{Base}) \qquad \frac{}{t \sim t}\ (\sim \text{Refl}) \qquad \frac{t_1 \sim t_2}{t_2 \sim t_1}\ (\sim \text{Sym})$$

$$\frac{t_1 \sim t_2 \quad t_2 \sim t_3}{t_1 \sim t_3}\ (\sim \text{Trans}) \qquad \frac{t_1 \sim u_1 \ldots t_k \sim u_k}{f(t_1,\ldots,t_k) \sim f(u_1,\ldots,u_k)}\ (\sim \text{Cong})$$

where f is an arbitrary function symbol of arity k in Σ in the rule $\sim$ Cong). We call $\sim$ the *equivalence relation induced by* $\Gamma_\omega \vdash \Delta_\omega$, and write $[t]$ for the equivalence class of t with respect to $\sim$, i.e. $[t] = \{u \mid t \sim u\}$.

Definition 3.3.4 (Counter-interpretation)**.** Define a first-order structure M_ω for Σ by:

- the domain of M_ω is $Terms(\Sigma)/\sim$, the set of $\sim$-equivalence classes of Σ-terms;
- for any constant symbol c, $c^{M_\omega} = [c]$;
- for any function symbol f in Σ of arity k, $f^{M_\omega}([t_1],\ldots,[t_k]) = [f(t_1,\ldots,t_k)]$;
- for any ordinary predicate Q in Σ, $Q^{M_\omega}([t_1],\ldots,[t_k]) \Leftrightarrow \exists u_1,\ldots,u_k.t_1 \sim u_1,\ldots,t_k \sim u_k$ and $Q(u_1,\ldots,u_k) \in \Gamma_\omega$;
- the interpretations of the inductive predicate symbols $P_1,\ldots,P_n$ of Σ is defined by: $(P_1^{M_\omega},\ldots,P_n^{M_\omega})$ is the smallest n-tuple of sets of tuples of $Terms(\Sigma)/\sim$ closed under the following conditions:

  1. for each $i \in \{1,\ldots,n\}$, if $P_i(t_1,\ldots,t_{k_i}) \in \Gamma_\omega$ then $P_i^{M_\omega}([t_1],\ldots,[t_{k_i}])$;
  2. $\varphi_\Phi(P_1^{M_\omega},\ldots,P_n^{M_\omega}) \subseteq (P_1^{M_\omega},\ldots,P_n^{M_\omega})$ (i.e. $(P_1^{M_\omega},\ldots,P_n^{M_\omega})$ is a prefixed point of φ_Φ).

  where φ_Φ is constructed with respect to M_ω and Σ. Note that this is not a circular definition, since the definition of φ_Φ (c.f. Definition 2.2.3) only requires the interpretation given to the constants, function symbols and ordinary predicates of Σ by M_ω, which we have already defined.

Also define an environment ρ_ω for M_ω by $\rho_\omega(x) = [x]$ for all variables x. Then (M_ω, ρ_ω) is called the *counter-interpretation for* $\Gamma_\omega \vdash \Delta_\omega$.

Proposition 3.3.5. *For all terms of* Σ *we have* $\rho_\omega(t) = [t]$.

Proof. By induction on the structure of t.

Case $t = x$. We have $\rho_\omega(x) = [x]$ by definition of ρ_ω.

Case $t = c$. We have $\rho_\omega(c) = c^{M_\omega} = [c]$ by definition of ρ_ω and M_ω.

Case $t = f(t_1,\ldots,t_k)$. We have:

$$\begin{array}{lcll} \rho_\omega(f(t_1,\ldots,t_k)) & = & f^{M_\omega}(\rho_\omega(t_1),\ldots,\rho_\omega(t_k)) & \text{c.f. Definition 2.1.4} \\ & = & f^{M_\omega}([t_1],\ldots,[t_k]) & \text{by induction hypothesis} \\ & = & [f(t_1,\ldots,t_k)] & \text{by definition of } M_\omega \end{array}$$

□

Proposition 3.3.6. *If $t(x_1,\ldots,x_k)$ is a term (whose variables are all contained in $\{x_1,\ldots,x_k\}$), and $t_1,\ldots,t_k$ are terms, we have $[t(t_1,\ldots,t_k)] = t^{M_\omega}([t_1],\ldots,[t_k])$.*

Proof. By induction on the structure of t.

Case $t = x$. As the variables of t are all contained in $\{x_1,\ldots,x_k\}$, we have $t = x_i$ for some $i \in \{1,\ldots,k\}$. We therefore require to prove $[x_i] = [x_i]$, which is trivial.

Case $t = c$. We are required to prove $[c] = c^{M_\omega}$, which is immediately the case by definition of M_ω.

Case $t = f(u_1,\ldots,u_m)$. By assumption we have $Var(u_i) \subseteq \{x_1,\ldots,x_k\}$ and so we can write u_i as $u_i(x_1,\ldots,x_k)$ for all $i \in \{1,\ldots,m\}$. Then we have:

$$\begin{array}{lcll} [(f(u_1,\ldots,u_m))(t_1,\ldots,t_k)] & = & [f(u_1(t_1,\ldots,t_k),\ldots,u_m(t_1,\ldots,t_k))] & \\ & = & f^{M_\omega}([u_1(t_1,\ldots,t_k)],\ldots,[u_m(t_1,\ldots,t_k)]) & \text{(defn of } M_\omega) \\ & = & f^{M_\omega}(u_1^{M_\omega}([t_1],\ldots,[t_k]),\ldots,u_m^{M_\omega}([t_1],\ldots,[t_k])) & \text{(by IH)} \\ & = & (f(u_1,\ldots,u_m))^{M_\omega}([t_1],\ldots,[t_k]) & \end{array}$$

which completes the proof. □

Lemma 3.3.7. *If $t \sim u$ then, for any formula F, it holds that $\Gamma_\omega \vdash F[t/x]$ is cut-free provable iff $\Gamma_\omega \vdash F[u/x]$ is cut-free provable.*

Proof. By rule induction on $t \sim u$.

Case ($\sim$ Base). We have $t = u \in \Gamma_\omega$. Suppose $\Gamma_\omega \vdash F[t/x]$ is cut-free provable, i.e., $\Gamma' \vdash F[t/x]$ is cut-free provable for some finite $\Gamma' \subseteq \Gamma_\omega$. Without loss of generality, we can assume $t = u \in \Gamma'$ (for if not, we can weaken the derivation of $\Gamma' \vdash F[t/x]$ everywhere by $t = u$ on the left, possibly renaming some eigenvariables). Then we can derive $\Gamma' \vdash F[u/x]$ as follows:

$$\dfrac{\dfrac{\vdots}{\Gamma' \vdash F[t/x]}}{\dfrac{\Gamma', t = u \vdash F[u/x]}{\Gamma' \vdash F[u/x]}\,(=\text{L})}\,(\text{ContrL})$$

whence it follows that $\Gamma_\omega \vdash F[u/x]$ is cut-free provable because $\Gamma' \subseteq \Gamma_\omega$. The converse is similar.

Case ($\sim$ Refl). We require to prove that $\Gamma_\omega \vdash F[t/x]$ is cut-free provable iff $\Gamma_\omega \vdash F[t/x]$ is cut-free provable, which is trivially the case.

Case ($\sim$ Sym). By induction hypothesis, we have that $\Gamma_\omega \vdash F[t_1/x]$ is cut-free provable iff $\Gamma_\omega \vdash F[t_2/x]$ is cut-free provable. It immediately follows that $\Gamma_\omega \vdash F[t_2/x]$ is cut-free provable iff $\Gamma_\omega \vdash F[t_1/x]$ is cut-free provable.

Case ($\sim$ Trans). By induction hypothesis, we have that $\Gamma_\omega \vdash F[t_1/x]$ is cut-free provable iff $\Gamma_\omega \vdash F[t_2/x]$ is cut-free provable iff $\Gamma_\omega \vdash F[t_3/x]$ is cut-free provable. It immediately follows that $\Gamma_\omega \vdash F[t_1/x]$ is cut-free provable iff $\Gamma_\omega \vdash F[t_3/x]$ is cut-free provable.

Case ($\sim$ Comp). By induction hypothesis we have that for all formulas F and all $i \in \{1,\ldots,k\}$, $\Gamma_\omega \vdash F[t_i/x]$ is cut-free provable iff $\Gamma_\omega \vdash F[u_i/x]$ is cut-free provable. We require to show that $\Gamma_\omega \vdash F[f(t_1,\ldots,t_k)/x]$ is cut-free provable iff $\Gamma_\omega \vdash F[f(u_1,\ldots,u_k)/x]$ is cut-free provable. By successively applying the induction hypotheses, we obtain:

$$\begin{array}{ll} & \Gamma_\omega \vdash F[f(t_1,\ldots,t_k)/x] \text{ cut-free provable} \\ \Leftrightarrow & \Gamma_\omega \vdash F[f(u_1,t_2,\ldots,t_k)/x] \text{ cut-free provable} \\ \Leftrightarrow & \Gamma_\omega \vdash F[f(u_1,u_2,t_3,\ldots,t_k)/x] \text{ cut-free provable} \\ & \vdots \\ \Leftrightarrow & \Gamma_\omega \vdash F[f(u_1,\ldots,u_k)/x] \text{ cut-free provable} \end{array}$$

which completes the case. □

Lemma 3.3.8. *For any $i \in \{1,\ldots,n\}$, if $P_i^{M_\omega}([t_1],\ldots,[t_{k_i}])$ then $\Gamma_\omega \vdash P_i(t_1,\ldots,t_{k_i})$ is cut-free provable.*

Proof. Define an n-tuple of sets $(X_1,\ldots,X_n)$ by:

$$X_i = \{([t_1],\ldots,[t_{k_i}]) \mid \Gamma_\omega \vdash P_i(t_1,\ldots,t_{k_i}) \text{ is cut-free provable}\}$$

for each $i \in \{1,\ldots,n\}$, where k_i is the arity of P_i. First, note that if $([t_1],\ldots,[t_{k_i}]) \in X_i$, then by definition of X_i there exist terms $u_1,\ldots,u_{k_i}$ such that $t_1 \sim u_1,\ldots,t_{k_i} \sim u_{k_i}$ and $\Gamma_\omega \vdash P_i(u_1,\ldots,u_{k_i})$ is cut-free provable. Then by Lemma 3.3.7 applied k_i times, it holds that $\Gamma_\omega \vdash P_i(t_1,\ldots,t_{k_i})$. We shall use this fact more than once in what follows.

We shall show that $(X_1,\ldots,X_n)$ satisfies:

1. for each $i \in \{1,\ldots,n\}$, if $P_i(t_1,\ldots,t_{k_i}) \in \Gamma_\omega$ then $([t_1],\ldots,[t_{k_i}]) \in X_i$;
2. $\varphi_\Phi(X_1,\ldots,X_n) \subseteq (X_1,\ldots,X_n)$.

Since $(P_1^{M_\omega},\ldots,P_n^{M_\omega})$ is defined to be the smallest tuple of sets satisfying these properties, it follows that if $P_i^{M_\omega}([t_1],\ldots,[t_{k_i}])$, then $([t_1],\ldots,[t_{k_i}]) \in X_i$ and thus by the argument above, it holds that $\Gamma_\omega \vdash P_i(t_1,\ldots,t_{k_i})$ is cut-free provable. We now proceed to verify that 1 and 2 above do indeed hold:

1. Let $i \in \{1,\ldots,n\}$ and suppose $P_i(t_1,\ldots,t_{k_i}) \in \Gamma_\omega$. It follows that $\Gamma_\omega \vdash P_i(t_1,\ldots,t_{k_i})$ is cut-free provable (via an application of (Ax)), and hence $([t_1],\ldots,[t_{k_i}]) \in X_i$ by definition of X_i.

2. By definition of φ_Φ, we are required to show $\bigcup_r \varphi_{i,r}(X_1,\ldots,X_n) \subseteq X_i$ for each $i \in \{1,\ldots,n\}$. Let $i \in \{1,\ldots,n\}$ and let $\Phi_{i,r}$ be an arbitrary production with P_i in the conclusion, say:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1,\ldots,j_m,i \in \{1,\ldots,n\}$$

   It suffices to show that $\varphi_{i,r}(X_1,\ldots,X_n) \subseteq X_i$, i.e.

$$\{\mathbf{t}^{\mathbf{M}_\omega}(\mathbf{x}) \mid Q_1^{M_\omega}\mathbf{u_1}^{\mathbf{M}_\omega}(\mathbf{x}),\ldots,Q_h^{M_\omega}\mathbf{u_h}^{\mathbf{M}_\omega}(\mathbf{x}),\mathbf{t_1}^{\mathbf{M}_\omega}(\mathbf{x}) \in X_{j_1},\ldots,\mathbf{t_m}^{\mathbf{M}_\omega}(\mathbf{x}) \in X_{j_m}\} \subseteq X_i$$

   which, by Proposition 3.3.6, rewrites to:

$$\{[\mathbf{t}(\mathbf{x})] \mid Q_1^{M_\omega}[\mathbf{u_1}(\mathbf{x})],\ldots,Q_h^{M_\omega}[\mathbf{u_h}(\mathbf{x})],[\mathbf{t_1}(\mathbf{x})] \in X_{j_1},\ldots,[\mathbf{t_m}(\mathbf{x})] \in X_{j_m}\} \subseteq X_i \ (*)$$

   Suppose $Q_1^{M_\omega}[\mathbf{u_1}(\mathbf{x})],\ldots,Q_h^{M_\omega}[\mathbf{u_h}(\mathbf{x})]$ all hold and $[\mathbf{t_1}(\mathbf{x})] \in X_{j_1},\ldots,[\mathbf{t_m}(\mathbf{x})] \in X_{j_m}$. By the argument at the start of the proof, there then exist cut-free proofs of $\Gamma_\omega \vdash P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,$ $\Gamma_\omega \vdash P_{j_m}\mathbf{t_m}(\mathbf{x})$. Similarly, for each $i \in \{1,\ldots,h\}$, letting Q_i have arity k_i and writing $([u_1(\mathbf{x})],\ldots,[u_{k_i}(\mathbf{x})])$ for $[\mathbf{u_i}(\mathbf{x})]$, we have by the definition of M_ω that there exist terms $w_1,\ldots,w_{k_i}$ such that $u_1(\mathbf{x}) \sim w_1,\ldots,u_{k_i}(\mathbf{x}) \sim w_{k_i}$ and $Q_i(w_1,\ldots,w_{k_i}) \in \Gamma_\omega$. So $\Gamma_\omega \vdash Q_i(w_1,\ldots,w_{k_i})$ is cut-free provable by an application of (Ax) and, by applying Lemma 3.3.7 k_i times, so is $\Gamma_\omega \vdash Q_i(u_1(\mathbf{x}),\ldots,u_{k_i}(\mathbf{x}))$, i.e., $\Gamma_\omega \vdash Q_i\mathbf{u_i}(\mathbf{x})$.

   We now observe that we can obtain a cut-free proof of $\Gamma_\omega \vdash P_i\mathbf{t}(\mathbf{x})$ using the right-unfolding rule corresponding to the production under consideration as follows:

$$\frac{\Gamma_\omega \vdash Q_1\mathbf{u_1}(\mathbf{x}) \ldots \Gamma_\omega \vdash Q_h\mathbf{u_h}(\mathbf{x}) \quad \Gamma_\omega \vdash P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots \Gamma_\omega \vdash P_{j_m}\mathbf{t_m}(\mathbf{x})}{\Gamma_\omega \vdash P_i\mathbf{t}(\mathbf{x})}\ (P_i\mathrm{R}_r)$$

   Hence we have $[\mathbf{t}(\mathbf{x})] \in X_i$ as required. This proves the set inclusion $(*)$ above and thus completes the proof.

□

Lemma 3.3.9. *If $F \in \Gamma_\omega$ then $M_\omega \models_{\rho_\omega} F$, and if $F \in \Delta_\omega$ then $M_\omega \not\models_{\rho_\omega} F$.*

Proof. By induction on the structure of F.

Case $F = P(t_1,\ldots,t_k)$. There are two subcases to consider: either P is an ordinary predicate or an inductive predicate.

Subcase $P = Q$, where Q is an ordinary predicate symbol of Σ. Suppose $Q(t_1,\ldots,t_k) \in \Gamma_\omega$. As $t_i \sim t_i$ for all $i \in \{1,\ldots,k\}$ by ($\sim$Refl), we thus have $Q^{M_\omega}([t_1],\ldots,[t_k])$ by definition of M_ω, i.e. $M_\omega \models_{\rho_\omega} Q(t_1,\ldots,t_k)$ as required.

Now suppose $Q(t_1,\ldots,t_k) \in \Delta_\omega$, and suppose for contradiction that $M_\omega \models_{\rho_\omega} Q(t_1,\ldots,t_k)$, i.e. $Q^{M_\omega}([t_1],\ldots,[t_k])$ holds. Then there exist terms $u_1,\ldots,u_k$ such that $t_1 \sim u_1,\ldots,t_k \sim u_k$ and $Q(u_1,\ldots,u_k) \in \Gamma_\omega$. So there is a cut-free proof of $\Gamma_\omega \vdash Q(u_1,\ldots,u_k)$ (via an application of (Ax)). By Lemma 3.3.7 (applied k times), it follows that there is a cut-free proof of $\Gamma_\omega \vdash Q(t_1,\ldots,t_k)$. Since $Q(t_1,\ldots,t_k) \in \Delta_\omega$, there is a cut-free proof of $\Gamma_\omega \vdash \Delta_\omega$, which is a contradiction. Hence $M_\omega \not\models_{\rho_\omega} Q(t_1,\ldots,t_k)$ as required.

Subcase $P = P_i$, where P_i is an inductive predicate symbol of Σ. If $P_i(t_1,\ldots,t_k) \in \Gamma_\omega$ then $P_i^{M_\omega}([t_1],\ldots,[t_k])$ by definition of $P_i^{M_\omega}$. By Proposition 3.3.5, $P_i^{M_\omega}(\rho_\omega(t_1),\ldots,\rho_\omega(t_k))$, i.e. $M_\omega \models_{\rho_\omega} P_i(t_1,\ldots,t_k)$ as required.

Now suppose $P_i(t_1,\ldots,t_k) \in \Delta_\omega$ and suppose for contradiction that $M_\omega \models_{\rho_\omega} P_i(t_1,\ldots,t_k)$, i.e. $P_i^{M_\omega}(\rho_\omega(t_1),\ldots,\rho_\omega(t_k))$. By Proposition 3.3.5, $P_i^{M_\omega}([t_1],\ldots,[t_k])$ and so by Lemma 3.3.8, $\Gamma_\omega \vdash P_i(t_1,\ldots,t_k)$ is cut-free provable. Since $P_i(t_1,\ldots,t_k) \in \Delta_\omega$, it holds that $\Gamma_\omega \vdash \Delta_\omega$ is cut-free provable, which is a contradiction. So $M_\omega \not\models_{\rho_\omega} P_i(t_1,\ldots,t_k)$ as required.

Case $F = (t_1 = t_2)$. If $t_1 = t_2 \in \Gamma_\omega$ then we have $t_1 \sim t_2$ (by ($\sim$Base)) and thus $[t_1] = [t_2]$. By Proposition 3.3.5 it follows that $\rho_\omega(t_1) = \rho_\omega(t_2)$ and so $M_\omega \models_{\rho_\omega} t_1 = t_2$.

Now suppose $t_1 = t_2 \in \Delta_\omega$, and suppose for contradiction that $M_\omega \models_{\rho_\omega} t_1 = t_2$, i.e. $\rho_\omega(t_1) = \rho_\omega(t_2)$. By Proposition 3.3.5, $[t_1] = [t_2]$ and so $t_1 \sim t_2$. Now, observe that $\Gamma_\omega \vdash t_1 = t_1$ is cut-free provable via an application of (=R). Hence by Lemma 3.3.7, $\Gamma_\omega \vdash t_1 = t_2$ is also cut-free provable. But since $t_1 = t_2 \in \Delta_\omega$, we would then have a cut-free proof of $\Gamma_\omega \vdash \Delta_\omega$, which is a contradiction. Hence $M_\omega \not\models_{\rho_\omega} t_1 = t_2$.

Case $F = \neg F'$. If $\neg F' \in \Gamma_\omega$, then by the construction of $\Gamma_\omega \vdash \Delta_\omega$ (c.f.. Definition 3.3.2), there is an $i \geq 0$ such that for all $j \geq i$, $\neg F' \in \Gamma_j$. Furthermore, as the element $\neg F'$ appears infinitely often on the schedule according to which $\Gamma_\omega \vdash \Delta_\omega$ is constructed, it follows that $F' \in \Delta_j$ for some $j \geq i$ and hence $F' \in \Delta_\omega$. By induction hypothesis, we thus have $M_\omega \not\models_{\rho_\omega} F'$, i.e. $M_\omega \models_{\rho_\omega} \neg F$ as required.

Now suppose $\neg F' \in \Delta_\omega$. By construction of $\Gamma_\omega \vdash \Delta_\omega$, $F' \in \Gamma_\omega$ so by induction hypothesis we have $M_\omega \models_{\rho_\omega} F'$, i.e. $M_\omega \not\models_{\rho_\omega} \neg F'$ as required.

Case $F = F_1 \wedge F_2$. If $F_1 \wedge F_2 \in \Gamma_\omega$ then by construction of $\Gamma_\omega \vdash \Delta_\omega$, $F_1 \in \Gamma_\omega$ and $F_2 \in \Gamma_\omega$. By induction hypothesis $M_\omega \models_{\rho_\omega} F_1$ and $M_\omega \models_{\rho_\omega} F_2$, i.e. $M_\omega \models_{\rho_\omega} F_1 \wedge F_2$ as required.

Now if $F_1 \wedge F_2 \in \Delta_\omega$ then by construction we have $F_1 \in \Delta_\omega$ or $F_2 \in \Delta_\omega$. In the former case

we have by induction hypothesis $M_\omega \not\models_{\rho_\omega} F_1$ whence it is clear that $M_\omega \not\models_{\rho_\omega} F_1 \wedge F_2$; the other case is similar.

Cases $F = F_1 \vee F_2$**,** $F = F_1 \to F_2$. These cases are similar to the previous case $F = F_1 \wedge F_2$ above.

Case $F = \exists x F'$. If $\exists x F' \in \Gamma_\omega$ then by construction of $\Gamma_\omega \vdash \Delta_\omega$, we have $F'[z/x] \in \Gamma_\omega$ for some variable z, whence $M_\omega \models_{\rho_\omega} F'[z/x]$ by induction hypothesis. By part 2 of Lemma 2.1.9, $M_\omega \models_{\rho_\omega[x \mapsto \rho_\omega(z)]} F'$, i.e. $M_\omega \models_{\rho_\omega} \exists x F'$.

Now suppose $\exists x F' \in \Delta_\omega$, and observe that by construction of $\Gamma_\omega \vdash \Delta_\omega$ (c.f. Definition 3.3.2) there is then an $i \geq 0$ such that $\exists F' \in \Gamma_j$ for all $j \geq i$. Now consider an arbitrary term t of Σ and note that the element $\langle \exists x F', t \rangle$ appears infinitely often on the schedule $(E_i)_{i \geq 0}$ according to which $\Gamma_\omega \vdash \Delta_\omega$ is constructed. So there is a $j \geq i$ such that $E_j = \langle \exists x F', t \rangle$ and thus we have $F'[t/x] \in \Delta_\omega$. As t was chosen arbitrarily, it follows that for every term t, $F'[t/x] \in \Delta_\omega$. So by induction hypothesis $M_\omega \not\models_{\rho_\omega} F'[t/x]$ for every term t. Suppose for contradiction that $M_\omega \models_{\rho_\omega} \exists x F'$. Then for some $t \in \mathit{Terms}(\Sigma)$, we would have $M_\omega \models_{\rho_\omega[x \mapsto [t]]} F'$, i.e. $M_\omega \models_{\rho_\omega[x \mapsto \rho_\omega(t)]} F'$ by Proposition 3.3.5. By part 2 of Lemma 2.1.9, it follows that $M_\omega \models_{\rho_\omega} F'[t/x]$, which contradicts our induction hypotheses. Hence $M_\omega \not\models_{\rho_\omega} \exists x F'$.

Case $F = \forall x F'$. This case is similar to the case $F = \exists x F'$ above. □

Definition 3.3.10 (Henkin counter-class)**.** Define $\mathcal{H}_\omega = \{H_k \mid k \in \mathbb{N}\}$ by:

$$H_k = \{\{([t_1], \ldots, [t_k]) \mid M_\omega \models_{\rho_\omega} F[t_1/x_1, \ldots, t_k/x_k]\} \mid F \text{ a formula}, x_1, \ldots, x_k \text{ distinct variables}\}$$

for each $k \geq 0$. (Notice that we do not place any restrictions on the free variables of the formula F.) Then $\mathcal{H}_\omega$ is said to be the *Henkin counter-class for* $\Gamma_\omega \vdash \Delta_\omega$.

Lemma 3.3.11. *The Henkin counter-class $\mathcal{H}_\omega$ for $\Gamma_\omega \vdash \Delta_\omega$ is indeed a Henkin class for M_ω.*

Proof. By Proposition 2.3.4, the class $\{H_k \mid k \in \mathbb{N}\}$ defined by:

$$H_k = \{\{(d_1, \ldots, d_k) \mid M_\omega \models_{\rho_\omega[x_1 \mapsto d_1, \ldots, x_k \mapsto d_k]} F\} \mid F \text{ a formula, } x_1, \ldots, x_k \text{ distinct variables}\}$$

for each $k \in \mathbb{N}$ is a Henkin class for M_ω. Now since the domain of M_ω is $\mathit{Terms}(\Sigma)/\sim$, an arbitrary element d_i of the domain can be written as $[t_i]$, where t_i is a term of Σ, and thus we can write an arbitrary member of H_k in the Henkin class above as:

$$\begin{array}{lll} & \{([t_1], \ldots, [t_k]) \mid M_\omega \models_{\rho_\omega[x_1 \mapsto [t_1], \ldots, x_k \mapsto [t_k]]} F\} & \\ \text{i.e.} & \{([t_1], \ldots, [t_k]) \mid M_\omega \models_{\rho_\omega[x_1 \mapsto \rho_\omega(t_1), \ldots, x_k \mapsto \rho_\omega(t_k)]} F\} & \text{by Proposition 3.3.5} \\ \text{i.e.} & \{([t_1], \ldots, [t_k]) \mid M_\omega \models_{\rho_\omega} F[t_1/x_1, \ldots, t_k/x_k]\} & \text{by Lemma 2.1.9, part 2} \end{array}$$

and thus $\mathcal{H}_\omega$ is a Henkin class for M_ω as required.

□

Lemma 3.3.12. *$(M_\omega, \mathcal{H}_\omega)$ is a Henkin model for (Σ, Φ).*

Proof. First observe that $\mathcal{H}_\omega$ is indeed a Henkin class for M_ω by Lemma 3.3.11. Now for each $i \in \{1,\ldots,n\}$, letting k_i be the arity of P_i and $x_1,\ldots,x_{k_i}$ be distinct variables, $P_i(x_1,\ldots,x_{k_i})$ is a formula of Σ, so by definition of H_{k_i}:

$$\{([t_1],\ldots,[t_{k_i}]) \mid M_\omega \models_{\rho_\omega} P_i(t_1,\ldots,t_{k_i})\} \in H_{k_i}$$
$$\text{i.e.} \quad \{([t_1],\ldots,[t_{k_i}]) \mid P_i^{M_\omega}([t_1],\ldots,[t_{k_i}])\} \in H_{k_i}$$

i.e. $P_i^{M_\omega} \in H_{k_i}$, so $(P_1^{M_\omega},\ldots,P_n^{M_\omega})$ is a $\mathcal{H}_\omega$-point. We also note that by definition of $(P_1^{M_\omega},\ldots,P_n^{M_\omega})$ we have $\varphi_\Phi(P_1^{M_\omega},\ldots,P_n^{M_\omega}) \subseteq (P_1^{M_\omega},\ldots,P_n^{M_\omega})$, so $(P_1^{M_\omega},\ldots,P_n^{M_\omega})$ is a prefixed $\mathcal{H}_\omega$-point of φ_Φ.

It therefore remains to show that $(P_1^{M_\omega},\ldots,P_n^{M_\omega})$ is the *least* prefixed $\mathcal{H}_\omega$-point of φ_Φ. Suppose that $(X_1,\ldots,X_n)$ is a prefixed $\mathcal{H}_\omega$-point of φ_Φ. We shall prove that $(X_1,\ldots,X_n)$ necessarily satisfies:

1. for each $i \in \{1,\ldots,n\}$, if $P_i(t_1,\ldots,t_{k_i}) \in \Gamma_\omega$ then $([t_1],\ldots,[t_{k_i}]) \in X_i$;
2. $\varphi_\Phi(X_1,\ldots,X_n) \subseteq (X_1,\ldots,X_n)$.

Since $(P_1^{M_\omega},\ldots,P_n^{M_\omega})$ is defined to be the smallest tuple of sets satisfying these properties, it then follows that $(P_1^{M_\omega},\ldots,P_n^{M_\omega}) \subseteq (X_1,\ldots,X_n)$, and so $(P_1^{M_\omega},\ldots,P_n^{M_\omega})$ is the least prefixed $\mathcal{H}_\omega$-point of φ_Φ. We observe that $(X_1,\ldots,X_n)$ immediately satisfies 2. since it is a prefixed point of φ_Φ by assumption. Also, since $(X_1,\ldots,X_n)$ is an $\mathcal{H}_\omega$-point (again by assumption), we have $X_i \in H_{k_i}$ for all $i \in \{1,\ldots,n\}$, so for each $i \in \{1,\ldots,n\}$ there exists a formula F_i and a tuple of k_i distinct variables $\mathbf{z_i} = (z_1,\ldots,z_{k_i})$ such that:

$$X_i = \{([t_1],\ldots,[t_{k_i}]) \mid M_\omega \models_{\rho_\omega} F_i[t_1/z_1,\ldots,t_{k_i}/z_{k_i}]\}$$

We now show that 1. above holds, that is, for each $i \in \{1,\ldots,n\}$, $P_i(t_1,\ldots,t_{k_i}) \in \Gamma_\omega$ implies $([t_1],\ldots,[t_{k_i}]) \in X_i$. We inductively assume that the statement holds for all predicates P_j such that $Prem^*(P_i,P_j)$ and $\neg Prem^*(P_j,P_i)$ hold, and show that the statement holds for P_i. (Technically, one is inducing over the ordering $\prec$ of inductive predicate symbols defined by $P_j \prec P_i \Leftrightarrow Prem^*(P_i,P_j)$ and $\neg Prem^*(P_j,P_i)$; one can easily see that $\prec$ is irreflexive, anti-symmetric, and transitive, and that there are no infinite decreasing $\prec$-chains as there are only finitely many inductive predicate symbols.) Suppose $P_i\mathbf{t} \in \Gamma_\omega$ (writing $\mathbf{t}$ for $(t_1,\ldots,t_{k_i})$), so that by construction of $\Gamma_\omega \vdash \Delta_\omega$, there is a j' such that $P_i\mathbf{t} \in \Gamma_j$ for all $j \geq j'$. Now note that the element $\langle P_i\mathbf{t}, \mathbf{z_1}, F_1, \ldots, \mathbf{z_n}, F_n\rangle$ occurs infinitely often on the schedule E according to which $\Gamma_\omega \vdash \Delta_\omega$ is constructed, so there is a $j \geq j'$ such that $E_j = \langle P_i\mathbf{t}, \mathbf{z_1}, F_1, \ldots, \mathbf{z_n}, F_n\rangle$ and $P_i\mathbf{t} \in \Gamma_j$, and so $\Gamma_{j+1} \vdash \Delta_{j+1}$ is one of the premises obtained from $\Gamma_j \vdash \Delta_j$ by applying the rule (Ind P_i) with induction hypotheses $F_1,\ldots,F_n$ and induction variables $\mathbf{z_1},\ldots,\mathbf{z_n}$ (c.f. Definition 3.3.2). It follows that either:

1. $\Gamma_{j+1} \vdash \Delta_{j+1}$ is the major premise $\Gamma_j, F_i\mathbf{t} \vdash \Delta_j$ of the induction rule instance, or

2. there is a production $\Phi_{k,r} \in \Phi$:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_k\mathbf{t}(\mathbf{x})} \qquad j_1, \ldots, j_m, k \in \{1, \ldots, n\}$$

   such that P_k and P_i are mutually dependent, and $\Gamma_{j+1} \vdash \Delta_{j+1}$ is the minor premise:

$$\Gamma, Q_1\mathbf{u_1}(\mathbf{x}), \ldots, Q_h\mathbf{u_h}(\mathbf{x}), G_{j_1}\mathbf{t_1}(\mathbf{x}), \ldots, G_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash F_k\mathbf{t}(\mathbf{x}), \Delta$$

   where G_j is defined for each j by: $G_j = \begin{cases} F_j & \text{if } P_j \text{ and } P_i \text{ are mutually dependent} \\ P_j\mathbf{z_j} & \text{otherwise} \end{cases}$

In the former case, we have $F_i\mathbf{t} \in \Gamma_\omega$ and thus $M_\omega \models_{\rho_\omega} F_i\mathbf{t}$ by Lemma 3.3.9, so by definition of X_i we have $[\mathbf{t}] = ([t_1], \ldots, [t_{k_i}]) \in X_i$ and are finished. In the latter case, we have:

$$Q_1\mathbf{u_1}(\mathbf{x}) \in \Gamma_\omega, \ldots, Q_h\mathbf{u_h}(\mathbf{x}) \in \Gamma_\omega, G_{j_1}\mathbf{t_1}(\mathbf{x}) \in \Gamma_\omega, \ldots, G_{j_m}\mathbf{t_m}(\mathbf{x}) \in \Gamma_\omega, F_k\mathbf{t}(\mathbf{x}) \in \Delta_\omega$$

Now, firstly we have $M_\omega \models_{\rho_\omega} Q_1\mathbf{u_1}(\mathbf{x}), \ldots, M_\omega \models_{\rho_\omega} Q_h\mathbf{u_h}(\mathbf{x})$ by Lemma 3.3.9 and so, by definition of M_ω and Proposition 3.3.5, $Q_1^{M_\omega}[\mathbf{u_1}(\mathbf{x})], \ldots, Q_h^{M_\omega}[\mathbf{u_h}(\mathbf{x})]$ all hold. Secondly, we have $M_\omega \not\models_{\rho_\omega} F_k\mathbf{t}(\mathbf{x})$, again by Lemma 3.3.9, and so $[\mathbf{t}(\mathbf{x})] \notin X_k$ by definition of X_k. Now consider the remaining statements $G_{j_l}\mathbf{t_l}(\mathbf{x}) \in \Gamma_\omega$ (where $l \in \{1, \ldots, m\}$). If P_{j_l} and P_i are mutually dependent, then $G_{j_l} = F_{j_l}$ and so $M_\omega \models_{\rho_\omega} F_{j_l}\mathbf{t_l}(\mathbf{x})$ by Lemma 3.3.9, i.e. $[\mathbf{t_l}(\mathbf{x})] \in X_{j_l}$ by definition of X_{j_l}. Otherwise, note that $Prem^*(P_i, P_{j_l})$ and $\neg Prem^*(P_{j_l}, P_i)$ both hold, i.e. $P_{j_l} \prec P_i$, because $Prem(P_k, P_{j_l})$ holds and P_k and P_i are mutually dependent, but P_{j_l} and P_i are not mutually dependent. Since $P_{j_l}\mathbf{t_l}(\mathbf{x}) \in \Gamma_\omega$ we then have $[\mathbf{t_l}(\mathbf{x})] \in X_{j_l}$ by induction hypothesis. In summary, we have the following:

$$Q_1^{M_\omega}[\mathbf{u_1}(\mathbf{x})], \ldots, Q_h^{M_\omega}[\mathbf{u_h}(\mathbf{x})], [\mathbf{t_1}(\mathbf{x})] \in X_{j_1}, \ldots, [\mathbf{t_m}(\mathbf{x})] \in X_{j_m}, [\mathbf{t}(\mathbf{x})] \notin X_k \ (*)$$

But since $\varphi_\Phi(X_1, \ldots, X_n) \subseteq (X_1, \ldots, X_n)$, we must have $\varphi_{k,r}(X_1, \ldots, X_n) \subseteq X_k$, i.e.

$$\{[\mathbf{t}(\mathbf{x})] \mid Q_1^{M_\omega}[\mathbf{u_1}(\mathbf{x})], \ldots, Q_h^{M_\omega}[\mathbf{u_h}(\mathbf{x})], [\mathbf{t_1}(\mathbf{x})] \in X_{j_1}, \ldots, [\mathbf{t_m}(\mathbf{x})] \in X_{j_m}\} \subseteq X_k$$

using Proposition 3.3.6, which contradicts $(*)$ above. This completes the case and thus the proof. □

Theorem 3.3.13 (Cut-free Henkin completeness of LKID)**.** *If $\Gamma \vdash \Delta$ is Henkin valid, then it is cut-free provable in LKID.*

Proof. Suppose for contradiction that $\Gamma \vdash \Delta$ is Henkin valid, but not cut-free provable, and let $\Gamma_\omega \vdash \Delta_\omega$ be the limit sequent for $\Gamma \vdash \Delta$ (c.f. Definition 3.3.2) with counter-interpretation (M_ω, ρ_ω) and Henkin counter-class $\mathcal{H}_\omega$ (c.f. Definitions 3.3.4 and 3.3.10). By Lemma 3.3.12, $(M_\omega, \mathcal{H}_\omega)$ is a Henkin model for (Σ, Φ), and by Lemma 3.3.9, we have $M_\omega \models_{\rho_\omega} F$ for all $F \in \Gamma_\omega$

and $M_\omega \not\models_{\rho_\omega} F$ for all $F \in \Delta_\omega$. As $\Gamma \subseteq \Gamma_\omega$ and $\Delta \subseteq \Delta_\omega$, we thus have $M_\omega \models_{\rho_\omega} F$ for all $F \in \Gamma$ and $M_\omega \not\models_{\rho_\omega} F$ for all $F \in \Delta$.

In other words, we have constructed a Henkin model $(M_\omega, \mathcal{H}_\omega)$ of (Σ, Φ) such that $\Gamma \vdash \Delta$ is not true in $(M_\omega, \mathcal{H}_\omega)$ and we have the required contradiction. Hence $\Gamma \vdash \Delta$ is cut-free provable in LKID. □

Theorem 3.3.14 (Eliminability of cut for LKID)**.** *If $\Gamma \vdash \Delta$ is provable in LKID then it is cut-free provable (i.e. provable without using the rules (Cut), (Subst) or (Wk)).*

Proof. Suppose $\Gamma \vdash \Delta$ is provable in LKID. By soundness (Proposition 3.2.8), $\Gamma \vdash \Delta$ is Henkin valid. Thus by cut-free completeness (Theorem 3.3.13), $\Gamma \vdash \Delta$ is cut-free provable in LKID. □

3.4 LKID, second-order logic, and Peano arithmetic

In order to put our results about LKID into a wider context, we now demonstrate, firstly, that LKID can be viewed as a subsystem of Takeuti's sequent calculus for classical second order logic and, secondly, that Peano arithmetic PA (c.f. Definition 2.3.9) can be formalised within LKID. It follows that the consistency of PA follows from the eliminability of cut in LKID, whence we deduce that there can be no elementary proof of Theorem 3.3.14.

3.4.1 Embedding LKID in L^2K

The main features of Takeuti's second-order sequent calculus L^2K (our name for the calculus is taken from Girard [29], although the system is named G^1LC in Takeuti's original formulation [69]), are as follows:

- the underlying language Σ has two types of objects: (first-order) individuals, and sets of individuals;
- we have accordingly two types of variables: standard (first-order) variables $x, y, z, \ldots$, which range over individuals; and second-order variables $X, Y, Z, \ldots$, which range over sets of (tuples of) individuals, and two versions of the $\forall$ and $\exists$ quantifiers: $\forall x, \forall X, \exists x, \exists X$;
- there is a special binary predicate $\in$ that takes as arguments one individual and one set (the intended interpretation of $x \in X$ is, of course, "x is a member of X");
- we may build *abstraction terms* from formulas: if F is a formula, and x is a first-order variable, then $\lambda x.F$ is an abstraction term (whose intended interpretation is "the set of x such that F"), and we may interchangeably write Ft and $t \in F$;

- the rules for L^2K are the rules of LK, augmented by the obvious rules for the second-order quantifiers:

$$\frac{\Gamma, F[T/X] \vdash \Delta}{\Gamma, \forall XF \vdash \Delta}\,(\forall^2\mathrm{L}) \qquad \frac{\Gamma \vdash F, \Delta}{\Gamma \vdash \forall XF, \Delta}\; X \notin FV(\Gamma \cup \Delta)\,(\forall^2\mathrm{R})$$

$$\frac{\Gamma, F \vdash \Delta}{\Gamma, \exists XF \vdash \Delta}\; X \notin FV(\Gamma \cup \Delta)\,(\exists^2\mathrm{L}) \qquad \frac{\Gamma \vdash F[T/X], \Delta}{\Gamma \vdash \exists XF, \Delta}\,(\exists^2\mathrm{R})$$

  where T is an abstraction term in the rules ($\forall^2$R) and ($\exists^2$L), and the set of free variables of a formula F, $FV(F)$, is defined in the obvious way (that is to say, analogously to Definition 2.1.7).

We claim that LKID is subsumed by the second-order sequent calculus L^2K. To see this, we translate equality and the inductive predicates of Σ as abstraction terms (i.e. sets) so that atomic formulas involving equality or inductive predicates are translated as membership statements for their respective sets. The (translations of the) rules for equality and for inductive predicates are then derivable in L^2K, whence the translation of every LKID-provable sequent is provable in L^2K.

To begin with, we translate equality, $=$, as the abstraction term $\lambda x.\lambda y.\forall X(x \in X \leftrightarrow x \in Y)$, and as usual write $t = u$ for $= tu$ (this is often known as *Leibnitz equality*). The rule ($=$R) is then straightforwardly derivable in L^2K (we use the rule label ($\equiv$) in proofs for rewriting according to a definition):

$$\dfrac{\dfrac{\dfrac{\dfrac{}{t \in X \vdash t \in X}\,(\mathrm{Ax})}{\vdash t \in X \to t \in X}\,(\to\mathrm{R}) \qquad \dfrac{\dfrac{}{t \in X \vdash t \in X}\,(\mathrm{Ax})}{\vdash t \in X \to t \in X}\,(\to\mathrm{R})}{\vdash t \in X \leftrightarrow t \in X}\,(\wedge\mathrm{R})}{\vdash \forall X(t \in X \leftrightarrow t \in X)}\,(\forall^2\mathrm{R})}{\Gamma \vdash t = t, \Delta}\,(\equiv)$$

Now we define the abstraction terms:

$$\begin{aligned} F_1 &\equiv \lambda z.\bigwedge \Gamma[z/x,u/y] \to \bigvee \Delta[z/x,u/y] \\ F_2 &\equiv \lambda z.\bigwedge \Gamma[u/x,z/y] \to \bigvee \Delta[u/x,z/y] \end{aligned}$$

whence we start the L^2K-derivation of the rule (=L) as follows:

$$\dfrac{\Gamma[t/x,u/y], t \in F_1 \vdash \Delta[t/x,u/y] \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{u \in F_2 \vdash u \in F_1}\,(\text{Ax}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma[u/x,t/y] \vdash \Delta[u/x,t/y]}{\vdots}\,(\wedge\text{L})}{\bigwedge\Gamma[u/x,t/y] \vdash \Delta[u/x,t/y]}\,(\wedge\text{L}) \quad \vdots \; (\vee\text{R})}{\bigwedge\Gamma[u/x,t/y] \vdash \bigvee\Delta[u/x,t/y]}\,(\vee\text{R})}{\vdash \bigwedge\Gamma[u/x,t/y] \to \bigvee\Delta[u/x,t/y]}\,(\to\text{R})}{\vdash t \in F_2}\,(\equiv)}{t \in F_2 \to u \in F_2 \vdash u \in F_1}\,(\to\text{L})}{t \in F_2 \leftrightarrow u \in F_2 \vdash u \in F_1}\,(\wedge\text{L})}{\forall X(t \in X \leftrightarrow u \in X) \vdash u \in F_1}\,(\forall^2\text{L})}{t = u \vdash u \in F_1}\,\equiv}{\dfrac{\dfrac{\dfrac{\Gamma[t/x,u/y], t = u, u \in F_1 \to t \in F_1 \vdash \Delta[t/x,u/y]}{\Gamma[t/x,u/y], t = u, t \in F_1 \leftrightarrow u \in F_1 \vdash \Delta[t/x,u/y]}\,(\wedge\text{L})}{\Gamma[t/x,u/y], t = u, \forall X(t \in X \leftrightarrow u \in X) \vdash \Delta[t/x,u/y]}\,(\forall^2\text{L})}{\dfrac{\Gamma[t/x,u/y], t = u, t = u \vdash \Delta[t/x,u/y]}{\Gamma[t/x,u/y], t = u \vdash \Delta[t/x,u/y]}\,(\text{ContrL})}\,\equiv}\,(\to\text{L})$$

Notice that in the above, $u \in F_1 \vdash u \in F_2$ is a logical axiom because $u \in F_1 = u \in F_2 = \bigwedge\Gamma[u/x,u/y] \to \bigvee\Delta[u/x,u/y]$. To complete the derivation of (=L), we observe that the upper sequent on the left-hand branch is L^2K-derivable as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{\{\Gamma[t/x,u/y] \vdash F \mid F \in \Gamma[t/x,u/y]\}}\,(\text{Ax})}{\vdots}\,(\wedge\text{R})}{\Gamma[t/x,u/y] \vdash \bigwedge\Gamma[u/x,t/y]}\,(\wedge\text{R}) \qquad \dfrac{\dfrac{\dfrac{}{\{F \vdash \Delta[t/x,u/y] \mid F \in \Delta[t/x,u/y]\}}\,(\text{Ax})}{\vdots}\,(\vee\text{L})}{\bigvee\Delta[u/x,t/y] \vdash \Delta[t/x,u/y]}\,(\vee\text{L})}{\dfrac{\Gamma[t/x,u/y], \bigwedge\Gamma[u/x,t/y] \to \bigvee\Delta[u/x,t/y] \vdash \Delta[t/x,u/y]}{\Gamma[t/x,u/y], t \in F_1 \vdash \Delta[t/x,u/y]}\,(\equiv)}\,(\to\text{L})$$

We now show how to translate the "natural number" predicate N defined in Example 2.2.5 into an abstraction term in L^2K and derive the corresponding LKID rules for N. Extending the translation to general inductive predicates is somewhat messy because of (possible) mutual dependency between inductive predicates, but should nevertheless work similarly. We translate the inductive predicate N as the abstraction term:

$$N = \lambda x.\forall X(0 \in X \wedge \forall y(y \in X \to sy \in X) \to x \in X)$$

We next show how to derive in L^2K any instance of the induction rule (Ind N) for N defined in Example 3.1.5. If the induction hypothesis of the rule instance is F and the induction variable is

z, we create an abstraction term $\lambda z.F$, whence we can derive the instance of (Ind N) as follows (we write C[X] for the formula $0 \in X \wedge \forall y(y \in X \to sy \in X)$):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma, Fx \vdash Fsx, \Delta}{\Gamma \vdash Fx \to Fsx, \Delta}(\to\mathrm{R})}{\Gamma \vdash \forall y(Fy \to Fsy), \Delta}(\forall\mathrm{R}) \quad \Gamma \vdash F0, \Delta}{\Gamma \vdash F0 \wedge \forall y(Fy \to Fsy), \Delta}(\wedge\mathrm{R})}{\Gamma \vdash C[F], \Delta}(\equiv) \quad \Gamma, Ft \vdash \Delta}{\Gamma, C[F] \to Ft \vdash \Delta}(\to\mathrm{L})}{\Gamma, \forall X(C[X] \to t \in X) \vdash \Delta}(\forall^2\mathrm{L})}{\Gamma, Nt \vdash \Delta}(\equiv)$$

The first right-introduction rule ($N\mathrm{R}_1$) is derivable in $\mathrm{L}^2\mathrm{K}$ as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{0 \in X \vdash 0 \in X}(\mathrm{Ax})}{0 \in X \wedge \forall y(y \in X \to sy \in X) \vdash 0 \in X}(\wedge\mathrm{L})}{\vdash 0 \in X \wedge \forall y(y \in X \to sy \in X) \to 0 \in X}(\to\mathrm{R})}{\vdash \forall X(0 \in X \wedge \forall y(y \in X \to sy \in X) \to 0 \in X)}(\forall^2\mathrm{R})}{\Gamma \vdash N0, \Delta}(\equiv)$$

The second right-introduction rule ($N\mathrm{R}_2$) is likewise $\mathrm{L}^2\mathrm{K}$-derivable:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{C[X] \vdash C[X]}(\mathrm{Ax}) \quad \dfrac{\dfrac{\dfrac{\dfrac{}{t \in X \vdash t \in X}(\mathrm{Ax}) \quad \dfrac{}{st \in X \vdash st \in X}(\mathrm{Ax})}{t \in X, t \in X \to st \in X \vdash st \in X}(\to\mathrm{L})}{t \in X, \forall y(y \in X \to sy \in X) \vdash st \in X}(\forall\mathrm{L})}{t \in X, 0 \in X \wedge \forall y(y \in X \to sy \in X) \vdash st \in X}(\wedge\mathrm{L})}{t \in X, C[X] \vdash st \in X}(\equiv)}{C[X] \to t \in X, C[X] \vdash st \in X}(\to\mathrm{L})}{\forall X(C[X] \to t \in X), C[X] \vdash st \in X}(\forall^2\mathrm{L})}{\forall X(C[X] \to t \in X) \vdash C[X] \to st \in X}(\to\mathrm{R})}{\forall X(C[X] \to t \in X) \vdash \forall X(C[X] \to st \in X)}(\forall^2\mathrm{R})}{Nt \vdash Nst}(\equiv) \quad \Gamma \vdash Nt, \Delta}{\Gamma \vdash Nst, \Delta}(\mathrm{Cut})$$

We remark that ($N\mathrm{R}_2$) is *not* derivable in $\mathrm{L}^2\mathrm{K}$ without the use of (Cut). If it were, then in particular it would be derivable for the case $\Gamma = \Delta = \emptyset$. Working backwards from $\vdash Nst$, one is forced to initially apply the rules ($\forall^2\mathrm{R}$), ($\to\mathrm{R}$) and ($\wedge\mathrm{L}$), thereby obtaining the upper sequent $0 \in X, \forall y(y \in X \to sy \in X) \vdash st \in X$. But there is clearly no way to introduce a second-order

quantifier from this sequent without a cut and thus we cannot obtain the rule premise $\vdash Nt$ as required. So our translation does not in general map cut-free proofs in LKID to cut-free proofs in L^2K. (However, there might exist a cleverer translation that does have this property.)

In 1953, Takeuti stated his well-known conjecture that cut-elimination holds in L^2K [69]. The conjecture was eventually proven correct by Tait [67] using semantic methods in 1966, although the first syntactic proof was not given until later by Girard [28]. Since L^2K subsumes LKID, one might think that cut-elimination for LKID follows from cut-elimination in L^2K. However, as cut-free proofs in LKID seemingly cannot be easily translated into cut-free proofs in L^2K, this would appear not to be the case. Even if cut-free proofs in LKID could be translated into cut-free proofs in L^2K, the result would not follow straightforwardly since existing proofs of Takeuti's conjecture (which are already complicated) do not give a strong normalisation result. However, the work of Parigot on strong normalisation for second order classical natural deduction may be applicable [53].

3.4.2 LKID and Peano arithmetic

LKID subsumes LK and LK_e by construction, and is subsumed by L^2K as shown above. This just confirms our intuition that the logic FOL_{ID} extends first-order logic but stops short of the full quantification over second-order variables permitted in second-order logic. We can do somewhat better than this rough characterisation, however: as advertised previously, we now show that Peano arithmetic can be translated into LKID augmented with some extra axioms (i.e. proof rules with no premises). We then obtain the consistency of Peano arithmetic from the eliminability of cut in LKID (Theorem 3.3.14) using a standard argument (as given in e.g. [29]).

For this section, we recall the definition of Peano arithmetic (PA) given in Definition 2.3.9. Further, we let Σ'_{PA} be the language obtained by extending Σ_{PA} with a unary inductive predicate symbol N, and let Φ_N be the inductive definition set consisting of the "natural number" productions for N defined in Example 2.2.5. (The corresponding right- and left-introduction rules for N are given respectively in Examples 3.1.1 and 3.1.5.)

Lemma 3.4.1. *Let t be a term of Σ'_{PA}. If $Var(t) \subseteq \{x_1, \ldots, x_k\}$ then the sequent $Nx_1, \ldots, Nx_k \vdash Nt$ is provable in LKID (using the definition set Φ_N).*

Proof. By structural induction on t. In the base cases, t is either the constant 0 or one of the variables $x_1, \ldots, x_k$, and we are immediately done by an application of ($N\mathrm{R}_1$) or (Ax) respectively. In the induction step case, we have $t = st'$, where the sequent $Nx_1, \ldots, Nx_k \vdash Nt'$ is provable in LKID by the induction hypothesis, and we are then done by an application of ($N\mathrm{R}_2$). □

Definition 3.4.2. Define the function R by induction on the structure of Σ'_{PA}-formulas of $\mathrm{FOL}_{\mathrm{ID}}$ as follows:

$$\begin{array}{rcll} R(Nt) & = & Nt & \\ R(s=t) & = & s=t & \\ R(\neg F) & = & \neg R(F) & \\ R(F_1 * F_2) & = & R(F_1) * R(F_2) & (* \in \{\wedge, \vee, \rightarrow\}) \\ R(\forall x F) & = & \forall x(Nx \rightarrow R(F)) & \\ R(\exists x F) & = & \exists x(Nx \wedge R(F)) & \end{array}$$

We extend R to multisets of formulas in the obvious way: $R(\Gamma) = \{R(F) \mid F \in \Gamma\}$.

We remark that the effect of applying the function R to a formula F is to "relativise" all the quantifiers appearing in F so that they quantify over the predicate N. The function R gives an embedding of PA into LKID, constructed in the next lemma. Note that we write LKID+ (PA1)–(PA6) for the "augmented" proof system obtained by adding to *LKID* the axioms: $\dfrac{}{\vdash (PAi)}$ for $i \in \{1, \ldots, 6\}$.

Lemma 3.4.3. *If a sequent $\Gamma \vdash \Delta$ is provable in PA, and $FV(\Gamma \cup \Delta) \subseteq \{x_1, \ldots, x_k\}$, then the sequent $Nx_1, \ldots, Nx_k, R(\Gamma) \vdash R(\Delta)$ is provable in LKID+ (PA1)–(PA6), where the language for LKID is Σ'_{PA} and the inductive definition set is Φ_N.*

Proof. (Sketch) The proof is by induction on the height of the PA derivation of $\Gamma \vdash \Delta$. For the PA axioms, we note that for each $i \in \{1, \ldots, 6\}$ the sequent $(PAi) \vdash R(PAi)$ is derivable in LKID, and that $\vdash R(PA7)$ is provable in LKID using the induction rule (Ind N). The remaining cases follow straightforwardly by the induction hypothesis; the cases ($\exists$R) and ($\forall$L) require the use of Lemma 3.4.1. □

Theorem 3.4.4. *Eliminability of cut in LKID implies consistency of PA.*

Proof. Suppose PA is inconsistent, i.e., there is a proof of the empty sequent $\vdash$ in $\mathrm{LK_e}$ from the Peano axioms (PA1)–(PA7). By Lemma 3.4.3, $\vdash$ is then provable in LKID+ (PA1)–(PA6). Since (PA1)–(PA6) are closed first-order formulas, it follows that the sequent (PA1), ..., (PA6) $\vdash$ is provable in LKID, and thus cut-free provable by cut-eliminability in LKID (Theorem 3.3.14). But every rule of LKID, except (Cut), having an inductive predicate in one of its premises also has an inductive predicate in its conclusion. Since the sequent (PA1), ..., (PA6) $\vdash$ contains no inductive predicates, there are no instances of the rules for inductive predicates occurring anywhere in its cut-free proof. We thus have a cut-free proof of (PA1), ..., (PA6) $\vdash$ in the system $\mathrm{LK_e}$, and so $\vdash$ is derivable in $\mathrm{LK_e}$ from the axioms (PA1)–(PA6), i.e. the axioms (PA1)–(PA6) are inconsistent. But this system can be proved consistent by elementary means (see e.g. [29]). Hence we have the required contradiction and conclude PA is consistent. □

It is reasonably obvious that the proof of Lemma 3.4.3 and thus the proof of Theorem 3.4.4 given above can be formalised in PA, whence we can infer the following:

Corollary 3.4.5. *The eliminability of cut in LKID (Theorem 3.3.14) is not provable in PA.*

Proof. The eliminability of cut in LKID implies the consistency of PA by Theorem 3.4.4. If the eliminability of cut in LKID is provable in PA, then the consistency of PA would thus be provable in PA (via a formalisation in PA of the proof of Theorem 3.4.4). However, we know by Gödel's famous second incompleteness theorem that PA cannot prove its own consistency. □

Gentzen's original proof of PA-consistency [26], as well as modern versions (see e.g. [70]), requires the use of transfinite induction up to the ordinal ε_0; ordinals less than ε_0 are assigned to proofs and it is shown that if there is a proof of the empty sequent then there exists an infinite sequence of "reductions" between proofs such that each successive proof has a lesser ordinal than its predecessor, contradicting the well-ordering property of ε_0. Unfortunately, the fact that such proofs exploit the many special simplifying properties of the consistency problem suggests that the techniques they use are not, by themselves, sufficient to establish a full cut-elimination result for LKID.

There is an analogy between cut-elimination in LKID and normalisation in Gödel's system T [30]; the latter roughly corresponds to cut-elimination for an intuitionistic version of LKID with just the "natural number" predicate N. It has been established that normalization in system T can be proved by transfinite induction up to ε_0 (see e.g. [34, 84]). However, such proofs are significantly more complicated than proofs of PA-consistency. It seems plausible that a direct proof of cut-elimination for LKID along these lines would be more complex still, due to the fact that LKID is a classical system.

Chapter 4

LKID$^\omega$: a proof system for infinite descent in $\mathbf{FOL}_{ID}$

In this chapter we formulate a second proof system, LKID$^\omega$, for reasoning in the logic FOL_{ID}. Whereas the system LKID formalised proof by induction inside FOL_{ID} by adding explicit induction and unfolding rules for the inductively defined predicates of the underlying language, LKID$^\omega$ can instead be seen as formalising (an extension of) Fermat's notion of proof by "infinite descent". In Fermat's original formulation, infinite descent arguments exploit the fact that the natural numbers are well-founded, which is to say that there are no infinite strictly decreasing sequences of natural numbers. Thus any case in a proof leading to the construction of such a sequence can be ignored as contradictory.

For LKID$^\omega$, we generalise the idea above to arbitrary inductively defined predicates by replacing the induction rules of LKID by simple "casesplit" rules on inductive predicates, but allowing proofs to be infinite trees, i.e., to contain infinite branches. While such proofs are not sound in general, the well-foundedness of inductive definitions allows us to disregard any infinite branch along which some inductive definition is unfolded infinitely often. Thus, by requiring that all infinite branches satisfy this property, we ensure that the remaining portion of proof is well-founded, and hence sound. (However, to formulate this requirement precisely and provide a formal proof of soundness is somewhat more complex.)

In Section 4.1, we formulate the casesplit rules of LKID$^\omega$ which replace the LKID induction rules. Similar rules have been considered previously by Schroeder-Heister [59, 60]. Otherwise, the rules of LKID$^\omega$ are as for LKID, and proofs in the system are infinite trees (in general) satisfying a global condition which ensures soundness by an infinite descent argument, as discussed above. We formulate the global condition on infinite proofs and prove the soundness of LKID$^\omega$ in Section 4.2. In Section 4.3 we prove that the cut-free fragment of LKID$^\omega$ is complete, this time with respect to standard models. Thus the infinitary proof system LKID$^\omega$ is (unsurprisingly) more powerful than the finitary proof system LKID. As is usual,

the completeness argument involves the construction of a model in which a given underivable sequent is false. However, there are interesting twists in the argument, due to working with infinite proofs satisfying a global soundness condition. Cut-admissibility for LKID$^\omega$ is then obtained as a corollary of soundness and completeness, just as for LKID.

As for the previous chapter, we assume a fixed language Σ with exactly n inductive predicate symbols $P_1, \ldots, P_n$, and a fixed inductive definition set Φ for Σ.

4.1 Sequent calculus proof rules for LKID$^\omega$

The proof rules of the system LKID$^\omega$ are the rules of LKID described in Section 3.1, except that for each inductive predicate P_i of Σ, the induction rule (Ind P_i) of LKID is replaced by the following *casesplit rule*:

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

where for each production having predicate P_i in its conclusion, say:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1, \ldots, j_m, i \in \{1, \ldots, n\}$$

there is a corresponding case distinction of the form:

$$\Gamma, \mathbf{u} = \mathbf{t}(\mathbf{x}), Q_1\mathbf{u_1}(\mathbf{x}), \ldots, Q_h\mathbf{u_h}(\mathbf{x}), P_{j_1}\mathbf{t_1}(\mathbf{x}), \ldots, P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash \Delta$$

subject to the restriction that $x \notin FV(\Gamma \cup \Delta \cup \{P_i\mathbf{u}\})$ for all $x \in \mathbf{x}$. For any instance of (Case P_i), the formula $P_i\mathbf{u}$ occurring in the conclusion is said to be the *active formula* of the rule instance and the formulas $P_1\mathbf{t_1}(\mathbf{x}), \ldots, P_m\mathbf{t_m}(\mathbf{x})$ occurring in a case distinction are said to be *case-descendents* of $P\mathbf{u}$.

Example 4.1.1. The casesplit rule for the "natural number" predicate N defined in Example 2.2.5 is:

$$\frac{\Gamma, t = 0 \vdash \Delta \qquad \Gamma, t = sx, Nx \vdash \Delta}{\Gamma, Nt \vdash \Delta}\ (\text{Case } N)$$

where $x \notin FV(\Gamma \cup \Delta \cup \{Nt\})$.

Example 4.1.2. The casesplit rule for the "even number" predicate E defined mutually with the "odd number" predicate O in Example 2.2.6 is:

$$\frac{\Gamma, t = 0 \vdash \Delta \qquad \Gamma, t = sx, Ox \vdash \Delta}{\Gamma, Et \vdash \Delta}\ (\text{Case } E)$$

where $x \notin FV(\Gamma \cup \Delta \cup \{Et\})$.

Example 4.1.3. The casesplit rule for the predicate R^+ (the transitive closure of the predicate R) defined in Example 2.2.7 is:

$$\frac{\Gamma, t = x, u = y, Rxy \vdash \Delta \qquad \Gamma, t = x, u = z, R^+xy, R^+yz \vdash \Delta}{\Gamma, R^+tu \vdash \Delta}\ (\text{Case } R^+)$$

where $x, y, z \notin FV(\Gamma \cup \Delta \cup \{R^+tu\})$.

Lemma 4.1.4. *For any inductive predicate symbol P_i, the casesplit rule (Case P_i) is admissible in LKID.*

Proof. We show how to derive an instance of (Case P_i) whose conclusion is the sequent $\Gamma, P_i\mathbf{u} \vdash \Delta$. First, for each $j \in \{1, \ldots, n\}$, let k_j be the arity of the inductive predicate P_j and let $\mathbf{z_j}$ be a vector of k_j distinct variables. Then, to the predicate P_i we associate the induction variables $\mathbf{z_i}$ and the induction hypothesis formula: $F_i = (\bigwedge\Gamma \wedge \mathbf{u} = \mathbf{z_i} \rightarrow \bigvee\Delta) \wedge P_i\mathbf{z_i}$. Further, to all predicates P_j such that $P_j \neq P_i$ and P_i and P_j are mutually dependent, we associate the induction variables $\mathbf{z_j}$ and induction hypothesis formula $F_j = P_j\mathbf{z_j}$.

To begin the derivation of (Case P_i), we apply the rule (Ind P_i) to the conclusion of (Case P_i) with the induction variables and hypotheses given above. (As before, double line inferences are used to indicate uses of the weakening rule (Wk), and we use the rule symbol ($\equiv$) to indicate the expansion of an abbreviated formula.) We then prove the major premise of the induction rule instance outright as follows:

$$\dfrac{\text{minor deductions} \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\{\Gamma \vdash J \mid J \in \Gamma\}}(\text{Ax})}{\{\Gamma \vdash J \mid J \in \Gamma\}}(\wedge\text{R})}{\vdots}}{\Gamma \vdash \bigwedge\Gamma}(\wedge\text{R}) \qquad \dfrac{}{\Gamma \vdash \mathbf{u} = \mathbf{u}}(=\text{R})}{\Gamma \vdash \bigwedge\Gamma \wedge \mathbf{u} = \mathbf{u}}(\wedge\text{R}) \qquad \dfrac{\dfrac{\dfrac{}{\{K \vdash \Delta \mid K \in \Delta\}}(\text{Ax})}{\{K \vdash \Delta \mid K \in \Delta\}}(\vee\text{R})}{\vdots}}{\bigvee\Delta \vdash \Delta}(\vee\text{R})}{\Gamma, \bigwedge\Gamma \wedge \mathbf{u} = \mathbf{u} \rightarrow \bigvee\Delta, P_i\mathbf{u} \vdash \Delta}(\rightarrow\text{L})}{\Gamma, (\bigwedge\Gamma \wedge \mathbf{u} = \mathbf{u} \rightarrow \bigvee\Delta) \wedge P_i\mathbf{u} \vdash \Delta}(\wedge\text{L})}{\Gamma, F_i\mathbf{u} \vdash \Delta}(\equiv)}{\Gamma, P_i\mathbf{u} \vdash \Delta}\ (\text{Ind } P_i)$$

Note that there is a minor premise for every production having in its conclusion a predicate that is mutually dependent with P_i. However, we only require a case distinction for every production having P_i in its conclusion. Now, for every production $\Phi_{j,r} \in \Phi$ having in its conclusion a predicate P_j such that P_j and P_i are mutually dependent but $P_i \neq P_j$, say:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_j\mathbf{t}(\mathbf{x})} \qquad j_1, \ldots, j_m, j \in \{1, \ldots, n\}$$

we observe that the corresponding minor deduction in the instance of (Ind P_i) above can also be proved outright as follows:

$$
\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{}{\{Q_k\mathbf{u_k}(\mathbf{x}) \vdash Q_k\mathbf{u_k}(\mathbf{x}) \mid k \in \{1,\ldots,h\}\}}\,(\text{Ax}) \qquad \cfrac{}{\{P_{j_k}\mathbf{t_k}(\mathbf{x}) \vdash P_{j_k}\mathbf{t_k}(\mathbf{x}) \mid k \in \{1,\ldots,m\}\}}\,(\text{Ax})}{Q_1\mathbf{u_1}(\mathbf{x}),\ldots,Q_h\mathbf{u_h}(\mathbf{x}),P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash P_j\mathbf{t}(\mathbf{x})}\,(P_j\text{R}_r)}{\vdots}\,(\wedge\text{L})}{Q_1\mathbf{u_1}(\mathbf{x}),\ldots,Q_h\mathbf{u_h}(\mathbf{x}),G_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,G_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash P_j\mathbf{t}(\mathbf{x})}\,(\wedge\text{L})}{\Gamma,Q_1\mathbf{u_1}(\mathbf{x}),\ldots,Q_h\mathbf{u_h}(\mathbf{x}),G_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,G_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash P_j\mathbf{t}(\mathbf{x}),\Delta}\,(\text{Wk})
$$

Note that each G_{j_k} is by definition either the atomic formula $P_{j_k}\mathbf{z_{j_k}}$ or the induction hypothesis formula F_{j_k} defined above (c.f. the induction rule description in Section 3.1). In the latter case, note F_{j_k} is, again by definition, either the formula F_i or the atomic formula $P_{j_k}\mathbf{z_{j_k}}$. So each formula $G_{j_k}\mathbf{t_k}(\mathbf{x})$ in the above derivation is either the atomic formula $P_{j_k}\mathbf{t_k}(\mathbf{x})$ or the formula $F_i\mathbf{t_k}(\mathbf{x}$, i.e. $(\bigwedge\Gamma \wedge \mathbf{u} = \mathbf{t_k}(\mathbf{x}) \rightarrow \bigvee\Delta) \wedge P_i\mathbf{t_k}(\mathbf{x})$, in which case $P_i\mathbf{t_k}(\mathbf{x})$ can be obtained from it by applying ($\wedge$L) and (Wk). This justifies the transformation of each G_{j_k} into P_{j_k} via applications of ($\wedge$L) and weakenings in the derivation above.

Finally, for each production $\Phi_{i,r}$ having P_i in its conclusion, the corresponding minor deduction in the instance of (Ind P_i) above yields the desired case distinction as follows (we abbreviate the set of formulas $Q_1\mathbf{u_1}(\mathbf{x}),\ldots,Q_h\mathbf{u_h}(\mathbf{x})$ by $\mathcal{Q}$):

$$
\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\cfrac{\Gamma,\mathbf{u}=\mathbf{t}(\mathbf{x}),\mathcal{Q},P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash \Delta}{\vdots}\,(\vee\text{R})}{\Gamma,\mathbf{u}=\mathbf{t}(\mathbf{x}),\mathcal{Q},P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash \bigvee\Delta}\,(\vee\text{R})}{\vdots}\,(\wedge\text{L})}{\bigwedge\Gamma\wedge\mathbf{u}=\mathbf{t}(\mathbf{x}),\mathcal{Q},P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash \bigvee\Delta}\,(\wedge\text{L})}{\mathcal{Q},P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash \bigwedge\Gamma\wedge\mathbf{u}=\mathbf{t}(\mathbf{x}) \rightarrow \bigvee\Delta}\,(\rightarrow\text{R}) \qquad \cfrac{}{\mathcal{Q},P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash P_i\mathbf{t}(\mathbf{x})}\,(P_i\text{R}_r),(\text{Ax})}{\mathcal{Q},P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash (\bigwedge\Gamma\wedge\mathbf{u}=\mathbf{t}(\mathbf{x}) \rightarrow \bigvee\Delta)\wedge P_i\mathbf{t}(\mathbf{x})}\,(\wedge\text{R})}{\vdots}\,(\wedge\text{L})}{\mathcal{Q},G_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,G_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash (\bigwedge\Gamma\wedge\mathbf{u}=\mathbf{t}(\mathbf{x}) \rightarrow \bigvee\Delta)\wedge P_i\mathbf{t}(\mathbf{x})}\,(\wedge\text{L})}{\Gamma,\mathcal{Q},G_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,G_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash (\bigwedge\Gamma\wedge\mathbf{u}=\mathbf{t}(\mathbf{x}) \rightarrow \bigvee\Delta)\wedge P_i\mathbf{t}(\mathbf{x}),\Delta}\,(\text{Wk})
$$

Note that the justification for obtaining each P_{j_k} from G_{j_k} is as for the previous case. To complete the proof, we simply observe that we have obtained a derivation of the rule (Case P_i) within the system LKID. □

The local soundness of the proof rules of LKID$^\omega$ with respect to the Henkin semantics of FOL$_{\text{ID}}$ now follows immediately from the soundness of LKID:

Lemma 4.1.5 (Local Henkin soundness of LKID$^\omega$)**.** *Let $(M,\mathcal{H})$ be a Henkin model for (Σ,Φ). If all of the premises of a rule of LKID$^\omega$ are true in $(M,\mathcal{H})$, then the conclusion of the rule is also true in $(M,\mathcal{H})$.*

Proof. We just need to check that each rule of LKID$^\omega$ has the desired property. All of the rules apart from the casesplit rules are already covered by Lemma 3.2.2; the fact that the rule (Case P_i) is sound for each inductive predicate P_i follows from Lemma 4.1.4 and the fact that the rules of LKID are locally sound with respect to Henkin models (Lemma 3.2.2 again). □

We remark that, although the proof rules of LKID$^\omega$ are locally sound with respect to Henkin models of FOL$_{\mathrm{ID}}$, we will eventually obtain soundness of the full (infinitary) system LKID$^\omega$ only with respect to standard models. In order to establish this soundness property, we will only actually use the fact that the proof rules of LKID$^\omega$ are locally sound with respect to standard models.

It is worth noting that the casesplit rule for an inductive predicate P_i may also be formulated without the use of equality, as follows:

$$\frac{\text{case distinctions}}{\Gamma[\mathbf{u}/\mathbf{y}], P_i\mathbf{u} \vdash \Delta[\mathbf{u}/\mathbf{y}]}\ (\text{Case } P_i)(2)$$

where $\mathbf{y}$ is a vector of appropriately many variables and for each production having P_i in its conclusion, say:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1,\ldots,j_m,i \in \{1,\ldots,n\}$$

there is a corresponding case distinction:

$$\Gamma[\mathbf{t}(\mathbf{x})/\mathbf{y}], Q_1\mathbf{u_1}(\mathbf{x}),\ldots,Q_h\mathbf{u_h}(\mathbf{x}),P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash \Delta[\mathbf{t}(\mathbf{x})/\mathbf{y}]$$

subject to the restriction that $x \notin FV(\Gamma \cup \Delta \cup \{P_i\mathbf{u}\})$ for all $x \in \mathbf{x}$. In the presence of equality, the two formulations of the casesplit rule for an inductive predicate are interderivable (in LKID or LKID$^\omega$). The formulation of casesplit without equality is obtainable using the formulation with equality via the derivation:

$$\dfrac{\dfrac{\ldots\ \Gamma[\mathbf{t}(\mathbf{x})/\mathbf{y}], Q_1\mathbf{u_1}(\mathbf{x}),\ldots,Q_h\mathbf{u_h}(\mathbf{x}),P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash \Delta[\mathbf{t}(\mathbf{x})/\mathbf{y}]\ \ldots}{\ldots\ \Gamma[\mathbf{u}/\mathbf{y}], \mathbf{u}=\mathbf{t}(\mathbf{x}), Q_1\mathbf{u_1}(\mathbf{x}),\ldots,Q_h\mathbf{u_h}(\mathbf{x}),P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash \Delta[\mathbf{u}/\mathbf{y}]\ \ldots}\ (=\mathrm{L})}{\Gamma[\mathbf{u}/\mathbf{y}], P_i\mathbf{u} \vdash \Delta[\mathbf{u}/\mathbf{y}]}\ (\text{Case } P_i)$$

and the formulation using equality is derivable from the formulation by first using a cut:

$$\dfrac{\Gamma, \mathbf{u}=\mathbf{u}, P_i\mathbf{u} \vdash \Delta \quad \dfrac{}{\Gamma, P_i\mathbf{u} \vdash \mathbf{u}=\mathbf{u}, \Delta}\ (=\mathrm{R})}{\Gamma, P_i\mathbf{u} \vdash \Delta}\ (\mathrm{Cut})$$

and then continuing on the left branch as follows:

$$\frac{\ldots\ \Gamma, \mathbf{u}=\mathbf{t}(\mathbf{x}), Q_1\mathbf{u_1}(\mathbf{x}),\ldots,Q_h\mathbf{u_h}(\mathbf{x}),P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash \Delta\ \ldots}{\Gamma, \mathbf{u}=\mathbf{u}, P_i\mathbf{u} \vdash \Delta}\ (\text{Case } P_i)(2)$$

4.2 Infinite proofs in LKID$^\omega$

The main feature of LKID$^\omega$ is that we allow infinite derivation trees to be proofs. However, it is readily seen that there are infinite derivation trees in LKID$^\omega$ with invalid endsequents, so not every such tree is a proof. For this reason, we call infinite LKID$^\omega$ derivation trees *pre-proofs*:

Definition 4.2.1 (LKID$^\omega$ pre-proof)**.** An *LKID$^\omega$ pre-proof* of a sequent $\Gamma \vdash \Delta$ is a (possibly infinite) LKID$^\omega$ derivation tree $\mathcal{D} = (V, s, r, p)$ whose endsequent is $\Gamma \vdash \Delta$ and such that $\mathrm{Bud}(\mathcal{D}) = \emptyset$ (i.e. every sequent in the proof tree is the conclusion of some proof rule application and so $\mathcal{D}$ is "finished").

At this point, it may be helpful for the reader to recall our example of an infinite proof from the introduction:

Example 4.2.2. Let N, E and O be the "natural", "even" and "odd number" predicates given in Examples 2.2.5 and 2.2.6. (The casesplit rule for N is given in Example 4.1.1 above.) The following is then (an initial part of) an LKID$^\omega$ pre-proof of the sequent $Nx \vdash Ex, Ox$:

$$\dfrac{\dfrac{\dfrac{}{\vdash E0, O0}\,(E\mathrm{R}_1)}{x = 0 \vdash Ex, Ox}\,(=\mathrm{L}) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\vdash O0, E0}\,(E\mathrm{R}_1)}{y = 0 \vdash Oy, Ey}\,(=\mathrm{L}) \qquad \dfrac{\dfrac{\dfrac{\vdots\ \text{(etc.)}}{Nz \vdash Osz, Esz}}{y = sz, Nz \vdash Oy, Ey}\,(=\mathrm{L})}{Ny \vdash Oy, Ey}\,(\text{Case } N)}{Ny \vdash Oy, Osy}\,(O\mathrm{R}_1)}{Ny \vdash Esy, Osy}\,(E\mathrm{R}_2)}{x = sy, Ny \vdash Ex, Ox}\,(=\mathrm{L})}{Nx \vdash Ex, Ox}\,(\text{Case } N)$$

Whereas when dealing with LKID we were concerned with Henkin validity of sequents, i.e. truth in all Henkin models of (Σ, Φ), for LKID$^\omega$ we shall only be concerned with the standard notion of validity, i.e. truth in all *standard* models. We shall say a sequent is *valid* just in case it is true in all standard models of (Σ, Φ). Of course, in a finite derivation tree, the validity of the endsequent is guaranteed by the local soundness of the proof rules (see e.g. Proposition 3.2.8). However, this argument does not extend to LKID$^\omega$ pre-proofs as they may be non-well-founded, i.e. contain infinite branches. In this situation, if the endsequent of the pre-proof is false in some model M, the local soundness of the proof rules implies the existence of an infinite path through the pre-proof such that each sequent along the path is also false in M (under some environment). Our aim is to formulate a soundness condition on pre-proofs that ensures that we can obtain a logical contradiction in such a circumstance.

Definition 4.2.3 (Trace)**.** Let (v_i) be a path in an LKID$^\omega$ pre-proof $\mathcal{D} = (V, s, r, p)$. A *trace following* (v_i) is a sequence (τ_i) such that, for all i:

- $\tau_i = P_{j_i}\mathbf{t_i} \in \Gamma_i$, where P_{j_i} is an inductive predicate (i.e. $j_i \in \{1,\ldots,n\}$) and $s(v_i) = \Gamma_i \vdash \Delta_i$;
- if $r(v_i)$ is (Subst) then $\tau_i = \tau_{i+1}[\theta]$, where θ is the substitution associated with the rule instance;
- if $r(v_i)$ is (=L) with active formula $t = u$ then there exists a formula F and variables x and y such that $\tau_i = F[t/x, u/y]$ and $\tau_{i+1} = F[u/x, t/y]$ (i.e. τ_{i+1} is obtained by (possibly) swapping some occurrences of t and u in τ_i);
- if $r(v_i)$ is some casesplit rule (Case P_k) then either $\tau_{i+1} = \tau_i$, or 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;
- if $r(v_i)$ is not (Subst), (=L) or a casesplit rule, then $\tau_{i+1} = \tau_i$.

As mentioned above, given the invalidity of the endsequent of an LKID$^\omega$ pre-proof $\mathcal{D}$ we can construct an infinite path in $\mathcal{D}$ and an infinite sequence of falsifying environments for the sequents on this path. Furthermore, any trace on (part of) this path can be understood as a monotonically decreasing sequence of ordinals via the approximant construction for inductive predicates in standard models (c.f. Definition 2.2.8). This property is made precise in the following lemma:

Lemma 4.2.4. *Let $\mathcal{D}$ be an LKID$^\omega$ pre-proof of $\Gamma_0 \vdash \Delta_0$, and let M be a standard model such that $\Gamma_0 \vdash \Delta_0$ is false in M under the environment ρ_0 (say). Then there is an infinite path $(v_i)_{i\geq 0}$ in $\mathcal{D}$ and an infinite sequence $(\rho_i)_{i\geq 0}$ of environments such that:*

1. *for all i, $s(v_i) = \Gamma_i \vdash \Delta_i$ is false in M under ρ_i;*
2. *if there is a trace $(\tau_i = P_{j_i}\mathbf{t_i})_{i\geq n}$ following some tail $(v_i)_{i\geq n}$ of $(v_i)_{i\geq 0}$, then the sequence $(\alpha_i)_{i\geq n}$ of ordinals defined by:*

$$\alpha_i = \textit{least ordinal } \alpha \textit{ such that } \rho_i(\mathbf{t_i}) \in P^\alpha_{j_i}$$

for all $i \geq n$, is a non-increasing sequence. Furthermore, if j is a progress point of $(\tau_i)_{i\geq n}$ then $\alpha_{j+1} < \alpha_j$.

Proof. Throughout this proof, we write $\Gamma \not\models_\rho \Delta$ to mean that the sequent $\Gamma \vdash \Delta$ is false in the model M under the environment ρ.

First note that the ordinal sequence $(\alpha_i)_{i\geq n}$ defined in property 2 of the lemma is well-defined, for, by the definition of trace, we have $\tau_i = P_{j_i}\mathbf{t_i} \in \Gamma_i$ for each $i \geq n$, and since $\Gamma_i \not\models_{\rho_i} \Delta_i$ for all i by property 1 of the lemma we must have $\models_{\rho_i} P_{j_i}\mathbf{t_i}$, i.e. $\rho_i(\mathbf{t_i}) \in \bigcup_\alpha P^\alpha_{j_i}$, for each $i \geq n$. Now $\rho_i(\mathbf{t_i}) \in \bigcup_\alpha P^\alpha_{j_i}$ iff $\rho_i(\mathbf{t_i}) \in P^\alpha_{j_i}$ for some ordinal α, and there is a least such α by the well-ordering property of the ordinals, so α_i is defined for each $i \geq n$.

The two properties required by the lemma are trivially true of the 1-element sequences $(v_0 = root(\mathcal{D}))$ and (ρ_0). We assume we have sequences $(v_i)_{0\leq i\leq k}$ and $(\rho_i)_{0\leq i\leq k}$ satisfying the two properties of the lemma and show how to construct v_{k+1} and ρ_{k+1}. We always choose v_{k+1} to be $p_j(v_k)$ for some j (i.e. such that $s(v_{k+1})$ is a premise of the rule instance of which $s(v_k)$ is the conclusion) so that $(v_i)_{i\geq 0}$ is an infinite path in $\mathcal{D}$ as required. To establish that property 2 holds of the constructed sequence, it suffices to assume the existence of an arbitrary trace (τ_k, τ_{k+1}) following (v_k, v_{k+1}), and show that $\alpha_{k+1} \leq \alpha_k$, and that if k is a progress point of the trace then $\alpha_{k+1} < \alpha_k$. It is clear that this process can be repeated infinitely often, thus yielding the required infinite sequences.

We note that since $\mathcal{D}$ is an LKID$^\omega$ derivation tree, the sequent $s(v_k)$ is the conclusion of an instance of the LKID$^\omega$ rule $r(v_k)$, which clearly cannot be a rule with no premises, as the conclusion of every such rule is a valid sequent by Lemma 4.1.5. We therefore distinguish a case for each of the remaining proof rules. In all cases the falsifiability of v_{k+1} by ρ_{k+1} follows immediately from the local soundness of the proof rule in question (Lemma 4.1.5). Furthermore, ρ_{k+1} can always be constructed in a manner consistent with the requirement for the second property. We examine only the interesting cases here:

Case $r(v_k) =$ (Subst): $s(v_k)$ is the conclusion $\Gamma[\theta] \vdash \Delta[\theta]$ of an instance of (Subst) and by induction hypothesis $\Gamma[\theta] \not\models_{\rho_k} \Delta[\theta]$. We choose v_{k+1} to be the node of $\mathcal{D}$ labelled with the only premise $\Gamma \vdash \Delta$ of the rule instance, and choose $\rho_{k+1} = \rho_k \circ \theta$. The sequent $s(v_{k+1})$ is thus falsifiable by ρ_{k+1} (satisfying property 1) since $\Gamma \not\models_{\rho_k\circ\theta} \Delta$ iff $\Gamma[\theta] \not\models_{\rho_k} \Delta[\theta]$ by part 2 of Lemma 2.1.9.

For property 2, we must have by the the definition of trace that $\tau_k = \tau_{k+1}[\theta]$, i.e. if $\tau_k = P_i\mathbf{t}$ then $\tau_{k+1} = P_i\mathbf{t}'$ and $\mathbf{t} = \mathbf{t}'[\theta]$. Note that $\rho_{k+1}(\mathbf{t}') = (\rho_k \circ \theta)(\mathbf{t}') = \rho_k(\mathbf{t}'[\theta]) = \rho_k(\mathbf{t})$. Thus, for any α, $\rho_{k+1}(\mathbf{t}') \in P_i^\alpha$ iff $\rho_k(\mathbf{t_k}) \in P_i^\alpha$ and so $\alpha_{k+1} = \alpha_k$.

Case $r(v_k) =$ (=L): $s(v_k)$ is the conclusion $\Gamma[t/x,u/y], t = u \vdash \Delta[t/x,u/y]$ of an instance of (=L) and by induction hypothesis $\Gamma[t/x,u/y], t = u \not\models_{\rho_k} \Delta[t/x,u/y]$. In particular, $M \models_{\rho_k} t = u$, so $\rho_k(t) = \rho_k(u)$. We choose v_{k+1} to be the node of $\mathcal{D}$ labelled with the only premise $\Gamma[u/x,t/y] \vdash \Delta[u/x,t/y]$ of the rule instance, and choose $\rho_{k+1} = \rho_k$. It then follows that $\Gamma[u/x,t/y] \not\models_{\rho_{k+1}} \Delta[u/x,t/y]$ since $\rho_{k+1}(t) = \rho_k(t) = \rho_k(u) = \rho_{k+1}(u)$, so property 1 of the lemma is satisfied.

For property 2 of the lemma, we must have that by the definition of trace that τ_{k+1} and τ_k are the same atomic formula, up to some possible swappings of occurrences of t and u. So we can write $\tau_{k+1} = P_i\mathbf{t_1}$ and $\tau_k = P_i\mathbf{t_2}$, where $\rho_k(\mathbf{t_1}) = \rho_k(\mathbf{t_2})$ since $\rho_k(t) = \rho_k(u)$, and so $\rho_{k+1}(\mathbf{t_1}) = \rho_k(\mathbf{t_2})$. Thus, for any α, $\rho_{k+1}(\mathbf{t_1}) \in P_i^\alpha$ iff $\rho_k(\mathbf{t_2}) \in P_i^\alpha$ and so $\alpha_{k+1} = \alpha_k$ as required.

Case $r(v_k) =$ (Case P_i): $s(v_k)$ is the conclusion $\Gamma, P_i\mathbf{u} \vdash \Delta$ of an application of rule (Case P_i)

(where $i \in \{1,\ldots,n\}$). As $s(v_k)$ is false in M under ρ_k by induction hypothesis, we have $M \models_{\rho_k} P_i\mathbf{u}$, i.e. $\rho_k(\mathbf{u}) \in \bigcup_\alpha P_i^\alpha$. Let α' be the least ordinal α such that $\rho_k(\mathbf{u}) \in P_i^\alpha = \pi_i^n(\varphi_\Phi^\alpha)$. By Definition 2.2.8 we thus have $\rho_k(\mathbf{u}) \in \pi_i^n(\bigcup_{\beta<\alpha'} \varphi_\Phi(\varphi_\Phi^\beta))$. By construction of φ_Φ (c.f. Definition 2.2.3), there is then a $\beta < \alpha'$ and a production $\Phi_{i,r} \in \Phi$ such that $\rho_k(\mathbf{u}) \in \varphi_{i,r}(\varphi_\Phi^\beta)$. Now $\Phi_{i,r}$ is a production with P_i in its conclusion, say:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1,\ldots,j_m,i \in \{1,\ldots,n\}$$

so by definition of $\varphi_{i,r}$ we have:

$$\rho_k(\mathbf{u}) \in \{\mathbf{t}^M(\mathbf{d}) \mid Q_1^M\mathbf{u_1}^M(\mathbf{d}),\ldots,Q_h^M\mathbf{u_h}^M(\mathbf{d}),\mathbf{t_1}^M(\mathbf{d}) \in \pi_{j_1}^n(\varphi_\Phi^\beta),\ldots,\mathbf{t_m}^M(\mathbf{d}) \in \pi_{j_m}^n(\varphi_\Phi^\beta)\}$$

$$\text{i.e.} \quad \exists\mathbf{d}.\ \rho_k(\mathbf{u}) = \mathbf{t}^M(\mathbf{d}) \text{ and } Q_1^M\mathbf{u_1}^M(\mathbf{d}),\ldots,Q_h^M\mathbf{u_h}^M(\mathbf{d}),\mathbf{t_1}^M(\mathbf{d}) \in P_{j_1}^\beta,\ldots,\mathbf{t_m}^M(\mathbf{d}) \in P_{j_m}^\beta \ (*)$$

Now define v_{k+1} to be the node of $\mathcal{D}$ such that $s(v_{k+1})$ is the (case distinction) premise:

$$\Gamma, \mathbf{u} = \mathbf{t}(\mathbf{x}), Q_1\mathbf{u_1}(\mathbf{x}),\ldots,Q_h\mathbf{u_h}(\mathbf{x}),P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash \Delta$$

where $x \notin FV(\Gamma \cup \Delta \cup \{P_i\mathbf{u}\})$ for all $x \in \mathbf{x}$, and define $\rho_{k+1} = \rho_k[\mathbf{x} \mapsto \mathbf{d}]$. For property 1 we need to show $s(v_{k+1})$ is falsified by ρ_{k+1}. It is clear that we have $M \models_{\rho_{k+1}} J$ for all $J \in \Gamma$ and $M \not\models_{\rho_{k+1}} K$ for all $K \in \Delta$ by the induction hypothesis and part 1 of Lemma 2.1.5, since ρ_{k+1} agrees with ρ_k on all variables free in $\Gamma \cup \Delta$. Also by part 1 of Lemma 2.1.5, we have $\rho_{k+1}\mathbf{u} = \rho_k\mathbf{u}$ since $x \notin FV(P_i\mathbf{u}) = Var(\mathbf{u})$ for all $x \in \mathbf{x}$. Now $\rho_{k+1}(\mathbf{t}(\mathbf{x})) = \mathbf{t}^M(\rho_{k+1}(\mathbf{x})) = \mathbf{t}^M(\mathbf{d}) = \rho_k(\mathbf{u})$ by the definition of ρ_{k+1} and the statement (*) above, so we have $\rho_{k+1}(\mathbf{u}) = \rho_{k+1}(\mathbf{t}(\mathbf{x}))$, i.e. $M \models_{\rho_{k+1}} \mathbf{u} = \mathbf{t}(\mathbf{x})$ as required. We then just need to show each of $M \models_{\rho_{k+1}} Q_1\mathbf{u_1}(\mathbf{x}),\ldots,M \models_{\rho_{k+1}} Q_h\mathbf{u_h}(\mathbf{x}),M \models_{\rho_{k+1}} P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,M \models_{\rho_{k+1}} P_{j_m}\mathbf{t_m}(\mathbf{x})$, which is clear from the statement $(*)$ above together with the definition of ρ_{k+1}. For property 2, there are two possibilities to consider:

- k is not a progress point of the trace (τ_k, τ_{k+1}) and so, by the definition of trace, we have $\tau_{k+1} = \tau_k$. Now, since $\tau_k = P_j\mathbf{t}$ (say) is a formula occurring in $s(v_k) = \Gamma, P_i\mathbf{u} \vdash \Delta$, and ρ_{k+1} agrees with ρ_k on variables free in $\Gamma \cup \Delta \cup \{P_i\mathbf{u}\}$, we have $\rho_{k+1}(\mathbf{t}) = \rho_k(\mathbf{t})$ and so $\rho_k(\mathbf{t}) \in P_j^\alpha$ iff $\rho_{k+1}(\mathbf{t}) \in P_j^\alpha$, i.e. $\alpha_{k+1} = \alpha_k$ and we are done.

- k is a progressing position of the trace (τ_k, τ_{k+1}). In that case, τ_k is the active formula $P_i\mathbf{u}$ of the rule instance and τ_{k+1} is a case-descendant of $P_i\mathbf{u}$, i.e. τ_{k+1} is one of the formulas $P_{j_1}\mathbf{t_1}(\mathbf{x}),\ldots,P_{j_m}\mathbf{t_m}(\mathbf{x})$. Now the discussion above shows that there is an ordinal β such that $\rho_{k+1}(\mathbf{t_1}(\mathbf{x})) \in P_{j_1}^\beta,\ldots,\rho_{k+1}(\mathbf{t_m}(\mathbf{x})) \in P_{j_m}^\beta$ and that furthermore, β is smaller than the least ordinal α satisfying $\rho_k(\mathbf{t}) \in P_i^\alpha$. We thus have $\alpha_{k+1} < \alpha_k$ as required.

□

Lemma 4.2.4 immediately gives rise to a natural soundness criterion for LKID$^\omega$ pre-proofs:

Definition 4.2.5 (LKID$^\omega$ proof)**.** An LKID$^\omega$ pre-proof $\mathcal{D}$ is said to be an LKID$^\omega$ *proof* if, for every infinite path in $\mathcal{D}$, there is an infinitely progressing trace following some tail of the path.

For convenience, we may restrict the quantification over all infinite paths in Definition 4.2.5 to all *rooted* infinite paths. It is obvious that there is an infinitely progressing trace on a tail of every infinite path in a pre-proof if and only if there is one on the tail of every infinite path starting from the root of the pre-proof, because every vertex is reachable from it.

Proposition 4.2.6 (Soundness of LKID$^\omega$)**.** *If there is an LKID$^\omega$ proof of $\Gamma \vdash \Delta$ then $\Gamma \vdash \Delta$ is valid (i.e. true in all standard models for (Σ, Φ)).*

Proof. Suppose that we have an LKID$^\omega$ proof $\mathcal{D}$ of $\Gamma \vdash \Delta$ but $\Gamma \vdash \Delta$ is not valid, i.e. there is a standard model M of (Σ, Φ) such that $\Gamma \vdash \Delta$ is not true in M. We can then apply Lemma 4.2.4 to obtain infinite sequences $(v_i)_{i\geq 0}$ and $(\rho_i)_{i\geq 0}$ satisfying properties 1 and 2 of the Lemma. By the definition of LKID$^\omega$ proof, as $(v_i)_{i\geq 0}$ is an infinite path through $\mathcal{D}$, there is an infinitely progressing trace following some tail of $(v_i)_{i\geq 0}$. So by property 2 of the Lemma, there is thus a non-increasing sequence of ordinals $(\alpha_i)_{i\geq n}$ that decreases infinitely often. As this contradicts the well-foundedness property of the ordinals we obtain the required contradiction and conclude that $\Gamma \vdash \Delta$ must be valid. □

4.2.1 Generalised trace-based infinite proof systems

The ideas used to develop the proof system LKID$^\omega$ for FOL$_{\text{ID}}$ above have been used previously to develop (sound) infinite proof systems for other logics, notably the μ-calculus (see e.g. [63, 58, 19, 52]). To show the versatility of the method, we now consider an essentially arbitrary infinite proof system S$^\omega$ and show that a sound notion of infinite proof exists for the system providing that an appropriate notion of "trace" exists. We shall make only the following general assumptions about the proof system S$^\omega$:

- The rules of S$^\omega$ are of the form:
$$\frac{S_1 \dots S_n}{S}\,(R)$$
where $n \in \mathbb{N}$. $S, S_1, \dots, S_n$ are called "sequents". We write *Seqs* for the set of well-formed sequents of S$^\omega$ and *Rules* for the set of proof rules of S$^\omega$.

- There is a set $\mathbb{I}$ of *interpretations*, and we have a semantic notion of satisfaction between interpretations and sequents. We write $I \models S$ to mean that the sequent S is satisfied by the interpretation I. We say a sequent S is *valid* if S is satisfied by every interpretation I. (Observe that for our systems LKID and LKID$^\omega$, an interpretation of a sequent is given by a first-order structure M and an environment ρ. In many systems, such as μ-calculus, an interpretation will just be a environment interpreting variables in some domain.)

An S$^\omega$ pre-proof is then just a possibly infinite S$^\omega$ derivation tree (with no buds). In general, as we saw previously, such pre-proofs are sound if the invalidity of the endsequent implies the existence of an infinite path in the proof from which an infinite descending chain of ordinals can be constructed. This motivates the formulation of the following definition of a *trace function*, which generalises the notion of trace for LKID$^\omega$ from Definition 4.2.3:

Definition 4.2.7 (Trace function)**.** Let $\mathcal{T}$ be a set, and let $TVal \subseteq \mathcal{T} \times Seqs$ satisfy: for any $S \in Seqs$, there are only finitely many $\tau \in \mathcal{T}$ such that $TVal(\tau, S)$. Let $TPair : (\mathcal{T} \times \mathcal{T}) \to (Seqs \times Rules \times Seqs) \to \{0,1,2\}$ be a computable function satisfying:

$$\forall \tau, \tau', s, r, s'. \neg TVal(\tau, s) \vee \neg TVal(\tau', s') \Rightarrow TPair(\tau, \tau')(s, r, s') = 0$$

Also suppose there exists a function $\sigma : \mathcal{T} \times \mathbb{I} \to O$, where O is some initial segment of the ordinals, such that for any S$^\omega$ pre-proof $\mathcal{D} = (V, s, r, p)$:

$$\begin{array}{lll} I \not\models s(v) & \Rightarrow & \exists I', v', j.\ v' = p_j(v) \\ & \text{and} & I' \not\models s(v') \\ & \text{and} & TPair(\tau, \tau')(s(v), r(v), s(v')) = 1 \Rightarrow \sigma(\tau', I') \leq \sigma(\tau, I) \\ & \text{and} & TPair(\tau, \tau')(s(v), r(v), s(v')) = 2 \Rightarrow \sigma(\tau', I') < \sigma(\tau, I) \end{array}$$

Then $TVal$ is called a *trace value relation* and $TPair$ is called a *trace pair function* for the proof system, and σ is the *ordinal trace function* associated with $TPair$ and $TVal$. (τ, τ') is said to be a *valid trace pair on* (v, v') if $TPair(\tau, \tau')(s(v), r(v), s(v')) \neq 0$, and is said to be a *progressing trace pair* on (v, v') if $TPair(\tau, \tau')(s(v), r(v), s(v')) = 2$. Further, a sequence (τ_i) is said to be a *trace* on a path (v_i) if for all i, (τ_i, τ_{i+1}) is a valid trace pair on (v_i, v_{i+1}). A trace (τ_i) is said to *progress* at j if (τ_j, τ_{j+1}) is a progressing trace pair, and is said to be *infinitely progressing* if there are infinitely many points at which it progresses.

Notice that, due to the properties required of a trace pair function, a trace pair function exists for S$^\omega$ only if the rules of S$^\omega$ are locally sound. We shall suppose there exists a trace pair function for the proof system S$^\omega$, in which case an analogous version of Lemma 4.2.4 holds for S$^\omega$:

Lemma 4.2.8. *Let $\mathcal{D}$ be a S$^\omega$ pre-proof of a sequent S_0, and suppose that for some interpretation I_0, S_0 is false in I_0. Then there is an infinite path $(v_i)_{i \geq 0}$ in $\mathcal{D}$ and an infinite sequence I_0 of interpretations such that:*

1. *for all i, $I_i \not\models s(v_i)$;*
2. *if there is a trace $(\tau_i)_{i \geq n}$ following some tail $(v_i)_{i \geq n}$ of $(v_i)_{i \geq 0}$, then the sequence of ordinals $(\alpha_i)_{i \geq n}$ defined by $\alpha_i = \sigma(\tau_i, I_i)$, where σ is the ordinal trace function associated with the trace pair function for S$^\omega$, is non-increasing. Furthermore, if j is a progress point of (τ_i) then $\alpha_{j+1} < \alpha_j$.*

Proof. First note that $s(root(\mathcal{D})) = S_0$ is false in I_0 by supposition, so the sequences $(v_0 = root(\mathcal{D}))$ and (I_0) trivially satisfy the properties required by the lemma. We inductively assume that we have constructed a path $(v_i)_{0\leq i\leq k}$ in $\mathcal{D}$ and a sequence $(I_i)_{0\leq i\leq k}$ of interpretations satisfying the two properties of the lemma. In particular, $s(v_k)$ is false in I_k, so by the definition of a trace pair function, we can find v_{k+1} such that (v_k, v_{k+1}) is an edge of $\mathcal{G}_\mathcal{P}$ and $I_{k+1} \not\models s(v_{k+1})$. Furthermore, if there is a trace following some tail $(v_i)_{n\leq i\leq k+1}$ of this path, then there is a trace $(\tau_i)_{n\leq i\leq k}$ following $(v_i)_{n\leq i\leq k}$ and a trace pair (τ_k, τ_{k+1}) on (v_k, v_{k+1}). By induction hypothesis there is a non-increasing sequence $(\alpha_i)_{n\leq i\leq k}$ that decreases whenever the trace progresses. By the definition of trace pair function, we have $\alpha_{k+1} = \sigma(\tau_{k+1}, I_{k+1}) \leq \sigma(\tau_k, I_k) = \alpha_k$ since (τ_k, τ_{k+1}) is a valid trace pair on (v_k, v_{k+1}). Furthermore, if k is a progressing point of the trace, then (τ_k, τ_{k+1}) is a progressing trace pair, and we have $\alpha_{k+1} < \alpha_k$ by the definition of trace pair function as required. This completes the induction step. □

Definition 4.2.9 (Generalised infinite proof)**.** An S$^\omega$ pre-proof $\mathcal{D}$ is said to be an S$^\omega$ *proof* if, for every infinite path in $\mathcal{D}$, there is an infinitely progressing trace following some tail of the path.

Proposition 4.2.10 (Soundness of generalised infinite proof)**.** *If there is a S$^\omega$ proof of S then S is valid.*

Proof. Suppose that we have a S$^\omega$ proof $\mathcal{D}$ of S but S is not valid, i.e. there is an interpretation I_0 such that $I_0 \not\models S$. We can then apply Lemma 4.2.8 to obtain infinite sequences (v_i) and (I_i) satisfying properties 1 and 2 of the lemma. By the definition of S$^\omega$ proof, as (v_i) is an infinite path through $\mathcal{D}$, there is an infinitely progressing trace following some tail of (v_i). So by property 2 of the lemma, there is thus a non-increasing sequence of ordinals $(\alpha_i)_{i\geq n}$ that decreases infinitely often. As this contradicts the well-foundedness property of the ordinals we obtain the required contradiction and conclude that S must be valid. □

4.3 Cut-free completeness of LKID$^\omega$

We now present a proof of cut-free completeness for LKID$^\omega$ with respect to standard models. The structure of the proof is similar to that of the cut-free Henkin completeness proof of LKID given previously (c.f. Section 3.3), but without the need to construct Henkin models. However, additional complications arise from the need to consider the soundness condition imposed on LKID$^\omega$ proofs.

As for the LKID completeness proof, we shall say that a sequent is *cut-free provable* (in LKID$^\omega$) if there is an LKID$^\omega$ proof of the sequent that does not include any instances of the rules (Cut), (Subst) or (Wk), and we shall show that every valid sequent (with respect to standard models) is cut-free provable in this sense. Thus, as before, we shall obtain from our sound-

ness and completeness results very slightly more than the usual notion of cut-eliminability, but the rules (Wk) and (Subst) are in any case easily shown admissible in LKID$^\omega$. Since LKID$^\omega$ proofs are infinite objects, it is of interest to consider the structural complexity of proof objects. In this regard, we show that every valid sequent actually has *recursive* cut-free LKID$^\omega$ proof, which is to say that the LKID$^\omega$ proof is a recursive tree in the standard sense (for a definition see e.g. [55]). The outline of the cut-free completeness proof for LKID$^\omega$ is as follows:

1. Given an arbitrary (possibly provable) sequent $\Gamma \vdash \Delta$ we construct a recursive *search tree* corresponding to an exhaustive infinitary search for a cut-free LKID$^\omega$ proof of $\Gamma \vdash \Delta$. This tree is constructed according to a schedule similar to that used in the LKID completeness proof, except that we also schedule the application of the rules for equality and for right-unfolding of inductive predicates.

2. If the search tree for $\Gamma \vdash \Delta$ is not an LKID$^\omega$ proof, then either it is not a pre-proof, and thus contains a bud node, or it contains an *untraceable branch* — an infinite path such that there is no infinitely progressing trace on any tail of the path. In the former case, we can straightforwardly construct a model in which $\Gamma \vdash \Delta$ is false. We continue with the remaining case.

3. We may now assume that the search tree contains an untraceable branch. We can then prove that no sequent on this branch, including $\Gamma \vdash \Delta$, is cut-free provable, by showing essentially that the existence of a cut-free proof would imply the existence of an infinitely progressing trace on the untraceable branch, which is a contradiction.

4. The untraceable branch in the search tree is used to construct a *limit sequent* $\Gamma_\omega \vdash \Delta_\omega$, which in turn is used to construct a first-order structure M_ω and an environment ρ_ω. The construction of this *counter-interpretation* is very similar to the construction in the LKID completeness proof, except that the interpretation of the inductive predicates in M_ω is defined so as to ensure that M_ω is a standard model for (Σ, Φ).

5. We show that $\Gamma_\omega \vdash \Delta_\omega$ is false in M_ω under the environment ρ_ω. This proof is identical to the analogous step in the LKID completeness proof except in the case where an inductive predicate formula occurs in Γ_ω, where the argument needs an extra twist relying on the soundness condition. In particular, $\Gamma \vdash \Delta$ is false in the standard model M_ω since it is a subsequent of $\Gamma_\omega \vdash \Delta_\omega$.

6. Now given any sequent $\Gamma \vdash \Delta$, if the search tree for $\Gamma \vdash \Delta$ is not an LKID$^\omega$ proof then it follows by steps 2–5 that $\Gamma \vdash \Delta$ is invalid. Thus if $\Gamma \vdash \Delta$ *is* valid, then the search tree is a recursive cut-free LKID$^\omega$ proof of $\Gamma \vdash \Delta$.

Definition 4.3.1 (Schedule)**.** An *LKID$^\omega$-schedule element* of Σ is defined as follows:

- any formula of the form $\neg F$, $F_1 \wedge F_2$, $F_1 \vee F_2$, or $F_1 \to F_2$ is an LKID$^\omega$-schedule element;
- for any term t of Σ, any variable $x \in \mathcal{V}$ and any formula F, the pairs $\langle \forall xF, t \rangle$ and $\langle \exists xF, t \rangle$ are LKID$^\omega$-schedule elements;
- for any terms t, u of Σ, any variables $x, y \in \mathcal{V}$, and any finite multisets of formulas Γ, Δ, the tuple $\langle t = u, x, y, \Gamma, \Delta \rangle$ is an LKID$^\omega$-schedule element;
- for any inductive predicate symbol P_i of arity k, any terms $t_1, \ldots, t_k$ of Σ, and for every production $\Phi_{i,r} \in \Phi$, say:

$$\frac{Q_1 \mathbf{u_1}(\mathbf{x}) \ldots Q_h \mathbf{u_h}(\mathbf{x}) \quad P_{j_1} \mathbf{t_1}(\mathbf{x}) \ldots P_{j_m} \mathbf{t_m}(\mathbf{x})}{P_i \mathbf{t}(\mathbf{x})} \qquad j_1, \ldots, j_m, i \in \{1, \ldots, n\}$$

  such that $(t_1, \ldots, t_k) = \mathbf{t}(\mathbf{t'})$ for some $\mathbf{t'}$, the pair $\langle P_i(t_1, \ldots, t_k), r \rangle$ is an LKID$^\omega$-schedule element.

An *LKID$^\omega$-schedule* for Σ is then an enumeration $(E_i)_{i \geq 0}$ of schedule elements of Σ such that every schedule element of Σ appears infinitely often in the enumeration.

An LKID$^\omega$-schedule is similar to an LKID-schedule (c.f. Definition 3.3.1) with two major differences. First, we have schedule elements for equality formulas, with appropriate extra information enabling the left rule for equality to be applied. Second, the extra information stored with atomic formulas involving inductive predicates is now simply an index which allows a particular right-unfolding rule for the inductive predicate to be selected. (The extra condition on this index ensures that only the indices of right-rules which can be applied to the inductive predicate formula are considered.) The justification for the existence of an LKID$^\omega$-schedule is similar to that for an LKID-schedule.

Definition 4.3.2 (Search tree)**.** Let $\Gamma \vdash \Delta$ be an arbitrary (possibly provable) sequent; note that there is an LKID$^\omega$-schedule for Σ.

We then define an infinite sequence of $(T_i)_{i \geq 0}$ of LKID$^\omega$ derivation trees such that T_i is a subtree of T_{i+1} for all $i \geq 0$. Moreover, each T_i has the property that the sequent at any node v of T_i is a subsequent of the sequent at any descendant of v. We define T_0 to be the single-node tree $\Gamma \vdash \Delta$ so, in particular, the endsequent of T_i will be $\Gamma \vdash \Delta$ for all $i \geq 0$. We inductively assume we have constructed the tree T_j and show how to construct T_{j+1}.

In general T_{j+1} will be obtained by replacing certain bud nodes of T_j with finite LKID$^\omega$ derivation trees, whence it is clear that T_{j+1} is also a finite LKID$^\omega$ derivation tree as required. Firstly, we replace any bud $\Gamma' \vdash \Delta'$ of T_j such that $\Gamma' \cap \Delta' \neq \emptyset$ with the derivation tree:

$$\frac{}{\Gamma' \vdash \Delta'}\,(\mathrm{Ax})$$

and we likewise replace any bud $\Gamma' \vdash \Delta'$ such that $t = t \in \Delta'$ for some term t with the derivation tree:

$$\frac{}{\Gamma' \vdash \Delta'}\,(=\text{R})$$

We then proceed by case distinction on E, the $(j+1)$th element in the schedule for Σ:

- $E = \neg F$. Then we replace every bud node $\Gamma' \vdash \Delta'$ such that $\neg F \in \Gamma'$ with the derivation tree:
$$\dfrac{\dfrac{\Gamma' \vdash F, \Delta'}{\Gamma', \neg F \vdash \Delta'}\,(\neg\text{L})}{\Gamma' \vdash \Delta'}\,(\text{ContrL})$$
We also replace every bud node $\Gamma' \vdash \Delta'$ such that $\neg F \in \Delta'$ with the derivation tree:
$$\dfrac{\dfrac{\Gamma', F \vdash \Delta'}{\Gamma' \vdash \neg F, \Delta'}\,(\neg\text{R})}{\Gamma' \vdash \Delta'}\,(\text{ContrR})$$

- $E = F_1 \wedge F_2$. Then we replace every bud node $\Gamma' \vdash \Delta'$ such that $F_1 \wedge F_2 \in \Gamma'$ with the derivation tree:
$$\dfrac{\dfrac{\Gamma', F_1, F_2 \vdash \Delta'}{\Gamma', F_1 \wedge F_2 \vdash \Delta'}\,(\wedge\text{L})}{\Gamma' \vdash \Delta'}\,(\text{ContrL})$$
We also replace every bud node $\Gamma' \vdash \Delta'$ such that $F_1 \wedge F_2 \in \Delta'$ with the derivation tree:
$$\dfrac{\dfrac{\Gamma' \vdash F_1, \Delta' \quad \Gamma' \vdash F_2, \Delta'}{\Gamma' \vdash F_1 \wedge F_2, \Delta'}\,(\wedge\text{R})}{\Gamma' \vdash \Delta'}\,(\text{ContrR})$$

- $E = F_1 \vee F_2$. Then we replace every bud node $\Gamma' \vdash \Delta'$ such that $F_1 \vee F_2 \in \Gamma'$ with the derivation tree:
$$\dfrac{\dfrac{\Gamma', F_1 \vdash \Delta' \quad \Gamma', F_2 \vdash \Delta'}{\Gamma', F_1 \vee F_2 \vdash \Delta'}\,(\vee\text{L})}{\Gamma' \vdash \Delta'}\,(\text{ContrL})$$
We also replace every bud node $\Gamma' \vdash \Delta'$ such that $F_1 \vee F_2 \in \Delta'$ with the derivation tree:
$$\dfrac{\dfrac{\Gamma' \vdash F_1, F_2, \Delta'}{\Gamma' \vdash F_1 \vee F_2, \Delta'}\,(\vee\text{R})}{\Gamma' \vdash \Delta'}\,(\text{ContrR})$$

- $F = F_1 \to F_2$. Then we replace every bud node $\Gamma' \vdash \Delta'$ such that $F_1 \to F_2 \in \Gamma'$ with the derivation tree:
$$\dfrac{\dfrac{\Gamma', F_2 \vdash \Delta' \quad \Gamma' \vdash F_1, \Delta'}{\Gamma', F_1 \to F_2 \vdash \Delta'}\,(\to\text{L})}{\Gamma' \vdash \Delta'}\,(\text{ContrL})$$

We also replace every bud node $\Gamma' \vdash \Delta'$ such that $F_1 \to F_2 \in \Delta'$ with the derivation tree:

$$\dfrac{\dfrac{\Gamma', F_1 \vdash F_2, \Delta'}{\Gamma' \vdash F_1 \to F_2, \Delta'}\,(\to\text{R})}{\Gamma' \vdash \Delta'}\,(\text{ContrR})$$

- $E = \langle \exists xF, t \rangle$. Then we replace every bud node $\Gamma' \vdash \Delta'$ such that $\exists xF \in \Gamma'$ with the derivation tree:

$$\dfrac{\dfrac{\Gamma', F[z/x] \vdash \Delta'}{\Gamma', \exists xF \vdash \Delta'}\,(\exists\text{L})}{\Gamma' \vdash \Delta'}\,(\text{ContrL})$$

where $z \notin FV(\Gamma' \cup \Delta')$.

We also replace every bud node $\Gamma' \vdash \Delta'$ such that $\exists xF \in \Delta'$ with the derivation tree:

$$\dfrac{\dfrac{\Gamma' \vdash F[t/x], \Delta'}{\Gamma' \vdash \exists xF, \Delta'}\,(\exists\text{R})}{\Gamma' \vdash \Delta'}\,(\text{ContrR})$$

- $E = \langle \forall xF, t \rangle$. Then we replace every bud node $\Gamma' \vdash \Delta'$ such that $\forall xF \in \Gamma'$ with the derivation tree:

$$\dfrac{\dfrac{\Gamma', F[t/x] \vdash \Delta'}{\Gamma', \forall xF \vdash \Delta'}\,(\forall\text{L})}{\Gamma' \vdash \Delta'}\,(\text{ContrL})$$

We also replace every bud node $\Gamma' \vdash \Delta'$ such that $\forall xF \in \Delta'$ with the derivation tree:

$$\dfrac{\dfrac{\Gamma' \vdash F[z/x], \Delta'}{\Gamma' \vdash \forall xF, \Delta'}\,(\forall\text{R})}{\Gamma' \vdash \Delta'}\,(\text{ContrL})$$

where $z \notin FV(\Gamma' \cup \Delta')$.

- $E = \langle t = u, x, y, \Gamma, \Delta \rangle$. Let $\Gamma' \vdash \Delta'$ be a bud node such that $t = u \in \Gamma'$, $\Gamma' \subseteq \Gamma[t/x, u/y] \cup \{t = u\}$ and $\Delta' \subseteq \Delta[t/x, u/y]$. So $\Gamma' = \Gamma''[t/x, u/y] \cup \{t = u\}$ for some $\Gamma'' \subseteq \Gamma$ and $\Delta' = \Delta''[t/x, u/y]$ for some $\Delta'' \subseteq \Delta$. Then we replace the bud node $\Gamma' \vdash \Delta'$ by the derivation tree:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\Gamma''[u/x,t/y], \Gamma''[t/x,u/y], t = u \vdash \Delta''[t/x,u/y], \Delta''[u/x,t/y]}{\Gamma''[t/x,u/y], \Gamma''[t/x,u/y], t = u, t = u \vdash \Delta''[t/x,u/y], \Delta''[t/x,u/y]}\,(=\text{L})}{\Gamma''[t/x,u/y], \Gamma''[t/x,u/y], t = u \vdash \Delta''[t/x,u/y], \Delta''[t/x,u/y]}\,(\text{ContrL})}{\vdots}}{\Gamma''[t/x,u/y], t = u \vdash \Delta''[t/x,u/y], \Delta''[t/x,u/y]}\,(\text{ContrL})}{\vdots}\,(\text{ContrR})}{\Gamma''[t/x,u/y], t = u \vdash \Delta''[t/x,u/y]}\,(\text{ContrR})$$

- $E = \langle P_i\mathbf{u}, r\rangle$, where P_i is an inductive predicate symbol, $\mathbf{u}$ is a tuple of terms of Σ and $r \in \{1,\ldots,|\Phi_i|\}$. We replace every bud node $\Gamma' \vdash \Delta'$ such that $P_i\mathbf{u} \in \Gamma'$ with the derivation tree:

$$\dfrac{\dfrac{\text{case distinctions}}{\Gamma', P_i\mathbf{u} \vdash \Delta'}\,(\text{Case } P_i)}{\Gamma' \vdash \Delta'}\,(\text{ContrL})$$

  Now note that $\Phi_{i,r}$ is a production in Φ, say:

$$\dfrac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1,\ldots,j_m,i \in \{1,\ldots,n\}$$

  and we have $\mathbf{u} = \mathbf{t}(\mathbf{t}')$ for some $\mathbf{t}'$. Then we replace every bud node $\Gamma' \vdash \Delta'$ such that $P_i\mathbf{u} \in \Delta'$ with the derivation tree:

$$\dfrac{\dfrac{\Gamma' \vdash Q_1\mathbf{u_1}(\mathbf{t}'),\Delta' \ \ldots\ \Gamma' \vdash Q_h\mathbf{u_h}(\mathbf{t}'),\Delta' \quad \Gamma' \vdash P_{j_1}\mathbf{t_1}(\mathbf{t}'),\Delta' \ \ldots\ \Gamma' \vdash P_{j_m}\mathbf{t_m}(\mathbf{t}'),\Delta'}{\Gamma' \vdash P_i\mathbf{t}(\mathbf{t}'),\Delta'}\,(P_i\mathrm{R}_r)}{\Gamma' \vdash \Delta'}\,(\text{ContrR})$$

The *search tree for* $\Gamma \vdash \Delta$ is defined to be the (possibly infinite) tree $T_\omega = \bigcup_{i\geq 0} T_i$ (note that it makes sense to consider the union of the derivation trees T_i since T_i is a subtree of T_{i+1} for all $i \geq 0$).

If T_ω is not an LKID$^\omega$ proof then either it is not even an LKID$^\omega$ pre-proof, or it is an LKID$^\omega$ pre-proof but does not satisfy the proof condition. The next proposition addresses the former situation.

Proposition 4.3.3. *If the search tree T_ω for $\Gamma \vdash \Delta$ is not a pre-proof then $\Gamma \vdash \Delta$ is invalid.*

Proof. Since T_ω is clearly an LKID$^\omega$ derivation tree, it must contain a bud node, say $\Gamma' \vdash \Delta'$. Now by construction of T_ω, it is clear that none of the rules for the logical connectives (including equality) or for the inductive predicates can be applied to it, otherwise there would be an element on the schedule allowing a rule to be applied to the bud. Therefore Γ' can contain only atomic formulas involving non-inductive predicates, and Δ' can contain only atomic formulas to which no rule can be applied, although they may still contain inductive predicates. Also, again by construction of T_ω, the endsequent $\Gamma \vdash \Delta$ of T_ω is a subsequent of every node occurring in T_ω, i.e. $\Gamma \subseteq \Gamma'$ and $\Delta \subseteq \Delta'$. Therefore we have:

$$\Gamma \vdash \Delta \;=\; Q_1\mathbf{t_1},\ldots,Q_m\mathbf{t_m} \vdash R_1\mathbf{u_1},\ldots,R_k\mathbf{u_k}$$

where $Q_1,\ldots,Q_m$ are ordinary predicates and $R_1,\ldots,R_k$ are either ordinary or inductive predicates. To see that $\Gamma \vdash \Delta$ is invalid, we construct a standard model of (Σ,Φ) in which the sequent

is false. Define a first-order structure M for Σ whose domain is $Terms(\Sigma)$, and define:

$$\begin{array}{rcll} c^M & = & c & \text{for each constant symbol } c \\ f^M & = & f & \text{for each function symbol } f \\ Q^M\mathbf{t} & \Leftrightarrow & Q\mathbf{t} \in \Gamma & \text{for each ordinary predicate symbol } Q \\ P_i^M & = & \bigcup_\alpha P_i^\alpha & \text{for each inductive predicate } P_i \end{array}$$

Note that the last clause of the definition ensures that M is a standard model for (Σ, Φ) as required. Now define an environment ρ for M by $\rho(x) = x$ for each variable symbol x. It is then clear that $\rho(t) = t$ for all terms t of Σ, and thus all the formulas of Γ are obviously true in M under ρ. Similarly, all the formulas of Δ not involving inductive predicates are false in M under ρ. It remains to show that any formula $P_i\mathbf{u}$ occurring in Δ, where P_i is an inductive predicate, is false in M under ρ, i.e., that $\mathbf{u} \notin \bigcup_\alpha P_i^\alpha$. But this is clear from the fact that no right-unfolding rule can be applied to $P_i\mathbf{u}$, and thus $\mathbf{u} \notin \varphi_\Phi^\alpha$ for any α. □

Now, suppose that T_ω is an LKID$^\omega$ pre-proof, but not a proof. From Definition 4.2.5 we immediately have that there must exist an infinite path in T_ω such that there does not exist an infinitely progressing trace following any tail of the path. (This path can then trivially be extended to a rooted path as required.) For the remainder of this section, up until the statement of the cut-free completeness result, we shall let T_ω be the search tree for a fixed sequent $\Gamma \vdash \Delta$, and we shall assume that T_ω is an LKID$^\omega$ pre-proof but not a proof, i.e., there is an infinite (rooted) path of T_ω such that there does not exist an infinitely progressing trace following a tail of the path.

Definition 4.3.4 (Untraceable branch / Limit sequent)**.** Let $\pi = (v_i)_{i\geq 0}$ be the rooted infinite path in T_ω such that there is no infinitely progressing trace following any tail of the branch. (Of course, there may be more than one such branch, in which case we make an arbitrary choice.) π is said to be the *untraceable branch* of T_ω and $\Gamma_\omega \vdash \Delta_\omega$ is said to be the *limit sequent* for $\Gamma \vdash \Delta$, where $s(v_i) = \Gamma_i \vdash \Delta_i$ for each i and $\Gamma_\omega = \bigcup_{i\geq 0} \Gamma_i$ and $\Delta_\omega = \bigcup_{i\geq 0} \Delta_i$. (Notice that these are well-defined since, by construction of the search tree T_ω, we have $\Gamma_i \subseteq \Gamma_{i+1}$ and $\Delta_i \subseteq \Delta_{i+1}$ for all $i \geq 0$.)

As for the LKID completeness proof, we observe that the limit sequent $\Gamma_\omega \vdash \Delta_\omega$ is not technically a sequent as Γ_ω and Δ_ω are, in general, infinite multisets of formulas. Our notion of (cut-free) provability in LKID$^\omega$ for an infinite "sequent" is, as before, (cut-free) provability for some finite subsequent[1]. For example, if we say that $\Gamma_\omega \vdash \Delta_\omega$ is cut-free provable in LKID$^\omega$

[1]This notion of provability for an infinite sequent implies but does not coincide with validity. For example, the infinite sequent $Nt \vdash t = 0, t = s0, t = ss0, \ldots, t = s^n0, \ldots$, is valid even though every one of its finite subsequents is invalid — a typical failure of compactness. Interestingly, if one instead were to allow infinite sequents to appear in LKID$^\omega$ proofs, a more powerful notion of cut-free provability would be obtained, which would coincide with validity by our completeness proof.

then we mean there exist finite $\Gamma' \subseteq \Gamma_\omega$ and $\Delta' \subseteq \Delta_\omega$ such that the sequent $\Gamma' \vdash \Delta'$ is cut-free provable in LKID$^\omega$.

Lemma 4.3.5. *Let $\pi = (v_i)_{i\geq 0}$ be the untraceable branch of T_ω. Then for all $i \geq 0$, the sequent $s(v_i)$ is not cut-free provable in LKID$^\omega$.*

Proof. The intuition behind this proof is that every possible attempt is made when constructing T_ω to either close a given proof branch or, if this is impossible, to construct an infinitely progressing trace following some tail of it (since formulas are never "thrown away" when constructing the search tree). Any putative proof of a sequent occurring on the untraceable branch would yield either a way of closing the branch or constructing an infinitely progressing trace following a tail of it, which is impossible.

We write $\Gamma_i \vdash \Delta_i$ for $s(v_i)$. Suppose for contradiction that for some $i \geq 0$, $\Gamma_i \vdash \Delta_i$ has a cut-free proof in LKID$^\omega$. So there is a (possibly infinite) LKID$^\omega$ derivation tree $T = (V', s', r', p')$ such that T has no bud nodes, $s'(root(T)) = \Gamma_i \vdash \Delta_i$, and there is an infinitely progressing trace following a tail of every infinite path in T.

Now let $j \geq i$, let $v' \in V'$ and $s'(v') = \Gamma' \vdash \Delta'$, and suppose $\Gamma' \subseteq \Gamma_j$ and $\Delta' \subseteq \Delta_j$. Recall that by construction of T_ω, we have $\Gamma_k \subseteq \Gamma_{k+1}$ and $\Delta_k \subseteq \Delta_{k+1}$ for all $k \geq 0$, and every LKID$^\omega$ schedule element occurs infinitely often on the schedule according to which T_ω is constructed. There is therefore a point along π where exactly the same rule is applied (with the same active formula and side conditions, if applicable) as is applied to $\Gamma' \vdash \Delta'$ in the proof T. (It is for this reason that it is crucial that we also schedule the rules for equality and right-unfolding of inductive definitions in the construction of the search tree.) It follows that there is a $k \geq j$ such that the rule $r'(v')$ is applied at v_k (i.e. $r(v_k) = r'(v')$) with the same active formula as in the rule instance at v' , and there is some premise $p'_x(v')$ of the rule instance at v' such that, writing $s'(p'_x(v')) = \Gamma'' \vdash \Delta''$, we have $\Gamma'' \subseteq \Gamma_{k+1}$ and $\Delta'' \subseteq \Delta_{k+1}$. (Notice that $p'_x(v')$ must exist since otherwise π is not an infinite path, which is a contradiction.) This situation is illustrated in Figure 4.1. Since $\Gamma_i \subseteq \Gamma_{i+1}$ and $\Delta_i \subseteq \Delta_{i+1}$ for all $i \geq 0$, and (τ, τ) is always a valid trace on any edge (v_i, v_{i+1}) such that $\tau \in \Gamma_i$ and $\tau \in \Gamma_{i+1}$ (except when the rule (Subst) is applied, which does not occur here since we consider cut-free proofs), it follows that if (τ, τ') is a (progressing) trace following $(v', p'_x(v'))$, then $(\tau, \ldots, \tau, \tau')$ is a (progressing) trace following $(v_j, \ldots, v_k, v_{k+1})$.

Now since $s'(root(T)) = \Gamma_i \vdash \Delta_i$, and trivially $\Gamma_i \subseteq \Gamma_i$ and $\Delta_i \subseteq \Delta_i$, we can repeatedly iterate the argument in the preceding paragraph to obtain a path $\pi' = (v'_i)_{i\geq 0}$ in T and a sequence $i < k_1 < k_2 < \ldots$ of natural numbers such that, for all $n \geq 1$, if (τ, τ') is a (progressing trace) following (v'_n, v'_{n+1}), then $(\tau, \ldots, \tau, \tau')$ is a (progressing) trace following $(v_{k_n}, \ldots, v_{k_{n+1}})$.

Since T is a proof, there is an infinitely progressing trace following some tail of the path π' in T. It follows that there then is an infinitely progressing trace following some tail of the untraceable path π in T_ω. But this contradicts the defining property of π. So there cannot exist a cut-free proof of $\Gamma_i \vdash \Delta_i$ in LKID$^\omega$. □

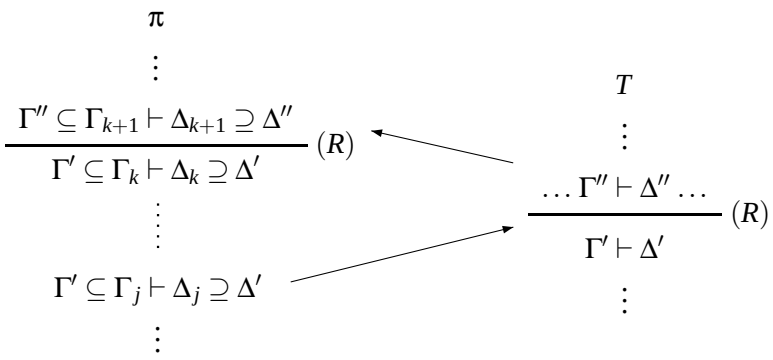


Figure 4.1: Part of the proof of Lemma 4.3.5. Given $\Gamma_j \vdash \Delta_j$ and $\Gamma' \vdash \Delta'$ with $\Gamma' \subseteq \Gamma_j$ and $\Delta' \subseteq \Delta_j$, we can construct $\Gamma'' \vdash \Delta''$, $\Gamma_k \vdash \Delta_k$ and $\Gamma_{k+1} \vdash \Delta_{k+1}$ as shown.

Definition 4.3.6. The *counter-interpretation for* $\Gamma_\omega \vdash \Delta_\omega$ is defined to be (M_ω, ρ_ω), where M_ω is a first-order structure for Σ and ρ_ω is an environment for M_ω both defined exactly as in Definition 3.3.4 except that for each $i \in \{1,\ldots,n\}$, we define $P_i^{M_\omega} = \bigcup_\alpha P_i^\alpha$. (This ensures that M_ω is thus a standard model for (Σ, Φ).)

Lemma 4.3.7. *If $M_\omega \models_{\rho_\omega} P_i(t_1,\ldots,t_{k_i})$, then $\Gamma_\omega \vdash P_i(t_1,\ldots,t_{k_i})$ is cut-free provable in LKID$^\omega$.*

Proof. Define an n-tuple of sets $(X_1,\ldots,X_n)$ by:

$$X_i = \{([t_1],\ldots,[t_{k_i}]) \mid \Gamma_\omega \vdash P_i(t_1,\ldots,t_{k_i}) \text{ is cut-free provable in LKID}^\omega\}$$

for each $i \in \{1,\ldots,n\}$. Suppose $(X_1,\ldots,X_n)$ is a prefixed point of φ_Φ and thus, since the ω^{th} approximant φ_Φ^ω is the least prefixed point of φ_Φ by Lemma 2.2.11, we have $\varphi_\Phi^\omega \subseteq (X_1,\ldots,X_n)$. Then, for any $i \in \{1,\ldots,n\}$:

$$\begin{array}{lll}
M_\omega \models_{\rho_\omega} P_i(t_1,\ldots,t_{k_i}) & \Leftrightarrow & (\rho_\omega(t_1),\ldots,\rho_\omega(t_{k_i})) \in \bigcup_\alpha P_i^\alpha = \pi_i^n.\varphi_\Phi^\omega \\
& \Leftrightarrow & ([t_1],\ldots,[t_{k_i}]) \in \pi_i^n.\varphi_\Phi^\omega \quad \text{by Prop. 3.3.5} \\
& \Rightarrow & ([t_1],\ldots,[t_{k_i}]) \in \pi_i^n(X_1,\ldots,X_n) \\
& \Leftrightarrow & ([t_1],\ldots,[t_{k_i}]) \in X_i \\
& \Leftrightarrow & \exists u_1,\ldots,u_{k_i}.\, u_1 \sim t_1,\ldots,u_{k_i} \sim t_{k_i} \\
& & \quad \text{and } \Gamma_\omega \vdash P_i(u_1,\ldots,u_{k_i}) \text{ cut-free provable in LKID}^\omega \\
& \Leftrightarrow & \Gamma_\omega \vdash P_i(t_1,\ldots,t_{k_i}) \text{ cut-free provable (by Lemma 3.3.7)}
\end{array}$$

It thus suffices to show that $(X_1,\ldots,X_n)$ is a prefixed point of φ_Φ, i.e. $\varphi_\Phi(X_1,\ldots,X_n) \subseteq (X_1,\ldots,X_n)$. The proof of this fact is very similar to the second part of the proof of Lemma 3.3.8. □

Lemma 4.3.8. *If $F \in \Gamma_\omega$ then $M_\omega \models_{\rho_\omega} F$, and if $F \in \Delta_\omega$ then $M_\omega \not\models_{\rho_\omega} F$.*

Proof. First note that $\Gamma_\omega \vdash \Delta_\omega$ is not cut-free provable by Lemma 4.3.5. The proof proceeds by structural induction on F. All of the cases are then exactly as in Lemma 3.3.9 except for the case $F = P_i\mathbf{u}$, where P_i is an inductive predicate. This case is dealt with as follows:

Case $F = P_i\mathbf{u}$, where P_i is an inductive predicate symbol of Σ (so $i \in \{1,\ldots,n\}$). Suppose first that $P_i\mathbf{u} \in \Delta_\omega$ and suppose for contradiction that $M_\omega \models_{\rho_\omega} P_i\mathbf{u}$. Then by Lemma 4.3.7, $\Gamma_\omega \vdash P_i\mathbf{u}$ is cut-free provable in LKID$^\omega$ and, since $P_i\mathbf{u} \in \Delta_\omega$, it holds that $\Gamma_\omega \vdash \Delta_\omega$ is cut-free provable, which is a contradiction. So $M_\omega \not\models_{\rho_\omega} P_i\mathbf{u}$ as required.

Now suppose that $P_i\mathbf{u} \in \Gamma_\omega$, and suppose for contradiction that $M_\omega \not\models_{\rho_\omega} P_i\mathbf{u}$. Let $(v_j)_{j\geq 0}$ be the untraceable branch of T_ω, and write $\Gamma_j \vdash \Delta_j$ for $s(v_j)$. Note that by construction of T_ω, there is an $j \geq 0$ such that for all $j' \geq j$, $P_i\mathbf{u} \in \Gamma_{j'}$. Furthermore, since every schedule element occurs infinitely often on the schedule according to which T_ω is constructed, there is a $k \geq j$ such that the rule (Case P_i) is applied at $\Gamma_k \vdash \Delta_k$ with active formula $P_i\mathbf{u}$ in the construction of T_ω. So for some production in Φ, say:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_i\mathbf{t}(\mathbf{x})} \qquad j_1,\ldots,j_m,i \in \{1,\ldots,n\}$$

we have the following:

$$\Gamma_{k+1} \vdash \Delta_{k+1} \;=\; \Gamma_k, \mathbf{u} = \mathbf{t}(\mathbf{x}), Q_1\mathbf{u_1}(\mathbf{x}), \ldots, Q_h\mathbf{u_h}(\mathbf{x}), P_{j_1}\mathbf{t_1}(\mathbf{x}), \ldots, P_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash \Delta_k$$

where $x \notin FV(\Gamma_k \cup \Delta_k)$ for all k. So the formulas $\mathbf{u} = \mathbf{t}(\mathbf{x})$,$Q_1\mathbf{u_1}(\mathbf{x}), \ldots, Q_h\mathbf{u_h}(\mathbf{x})$,$P_{j_1}\mathbf{t_1}(\mathbf{x})$,…, $P_{j_m}\mathbf{t_m}(\mathbf{x})$ are all in Γ_ω. Note that since $\mathbf{u} = \mathbf{t}(\mathbf{x}) \in \Gamma_\omega$, we have $\mathbf{u} \sim \mathbf{t}(\mathbf{x})$ by definition of $\sim$ (c.f. Definition 3.3.3) and so $[\mathbf{u}] = [\mathbf{t}(\mathbf{x})]$. Also, since $Q_1\mathbf{u_1}(\mathbf{x}), \ldots, Q_h\mathbf{u_h}(\mathbf{x}) \in \Gamma_\omega$, all of $Q_1^{M_\omega}[\mathbf{u_1}(\mathbf{x})], \ldots, Q_h^{M_\omega}[\mathbf{u_h}(\mathbf{x})]$ hold by definition of M_ω. Thus if $P_{j_1}^{M_\omega}[\mathbf{t_1}(\mathbf{x})], \ldots, P_{j_m}^{M_\omega}[\mathbf{t_m}(\mathbf{x})]$ also all hold then $P_i^{M_\omega}[\mathbf{t}(\mathbf{x})]$ holds by definition of $P_i^{M_\omega}$ and φ_Φ, i.e. $P_i^{M_\omega}\mathbf{u}$ holds so $M_\omega \models_{\rho_\omega} P_i\mathbf{u}$, which contradicts our initial assumption that $M_\omega \not\models_{\rho_\omega} P_i\mathbf{u}$.

Therefore, for some $k \in \{1,\ldots,m\}$, we have that $P_{j_k}^{M_\omega}[\mathbf{t_k}(\mathbf{x})]$ does not hold. Note that $(P_i\mathbf{u}, \ldots, P_i\mathbf{u}, P_{j_k}\mathbf{t_k}(\mathbf{x}))$ is a progressing trace following some finite segment of the untraceable branch π of T_ω (starting with the first sequent on the branch such that $P_i\mathbf{u}$ occurs in its left-hand side, and finishing with the above premise of the considered instance of (Case P_i) in T_ω). By repeating the argument, this time starting with $P_{j_k}\mathbf{t_k}(\mathbf{x})$, we can find a point on the untraceable branch where $P_{j_k}\mathbf{t_k}(\mathbf{x})$ is unfolded using the rule (Case P_{j_k}) and another progressing trace, beginning with $P_{j_k}\mathbf{t_k}(\mathbf{x})$, following some further segment of the untraceable branch. By iterating the argument, we obtain an infinite sequence of atomic formulas $(F_i)_{i\geq 0}$ occurring on π, with $F_0 = P_i\mathbf{u}$, $F_1 = P_{j_k}\mathbf{t_k}(\mathbf{x})$, etc., such that for all $i \geq 0$, $F_{i+1}$is a case-descendant of F_i. Thus there exists an infinitely progressing trace $(F_0, \ldots, F_0, F_1, \ldots, F_1, F_2, \ldots)$ following some tail of π, which contradicts the fact that π is the untraceable branch of T_ω. Thus we must have $M_\omega \models_{\rho_\omega} P_i\mathbf{u}$ as required, which completes the case and thus the proof.

□

Theorem 4.3.9 (Cut-free completeness of LKID$^\omega$)**.** *If $\Gamma \vdash \Delta$ is valid with respect to standard models of (Σ, Φ), then it has a recursive cut-free proof in LKID$^\omega$.*

Proof. Suppose that $\Gamma \vdash \Delta$ is valid, i.e. true in every standard model of (Σ, Φ), but that the search tree T_ω for $\Gamma \vdash \Delta$ is not an LKID$^\omega$ proof. So either T_ω is not even an LKID$^\omega$ pre-proof — in which case $\Gamma \vdash \Delta$ is invalid by Proposition 4.3.3,which is a contradiction — or T_ω has an untraceable branch (c.f. Definition 4.3.4). In the latter case, let $\Gamma_\omega \vdash \Delta_\omega$ be the limit sequent for $\Gamma \vdash \Delta$ with counter-interpretation (M_ω, ρ_ω) (c.f. Definition 4.3.6). By Lemma 4.3.8, we have $M_\omega \models_{\rho_\omega} F$ for all $F \in \Gamma_\omega$ and $M_\omega \not\models_{\rho_\omega} F$ for all $F \in \Delta_\omega$. As $\Gamma \subseteq \Gamma_\omega$ and $\Delta \subseteq \Delta_\omega$, we thus have $M_\omega \models_{\rho_\omega} F$ for all $F \in \Gamma$ and $M_\omega \not\models_{\rho_\omega} F$ for all $F \in \Delta$. In other words, we have constructed a standard model M_ω of (Σ, Φ) such that $\Gamma \vdash \Delta$ is false in M_ω, so $\Gamma \vdash \Delta$ is again invalid, which is a contradiction.

Thus T_ω is an LKID$^\omega$ proof of the sequent $\Gamma \vdash \Delta$. By construction, it is both recursive and cut-free. □

Theorem 4.3.10 (Cut-eliminability in LKID$^\omega$)**.** *If $\Gamma \vdash \Delta$ is provable in LKID$^\omega$ then it has a recursive cut-free proof in LKID$^\omega$ (i.e. a recursive proof that does not employ the rules (Cut), (Wk) or (Subst)).*

Proof. Suppose $\Gamma \vdash \Delta$ is provable in LKID$^\omega$. By soundness (Proposition 4.2.6), $\Gamma \vdash \Delta$ is valid with respect to standard models of (Σ, Φ). Thus by Theorem 4.3.9, the search tree for $\Gamma \vdash \Delta$ is a recursive cut-free proof of $\Gamma \vdash \Delta$ in LKID$^\omega$. □

The fact that LKID$^\omega$ is complete with respect to standard models, together with the fact that LKID cannot have this property, means that the infinitary system LKID$^\omega$ is more powerful than the finitary system LKID, because there are sequents that are valid in the standard sense but not Henkin valid (this essentially follows from the fact that there are non-standard Henkin models, c.f. Corollary 2.3.11). The fact that LKID$^\omega$ is complete in the standard sense, and can interpret Peano arithmetic via a translation similar to that used for LKID in the previous chapter, implies by Gödel's incompleteness theorem that it is impossible to recursively enumerate a complete set of LKID$^\omega$ proofs. Moreover, cut-free proofs in LKID$^\omega$ enjoy a property akin to the subformula property for cut-free proofs in LK, when the notion of subformula is appropriately generalised. That is to say, if one allows that an "inductive-subformula" (say) of a formula F is either a subformula of F in the usual sense or related to such a subformula by equality or by unfolding of inductively defined predicates, then every formula occurring in a cut-free LKID$^\omega$ proof of a sequent $\Gamma \vdash \Delta$ is an inductive-subformula of a formula occurring in $\Gamma \vdash \Delta$. The eliminability of cut in LKID$^\omega$ thus corresponds, at least in some sense, to Girard's ideal of "purity of methods" [29] for FOL$_{\text{ID}}$.

Chapter 5

Cyclic proofs in trace-based infinitary proof systems

In the previous chapter, we considered an infinitary proof system LKID^ω for FOL_{ID} whose soundness is guaranteed by a global proof condition based on *traces* in the proof, and we established that any valid sequent of FOL_{ID} has a recursive cut-free LKID^ω proof. Unfortunately, the system LKID^ω is not suitable for practical formal reasoning. Note that it is not possible to recursively enumerate the set of LKID^ω proofs, for otherwise — via an embedding of Peano arithmetic into LKID^ω similar to that in Section 3.4.2 — it would be possible to recursively enumerate the true statements of arithmetic, which is impossible. In particular, although recursive LKID^ω proofs can be encoded as natural numbers (for example), it is not even semidecidable whether a given natural number encodes an LKID^ω proof for, if this were the case, we could recursively enumerate the set of LKID^ω proofs. Thus we do not have a *useful* finite representation for LKID^ω (pre-)proofs in general.

We saw in Section 4.2.1 that the LKID^ω proof condition — that an infinitely progressing trace exists on some tail of every infinite path in the proof — yields a sound notion of infinite proof for any proof system S^ω for which a suitable notion of trace can be found (c.f. Definition 4.2.7). For sufficiently powerful systems S^ω, the same practical problems for formal reasoning arise as in the case of LKID^ω above.

We recall that an infinite tree is said to be *regular* if it contains only finitely many distinct subtrees [18], and it is well-known that a tree is regular exactly if it can be represented as a finite (cyclic) graph [16, 46, 76]. Since formal reasoning requires a convenient finite representation for proofs, it is natural to consider the restriction of an infinitary trace-based system S^ω to those (pre-)proofs given by regular derivation trees, i.e., those derivation trees representable by finite cyclic graphs. We use the generic term *cyclic proof systems* for systems obtained from infinitary proof systems in this way.

In Section 5.1, we show how to obtain the restricted proof system CS^ω, which *is* suitable

for formal reasoning, from an arbitrary trace-based proof system S^ω, and we also show that in the setting of CS^ω, the proof condition for S^ω is decidable. In Section 5.2, we consider the particular case of the system CLKID^ω obtained by so restricting LKID^ω, and give some illustrative examples of cyclic CLKID^ω proofs. We conjecture that cut is not eliminable in the restricted system CLKID^ω (despite being eliminable in the full infinitary system LKID^ω).

5.1 The cyclic restriction of a trace-based infinitary proof system

Given an infinite proof system S^ω equipped with a suitable trace pair function (c.f. Definition 4.2.7), we obtain its cyclic restriction, CS^ω, as follows. A CS^ω pre-proof is a finite S^ω derivation tree, possibly containing bud nodes, together with a function assigning to each bud node an interior *companion* node in the tree with the same sequent labelling; this allows us to identify cyclicity in derivation trees. Note that equality on formulas is standard syntactic equality, and so equality on sequents is defined by extending this notion to multisets in the standard way. CS^ω pre-proofs can be viewed as finite (cyclic) graph representations of regular infinite trees, i.e., regular S^ω pre-proofs, by identifying bud nodes with their assigned companions. The soundness condition on S^ω pre-proofs can thus be straightforwardly applied to CS^ω pre-proofs, and moreover, this condition is decidable in the restricted setting of CS^ω.

Definition 5.1.1 (Companion)**.** Let $\mathcal{D} = (V, s, r, p)$ be a derivation tree and let $B \in \mathrm{Bud}(\mathcal{D})$. A node $C \in V$ is said to be a *companion for* B if $r(C)$ is defined and $s(C) = s(B)$.

We remark that, in contrast to some other formal systems employing notions of cyclic or circular proof (e.g. [63, 32]), we do not require companions to be ancestors of the bud nodes to which they are assigned, i.e. C need not appear on the unique path in $\mathcal{D}$ from $root(\mathcal{D})$ to B in the definition above. (However, we show in the next chapter that any CS^ω pre-proof can be transformed into an "equivalent" pre-proof in which the buds and companions *are* arranged in this form.)

Definition 5.1.2 (CS^ω pre-proof)**.** A *CS^ω pre-proof* of a sequent S is a pair $(\mathcal{D} = (V, s, r, p), \mathcal{R})$, where $\mathcal{D}$ is a finite S^ω derivation tree with endsequent S, and $\mathcal{R} : V \to V$ is a partial function assigning a companion to every bud node in $\mathcal{D}$.

Next, we show how a CS^ω pre-proof can be understood as a cyclic graph. We write $f(x) \simeq g(y)$, where f and g are (partial) functions, to mean that $f(x)$ is defined iff $g(y)$ is defined and if $f(x)$ is defined then $f(x) = g(y)$.

Definition 5.1.3 (Pre-proof graph)**.** Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a CS^ω pre-proof, where $\mathcal{D} = (V, s, r, p)$. Then the *graph* of $\mathcal{P}$, written $\mathcal{G}_\mathcal{P}$, is the derivation graph obtained from $\mathcal{D}$ by identifying each bud node $B \in \mathrm{Bud}(\mathcal{D})$ with its companion $\mathcal{R}(B)$. That is to say, $\mathcal{G}_\mathcal{P} = (V', s, r, p')$, where

$V' = V \setminus \mathrm{Bud}(\mathcal{D})$ and p' is defined by:

$$p'_j(v) \simeq \begin{cases} \mathcal{R}(p_j(v)) & \text{if } p_j(v) \in \mathrm{Bud}(\mathcal{D}) \\ p_j(v) & \text{otherwise} \end{cases}$$

for each $j \in \mathbb{N}$. (Note that $\mathcal{G}_\mathcal{P}$ contains no bud nodes, i.e., the rule labelling function r is total on V'.) A path $v_0 v_1 v_2 \ldots$ in $\mathcal{G}_\mathcal{P}$ is said to be *rooted* iff $v_0 = root(\mathcal{D})$.

We now demonstrate how a CS^ω pre-proof can be understood as a S^ω pre-proof. Note that we use the symbol $\cdot$ to indicate concatenation of sequences.

Definition 5.1.4 (Tree-unfolding)**.** Let $\mathcal{P} = (\mathcal{D} = (V, s, r, p), \mathcal{R})$ be a CS^ω pre-proof with graph $\mathcal{G}_\mathcal{P} = (V', s, r, p')$. Define $Path(\mathcal{G}_\mathcal{P})$, the set of rooted finite paths through $\mathcal{G}_\mathcal{P}$, as follows:

$$Path(\mathcal{G}_\mathcal{P}) = \{(v_i)_{0 \le i \le n} \mid n \in \mathbb{N} \text{ and } v_0 = root(\mathcal{D}) \text{ and } \forall i \in \{1, \ldots, n\}. \exists j.\ v_i = p'_j(v_{i-1})\}$$

Then the *tree-unfolding* of $\mathcal{P}$ is $\mathcal{T}_\mathcal{P} = (Path(\mathcal{G}_\mathcal{P}), s^*, r^*, p^*)$, where:

- $s^*((v_i)_{0 \le i \le n}) = s(v_n)$
- $r^*((v_i)_{0 \le i \le n}) = r(v_n)$ (note that r is total on V', so $r(v_n)$ is always defined)
- $p^*_j((v_i)_{0 \le i \le n}) \simeq ((v_i)_{0 \le i \le n} \cdot p'_j(v_n))$

Lemma 5.1.5. *For any CS^ω pre-proof $\mathcal{P}$, its tree-unfolding $\mathcal{T}_\mathcal{P}$ is a S^ω pre-proof.*

Proof. Let $\mathcal{G}_\mathcal{P} = (V', s, r, p')$ and $\mathcal{T}_\mathcal{P} = (Path(\mathcal{G}_\mathcal{P}), s^*, r^*, p^*)$. First we need to establish that $\mathcal{T}_\mathcal{P}$ is a derivation graph. To see this, we observe that:

$$\begin{aligned}
& p^*_j((v_i)_{0 \le i \le n}) \text{ is defined} \\
\Leftrightarrow\ & p'_j(v_n) \text{ is defined} \\
\Leftrightarrow\ & r(v_n) = (R) \text{ and } 1 \le j \le m \text{ and } \frac{s(p'_1(v_n)) \quad \ldots \quad s(p'_m(v_n))}{s(v_n)} \text{ is an instance of rule } (R) \\
\Leftrightarrow\ & r^*((v_i)_{0 \le i \le n}) = (R) \text{ and } 1 \le j \le m \text{ and} \\
& \frac{s^*(p^*_1((v_i)_{0 \le i \le n})) \quad \ldots \quad s^*(p^*_m((v_i)_{0 \le i \le n}))}{s^*((v_i)_{0 \le i \le n})} \text{ is an instance of rule } (R)
\end{aligned}$$

as required. (Note that the second equivalence holds since $\mathcal{G}_\mathcal{P}$ is a derivation graph.)

Now we define the root of $\mathcal{T}_\mathcal{P}$ to be the one-element sequence $(root(\mathcal{D}))$, and need to establish that there is a unique path from $(root(\mathcal{D}))$ to any other node $(v_i)_{0 \le i \le n} \in Path(\mathcal{G}_\mathcal{P})$. We proceed by induction on n. In the case $n = 0$ there is clearly a unique path (of length 0) from $(root(\mathcal{D}))$ to $(root(\mathcal{D}))$ in $\mathcal{T}_\mathcal{P}$ and we are done. For the step case, we assume there is a unique path from $(root(\mathcal{D}))$ to any node of the form $(v_i)_{0 \le i \le k}$ and proceed as follows:

$$\begin{aligned}(v_i)_{0\leq i\leq k+1} \in \mathit{Path}(\mathcal{G}_{\mathcal{P}}) &\Rightarrow \exists \text{ unique } j.\ v_{k+1} = p'_j(v_k)\\ &\Leftrightarrow \exists \text{ unique } j.\ (v_i)_{0\leq i\leq k+1} = (v_i)_{0\leq i\leq k} \cdot p'_j(v_k)\\ &\Leftrightarrow \exists \text{ unique } j.\ (v_i)_{0\leq i\leq k+1} = p^*_j((v_i)_{0\leq i\leq k})\end{aligned}$$

The uniqueness of the path from $(\mathit{root}(\mathcal{D}))$ to $(v_i)_{0\leq i\leq k+1}$ follows from the uniqueness of the path from $(\mathit{root}(\mathcal{D}))$ to $(v_i)_{0\leq i\leq k}$ together with the uniqueness of j in the above. So $\mathcal{T}_{\mathcal{P}}$ is indeed a derivation tree. It is clear that it is a S^ω derivation tree, because $\mathcal{D}$ is a S^ω derivation tree, and the sequent and rule labellings appearing in $\mathcal{T}_{\mathcal{P}}$ are just those appearing in $\mathcal{D}$. Finally, to see that $\mathcal{T}_{\mathcal{P}}$ is a S^ω pre-proof, we just need to establish that $\mathcal{T}_{\mathcal{P}}$ has no bud nodes, which is clear since $r^*((v_i)_{0\leq i\leq n})$ is defined whenever $r(v_n)$ is defined, and r is total on the nodes V' of $\mathcal{G}_{\mathcal{P}}$. □

Definition 5.1.6 (CS^ω proof)**.** A CS^ω pre-proof $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ is said to be a CS^ω *proof* if, for every infinite path in $\mathcal{G}_{\mathcal{P}}$, there is an infinitely progressing trace following some tail of the path.

Of course, as in the case of our infinitary systems in the previous chapter, we can restrict to rooted infinite paths in the definition above without loss of generality.

Proposition 5.1.7. *For any CS^ω pre-proof $\mathcal{P}$, $\mathcal{P}$ is a CS^ω proof if and only if $\mathcal{T}_{\mathcal{P}}$ is a S^ω proof.*

Proof. Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a CS^ω pre-proof and note that its tree-unfolding $\mathcal{T}_{\mathcal{P}}$ is a S^ω pre-proof by Lemma 5.1.5 above. We then require to show that there exists an infinitely progressing trace on some tail of every infinite path in $\mathcal{G}_{\mathcal{P}} = (V', s, r, p')$ just in case there is one on some tail of every infinite path in $\mathcal{T}_{\mathcal{P}} = (\mathit{Path}(\mathcal{G}_{\mathcal{P}}), s^*, r^*, p^*)$. Without loss of generality, we may restrict our attention to *rooted* infinite paths in $\mathcal{G}_{\mathcal{P}}$ and $\mathcal{T}_{\mathcal{P}}$ (c.f. our remarks immediately after Definition 4.2.5).

Now suppose $\mathcal{T}_{\mathcal{P}}$ is a S^ω proof, and let $\pi = (v_i)_{i\geq 0}$ be a rooted infinite path in $\mathcal{G}_{\mathcal{P}}$. Now we have for all $i \geq 0$ and for all $j \in \mathbb{N}$:

$$v_{i+1} = p'_j(v_i) \Leftrightarrow (v_k)_{0\leq k\leq i+1} = p^*_j((v_k)_{0\leq k\leq i})$$

and since $(v_0) = (\mathit{root}(\mathcal{D})) = \mathit{root}(\mathcal{T}_{\mathcal{P}})$, it follows that $((v_k)_{0\leq k\leq i})_{i\geq 0}$ is a rooted infinite path in $\mathcal{T}_{\mathcal{P}}$. Moreover, we have for each $i \geq 0$:

$$s^*((v_k)_{0\leq k\leq i}) = s(v_i) \text{ and } r^*((v_k)_{0\leq k\leq i}) = r(v_i)$$

so the sequent and rule labelling of the path $((v_k)_{0\leq k\leq i})_{i\geq 0}$ in $\mathcal{T}_{\mathcal{P}}$ is identical to the sequent and rule labelling of the path $(v_i)_{i\geq 0}$ in $\mathcal{G}_{\mathcal{P}}$. Since $\mathcal{T}_{\mathcal{P}}$ is an S^ω proof, there is an infinitely progressing trace following some tail of $((v_k)_{0\leq k\leq i})_{i\geq 0}$, whence it is immediate by the above that this trace is also a infinitely progressing trace following a tail of $(v_i)_{i\geq 0}$, because the trace pair function for the system only depends on the sequents and rule labelling a given edge

(c.f. Definition 4.2.7). Hence there is an infinitely progressing trace on some tail of every infinite path in $\mathcal{G}_\mathcal{P}$ as required.

The argument that if $\mathcal{G}_\mathcal{P}$ is a CS^ω proof then one can construct an infinitely progressing trace on a tail of any rooted infinite path in $\mathcal{T}_\mathcal{P}$ is very similar to the above. □

Corollary 5.1.8 (Soundness of CS^ω)**.** *If there is a CS^ω proof $\mathcal{P}$ of a sequent S then S is valid (according to the semantics of the infinitary system S^ω).*

Proof. If $\mathcal{P}$ is a CS^ω proof then its tree-unfolding $\mathcal{T}_\mathcal{P}$ is a S^ω proof by Proposition 5.1.7. The endsequent S of $\mathcal{P}$ is the endsequent of $\mathcal{T}_\mathcal{P}$ (since $s^*(root(\mathcal{T}_\mathcal{P})) = s^*((root(\mathcal{D}))) = s(root(\mathcal{D})) = S$), so S is valid by soundness of the full infinitary system S^ω (Proposition 4.2.10). □

Proposition 5.1.7 establishes that a CS^ω (pre-)proof $\mathcal{P}$ can be considered a finite representation of the S^ω (pre-)proof $\mathcal{T}_\mathcal{P}$. In fact, a CS^ω pre-proof always represents a *regular* S^ω proof, and every regular S^ω proof can be so represented; this follows from the well-known fact that a tree is regular exactly if it is representable by a cyclic graph [16, 46, 76]. We therefore only state, and do not prove, the following proposition:

Proposition 5.1.9. *Let $\mathcal{D}$ be a S^ω pre-proof. Then $\mathcal{D}$ is a regular tree if and only if there exists a CS^ω pre-proof $\mathcal{P}$ with $\mathcal{T}_\mathcal{P} = \mathcal{D}$.*

The condition for a CS^ω pre-proof to be a proof (c.f. Definition 5.1.6) is of course a global condition in the sense that it can be determined only by examining the entire pre-proof structure (in general, anyway). However, in contrast to the situation of S^ω — as discussed at the start of this chapter, it is not decidable whether a given S^ω pre-proof is a proof — the soundness condition for CS^ω is decidable:

Proposition 5.1.10. *It is decidable whether a CS^ω pre-proof is a CS^ω proof.*

Proof. (Sketch) We show that the property that every infinite path possesses a tail on which an infinitely progressing trace exists is an ω-regular property (similar arguments appear in [52, 62, 42]), and hence reducible to the emptiness of a Büchi automaton.

We write $\mathcal{L}(B)$ for the language accepted by a Büchi automaton B. Given a CS^ω pre-proof $\mathcal{P}$, we first construct a Büchi automaton $Trace$ such that $\mathcal{L}(Trace)$ is the set of strings of vertices of $\mathcal{G}_\mathcal{P}$ such that an infinitely progressing trace can be found on a suffix of the string (irrespective of whether the string is actually a path in $\mathcal{G}_\mathcal{P}$). We then construct the automaton $\overline{Trace}$ accepting the complemented language $\overline{\mathcal{L}(Trace)}$, i.e. the set of strings of vertices of $\mathcal{G}_\mathcal{P}$ such that no infinitely progressing trace exists on any suffix of the string. From $\overline{Trace}$ we build a final Büchi automaton $PrfDec$ accepting only those strings that are in $\mathcal{L}(\overline{Trace})$ and that also are valid paths in $\mathcal{G}_\mathcal{P}$. One can then easily see that $\mathcal{P}$ is a CS^ω proof if and only if $\mathcal{L}(PrfDec) = \emptyset$, which is a decidable problem. □

A full proof of Proposition 5.1.10 is given in Appendix A. It contains full details of the construction of the Büchi automaton $PrfDec$ for proof decision in CS^{ω}, and thus may be of interest to the reader concerned with the implementation of trace-based cyclic proof systems.

5.2 CLKID^{ω}: a cyclic proof system for $\mathbf{FOL_{ID}}$

In this section, we consider the cyclic proof system CLKID^{ω} for $\mathrm{FOL}_{\mathrm{ID}}$ obtained as the restriction of the full infinitary system LKID^{ω} to regular trees, as per Section 5.1 above. We give some concrete examples of cyclic proofs in CLKID^{ω}, and state our conjecture that the cut-eliminability property of LKID^{ω} does not hold in the restricted system CLKID^{ω}.

Example 5.2.1. Let Φ_{NEO} be the inductive definition set consisting of the productions for the "natural number" predicate N given in Example 2.2.5 and the productions for the "even and odd number" predicates E and O given in Example 2.2.6. The following is then a CLKID^{ω} proof of $Nx \vdash Ex \vee Ox$:

$$\dfrac{\dfrac{}{\vdash E0, O0}\,(E\mathrm{R}_1) \qquad \dfrac{\dfrac{\dfrac{\dfrac{Nx \vdash Ox, Ex\ (\dagger)}{Ny \vdash Oy, Ey}\,(\mathrm{Subst})}{Ny \vdash Oy, Osy}\,(O\mathrm{R}_1)}{Ny \vdash Esy, Osy}\,(E\mathrm{R}_2)}{x = sy, Ny \vdash Ex, Ox}\,(=\mathrm{L})}{\dfrac{Nx \vdash Ex, Ox\ (\dagger)}{Nx \vdash Ex \vee Ox}\,(\vee\mathrm{R})}\,(\text{Case } N)$$

We use the symbol $(\dagger)$ to indicate the pairing of the bud in the pre-proof above with a suitable companion. To see that the pre-proof satisfies the CLKID^{ω} proof condition, observe that any infinite path through the pre-proof graph necessarily has a tail consisting of repetitions of the path $(\dagger)\ldots(\dagger)$ from the companion to the bud in this proof, and there is a progressing trace following this path: (Nx, Ny, Ny, Ny, Ny, Nx). By concatenating copies of this trace one can thus obtain an infinitely progressing trace on a tail of any infinite path as required.

Example 5.2.2. The following is a CLKID^{ω} proof of the converse statement to the previous example, $Ex \vee Ox \vdash Nx$ (we use the symbols $(\dagger)$ and $(*)$ to indicate the pairing of companions with buds):

$$\dfrac{\dfrac{\dfrac{}{\vdash N0}\,(N\mathrm{R}_1)}{x = 0 \vdash Nx}\,(=\mathrm{L}) \qquad \dfrac{\dfrac{\dfrac{Ox \vdash Nx\ (\dagger)}{Oy \vdash Ny}\,(\mathrm{Subst})}{Oy \vdash Nsy}\,(N\mathrm{R}_2)}{x = sy, Oy \vdash Nx}\,(=\mathrm{L})}{Ex \vdash Nx\ (*)}\,(\text{Case } E) \qquad \dfrac{\dfrac{\dfrac{Ex \vdash Nx\ (*)}{Ey \vdash Ny}\,(\mathrm{Subst})}{Ey \vdash Nsy}\,(N\mathrm{R}_2)}{x = sy, Ey \vdash Nx}\,(=\mathrm{L})}{Ox \vdash Nx\ (\dagger)}\,(\text{Case } O)$$

$$\dfrac{Ex \vdash Nx\ (*) \qquad Ox \vdash Nx\ (\dagger)}{Ex \vee Ox \vdash Nx}\,(\vee\mathrm{L})$$

To see that this satisfies the CLKID$^\omega$ proof condition, observe that any infinite path through the pre-proof graph necessarily has a tail consisting of repetitions of the "figure-of-8" loop $(\dagger)\ldots(*)\ldots(\dagger)$ in this proof, and there is a trace with two progress points following this path: $(Ox, Ey, Ey, Ey, Ex, Oy, Oy, Oy, Ox)$. (Note that bud nodes are identified with their companions in the pre-proof graph.) As in the previous example, we can construct the required infinitely progressing trace on this tail by concatenating copies of this trace.

Example 5.2.3 ("The $P\&Q$ Example", Wirth [85] pp. 43–47)**.** Let Φ_{NPQ} be the inductive definition set consisting of the usual productions for the "natural number" predicate N (c.f. Example 2.2.5) together with the following productions for the unary predicate P and binary predicate Q:

$$\frac{}{P0} \qquad \frac{Px \quad Q(x,sx)}{Psx} \qquad \frac{}{Q(x,0)} \qquad \frac{Q(x,y) \quad Px}{Q(x,sy)}$$

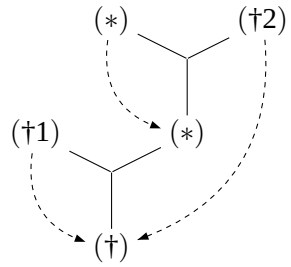
We wish to provide a CLKID$^\omega$ proof of the sequent $Nx, Ny \vdash Q(x,y)$, and begin as follows:

$$\dfrac{\dfrac{}{Nx \vdash Q(x,0)}\,(QR_1)}{Nx, y = 0 \vdash Q(x,y)}\,(=\!L) \qquad \dfrac{\dfrac{\dfrac{Nx, Ny \vdash Q(x,y)\ (\dagger 1)}{Nx, Nz \vdash Q(x,z)}\,(\text{Subst}) \qquad Nx \vdash Px\ (*)}{Nx, Nz \vdash Q(x,sz)}\,(QR_2)}{Nx, y = sz, Nz \vdash Q(x,y)}\,(=\!L)$$
$$\dfrac{\qquad\qquad}{Nx, Ny \vdash Q(x,y)\ (\dagger)}\,(\text{Case } N)$$

We then continue on the rightmost branch of the proof $(*)$ as follows:

$$\dfrac{\dfrac{}{\vdash P0}\,(PR_1)}{x = 0 \vdash Px}\,(=\!L) \qquad \dfrac{\dfrac{Nx \vdash Px\ (*)}{Nz \vdash Pz}\,(\text{Subst}) \qquad \dfrac{Ny, Nx \vdash Q(x,y)\ (\dagger 2)}{Nsz, Nz \vdash Q(z,sz)}\,(\text{Subst})}{Nsz, Nz \vdash Psz}\,(PR_2)}{Nx, x = sz, Nz \vdash Px}\,(=\!L)$$
$$\dfrac{\qquad\qquad}{\dfrac{Nx, Nx \vdash Px}{Nx \vdash Px\ (*)}\,(\text{ContrL})}\,(\text{Case } N)$$

Note that both of the buds $(\dagger 1)$ and $(\dagger 2)$ are assigned the companion $(\dagger)$. The following is a schematic representation of the pre-proof given above (solid lines denote paths in the derivation tree and dashed arrows indicate the assignation of companions to buds):



To see that this pre-proof is a CLKID$^\omega$ proof, consider an infinite path π through the pre-proof graph. There are two cases to consider:

- Neither the bud $(*)$ nor the bud $(\dagger 2)$ occur infinitely often on π, in which case π has a tail π' consisting only of repetitions of the tree path from the companion $(\dagger)$ to the bud $(\dagger 1)$. Note that there is a progressing trace following this path: (Ny, Nz, Nz, Nz, Ny), and we can construct an infinitely progressing trace on π' by composing copies of this trace.

- At least one of the buds $(*), (\dagger 2)$ occur infinitely often on π. Note that we have:

  - a non-progressing trace (Nx, Nx, Nx, Nx, Nx) following the path from $(\dagger)$ to $(\dagger 1)$;
  - a progressing trace $(Nx, Nx, Nx, Nx, Nx, Nz, Nz, Nz, Nx)$ following the path from $(\dagger)$ to $(\dagger 2)$;
  - a progressing trace (Nx, Nx, Nz, Nz, Nz, Nx) following the path from $(*)$ to $(*)$.

  Noting that the first two of these traces have the same value at the companion node $(\dagger)$, and the last two traces have the same value at the companion node $(*)$, it is then clear by inspection of the pre-proof that one can construct a trace following a tail of π by composing components of these traces as required. This trace is infinitely progressing because at least one of the two progressing traces above must occur infinitely often in it.

(In fact, the above is an informal argument that the pre-proof above has a *trace manifold*, which is sufficient to ensure it is a CLKID$^\omega$ proof. See Section 7.2 for full details.)

In the two previous chapters, we obtained cut-eliminability results for the finitary system LKID and for the infinitary system LKID$^\omega$. However, it seems probable that cut is not eliminable in the system CLKID$^\omega$, though providing a proof of this fact is apparently rather challenging. Note that the eliminability of cut in LKID$^\omega$ does not imply the eliminability of cut in CLKID$^\omega$. If a sequent has a CLKID$^\omega$ proof then this proof is also a LKID$^\omega$ proof, and thus there exists a cut-free LKID$^\omega$ proof of the same sequent; but there is no guarantee that the cut-free proof is a regular tree, and so need not be a CLKID$^\omega$ proof in general. In Chapter 7 we show that LKID proofs can be translated into CLKID$^\omega$ proofs by replacing each induction rule instance in an LKID proof with a CLKID$^\omega$ derivation. However, the CLKID$^\omega$ derivation of an induction rule necessarily involves an instance of the cut rule (because the induction rule for a predicate introduces an arbitrary formula in its premises). This does not mean that cut is not eliminable in CLKID$^\omega$, because there might nevertheless exist a different cut-free CLKID$^\omega$ proof of an arbitrary LKID-provable sequent. In any case, cut-free LKID proofs do not straightforwardly correspond to cut-free CLKID$^\omega$ proofs, and so even if LKID and CLKID$^\omega$ were equivalent systems (as is conjectured formally in Chapter 7), it would seem that the eliminability of cut in CLKID$^\omega$ would not follow from its eliminability in LKID. Also, note that CLKID$^\omega$ cannot be complete with respect to standard models (because the set of CLKID$^\omega$ proofs can be recursively enumerated), and in particular it is not cut-free complete

with respect to standard models so a semantic proof of cut-eliminability by soundness and cut-free completeness (as in the cut-eliminability proofs for LKID and LKID$^\omega$) is impossible.

Conjecture 5.2.4. *Cut is not eliminable in the system CLKID$^\omega$. That is to say, there exists a sequent $\Gamma \vdash \Delta$ of FOL_{ID} which is provable in CLKID$^\omega$, but is not provable without the use of the rule (Cut).*

The discussion above provides some indication that Conjecture 5.2.4 is likely to hold, but it is nevertheless far from clear how to supply a formal proof of the conjecture. Of course, one would need to demonstrate that there is a sequent that is provable in CLKID$^\omega$, but not provable without cut. A suitable candidate sequent might be one whose standard inductive proof (in LKID, say) requires the induction hypothesis to be a generalisation of the sequent, i.e. the inductive proof necessarily uses a formula of strictly greater logical complexity than any formula occurring in the root sequent. For examples demonstrating the need for generalisation in inductive proof see e.g. [12]. However, it does not immediately follow that a similar generalisation of the root sequent is then required for the construction of a CLKID$^\omega$ proof. Since any LKID-provable sequent is also cut-free LKID$^\omega$-provable, the crux of the argument would presumably lie in showing that no *regular* cut-free proof of the sequent can exist. Unfortunately, it is not completely obvious how one would make such an argument, and Conjecture 5.2.4 will for now have to be left as a possible direction for future work. Of course, while cut-elimination is a desirable property for proof systems generally, the non-eliminability of cut for CLKID$^\omega$ (if established) would not necessarily be disastrous. Rather, it would imply that the search for appropriate generalisations of inductive conjectures must be performed via cuts in CLKID$^\omega$, and not necessarily in conjunction with the application of induction, as is the case in LKID.

There is also a question of whether the substitution rule (Subst) is admissible in CLKID$^\omega$. In the example proofs given above, it is used as the last rule on a particular branch in order to ensure that a bud sequent is syntactically identical to a particular interior node, which can then be used as a companion for the bud. It appears very likely that this is a necessary step in general (at least in cut-free proofs), i.e. that (Subst) is not an eliminable rule in CLKID$^\omega$, but we have not investigated this in detail[1].

[1] It would be possible to work with a much weaker definition of companion (c.f. Defn 5.1.1) incorporating substitution and / or weakening, so that a companion for a bud labelled $\Gamma \vdash \Delta$ would be any node labelled $\Gamma' \vdash \Delta'$ such that, for some substitution θ, we have $\Gamma' \subseteq \Gamma[\theta]$ and $\Delta' \subseteq \Delta[\theta]$ (similar definitions are used by Sprenger and Dam [62] and Schöpp and Simpson [58]). In this case, it is easy to see that (Subst) is then eliminable, and the system may be more suitable for proof search; but from our theoretical standpoint there is the technical inconvenience that buds cannot be straightforwardly identified with their companions when forming the pre-proof graph, and in order to view cyclic proofs as representations of infinite proofs, one would have to add extra rule applications between bud nodes and their companions, or use some other suitable conversion.

Chapter 6

Cycle normalisation for cyclic proofs

In this chapter we shall examine the structure of (pre-)proofs in an arbitrary cyclic proof system CS^ω obtained as the cyclic restriction of an infinitary proof system S^ω equipped with a suitable trace pair function, as discussed in Chapter 5. As many CS^ω pre-proofs can represent the same infinite S^ω pre-proof, i.e. have the same tree-unfolding (up to isomorphism), these tree-unfoldings give us a natural notion of equivalence on CS^ω pre-proofs. Our primary tool for establishing such equivalences between pre-proofs is the *derivation graph homomorphism* (which is analogous to homomorphism on ordinary graphs). We develop the necessary proof machinery for establishing equivalences between CS^ω pre-proofs in Section 6.1.

Our definition of a CS^ω pre-proof allows essentially arbitrary complexity in their cycle structure; to form a pre-proof, bud nodes in a derivation tree may be assigned companions occurring anywhere in the tree. For example, in Example 5.2.2 of the previous chapter we saw a natural example of a cyclic proof in the system CLKID^ω with a "figure-of-8" structure. It is natural to investigate the extent to which this complexity is necessary: can the structure of an arbitrary CS^ω (pre)-proof be simplified in general? We give a positive answer to this question by demonstrating the *cycle-normalisation* property: every CS^ω pre-proof can be transformed into a pre-proof in which the repeat function assigns to each bud node an ancestor of the bud as a companion. Such pre-proofs are said to be in *cycle normal form*. Furthermore, the pre-proof thus obtained is equivalent to the original pre-proof in the aforementioned sense of having the same tree-unfolding. In one sense, cycle-normalisation is almost an obvious property: intuitively, since a CS^ω pre-proof represents a regular (infinite) tree, one can obtain a pre-proof in cycle normal form by traversing each infinite branch of the tree-unfolding and forming a cycle as soon as one encounters the same subtree twice along the branch. However, providing a formal proof along these lines is somewhat intricate and involves the consideration of many details. We give a complete proof of the equivalence-preserving cycle-normalisation property for arbitrary CS^ω pre-proofs via a folding operation on the infinite S^ω pre-proof represented by a CS^ω pre-proof in Section 6.2.

Finally, in Section 6.3 we give an alternative proof of cycle-normalisation via an iterative unfolding operation on pre-proofs (again yielding an equivalent pre-proof). Despite a large complexity bound on the size of the pre-proof thereby obtained, this proof nevertheless gives a direct algorithm for transforming a pre-proof into another in cycle-normal form.

6.1 Tree-unfolding equivalence on cyclic proofs

Definition 6.1.1 (Derivation graph homomorphism)**.** Let $G = (V, s, r, p)$ and $G' = (V', s', r', p')$ be derivation graphs. A *derivation graph homomorphism from G to G'* is a total function $f : V \to V'$ satisfying, for all $v \in V$:

$$\begin{array}{rcl} s'(f(v)) & = & s(v) \\ r'(f(v)) & \simeq & r(v) \\ p'_j(f(v))) & \simeq & f(p_j(v)) \quad \text{(for each } j \in \mathbb{N}) \end{array}$$

If G, G' are derivation graphs, we write $f : G \to G'$ to mean that f is a derivation graph homomorphism from G to G'.

Proposition 6.1.2 (Composition of derivation graph homomorphisms)**.** *Let $f_1 : G_1 \to G_2$ and $f_2 : G_2 \to G_3$ be derivation graph homomorphisms. Then their composition, $f_2 \circ f_1 : G_1 \to G_3$, is also a derivation graph homomorphism.*

Proof. The proof is an easy verification, but we include the details for completeness. For $i \in \{1,2,3\}$, let $G_i = (V_i, s_i, r_i, p_i)$. We just need to check the properties required for $f_2 \circ f_1$ to be a derivation graph homomorphism, which all follow from the fact that f_1 and f_2 are derivation graph homomorphisms:

- $s_1(v) = s_2(f_1(v)) = s_3(f_2(f_1(v)))$
- $r_1(v) \simeq r_2(f_1(v)) \simeq r_3(f_2(f_1(v)))$
- for any $j \in \mathbb{N}$, $f_2(f_1(p_{1_j}(v))) \simeq f_2(p_{2_j}(f_1(v))) \simeq p_{3_j}(f_2(f_1(v)))$

□

The fundamental property of derivation graph homomorphisms is that they preserve the sequent and rule labellings of paths in a derivation graph, and thus traces following those paths:

Lemma 6.1.3 (Path preservation under homomorphism)**.** *Let $G = (V, s, r, p)$ and $G' = (V', s', r', p')$ be derivation graphs such that there exists a derivation graph homomorphism $f : G \to G'$. Then for every infinite path $v_0 v_1 v_2 \ldots$ in G there is an infinite path $x_0 x_1 x_2 \ldots$ in G' with the same rule and sequent labelling, i.e. $s(v_i) = s'(x_i)$ and $r(v_i) = r'(x_i)$ for all $i \geq 0$. Furthermore, if f is surjective, then the converse holds.*

Proof. First, let $\pi_1 = v_0 j_0 v_1 j_1 v_2 j_2 \ldots$ be a path in G so that $r(v_i)$ is defined and $v_{i+1} = p_{j_i}(v_i)$ for all $i \geq 0$. As $f(v_{i+1}) = f(p_{j_i}(v_i)) = p'_{j_i}(f(v_i))$, it holds that that $f(v_0) j_0 f(v_1) j_1 f(v_2) j_2 \ldots$ is a path in G'. Furthermore, we have $s(v_i) = s'(f(v_i))$ and $r(v_i) = r'(f(v_i))$ as required.

Now suppose f is surjective and let $\pi_2 = x_0 j_0 x_1 j_1 x_2 j_2 \ldots$ be a path in G' so that $x_{i+1} = p'_{j_i}(x_i)$ for each i. Consider the path in G inductively defined by: let $v_0 \in V$ be such that $f(v_0) = x_0$ and define $v_{i+1} = p_{j_i}(v_i)$ for each $i \geq 0$. We prove by induction on i that v_i is well-defined and satisfies $f(v_i) = x_i$ for all $i \geq 0$. If $i = 0$ we are immediately done by the surjectivity of f. For $i = k+1$ we have $x_{k+1} = p'_{j_k}(x_k) = p'_{j_k}(f(v_k))$ (by IH) $= f(p_{j_k}(v_k)) = f(v_{k+1})$ and are done. So $v_0 j_0 v_1 j_1 v_2 j_2 \ldots$ is indeed a path in G. Furthermore, as f is a derivation graph homomorphism, we have $s(v_i) = s'(f(v_i)) = s'(x_i)$ and $r(v_i) = r'(f(v_i)) = r'(x_i)$ as required. □

Lemma 6.1.4 (Trace preservation under homomorphism)**.** *Let G and G' be derivation graphs and suppose there exists a derivation graph homomorphism $f : G \to G'$. Then if there is an infinitely progressing trace on some tail of every infinite path in G', then there is an infinitely progressing trace on some tail of every infinite path in G. Moreover, if f is surjective then the converse implication also holds.*

Proof. Let $G = (V, s, r, p)$ and $G' = (V', s', r', p')$. Suppose there exists an infinitely progressing trace on some tail of every infinite path in G', and consider an infinite path $v_0 v_1 v_2 \ldots$ in G. By Lemma 6.1.3 above, there exists an infinite path $x_0 x_1 x_2 \ldots$ in G' such that $s(v_i) = s'(v_i)$ and $r(v_i) = r'(x_i)$ for all $i \geq 0$. By assumption, there exists an infinitely progressing trace $\tau = \tau_n \tau_{n+1} \tau_{n+2} \ldots$ on some tail $x_n x_{n+1} x_{n+2} \ldots$ of this path. Now for any $i \geq n$:

$$\begin{aligned}
(\tau_i, \tau_{i+1}) \text{ is a valid trace pair on } (v_i, v_{i+1}) &\Leftrightarrow TPair(\tau_i, \tau_{i+1})(s(v_i), r(v_i), s(v_{i+1})) \neq 0 \\
&\Leftrightarrow TPair(\tau_i, \tau_{i+1})(s'(x_i), r'(x_i), s'(x_{i+1}) \neq 0 \\
&\Leftrightarrow (\tau_i, \tau_{i+1}) \text{ is a valid trace pair on } (x_i, x_{i+1})
\end{aligned}$$

where *TPair* is the trace pair function for S^ω (c.f. Definition 4.2.7). Similarly, (τ_i, τ_{i+1}) is a progressing trace pair on (v_i, v_{i+1}) exactly if it is on (x_i, x_{i+1}). Thus τ is the required infinitely progressing trace on a tail of $v_0 v_1 v_2 \ldots$.

For the converse implication, suppose that the derivation graph homomorphism f is surjective and that there is an infinitely progressing trace on some tail of every infinite path in G, and consider an infinite (rooted) path $x_0 x_1 x_2 \ldots$ in G'. Since f is surjective, by Lemma 6.1.3 there is an infinite path $v_0 v_1 v_2 \ldots$ in G such that $s(v_i) = s'(v_i)$ and $r(v_i) = r'(x_i)$ for all $i \geq 0$. The remainder of the argument is then similar to the case above. □

As indicated previously, we obtain a natural notion of equivalence on CS^ω pre-proofs by considering isomorphism (i.e. invertible derivation graph homomorphism) between their respective tree-unfoldings (c.f. Definition 5.1.4). Any two pre-proofs that are equivalent in this sense are also equivalent in the weaker sense that one satisfies the CS^ω proof condition if and only if the other does.

Definition 6.1.5 (Equivalence for cyclic pre-proofs)**.** Two CS^ω pre-proofs $\mathcal{P}$ and $\mathcal{P}'$ are said to be *equivalent*, written $\mathcal{P} \approx \mathcal{P}'$, if there is an invertible derivation graph homomorphism from $\mathcal{T}_\mathcal{P}$ to $\mathcal{T}_{\mathcal{P}'}$.

Proposition 6.1.6. *Let $\mathcal{P}$ and $\mathcal{P}'$ be CS^ω pre-proofs such that $\mathcal{P} \approx \mathcal{P}'$. Then $\mathcal{P}$ is a CS^ω proof iff $\mathcal{P}'$ is.*

Proof. First note that, by Proposition 5.1.7, $\mathcal{P}$ is a CS^ω proof iff $\mathcal{T}_\mathcal{P}$ is an S^ω proof, and $\mathcal{P}'$ is a CS^ω proof iff $\mathcal{T}_{\mathcal{P}'}$ is an S^ω proof. It therefore suffices to show that $\mathcal{T}_\mathcal{P}$ is an S^ω proof iff $\mathcal{T}_{\mathcal{P}'}$ is. Now since $\mathcal{P} \approx \mathcal{P}'$, there is a surjective (and injective) derivation graph homomorphism from $\mathcal{T}_\mathcal{P}$ to $\mathcal{T}_{\mathcal{P}'}$. By Lemma 6.1.4, there is an infinitely progressing trace on some tail of every infinite path in $\mathcal{T}_\mathcal{P}$ if and only if there is an infinitely progressing trace on some tail of every infinite path in $\mathcal{T}_{\mathcal{P}'}$, and thus indeed $\mathcal{T}_\mathcal{P}$ is an S^ω proof iff $\mathcal{T}_{\mathcal{P}'}$ is. □

Having defined a notion of equivalence between CS^ω pre-proofs, we now naturally are interested in transformations on a CS^ω pre-proof that yield an equivalent pre-proof. It turns out that two pre-proofs are equivalent if there exists a surjective derivation graph homomorphism between them that also preserves the root of their underlying derivation trees.

Proposition 6.1.7. *For any CS^ω pre-proof $\mathcal{P}$, there is a surjective derivation graph homomorphism $f_\mathcal{P}$ from $\mathcal{T}_\mathcal{P}$ to $\mathcal{G}_\mathcal{P}$.*

Proof. Let $\mathcal{P} = (\mathcal{D} = (V, s, r, p), \mathcal{R})$, so we have $\mathcal{G}_\mathcal{P} = (V', s, r, p')$ and $\mathcal{T}_\mathcal{P} = (\mathit{Path}(\mathcal{G}_\mathcal{P}), s^*, r^*, p^*)$ constructed according to Definitions 5.1.3 and 5.1.4 respectively. Now define $f_\mathcal{P} : \mathit{Path}(\mathcal{G}_\mathcal{P}) \to V'$ by: $f_\mathcal{P}((v_i)_{0 \leq i \leq n}) = v_n$. It is clear that $f_\mathcal{P}$ is well-defined and total on $\mathit{Path}(\mathcal{G}_\mathcal{P})$. Now for any $v \in V'$, there is a unique path in $\mathcal{D}$ from $\mathit{root}(\mathcal{D})$ to v (because $\mathcal{D}$ is a derivation tree). Since $v \in V' = V \setminus \mathrm{Bud}(\mathcal{D})$, it is not a bud node of $\mathcal{D}$, so this path is also a path in $\mathcal{G}_\mathcal{P}$ and thus there exists $(v_i)_{0 \leq i \leq n}$ in $\mathit{Path}(\mathcal{G}_\mathcal{P})$ with $v = v_n$. So $f_\mathcal{P}$ is surjective.

Now let $(v_i)_{0 \leq i \leq n}$ be an arbitrary element of $\mathit{Path}(\mathcal{G}_\mathcal{P})$. We need to check that $f_\mathcal{P}$ satisfies the properties required of a derivation graph homomorphism:

- By definition, we have $s^*((v_i)_{0 \leq i \leq n}) = s(v_n) = s(f_\mathcal{P}((v_i)_{0 \leq i \leq n}))$;
- Likewise, we have $r^*((v_i)_{0 \leq i \leq n}) = r(v_n) = r(f_\mathcal{P}((v_i)_{0 \leq i \leq n}))$;
- For any $j \in \mathbb{N}$, we have $f_\mathcal{P}(p^*_j((v_i)_{0 \leq i \leq n})) \simeq f_\mathcal{P}((v_i)_{0 \leq i \leq n}.p'_j(v_n)) \simeq p'_j(v_n) \simeq p'_j(f_\mathcal{P}((v_i)_{0 \leq i \leq n}))$ as required, which completes the proof.

□

Lemma 6.1.8. *Let G be a derivation graph, and T_1, T_2 be derivation trees, and let $f_1 : T_1 \to G$ and $f_2 : T_2 \to G$ be derivation graph homomorphisms such that $f_1(\mathit{root}(T_1)) = f_2(\mathit{root}(T_2))$. Then there is a map h such that $f_1 = f_2 \circ h$ and $h(\mathit{root}(T_1)) = \mathit{root}(T_2)$, and furthermore h is the unique root-preserving derivation graph homomorphism from T_1 to T_2.*

Proof. Let $G = (V, s, r, p)$, $T_1 = (V_1, s_1, r_1, p_1)$, and $T_2 = (V_2, s_2, r_2, p_2)$. Recall that the *height* of a node v in a tree T is the length of the unique path $root(\mathcal{D})\ldots v$ in T. Now for all $v \in V_1$, we define $h(v)$ satisfying $f_1(v) = (f_2(h(v)))$ and $h(root(T_1)) = root(T_2)$ by induction on the height n of v in T_1 as follows:

Case $n = 0$**:** We have $v = root(T_1)$ and define $h(v) = root(T_2)$, which is clearly the unique choice satisfying $h(root(T_1)) = root(T_2)$. We also have $f_1(root(T_1)) = f_2(root(T_2))$ by assumption, i.e. $f_1(v) = f_2(h(v))$ as required.

Case $n > 0$**:** We have $v = p_{1_j}(v')$ for a unique $j \in \mathbb{N}$ and $v' \in V_1$, where v' has height $n-1$ in T_1, and so by the induction hypothesis, $f_1(v') = f_2(h(v'))$, where $h(v')$ is uniquely defined. We thus have $f_1(v) = f_1(p_{1_j}(v')) = p_j(f_1(v')) = p_j(f_2(h(v'))) = f_2(p_{2_j}(h(v')))$ by the derivation graph homomorphism properties of f_1 and f_2 and by the induction hypothesis.

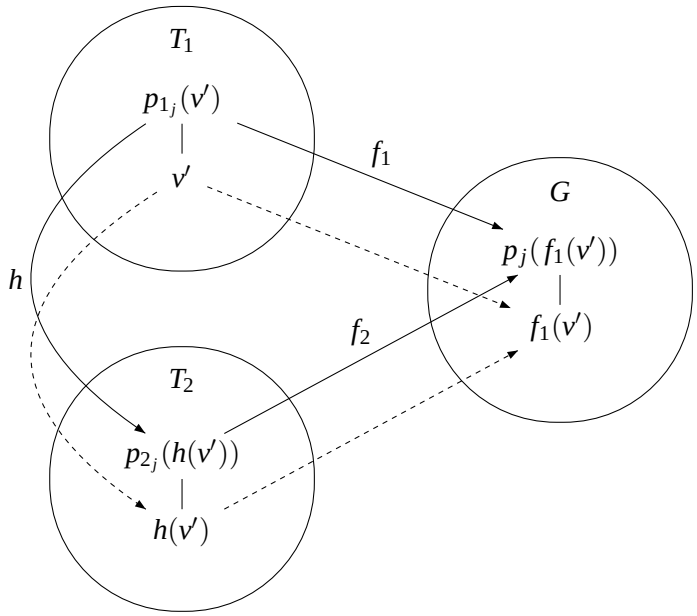


Figure 6.1: Inductive step of the construction of the map h in Lemma 6.1.8. The mappings assumed for the induction hypothesis are denoted by dashed arrows and those we construct by solid arrows.

We thus define $h(v) = h(p_{1_j}(v')) = p_{2_j}(h(v'))$ which clearly satisfies $f_1(v) = f_2(h(v))$ by the above. Note that $h(v)$ is thus uniquely defined since $h(v')$ and j are uniquely defined, and T_2 is a derivation tree. The construction of h in this case is illustrated in Figure 6.1. We also have $h(root(T_1)) = root(T_2)$ by induction hypothesis and are therefore done.

We show that h is a derivation graph homomorphism from T_1 to T_2. First, for all $v \in V_1$, we have $s_1(v) = s(f_1(v)) = s(f_2(h(v))) = s_2(h(v))$ by the construction of h and using the derivation

graph homomorphism properties of f_1 and f_2. By a similar argument we also have $r_1(v) \simeq r_2(h(v))$. We then just need to substantiate $h(p_{1_j}(v)) = p_{2_j}(h(v))$, which follows immediately from the step-case definition of h. This completes the proof.

□

Lemma 6.1.9. *Let G be a derivation graph, and T_1, T_2 be derivation trees, and let $f_1 : T_1 \to G$ and $f_2 : T_2 \to G$ be derivation graph homomorphisms such that $f_1(root(T_1)) = f_2(root(T_2))$. Then there is an invertible derivation graph homomorphism from T_1 to T_2.*

Proof. By applying Lemma 6.1.8 there are unique root-preserving derivation graph homomorphisms h, h' from T_1 to T_2 and from T_2 to T_1 respectively, and there are also unique root-preserving derivation graph homomorphisms from T_1 to T_1 and from T_2 to T_2. This situation is illustrated in Figure 6.2:

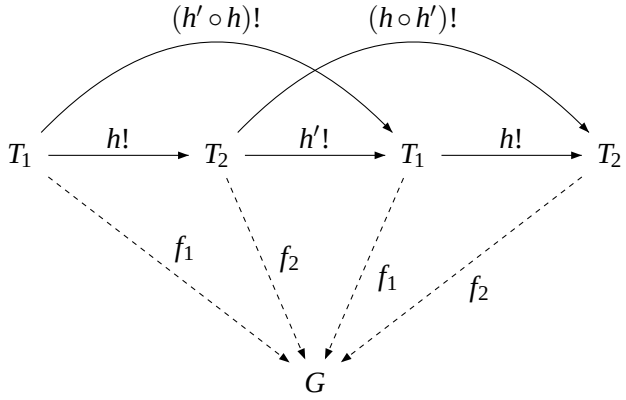


Figure 6.2: Proof of Lemma 6.1.9. The dashed lines denote the derivation graph homomorphisms assumed for the theorem and the solid lines denote the unique derivation graph homomorphisms constructed by applying Lemma 6.1.8.

As $h' \circ h$ is a (root-preserving) derivation graph homomorphism from T_1 to T_1 by Proposition 6.1.2, and the identity function *id* is trivially a (root-preserving) derivation graph homomorphism from T_1 to T_1, we must have $h' \circ h = id$ by uniqueness of the derivation graph homomorphism from T_1 to T_1. Similarly, we deduce $h \circ h' = id$, and so h is an invertible derivation graph homomorphism (with inverse h') from T_1 to T_2. □

Theorem 6.1.10 (Pre-proof equivalence theorem)**.** *Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ and $\mathcal{P}' = (\mathcal{D}', \mathcal{R}')$ be CS^ω pre-proofs. If there exists a surjective derivation graph homomorphism $g : \mathcal{G}_{\mathcal{P}'} \to \mathcal{G}_{\mathcal{P}}$ satisfying $g(root(\mathcal{D}')) = root(\mathcal{D})$, then $\mathcal{P} \approx \mathcal{P}'$.*

Proof. The proof is illustrated in Figure 6.3. By Proposition 6.1.7 we have a surjective derivation graph homomorphisms $f_{\mathcal{P}'} : \mathcal{T}_{\mathcal{P}'} \to \mathcal{G}_{\mathcal{P}'}$ and $f_{\mathcal{P}} : \mathcal{T}_{\mathcal{P}} \to \mathcal{G}_{\mathcal{P}}$ such that $f_{\mathcal{P}'}(root(\mathcal{T}_{\mathcal{P}'})) = root(\mathcal{D}')$ and $f_{\mathcal{P}}(root(\mathcal{T}_{\mathcal{P}})) = root(\mathcal{D})$. By Proposition 6.1.2, we can compose $f_{\mathcal{P}'}$ and the

given homomorphism g to obtain a surjective derivation graph homomorphism $(g \circ f_{\mathcal{P}'}) : \mathcal{T}_{\mathcal{P}'} \to \mathcal{G}_{\mathcal{P}}$ such that $(g \circ f_{\mathcal{P}'})(root(\mathcal{T}_{\mathcal{P}'})) = g(root(\mathcal{D}')) = root(\mathcal{D}) = f_{\mathcal{P}}(root(\mathcal{T}_{\mathcal{P}}))$. Hence we can apply Lemma 6.1.9 to obtain an invertible derivation graph homomorphism from $\mathcal{T}_{\mathcal{P}}$ to $\mathcal{T}_{\mathcal{P}'}$.

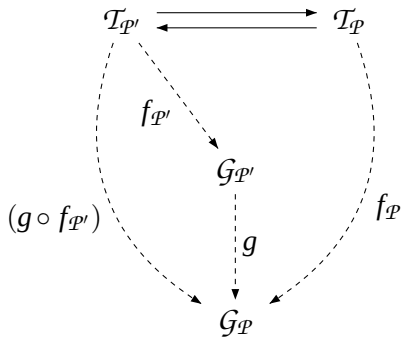


Figure 6.3: Proof of Theorem 6.1.10. The dashed arrows denote the surjective derivation graph homomorphisms given by Proposition 6.1.7 and by assumption, and the solid arrows denote the invertible derivation graph homomorphisms constructed by applying Lemma 6.1.9.

□

6.2 Cycle normalisation via tree-unfoldings

In this section we give a formal proof of the approach to the cycle-normalisation problem informally described at the beginning of the chapter. Given a CS^{ω} pre-proof $\mathcal{P}$, we show how to perform a folding operation on its infinite tree-unfolding $\mathcal{T}_{\mathcal{P}}$ (which is an S^{ω} pre-proof) to obtain a new CS^{ω} pre-proof, equivalent to the original one, in *cycle normal form*:

Definition 6.2.1 (Cycle normal form)**.** Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a CS^{ω} pre-proof. $\mathcal{P}$ is said to be in *cycle normal form* if, for all $B \in \mathrm{Bud}(\mathcal{D})$, the companion $\mathcal{R}(B)$ is an ancestor of B in $\mathcal{D}$.

Our main result of this section states that an S^{ω} pre-proof can be "folded" into an equivalent CS^{ω} pre-proof in cycle normal form provided one can find suitable "folding points" on each infinite branch of the S^{ω} pre-proof. In particular, for any CS^{ω} pre-proof $\mathcal{P}$, folding points are given for its tree-unfolding $\mathcal{T}_{\mathcal{P}}$ by the homomorphism $f_{\mathcal{P}}$, whence it follows that any CS^{ω} pre-proof $\mathcal{P}$ can be transformed into an equivalent pre-proof in cycle normal form by folding $\mathcal{T}_{\mathcal{P}}$.

Theorem 6.2.2 (Tree-folding theorem)**.** *Let T be an S^{ω} pre-proof, let G be a finite derivation graph, and let f be a surjective derivation graph homomorphism from T to G. Also, for each infinite rooted branch $\pi = v_0 v_1 v_2 \ldots$ in T, let $m_\pi, n_\pi \in \mathbb{N}$ be such that $m_\pi < n_\pi$ and $f(v_{m_\pi}) = f(v_{n_\pi})$. Note that m_π, n_π must exist since G is finite.*

Then there exists a CS$^\omega$ pre-proof $\mathcal{P}$ in cycle normal form (whose endsequent is the endsequent of T), and there are surjective derivation graph homomorphisms $T \to \mathcal{G}_\mathcal{P} \to G$. Furthermore, the homomorphism from $\mathcal{G}_\mathcal{P}$ to G maps $root(\mathcal{D})$ to $f(root(T))$.

Proof. Let $T = (V_T, s_T, r_T, p_T)$ and $G = (V_G, s_G, r_G, p_G)$, and define $\mathcal{D} = (V, s, r, p)$ and $\mathcal{R}$ as follows:

- $V = \{v \in V_T \mid$ for all infinite rooted branches $\pi = v_0 v_1 v_2 \ldots$in T, if $v = v_k$ then $k \leq n_\pi\}$;
- $s(v) = s_T(v)$ for all $v \in V$;
- if $v \in V$ and $v = v_{n_\pi}$ for some infinite rooted branch $\pi = v_0 v_1 v_2 \ldots$ in T, then $r(v)$ and $p_j(v)$ (for all j) are undefined, i.e. v is a bud node of $\mathcal{D}$ and we define $\mathcal{R}(v) = v_{m_\pi}$. (If there is more than one branch π meeting this criterion, we may choose any suitable m_π.) Otherwise we define $r(v) = r_T(v)$ and $p_j(v) \simeq p_{T_j}(v)$ (for all j).

First, to verify that $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ is a CS$^\omega$ pre-proof we need to show that $\mathcal{D}$ is a finite S$^\omega$ derivation tree whose buds are assigned suitable companions by $\mathcal{R}$. $\mathcal{D}$ is clearly a S$^\omega$ derivation graph; to see that it is a derivation tree, we observe that as $root(T)$ is clearly in V, and since $p \simeq p_T$ except on bud nodes of $\mathcal{D}$, the unique path in T from $root(T)$ to $v \in V$ is the unique path in $\mathcal{D}$ from $root(\mathcal{D}) = root(T)$ to v. Since $\mathcal{D}$ is a derivation tree, it is finitely branching, and as every infinite branch $\pi = v_0 v_1 v_2 \ldots$ in T is "cut off" at the point v_{n_π} in $\mathcal{D}$ (or before), it contains no infinite branches. $\mathcal{D}$ is thus a finite tree by the well-known König's Lemma (which states that any finitely branching tree is infinite iff it contains an infinite branch). To verify that $\mathcal{R}$ is a repeat function for $\mathcal{D}$, we need to show that each bud node B of $\mathcal{D}$ is assigned a companion C such that $r(C)$ is defined and $s(C) = s(B)$. Observe that, as T is an S$^\omega$ pre-proof and so has no bud nodes, each bud of $\mathcal{D}$ is of the form v_{n_π} for some infinite rooted branch $\pi = v_o v_1 v_2 \ldots$ in T and we have by construction $\mathcal{R}(v_{n_\pi}) = v_{m_\pi}$, where $m_\pi < n_\pi$ and $f(v_{n_\pi}) = f(v_{m_\pi})$. Thus $\mathcal{R}(B)$ is always an ancestor of B as required. As $m_\pi < n_\pi$, we have that v_{m_π} cannot be a bud node of $\mathcal{D}$, and so $r(v_{m_\pi}) = r_T(v_{m_\pi})$ is defined (notice that r_T is defined on all $v \in V_T$ since T is an S$^\omega$ pre-proof). Also, since f is a derivation graph homomorphism from T to G we have $s_T(v_{n_\pi}) = s_G(f(v_{n_\pi})) = s_G(f(v_{m_\pi})) = s_T(v_{m_\pi})$, i.e. $s(v_{n_\pi}) = s(v_{m_\pi})$ as required. Thus $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ is indeed a CS$^\omega$ pre-proof in cycle normal form as claimed.

Now, to establish the existence of the required derivation graph homomorphisms $T \to \mathcal{G}_\mathcal{P} \to G$, we first establish the following auxiliary property:

$$\forall v \in V_T . \exists x \in V.\ f(v) = f(x)$$

To see this, we let $v \in V_T$ and proceed by induction on the height h of v in T. If $h = 0$, then $v = root(T)$, which is clearly in V, so we are immediately done. If $h > 0$ then $v = p_{T_j}(v')$ for

some $j \in \mathbb{N}$ and $v' \in V_T$. Since v' has height $h-1$ in T, by induction hypothesis there exists $x' \in V$ such that $f(x') = f(v')$. Note that we have $f(v) = f(p_{T_j}(v')) = p_{G_j}(f(v')) = p_{G_j}(f(x'))$, using the induction hypothesis and the fact that f is a derivation graph homomorphism from T to G. It therefore suffices to show that $f(x) = p_{G_j}(f(x'))$ for some $x \in V$. Now there are two cases to consider:

- Suppose that x' is not a bud node of $\mathcal{D}$ and let $x = p_{T_j}(x')$. As f is a derivation graph homomorphism from T to G, we have $f(x) = f(p_{T_j}(x')) = p_{G_j}(f(x'))$, and as x' is not a bud node of $\mathcal{D}$, we thus have $x = p_{T_j}(x') \in V$ and are done;

- If x' is a bud node of $\mathcal{D}$, then there is an infinite branch π in T such that $x' = v_{n_\pi}$. Let $x = p_{T_j}(v_{m_\pi})$ and observe that since f is a derivation graph homomorphism from T to G, we have $f(x) = f(p_{T_j}(v_{m_\pi})) = p_{G_j}(f(v_{m_\pi})) = p_{G_j}(f(v_{n_\pi})) = p_{G_j}(f(x'))$. Also, as v_{m_π} is a companion node in $\mathcal{D}$, it is not a bud of $\mathcal{D}$ and thus we have $x = p_{T_j}(v_{m_\pi}) \in V$ as required.

This completes the proof of the auxiliary property and we can now show the existence of the required derivation graph homomorphisms.

We claim that f (restricted to V') is a surjective derivation graph homomorphism from $\mathcal{G}_{\mathcal{P}} = (V', s, r, p')$ to G. (Note that $\mathcal{G}_{\mathcal{P}}$ is obtained by identifying bud nodes with their companions in $\mathcal{P}$, c.f. Definition 5.1.3.) We just need to check the three homomorphism properties and surjectivity:

- For all $v \in V'$, we have $s(v) = s_T(v) = s_G(f(v))$, since f is a derivation graph homomorphism from T to G;

- For all $v \in V' (= V \setminus \mathrm{Bud}(\mathcal{D}))$, v is not a bud of $\mathcal{D}$ and so we have $r(v) = r_T(v) = r_G(f(v))$ since f is a derivation graph homomorphism from T to G;

- For all $v \in V'$ and $j \in \mathbb{N}$, v is not a bud node of $\mathcal{D}$ and so we have $p_j(v) \simeq p_{T_j}(v)$ by construction. As f is a derivation graph homomorphism from T to G, we have $f(p_j(v)) \simeq f(p_{T_j}(v)) \simeq p_{G_j}(f(v))$. It therefore suffices to show $f(p'_j(v)) \simeq f(p_j(v))$. If $p_j(v)$ is not a bud of $\mathcal{D}$ then $p'_j(v) \simeq p_j(v)$ and we are done. If $p_j(v)$ is a bud of $\mathcal{D}$, then $p'_j(v) \simeq \mathcal{R}(p_j(v))$ and by construction of $\mathcal{R}$ (as discussed above) we have $f(\mathcal{R}(p_j(v))) = f(p_j(v))$ and are likewise done. This completes the proof that f is a derivation graph homomorphism from $\mathcal{G}_{\mathcal{P}}$ to G.

- For all $v \in V_G$, there exists $v' \in V_T$ such that $f(v') = v$ since f is surjective from T to G. By the auxiliary property proven above, there is then an $x \in V$ such that $f(x) = f(v') = v$. If x is not a bud of $\mathcal{D}$, then $x \in V'$ and we are done. If x is a bud of $\mathcal{D}$, then notice that

$\mathcal{R}(x) \in V'$ and by construction of $\mathcal{R}$ we have $f(\mathcal{R}(x)) = f(x) = v$ as required. f is thus surjective from $\mathcal{G}_{\mathcal{P}}$ to G.

Next, we define a total function $g : V_T \to V'$ by induction on the height h of $v \in V_T$ in T, and simultaneously prove that it satisfies $f(g(v)) = f(v)$ for all $v \in V_T$ as follows:

For $h = 0$, we have $v = \mathit{root}(T)$ and we define $g(v) = g(\mathit{root}(T)) = \mathit{root}(T)$, which is clearly in V'. Note that we thus have $f(g(\mathit{root}(T))) = f(\mathit{root}(T))$ as required. For $h > 0$, we have $v = p_{T_j}(v')$ where v' has a height of $h-1$ in T and so by the induction hypothesis, $g(v') \in V'$ is defined and we have $f(g(v')) = f(v')$. Note that, using the induction hypothesis and the fact that f is a derivation graph homomorphism, we have:

$$f(v) = f(p_{T_j}(v')) = p_{G_j}(f(v')) = p_{G_j}(f(g(v'))) = f(p_{T_j}(g(v')))$$

so $p_{T_j}(g(v')) \in V_T$ is defined. Now we define $g(v) = p'_j(g(v'))$. Since $g(v') \in V'$, we have that $g(v')$ is not a bud of $\mathcal{D}$, so $p_j(g(v')) = p_{T_j}(g(v'))$, and so $p'_j(g(v')) \in V'$ is also defined (since $p'_j(v)$ is defined iff $p_j(v)$ is for all $v \in V'$). To see that $f(g(v)) = f(v)$, it suffices by the above to establish that $f(p'_j(g(v'))) = f(p_{T_j}(g(v')))$. If $p_j(g(v'))$ is not a bud of $\mathcal{D}$, then $p'_j(g(v')) = p_j(g(v')) = p_{T_j}(g(v'))$ and we are immediately done. If on the other hand $p_j(g(v'))$ is a bud of $\mathcal{D}$, then we have $p'_j(g(v')) = \mathcal{R}(p_j(g(v')))$, and by construction of $\mathcal{R}$ we thus have

$$f(p'_j(g(v'))) = f(\mathcal{R}(p_j(g(v')))) = f(p_j(g(v'))) = f(p_{T_j}(g(v'))$$

as required.

Finally, we just need to show that g is a derivation graph homomorphism and surjective from T to $\mathcal{G}_{\mathcal{P}}$:

- For all $v \in V_T$, since f is a derivation graph homomorphism from T to G, we have $s_T(v) = s_G(f(v)) = s_G(f(g(v)))$ by construction of g, and since f is also a derivation graph homomorphism from $\mathcal{G}_{\mathcal{P}}$ to G, we have $s_G(f(g(v))) = s(g(v))$. Thus $s_T(v) = s(g(v))$ as required;

- By a similar argument to the above, $r_T(v) = r_G(f(v)) = r_G(f(g(v)))$, and $r_G(f(g(v))) = r(g(v))$, so $r_T(v) = r(g(v))$ for all $v \in V_T$ as required;

- By construction of g we have $g(p_{T_j}(v)) \simeq p'_j(g(v))$ for all $v \in V_T$ as required.

- It is not hard to see that g is the identity function on all $v \in V' \subseteq V_T$ and is thus surjective from V_T to V'. We prove this by induction on the height h of v in T; if $h = 0$, $v = \mathit{root}(T)$ and we have $g(\mathit{root}(T)) = \mathit{root}(T)$. If $h > 0$, then $v = p_{T_j}(v')$ for some $v' \in V'$ and $j \in \mathbb{N}$, and by the induction hypothesis we have $g(v') = v'$. Thus by definition of g we have $g(v) = g(p_{T_j}(v')) = p'_j(g(v')) = p'_j(v')$. Now, v' is not a bud of $\mathcal{D}$ since $v' \in V' = V \setminus \mathrm{Bud}(\mathcal{D})$, so we also have $v = p_{T_j}(v') = p_j(v') = p'_j(v')$ and are done (the final equality holds since $v \in V'$ and so $p_j(v')$ is not a bud of $\mathcal{D}$).

This completes the proof. □

Corollary 6.2.3 (Equivalence-preserving cycle-normalisation)**.** *For any CS$^\omega$ pre-proof $\mathcal{P}$ not in cycle normal form, there exists an CS$^\omega$ pre-proof $\mathcal{P}'$ in cycle normal form such that $\mathcal{P} \approx \mathcal{P}'$. Furthermore, if n is the number of nodes in the derivation tree of $\mathcal{P}$, then the derivation tree of $\mathcal{P}'$ contains no more than $\Sigma_{0\leq i\leq n+1} c^i$ nodes, for some constant c.*

Proof. Let $\mathcal{P}$ be a CS$^\omega$ proof not in cycle normal form, and take $T = \mathcal{T}_\mathcal{P}$, $G = \mathcal{G}_\mathcal{P}$ and $f = f_\mathcal{P}$ in Theorem 6.2.2 above (note that $f_\mathcal{P}$ is indeed a surjective derivation graph homomorphism by Proposition 6.1.7). Let $\mathcal{P}'$ be the "folded" CS$^\omega$ pre-proof in cycle normal form constructed by the theorem, and note that, by the theorem, $\mathcal{P}'$ has the same endsequent as $\mathcal{P}$ and we have a surjective derivation graph homomorphism from $\mathcal{G}_{\mathcal{P}'}$ to $\mathcal{G}_\mathcal{P}$ mapping $root(\mathcal{D}')$ to $f_\mathcal{P}(root(\mathcal{T}_\mathcal{P})) = root(\mathcal{D})$. Thus we have $\mathcal{P} \approx \mathcal{P}'$ by Theorem 6.1.10.

For the complexity bound, we observe that the length of each branch of the derivation tree $\mathcal{D}'$ need be no more than $n+1$, since we only need to find for each infinite branch π of $\mathcal{T}_\mathcal{P}$ distinct nodes v, v' on the branch such that $f_\mathcal{P}(v) = f_\mathcal{P}(v')$, i.e. such that v and v' correspond to the same original node of $\mathcal{P}$. Thus the size of the tree $\mathcal{D}'$ is bounded above by $\Sigma_{0\leq i\leq n+1} c^i$, where the constant c is the maximum branching factor of $\mathcal{D}'$, i.e., the maximum number of premises of any rule of S$^\omega$. □

6.3 Cycle normalisation via iterated unfoldings

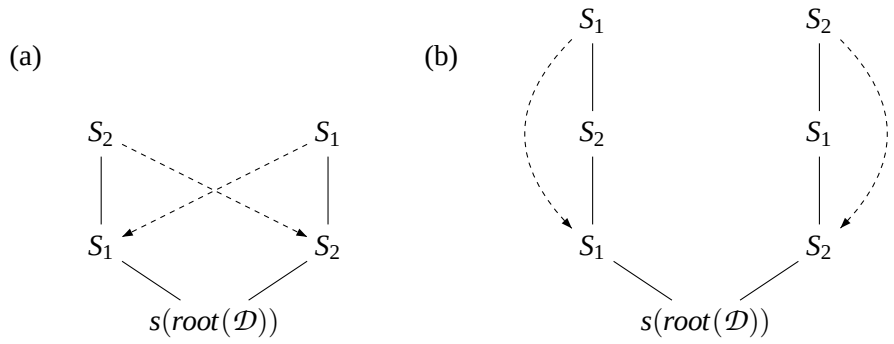


Figure 6.4: (a) A schematic pre-proof not in cycle normal form. Solid lines denote paths in the derivation tree and dashed arrows indicate the assignation of companions to buds. By unfolding this pre-proof we can obtain one that is in cycle normal form (b). Moreover, the unfolded pre-proof clearly has the same infinite tree-unfolding as the original pre-proof.

In this section, we give an alternative proof of the cycle-normalisation property that yields an algorithm for transforming an arbitrary CS$^\omega$ pre-proof into an equivalent pre-proof in cycle normal form. Figure 6.4 shows how a CS$^\omega$ pre-proof with 2 buds and companions arranged

in a "figure-of-8" configuration may be unfolded so that each bud node in the unfolded pre-proof has an ancestor node as companion (recall that we gave an example of a CLKID$^\omega$ proof in this form in Example 5.2.1). Furthermore, it is clear that the unfolded pre-proof has the same infinite tree-unfolding as the original pre-proof, i.e. the unfolded pre-proof is equivalent to the original one. This example motivates our alternative cycle-normalisation proof, which transforms a cyclic pre-proof into an equivalent pre-proof in cycle normal form via a finite number of such unfoldings. Before we supply the proof, we first develop some machinery for describing the unfolding process:

Definition 6.3.1 (Tangled companion)**.** Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a CS$^\omega$ pre-proof and let C be a companion node appearing in $\mathcal{P}$. Then C is said to be *tangled* (in $\mathcal{P}$) if there is a bud node $B \in \text{Bud}(\mathcal{D})$ such that $\mathcal{R}(B) = C$ and C is not an ancestor of B in $\mathcal{D}$.

Definition 6.3.2 (Entanglement set)**.** Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a CS$^\omega$ pre-proof and define the *entanglement set of* $\mathcal{P}$, written $Etgl(\mathcal{P})$, by:

$$Etgl(\mathcal{P}) = \{C \mid \exists B.\ \mathcal{R}(B) = C \text{ and either } C \text{ is tangled or } C \text{ has a tangled descendant in } \mathcal{D}\}$$

(Note that $Etgl(\mathcal{P})$ is a subset of the companion nodes occurring in $\mathcal{P}$.)

Lemma 6.3.3. *A CS$^\omega$ pre-proof $\mathcal{P}$ is in cycle normal form iff $Etgl(\mathcal{P}) = \emptyset$.*

Proof. Letting $\mathcal{P} = (\mathcal{D}, \mathcal{R})$, it is clear that $\mathcal{P}$ is in cycle normal form, i.e. $\mathcal{R}(B)$ is an ancestor of B for all bud nodes B, iff no companion node in $\mathcal{P}$ is tangled. Now if $Etgl(\mathcal{P}) = \emptyset$, then we immediately have that no companion node in $\mathcal{P}$ is tangled and we are done. Conversely, if no companion node in $\mathcal{P}$ is tangled, then it is clear that also no companion node in $\mathcal{P}$ can have a tangled descendant, and we thus have $Etgl(\mathcal{P}) = \emptyset$ as required. □

Lemma 6.3.4. *Let $\mathcal{P}$ be a CS$^\omega$ pre-proof with $Etgl(\mathcal{P}) \neq \emptyset$. Then there is some companion node $C \in Etgl(\mathcal{P})$ such that C is tangled, but has no tangled descendants.*

Proof. Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$. Pick any $C \in Etgl(\mathcal{P})$ of maximal height in $\mathcal{D}$ (we can do so since $Etgl(\mathcal{P})$ is non-empty); it is then clear that no descendant of C can be in $Etgl(\mathcal{P})$, since any descendant of C would have a greater height in $\mathcal{D}$. C must therefore be tangled in $\mathcal{P}$. □

We now prove the main lemma required for (equivalence-preserving) cycle-normalisation: any CS$^\omega$ pre-proof not in cycle normal form can be transformed to an equivalent pre-proof with the same endsequent and a smaller entanglement set. It follows that any CS$^\omega$ pre-proof can be transformed into an equivalent pre-proof in cycle-normal form by iterating this transformation a finite number of times.

Lemma 6.3.5. *There is an operation, $Untangle$, defined on CS$^\omega$ pre-proofs $\mathcal{P}$ not in cycle normal form such that all of the following hold:*

1. *$Untangle(\mathcal{P})$ is a CS^{ω} pre-proof with the same endsequent as $\mathcal{P}$;*
2. *$\mathcal{P} \approx Untangle(\mathcal{P})$;*
3. *$Etgl(Untangle(\mathcal{P})) \subset Etgl(\mathcal{P})$;*
4. *if n is the number of nodes in the derivation tree of $\mathcal{P}$, then the number of nodes in the derivation tree of $Untangle(\mathcal{P})$ is bounded by n^2.*

Proof. Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a CS^{ω} pre-proof not in cycle normal form, and note that $Etgl(\mathcal{P}) \neq \emptyset$ by Lemma 6.3.3. By Lemma 6.3.4, there is then a companion node C occurring in $\mathcal{P}$ such that C is tangled in $\mathcal{P}$ but has no tangled descendants. Now we define a non-empty subset of $\mathrm{Bud}(\mathcal{D})$ by:

$$\mathcal{B}_C = \{B \in \mathrm{Bud}(\mathcal{D}) \mid \mathcal{R}(B) = C \text{ and } C \text{ is not an ancestor of } B\}$$

and write ∇_C for the subtree of $\mathcal{D}$ rooted at C. We then define $Untangle(\mathcal{P}) = (\mathcal{D}_U, \mathcal{R}_U)$, where $\mathcal{D}_U, \mathcal{R}_U$ are obtained as follows:

- $\mathcal{D}_U = (V_U, s_U, r_U, p_U)$ is the derivation tree obtained from $\mathcal{D} = (V, s, r, p)$ by replacing each bud node $B \in \mathcal{B}_C$ with a copy of the subtree ∇_C. Technically, for all $B \in \mathcal{B}_C$ and for each node $v \in \nabla_C$ we create a new node $\mathrm{cp}_B(v)$ and define:

$$V_U = V \setminus \mathcal{B}_C \cup \{\mathrm{cp}_B(v) \mid B \in \mathcal{B}_C, v \in \nabla_C\}$$

$$s_U(v) = \begin{cases} s(v) & \text{if } v \in V \setminus \mathcal{B}_C \\ s(v') & \text{if } v = \mathrm{cp}_B(v') \text{ for some } B \in \mathcal{B}_C \text{ and } v' \in \nabla_C \end{cases}$$

$$r_U(v) \simeq \begin{cases} r(v) & \text{if } v \in V \setminus \mathcal{B}_C \\ r(v') & \text{if } v = \mathrm{cp}_B(v') \text{ for some } B \in \mathcal{B}_C \text{ and } v' \in \nabla_C \end{cases}$$

$$\forall j \in \mathbb{N}.\ p_{U_j}(v) \simeq \begin{cases} p_j(v) & \text{if } v \in V \setminus \mathcal{B}_C \text{ and } p_j(v) \in V \setminus \mathcal{B}_C \\ \mathrm{cp}_B(C) & \text{if } v \in V \setminus \mathcal{B}_C \text{ and } p_j(v) = B \in \mathcal{B}_C \\ \mathrm{cp}_B(p_j(v')) & \text{if } v = \mathrm{cp}_B(v') \text{ for some } B \in \mathcal{B}_C \text{ and } v' \in \nabla_C \end{cases}$$

- $\mathcal{R}_U$ is defined on bud nodes $B_U \in \mathrm{Bud}(\mathcal{D}_U)$ as follows:

$$\mathcal{R}_U(B_U) = \begin{cases} \mathcal{R}(B_U) & \text{if } B_U \in \mathrm{Bud}(\mathcal{D}) \\ \mathcal{R}(B') & \text{if } B_U = \mathrm{cp}_B(B') \text{ for some } B \in \mathcal{B}_C \text{ and } B' \in \mathrm{Bud}(\nabla_C), \\ & \text{and } \mathcal{R}(B') \notin \nabla_C \\ \mathrm{cp}_B(\mathcal{R}(B')) & \text{if } B_U = \mathrm{cp}_B(B') \text{ for some } B \in \mathcal{B}_C \text{ and } B' \in \mathrm{Bud}(\nabla_C), \\ & \text{and } \mathcal{R}(B') \in \nabla_C \end{cases}$$

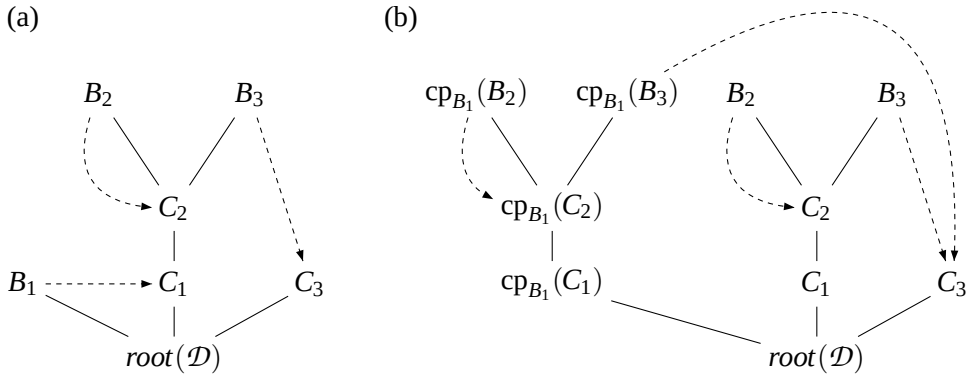


Figure 6.5: (a) A schematic pre-proof containing a tangled companion node C_1. Solid lines denote paths in the derivation tree and dashed arrows indicate the assignation of companions to buds. (b) The same pre-proof after applying the operation $Untangle$ defined in Lemma 6.3.5. The bud node B_1 has been replaced by a copy of the subtree rooted at C_1.

The $Untangle$ operation on a pre-proof not in cycle normal form is illustrated in Figure 6.5. It now remains to establish the properties of $Untangle(\mathcal{P})$ required by the lemma:

1. Note that $\mathcal{D}_U$ has been obtained from the derivation tree $\mathcal{D}$ by replacing each bud node $B \in \mathcal{B}_C$ with a copy of the derivation tree ∇_C. This replacement does not change the sequent labelling of the node being replaced, since for all $B \in \mathcal{B}_C$, the endsequent of ∇_C is $C = \mathcal{R}(B)$ and we have $s(B) = s(C)$. It is thus clear that $\mathcal{D}_U$ is a finite S^ω derivation tree and that $root(\mathcal{D}_U) = root(\mathcal{D})$.

   To show that $\mathcal{R}_U$ is a repeat function for $\mathcal{D}_U$, we must show that $s_U(\mathcal{R}_U(B_U)) = s_U(B_U)$ for all bud nodes $B_U \in \mathrm{Bud}(\mathcal{D}_U)$. If $B_U \in V \setminus \mathcal{B}_C$ then $B_U \in \mathrm{Bud}(\mathcal{D})$, so we have $s_U(\mathcal{R}_U(B_U)) = s(\mathcal{R}(B_U)) = s(B_U)$ (the last equality follows since $\mathcal{R}$ is a repeat function) and $s_U(B_U) = s(B_U)$, and are done. Otherwise, $B_U = \mathrm{cp}_B(B')$ for some $B \in \mathcal{B}_C$ and $B' \in \mathrm{Bud}(\nabla_C)$. We thus have $s_U(B_U) = s(B')$. Now if $\mathcal{R}(B') \notin \nabla_C$ then:

   $$s_U(\mathcal{R}_U(B_U)) = s_U(\mathcal{R}(B')) = s(R(B')) = s(B')$$

   as required (note that $\mathcal{R}(B') \in V \setminus \mathcal{B}_C$). Otherwise, if $\mathcal{R}(B') \in \nabla_C$ then:

   $$s_U(\mathcal{R}_U(B_U)) = s_U(\mathrm{cp}_B(\mathcal{R}(B'))) = s(\mathcal{R}(B')) = s(B')$$

   and we are again done.

   Thus $Untangle(\mathcal{P}) = (\mathcal{D}_U, \mathcal{R}_U)$ is a CS^ω pre-proof with the same endsequent as $\mathcal{P}$.

2. By Theorem 6.1.10, it suffices to establish a surjective derivation graph homomorphism from $\mathcal{G}_{Untangle(\mathcal{P})} = (V'_U, s_U, r_U, p'_U)$ to $\mathcal{G}_\mathcal{P} = (V', s, r, p')$ that maps $root(\mathcal{D}_U) = root(\mathcal{D})$ to $root(\mathcal{D})$. Note that $\mathcal{G}_\mathcal{P}$ and $\mathcal{G}_{Untangle(\mathcal{P})}$ are obtained by identifying each bud node with

its companion in $\mathcal{D}$ and $\mathcal{D}_U$ respectively (c.f. Definition 5.1.3). Now define $f : V'_U \to V'$ by:

$$f(v) = \begin{cases} v & \text{if } v \in V' \\ v' & \text{if } v = \mathrm{cp}_B(v') \text{ for some } B \in \mathcal{B}_C \text{ and } v' \in \nabla_C \end{cases}$$

Note that we immediately have $f(\mathit{root}(\mathcal{D})) = \mathit{root}(\mathcal{D})$ as required since $\mathit{root}(\mathcal{D})$ is obviously in V'. Now let $v' \in V' = V \setminus \mathrm{Bud}(\mathcal{D})$. As v' is not a bud of $\mathcal{D}$, we must have $v' \in V \setminus \mathcal{B}_C$ and so $v' \in V_U$. As the only buds of $\mathcal{D}_U$ are buds of $\mathcal{D}$ or copies of buds of $\mathcal{D}$, it holds that v' is not a bud of $\mathcal{D}_U$ so $v' \in V'_U = V_U \setminus \mathrm{Bud}(\mathcal{D}_U)$ and, as $v' \in V'$, we have $f(v') = v'$. Thus f is indeed surjective.

To show that f is a homomorphism we require to prove, for all $v \in V'_U, s_U(v) = s(f(v))$, $r_U(v) \simeq r(f(v))$ and $f(p'_{U\,j}(v)) \simeq p'_j(f(v))$ for all j. Let $v \in V'_U$, and note that we have:

$$V'_U = (V \setminus \mathcal{B}_C \cup \{\mathrm{cp}_B(v') \mid B \in \mathcal{B}_C, v' \in \nabla_C\}) \setminus \mathrm{Bud}(\mathcal{D}_U)$$

so as v is not a bud of $\mathcal{D}_U$ and not in $\mathcal{B}_C$, v cannot be a bud of $\mathcal{D}$. We now divide into cases as follows:

- **Case** $v \in V \setminus \mathcal{B}_C$**.** Note that, as v is not a bud of $\mathcal{D}$, we also have $v \in V'$ and so $f(v) = v$. We therefore require to prove $s_U(v) = s(v), r_U(v) \simeq r(v)$ and $f(p'_{U_j}(v)) \simeq p'_j(v)$ for all j. It is immediate from the definitions and the case assumption that $s_U(v) = s(v)$ and $r_U(v) \simeq r(v)$. Note that since $p_j(v) \in V$, either $p_j(v) \in V \setminus \mathcal{B}_C$ or $p_j(v) \in \mathcal{B}_C$, so we divide into further subcases.

  First suppose $p_j(v) \in V \setminus \mathcal{B}_C$, so we have $p_{U_j}(v) = p_j(v)$ and we require to show $f(p'_j(v)) \simeq p'_j(v)$. If $p_j(v) \in \mathrm{Bud}(\mathcal{D})$ then $p'_j(v) = \mathcal{R}(p_j(v)) \in V'$, and if $p_j(v) \notin \mathrm{Bud}(\mathcal{D})$ then $p'_j(v) = p_j(v) \in V'$. In either case we thus have $f(p'_j(v)) = p'_j(v)$ and are done.

  Now suppose $p_j(v) = B \in \mathcal{B}_C$, so that we have $p_{U_j}(v) = \mathrm{cp}_B(C)$. Note that $\mathrm{cp}_B(C)$ cannot be a bud of $\mathcal{D}_U$ since C cannot be a bud of $\mathcal{D}$, so:

  $$f(p'_{U_j}(v)) = f(p'_{U_j}(v)) = f(\mathrm{cp}_B(C)) = C$$

  But also $p'_j(v) = \mathcal{R}(B) = C$, so we are done.

- **Case** $v = \mathrm{cp}_B(v')$ for some $B \in \mathcal{B}_C$ and $v' \in \nabla_C$. We thus have $f(v) = v'$ and therefore require to prove $s_U(v) = s(v'), r_U(v) \simeq r(v')$ and $f(p'_{U_j}(v)) \simeq p'_j(v')$ for all j. It is immediate from the definitions and the case assumption that $s_U(v) = s(v')$ and $r_U(v) \simeq r(v')$. We now divide into further subcases.

First suppose that $p_j(v')$ is not a bud of $\mathcal{D}$, so $p'_j(v') = p_j(v')$. Then $\mathrm{cp}_B(p_j(v'))$ cannot be a bud of $\mathcal{D}_U$, and thus:

$$f(p'_{U_j}(v')) = f(p_{U_j}(v')) = f(\mathrm{cp}_B(p_j(v'))) = p_j(v')$$

as required. Now suppose that $p_j(v')$ is a bud of $\mathcal{D}$, so $p'_j(v) = \mathcal{R}(p_j(v))$. Then $\mathrm{cp}_B(p_j(v'))$ is a bud of $\mathcal{D}_U$, so we have:

$$f(p'_{U_j}(v')) = f(\mathcal{R}_U(p_{U_j}(v'))) = f(\mathcal{R}_U(\mathrm{cp}_B(p_j(v'))))$$

Now if $p_j(v') \notin \nabla_C$ then:

$$f(\mathcal{R}_U(\mathrm{cp}_B(p_j(v')))) = f(\mathcal{R}(p_j(v'))) = \mathcal{R}(p_j(v'))$$

since $\mathcal{R}(p_j(v')) \in V'$ as required. On the other hand, if $p_j(v') \in \nabla_C$ then

$$f(\mathcal{R}_U(\mathrm{cp}_B(p_j(v')))) = f(\mathrm{cp}_B(\mathcal{R}(p_j(v')))) = \mathcal{R}(p_j(v'))$$

and we are likewise done. This completes the case.

Thus f is a derivation graph homomorphism, as required.

3. We first show the non-strict inclusion. Suppose for contradiction that some companion is in $Etgl(Untangle(\mathcal{P}))$ but not in $Etgl(\mathcal{P})$. So for some bud node $B_U \in \mathrm{Bud}(\mathcal{D}_U)$, $\mathcal{R}_U(B_U) \notin Etgl(\mathcal{P})$ and $\mathcal{R}_U(B_U)$ is not an ancestor of B_U in $\mathcal{D}_U$. By definition of $\mathcal{R}_U$, there are three cases to consider:

   - $B_U \in \mathrm{Bud}(\mathcal{D})$ and $\mathcal{R}_U(B_U) = \mathcal{R}(B_U)$, so $\mathcal{R}(B_U)$ is not an ancestor of B_U in $\mathcal{D}_U$. By construction of $\mathcal{D}_U$, it is readily seen that $\mathcal{R}(B_U)$ cannot be an ancestor of B_U in $\mathcal{D}$ either, i.e., $\mathcal{R}(B_U)$ is tangled in $\mathcal{P}$. Thus $\mathcal{R}_U(B_U) = \mathcal{R}(B_U) \in Etgl(\mathcal{P})$ and we have a contradiction;
   - $B_U = \mathrm{cp}_B(B')$ for some $B \in \mathcal{B}_C$ and $B' \in \mathrm{Bud}(\nabla_C)$, and $\mathcal{R}(B') \notin \nabla_C$, so $\mathcal{R}_U(B_U) = \mathcal{R}(B')$. Now if $\mathcal{R}(B')$ is an ancestor of C in $\mathcal{D}$, then as C is tangled in $\mathcal{P}$, $\mathcal{R}(B')$ has a tangled descendant in $\mathcal{P}$. On the other hand, if $\mathcal{R}(B')$ is not an ancestor of C in $\mathcal{D}$ then, as $B' \in \nabla_C$, $\mathcal{R}(B')$ is not an ancestor of B' in $\mathcal{D}$, so $\mathcal{R}(B')$ is tangled in $\mathcal{P}$. In either case, $\mathcal{R}_U(B_U) = \mathcal{R}(B') \in Etgl(\mathcal{P})$ and we again have a contradiction.
   - $B_U = \mathrm{cp}_B(B')$ for some $B \in \mathcal{B}_C$ and $B' \in \mathrm{Bud}(\nabla_C)$, and $\mathcal{R}(B') \in \nabla_C$. Note that $\mathcal{R}(B')$ is thus a descendant of C and so must be an ancestor of B' in $\mathcal{D}$, for otherwise $\mathcal{R}(B')$ would be a tangled descendant of C in $\mathcal{P}$, contrary to our initial choice of C. In that case, it is clear that $\mathcal{R}_U(B_U) = \mathrm{cp}_B(\mathcal{R}(B'))$ is then an ancestor of B_U in $\mathcal{D}_U$. But this is a contradiction, which finally establishes the inclusion $Etgl(Untangle(\mathcal{P})) \subseteq Etgl(\mathcal{P})$.

To see that the strict inclusion holds, we first observe that we have $C \in Etgl(\mathcal{P})$ by construction. Also, since C was chosen to have no tangled descendants in $\mathcal{P}$, it cannot have any tangled descendants in $Untangle(\mathcal{P})$, otherwise there would be some companion node appearing in ∇_C that is in $Etgl(Untangle(\mathcal{P}))$ but not in $Etgl(\mathcal{P})$. Finally, to see that C is not tangled in $Untangle(\mathcal{P})$, we observe that $\mathcal{R}_u$ only maps bud nodes appearing in ∇_C to C, since no $B \in \mathcal{B}_C$ exists in the tree $\mathcal{D}_U$. We thus have $C \notin Etgl(Untangle(\mathcal{P}))$ and are done.

4. Let $n = |V|$, $b = |\mathcal{B}_C|$ and c be the number of nodes in ∇_C. Then $|V_U| = n - b + bc = n + b(c-1)$. Clearly we have $b + c \leq n$, i.e. $c \leq n - b$, so $|V_U| \leq n - b + n(n-b) = (n-b)(1+n)$. Since $\mathcal{P}$ is not in cycle normal form we have $b \geq 1$, in which case $(n-b)(1+n) \leq n^2$ and so $|V_U| \leq n^2$ as required.

□

Theorem 6.3.6 (Equivalence-preserving cycle-normalisation (2))**.** *Any CS$^\omega$ pre-proof $\mathcal{P}$ not in cycle normal form can be transformed into a CS$^\omega$ pre-proof $\mathcal{P}'$ in cycle normal form such that $\mathcal{P} \approx \mathcal{P}'$. Furthermore, if n is the number of nodes in the derivation tree of $\mathcal{P}$, then the derivation tree of $\mathcal{P}'$ has no more than $n^{2^{n/2}}$ nodes.*

Proof. Given a CS$^\omega$ pre-proof $\mathcal{P}$ not in cycle normal form, we define a sequence $(P_i)_{i\geq 0}$ of CS$^\omega$ pre-proofs by $\mathcal{P}_0 = \mathcal{P}$ and $\mathcal{P}_{i+1} = Untangle(\mathcal{P}_i)$ if $\mathcal{P}_i$ is not in cycle normal form, and $\mathcal{P}_{i+1} = \mathcal{P}_i$ otherwise for each $i \geq 0$. The first 3 properties of Lemma 6.3.5 then respectively establish the following, for all $i \geq 0$:

1. $\mathcal{P}_i$ is a CS$^\omega$ pre-proof with the same endsequent as $\mathcal{P}$;
2. $\mathcal{P}_i \approx \mathcal{P}$;
3. if $\mathcal{P}_i$ is not in cycle normal form then $Etgl(\mathcal{P}_{i+1}) \subset Etgl(\mathcal{P}_i)$.

As $Etgl(\mathcal{P})$ is certainly finite, there is thus a $j \geq 0$ such that $Etgl(\mathcal{P}_j) = \emptyset$, i.e. $\mathcal{P}_j$ is in cycle normal form by Lemma 6.3.3.

Finally, we observe that the operation $Untangle$ need be applied no more than $e = |Etgl(\mathcal{P})|$ times to $\mathcal{P}$ to obtain a proof in cycle normal form, i.e., $j \leq e$. By property 4 of Lemma 6.3.5, the size of the derivation tree of $\mathcal{P}_i$ has size at most n^{2^i} for each $i \geq 0$. Notice that since there is at least one bud for each companion, we must have $e \leq n/2$, so the derivation tree of $\mathcal{P}_j$ has size bounded by $n^{2^{n/2}}$. □

We remark that we have undertaken only a very crude complexity analysis of cycle-normalisation via the $Untangle$ operation in the proofs above, and we would expect cycle-normalisation to yield proofs rather smaller than the bound given above might suggest.

In concluding this chapter, we remark that cycle-normalisation is of interest for three main reasons:

1. Cycle-normalisation has implications for proof search in implementations of CLKID$^\omega$ and related cyclic proof systems. In particular, the fact that every sequent provable in a given cyclic proof system CS$^\omega$ also has an equivalent proof in cycle normal form implies that it is sufficient, when searching for a proof, to consider only ancestors of a bud node as potential companions. However, our rough complexity analysis indicates that proofs in cycle normal form are (in the worst case) exponentially larger than the smallest possible proof not in cycle normal form.

2. Further, from the point of view of structural analysis of cyclic (pre-)proofs, it is convenient to be able to assume a simplified proof structure. We can from now on assume that a given cyclic pre-proof is in cycle normal form without loss of generality, which will be essential to our definition of a simplified soundness condition for cyclic proof systems in the next chapter.

3. Finally, the proof machinery developed in order to prove cycle-normalisation is of independent interest and may turn out to be of use in the future for proving further results concerning the structure of proofs.

Chapter 7

Trace manifolds

In this chapter we continue our investigation into the structure of cyclic proofs in an arbitrary cyclic proof system CS^ω equipped with an appropriate notion of trace, as defined in Sections 5.1 and 4.2.1. Specifically, we shall be concerned with analysing the general trace condition qualifying CS^ω pre-proofs as proofs (c.f. Definition 5.1.6), which is both computationally and combinatorially complex due to the quantification over all infinite paths in the pre-proof graph. In order to simplify our structural analysis of cyclic proofs, we consider the formulation of an alternative condition that implies the general trace condition while providing a greater degree of explicit structural information on pre-proofs.

First of all, in Section 7.1 we analyse the composition of infinite paths in the graph of a CS^ω pre-proof. Since pre-proofs can be assumed to be in cycle normal form by our results in the previous chapter, every infinite path in the pre-proof graph can be shown to have a tail composed of segments of the elementary "cycles" from companions to buds in a pre-proof. Furthermore, the involved cycles are weakly connected by a certain relation. In Section 7.2 we exploit this fact in order to formulate an alternative soundness condition for CS^ω pre-proofs — the existence of a *trace manifold* for the pre-proof — which is more restrictive than the general soundness condition, but with the corresponding benefits of being finitary and more explicit. A trace manifold consists of traces following finite path segments in the derivation tree together with conditions ensuring that for any infinite path, the segments can be "glued together" to yield an infinitely progressing trace on a tail of that path. We give two formulations of a trace manifold — the first employing a direct quantification over all strongly connected subgraphs of the pre-proof graph, and the second employing the auxiliary notion of an *induction order* as used in [57, 63, 62] — and prove them equivalent.

Finally, in Section 7.3, we return once more to the setting of our original proof systems for $\mathrm{FOL}_{\mathrm{ID}}$, and demonstrate that CLKID^ω subsumes LKID by giving a translation from LKID proofs to CLKID^ω proofs. Furthermore, the translated proofs satisfy the trace manifold condition. We conclude the chapter by stating our main open conjecture, which is that a sequent is

provable in LKID if and only if it is provable in CLKID$^\omega$. A positive answer to this conjecture would give substance to a claim that proof by induction is equivalent to regular proof by infinite descent, and clarify the status of the principles of induction and infinite descent as used by mathematicians and in mechanised theorem proving.

7.1 Analysing the general trace condition

Definition 7.1.1 (Strong / weak graph connectivity)**.** A directed graph G with vertex set V and edge set E is said to be *strongly connected* if for any $v, v' \in V$, there is a path in G from v to v'. G is said to be *weakly connected* if its "undirected version" G' with vertex set V and edge set $E \cup \{(v', v) \mid (v, v') \in E\}$ is strongly connected. G is said to be *non-trivial* if $|V| > 1$ or if it contains exactly one vertex v and there is an edge $(v, v) \in E$.

Definition 7.1.2 (Induced subgraph)**.** Let $G = (V, s, r, p)$ be a derivation graph and let $V' \subseteq V$. Then the *subgraph of G induced by V'* is defined to be $S = (V', s, r, p')$, where p' is defined by:

$$p'_j(v) = \begin{cases} p_j(v) & \text{if } v \in V' \text{ and } p_j(v) \in V' \\ \text{undefined} & \text{otherwise} \end{cases}$$

for each $j \in \mathbb{N}$. Equivalently, viewing G as a (labelled) graph (V, E) (c.f. Definition 5.1.3), the subgraph of G induced by V' is (V', E'), where $E' = \{(v, v') \in E \mid v \in V' \text{ and } v' \in V'\}$. In a slight abuse of notation, we write $S \subseteq G$ to denote that S is a subgraph of G.

Note that a subgraph of a derivation graph is not technically a derivation graph in general, as it need not contain only well-formed proof rule instances.

The general CS$^\omega$ proof condition (c.f. Definition 5.1.6) requires that an infinitely progressing trace exists on some tail of every infinite path in the pre-proof graph. We can analyse this requirement in greater detail. First, we show that every infinite path has a tail that is a path in some strongly connected subgraph of the pre-proof graph:

Lemma 7.1.3. *Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a CS$^\omega$ pre-proof. Then for every infinite path π through $\mathcal{G}_\mathcal{P}$ there is a non-trivial strongly connected subgraph $S_\pi \subseteq \mathcal{G}_\mathcal{P}$ and a tail π' of π such that π' is an infinite path through S_π on which every vertex of S_π occurs infinitely often.*

Proof. Let π be an infinite path through $\mathcal{G}_\mathcal{P}$ and let V be the set of vertices of $\mathcal{G}_\mathcal{P}$. As V is finite, at least one vertex in V occurs infinitely often on π. Let $V' \subseteq V$ be the set of vertices which occur infinitely often on π, and observe that there is a tail π' of π such that only the nodes in V' appear on π'. Now let S_π be the subgraph of $\mathcal{G}_\mathcal{P}$ induced by V' and note that, since all vertices occurring on π' are in V', we must have that π' is an infinite path in S_π. To see that S_π is strongly connected, let $v, v' \in V'$, and observe that as v, v' occur infinitely often on π', there is a path from v to v' in S_π. Further, there is a path of length > 0, so S_π is non-trivial.

□

We remark that to satisfy the general proof condition it suffices to find, for any infinite path π in the pre-proof graph, an infinitely progressing trace on the tail π' of π constructed in Lemma 7.1.3 above.

Definition 7.1.4 (Path composition)**.** Let G be a graph, let $\pi_1 = v_0 \dots v_m$ be a finite path in G and let $\pi_2 = x_0 x_1 x_2 \dots$ be a (possibly infinite) path in G. We say that π_1 *composes with* π_2 if $v_m = x_0$, and define the composed path $\pi_1\pi_2$ to be $v_0 \dots v_{m-1} x_0 x_1 x_2 \dots$.

Definition 7.1.5 (Basic cycle)**.** Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a CS^ω pre-proof in cycle normal form and let $B \in \mathrm{Bud}(\mathcal{D})$. Then the *basic cycle* $\mathcal{C}_B$ *in* $\mathcal{G}_\mathcal{P}$ is the path in $\mathcal{G}_\mathcal{P}$ obtained from the unique path from $\mathcal{R}(B)$ to B in $\mathcal{D}$ by replacing the unique edge (v, B) on this path (for some v) with the edge $(v, \mathcal{R}(B))$ of $\mathcal{G}_\mathcal{P}$. (Note that $\mathcal{C}_B$ is thus both a path in $\mathcal{G}_\mathcal{P}$ and a non-trivial strongly connected subgraph of $\mathcal{G}_\mathcal{P}$.)

We remark that the notion of a *basic cycle* is only well defined for pre-proofs in cycle normal form, as the existence of a path in $\mathcal{D}$ from $\mathcal{R}(B)$ to B requires that $\mathcal{R}(B)$ is an ancestor of B. The basic cycles of a graph then induce the following relation on its bud nodes (adapted very slightly from its original usage in [63]):

Definition 7.1.6 (Structural connectivity, Sprenger and Dam [63])**.** Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a CS^ω pre-proof in cycle normal form. Define the relation $\leq_\mathcal{P}$ on $\mathrm{Bud}(\mathcal{D})$ by: $B_2 \leq_\mathcal{P} B_1$ iff $\mathcal{R}(B_2)$ appears on the basic cycle $\mathcal{C}_{B_1}$ in $\mathcal{G}_\mathcal{P}$.

We observe that the relation $\leq_\mathcal{P}$ is *not* a partial order in general. In particular, if two distinct buds B_1 and B_2 share the same companion then we have $B_1 \leq_\mathcal{P} B_2 \leq_\mathcal{P} B_1$ but $B_1 \neq B_2$, so $\leq_\mathcal{P}$ is not necessarily antisymmetric. Also, as shown in Figure 7.1, it is not necessarily transitive either.

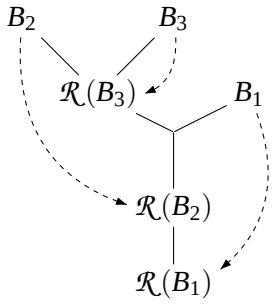


Figure 7.1: A schematic pre-proof (in cycle normal form). Solid lines indicate paths in the derivation tree and dashed arrows indicate the assignation of companions to buds. We have $B_2 \leq_\mathcal{P} B_1$ since $\mathcal{R}(B_2)$ occurs on the basic cycle $\mathcal{C}_{B_1}$, and likewise we have $B_3 \leq_\mathcal{P} B_2$. However, note that we do *not* then have $B_3 \leq_\mathcal{P} B_1$, because $\mathcal{R}(B_3)$ does not occur on the cycle $\mathcal{C}_{B_1}$.

Our next lemma establishes that any (non-trivial) strongly connected subgraph in the pre-proof graph can be viewed as a union of basic cycles, and furthermore, that the bud nodes

corresponding to these cycles are weakly connected by the relation $\leq_{\mathcal{P}}$. (Note that cycles are graphs, so a union of cycles is given by the usual union on graphs.)

Lemma 7.1.7. *Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a pre-proof in cycle normal form and let S be a non-trivial subgraph of $\mathcal{G}_{\mathcal{P}}$. Then S is strongly connected if and only if there exists a non-empty set of buds $\mathcal{B}_S \subseteq Bud(\mathcal{D})$ such that $\mathcal{B}_S$ is weakly $\leq_{\mathcal{P}}$-connected and $S = \bigcup_{B \in \mathcal{B}_S} \mathcal{C}_B$.*

Proof. Let S be a non-trivial subgraph of $\mathcal{G}_{\mathcal{P}}$. We prove each direction of the required bi-implication separately:

($\Leftarrow$) Let $\mathcal{B}_S \subseteq \mathrm{Bud}(\mathcal{D})$ be non-empty and weakly $\leq_{\mathcal{P}}$-connected and let $S = \bigcup_{B \in \mathcal{B}_S} \mathcal{C}_B$. To show that S is strongly connected, we let v, v' be vertices of S and show that there exists a path in S from v to v'. First note that as $S = \bigcup_{B \in \mathcal{B}_S} \mathcal{C}_B$, there exist $B, B' \in \mathcal{B}_S$ such that v occurs in $\mathcal{C}_B$ and v' occurs in $\mathcal{C}_{B'}$. Since any basic cycle is strongly connected and $\mathcal{C}_B, \mathcal{C}_{B'}$ are paths in S, there are paths in S from v to $\mathcal{R}(B)$ and from $\mathcal{R}(B')$ to v', so it suffices to show that there is a path in S from $\mathcal{R}(B)$ to $\mathcal{R}(B')$. Now as $\mathcal{B}_S$ is weakly $\leq_{\mathcal{P}}$-connected, there exists a chain $B_0 B_1 \dots B_n$ in $\mathcal{B}_S$ such that $B_0 = B$, $B_n = B'$ and, for all $i \in \{0, \dots, n-1\}$, either $B_i \leq_{\mathcal{P}} B_{i+1}$ or $B_{i+1} \leq_{\mathcal{P}} B_i$. We proceed by induction on the length n of this chain. If $n = 0$ then we have $B = B'$, and are done as there is trivially a path in S from $\mathcal{R}(B)$ to $\mathcal{R}(B)$. If $n > 0$, then by the induction hypothesis there is a path in S from $\mathcal{R}(B)$ to $\mathcal{R}(B_{n-1})$, so it suffices to show that there is a path in S from $\mathcal{R}(B_{n-1})$ to $\mathcal{R}(B_n) = \mathcal{R}(B')$. Now either $B_{n-1} \leq_{\mathcal{P}} B_n$ or $B_n \leq_{\mathcal{P}} B_{n-1}$, i.e., either $\mathcal{R}(B_{n-1})$ occurs on $\mathcal{C}_{B_n}$ or $\mathcal{R}(B_n)$ occurs on $\mathcal{C}_{B_{n-1}}$. In either case, there is a path in $\mathcal{C}_{B_{n-1}} \cup \mathcal{C}_{B_n}$ from $\mathcal{R}(B_{n-1})$ to $\mathcal{R}(B_n)$ and this path is a path in S since $\mathcal{C}_{B_{n-1}} \cup \mathcal{C}_{B_n} \subseteq S$. This completes the case.

($\Rightarrow$) Suppose S is strongly connected. We require to prove that there exists a weakly $\leq_{\mathcal{P}}$-connected set $\mathcal{B}_S \subseteq \mathrm{Bud}(\mathcal{D})$ such that $S = \bigcup_{B \in \mathcal{B}_S} \mathcal{C}_B$. Suppose the following claims hold:

1. for every node v in S, there exists $B \in \mathrm{Bud}(\mathcal{D})$ such that v occurs on $\mathcal{C}_B$ and $\mathcal{C}_B \subseteq S$;
2. if $\mathcal{C}_B \subseteq S$ and $\mathcal{C}_{B'} \subseteq S$ then there exists a chain $B_0 B_1 \dots B_n$ such that $\mathcal{C}_{B_i} \subseteq S$ for all i, $B_0 = B$ and $B_n = B'$, and for all i, either $B_i \leq_{\mathcal{P}} B_{i+1}$ or $B_{i+1} \leq_{\mathcal{P}} B_i$.

Now define $\mathcal{B}_S = \{B \mid \mathcal{C}_B \subseteq S\}$. Note that $\mathcal{B}_S$ is non-empty due to the non-triviality of S. We then have $\bigcup_{B \in \mathcal{B}_S} \mathcal{C}_B \subseteq S$ since $\mathcal{C}_B \subseteq S$ for each $B \in \mathcal{B}_S$. By claim 1 above, $v \in S$ implies there is a $B \in \mathcal{B}_S$ such that v is on $\mathcal{C}_B$. Thus $S \subseteq \bigcup_{B \in \mathcal{B}_S} \mathcal{C}_B$ and so $S = \bigcup_{B \in \mathcal{B}_S} \mathcal{C}_B$ as required. Now let $B, B' \in \mathcal{B}_S$, so that $\mathcal{C}_B \subseteq S$ and $\mathcal{C}_{B'} \subseteq S$. Then by claim 2 above, there is a chain from B to B' in $\mathcal{B}_S$ weakly connected by $\leq_{\mathcal{P}}$, so $\mathcal{B}_S$ is weakly $\leq_{\mathcal{P}}$-connected as required. To finish the proof, it remains to supply the proofs of claims 1 and 2 above:

1. Let v be a node of S. As S is strongly connected and non-trivial, there is a path of length > 0 from v back to v in S. Let h be the height of v in $\mathcal{D}$, and note that since $\mathcal{D}$ is a tree, every edge in $\mathcal{D}$ connects a node to one of strictly greater height. It follows that there is a path in $S \cap \mathcal{D}$ from v to a node v' of height $\geq h$ in $\mathcal{D}$, and an edge in $S \setminus \mathcal{D}$ from v' to a node of height $\leq h$ in $\mathcal{D}$. Since $S \subseteq \mathcal{G}_{\mathcal{P}}$, this edge is of the form $(v', \mathcal{R}(B))$ for some $B \in \mathrm{Bud}(\mathcal{D})$, and there is a path in S from $\mathcal{R}(B)$ to v. Now since $\mathcal{P}$ is in cycle normal form, $\mathcal{R}(B)$ is a strict ancestor of B in $\mathcal{D}$, and by construction of $\mathcal{G}_{\mathcal{P}}$, $\mathcal{R}(B)$ is therefore an ancestor of v' in $\mathcal{D}$. As $\mathcal{R}(B)$ has height $\leq h$ in $\mathcal{D}$, and v is an ancestor of v' in $\mathcal{D}$, it holds that $\mathcal{R}(B)$ is also an ancestor of v in $\mathcal{D}$. As there is a path in $S \cap \mathcal{D}$ from $\mathcal{R}(B)$ to v, and only edges in $\mathcal{D}$ connect nodes to nodes of greater height, the unique path in $\mathcal{D}$ from $\mathcal{R}(B)$ to v in $\mathcal{D}$ is in S.

   In summary, the unique paths in $\mathcal{D}$ from $\mathcal{R}(B)$ to v and from v to v' are in S, as is the edge $(v', \mathcal{R}(B))$ not in $\mathcal{D}$. In other words, we have a $B \in \mathrm{Bud}(\mathcal{D})$ such that v is on $\mathcal{C}_B$ and $\mathcal{C}_B \subseteq S$ as required.

2. Let $B, B' \in \mathrm{Bud}(\mathcal{D})$ satisfy $\mathcal{C}_B \subseteq S$ and $\mathcal{C}_{B'} \subseteq S$. Since S is assumed strongly connected, there is a path $\pi = v_0 v_1 v_2 \dots v_n$ in S with $v_0 = \mathcal{R}(B)$ and $v_n = \mathcal{R}(B')$. Now define a sequence $B_0 B_1 \dots B_{n+1}$ of bud nodes by $B_0 = B$, $B_{n+1} = B'$, and for each $i \in \{1, \dots, n\}$, let B_i be any bud such that $\mathcal{C}_{B_i} \subseteq S$ and (v_{i-1}, v_i) is an edge of $\mathcal{C}_{B_i}$. (Note that there is always such a B_i, because we have established in 1. above that every node of S is a vertex of some $\mathcal{C}_B \subseteq S$.) Note that the vertex v_i is thus a vertex of $\mathcal{C}_{B_i}$ for all $i \in \{0, \dots, n\}$. Also, since $(\mathcal{R}(B), v_1)$ is an edge of $\mathcal{C}_{B_1}$, we have that $\mathcal{R}(B)$ is a vertex of $\mathcal{C}_{B_1}$, so $B = B_0 \leq_{\mathcal{P}} B_1$. Similarly, since $(v_{n-1}, \mathcal{R}(B'))$ is an edge of $\mathcal{C}_{B_n}$, we have that $\mathcal{R}(B')$ is a vertex of $\mathcal{C}_{B_n}$, so $B' = B_{n+1} \leq_{\mathcal{P}} B_n$. It thus remains to show for each $i \in \{2, \dots, n\}$ that either $B_i \leq_{\mathcal{P}} B_{i-1}$ or vice versa, i.e. that $\mathcal{R}(B_i)$ is a vertex of $\mathcal{C}_{B_{i+1}}$ or $\mathcal{R}(B_{i+1})$ is a vertex of $\mathcal{C}_{B_i}$.

   Now, we have for any $i \in \{2, \dots, n\}$ that $\mathcal{C}_{B_i} \cup \mathcal{C}_{B_{i-1}} \subseteq S$ and (v_{i-1}, v_i) is an edge of $\mathcal{C}_{B_i}$. Note that $\mathcal{R}(B_i)$ is an ancestor of v_{i-1} in $\mathcal{D}$ since (v_{i-1}, v_i) is an edge of $\mathcal{C}_{B_i}$. Also, $\mathcal{R}(B_{i-1})$ is an ancestor of v_{i-1} in $\mathcal{D}$ since v_{i-1} is a vertex of $\mathcal{C}_{B_{i-1}}$. As $\mathcal{D}$ is a tree, either $\mathcal{R}(B_{i-1})$ is an ancestor of $\mathcal{R}(B_i)$ in $\mathcal{D}$ or vice versa. In the former case, $\mathcal{R}(B_i)$ is a vertex of $\mathcal{C}_{B_{i-1}}$, and in the latter case $\mathcal{R}(B_{i-1})$ is a vertex of $\mathcal{C}_{B_i}$, so we are done. This completes the proof.

□

Our aim is to formulate a localised notion of proof that is sufficient for the general CS^{ω} proof condition to hold. (Of course, it would be most desirable to find such a notion that is both sufficient *and* necessary, but it is not clear whether such a notion exists.) In other words, we need a condition that for any infinite path in the pre-proof graph yields an infinitely progressing

trace on some tail of that path. By Lemma 7.1.3, we can view a general infinite path in $\mathcal{G}_\mathcal{P}$ as an infinite path through a non-trivial strongly connected subgraph $S \subseteq \mathcal{G}_\mathcal{P}$ that visits all nodes of S infinitely often. Furthermore, by Lemma 7.1.7, S is the union of the basic cycles of a set of bud nodes weakly connected by the "structural connectivity" relation $\leq_\mathcal{P}$. This allows us to further analyse the composition of the infinite path through S:

Lemma 7.1.8. *Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a CS$^\omega$ pre-proof in cycle normal form, and let S be a non-trivial strongly connected subgraph of $\mathcal{G}_\mathcal{P}$. Note that $S = \bigcup_{B \in \mathcal{B}_S} \mathcal{C}_B$ by Lemma 7.1.7. Then any infinite path in S, starting from the companion of a bud in $\mathcal{B}_S$, is composed of paths (of length > 0) each of the form $\mathcal{R}(B_j) \ldots \mathcal{R}(B_k)$, where $B_j, B_k \in \mathcal{B}_S$ and $\mathcal{R}(B_j) \ldots \mathcal{R}(B_k)$ is a subpath of some cycle $\mathcal{C}_B \subseteq S$.*

Proof. By Lemma 7.1.7, $S = \bigcup_{B \in \mathcal{B}_S} \mathcal{C}_B$ where $\mathcal{B}_S \subseteq \mathrm{Bud}(\mathcal{D})$ is a weakly $\leq_\mathcal{P}$-connected set. It follows that any infinite path through S is composed of subpaths of the cycles $\{\mathcal{C}_B \mid B \in \mathcal{B}_S\}$. We just need to establish that we can confine our attention to subpaths of the form given above. To see this, consider an infinite path π through S starting from some companion node $\mathcal{R}(B_j)$, where $B_j \in \mathcal{B}_S$. Clearly we can write an initial finite segment of π as $\mathcal{R}(B_j) \ldots v, \mathcal{R}(B_k)$, where $\mathcal{R}(B_j) \ldots v$ is a path in $S \cap \mathcal{D}$ and $\mathcal{R}(B_k)$ is the next companion of a bud in $\mathcal{B}_S$ to occur on π after $\mathcal{R}(B_j)$. Now $(v, \mathcal{R}(B_k))$ is an edge of $\mathcal{C}_B$ for some $B \in \mathcal{B}_S$ (since $S = \bigcup_{B \in \mathcal{B}_S} \mathcal{C}_B$ and π is a path in S), and so $\mathcal{R}(B)$ is an ancestor of v in $\mathcal{D}$. As $\mathcal{D}$ is a tree and $\mathcal{R}(B_j)$ is also an ancestor of v in $\mathcal{D}$, and $\mathcal{R}(B)$ does not occur on the $\mathcal{D}$-path between $\mathcal{R}(B_j)$ and v by assumption, we have therefore that $\mathcal{R}(B)$ is an ancestor of $\mathcal{R}(B_j)$ in $\mathcal{D}$ and so $\mathcal{R}(B_j) \ldots v, \mathcal{R}(B_k)$ is a subpath of $\mathcal{C}_B$ as required. It is clear that we can now repeat the argument starting from the tail of π beginning with $\mathcal{R}(B_k)$, and so π can be decomposed into paths of the required type. □

Hence for a CS$^\omega$ pre-proof to satisfy the general proof condition it is both sufficient and necessary to find, for each strongly connected subgraph $S \subseteq \mathcal{G}_\mathcal{P}$, traces following all finite paths of the form described in Lemma 7.1.8, together with the stipulation that there are sufficiently many such traces to facilitate the composition of an infinitely progressing trace following any infinite path in S. Thus we may obtain conditions that are merely sufficient to satisfy the general proof condition by placing restrictions on the form and number of such traces.

7.2 Trace manifolds

In this section we develop two equivalent localised soundness conditions for CS$^\omega$ pre-proofs that exploit our results of the previous section. While more restrictive than the general soundness condition, these so-called *trace manifold* conditions are nevertheless combinatorially less

complex, and provide a greater degree of explicit information on the form of the implicit infinite descent arguments in pre-proofs.

Definition 7.2.1 (Trace manifold)**.** Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a CS$^\omega$ pre-proof in cycle normal form, and let $\mathrm{Bud}(\mathcal{D})$ be indexed by i. A *trace manifold* for $\mathcal{P}$ is a set of traces:

$$\{\tau_{S,i} \mid S = \bigcup_{B \in \mathcal{B}_S} \mathcal{C}_B \text{ where } \mathcal{B}_S \subseteq \mathrm{Bud}(\mathcal{D}) \text{ weakly } \leq_{\mathcal{P}}\text{-connected}, B_i \in \mathcal{B}_S\}$$

satisfying:

1. for all S and i, $\tau_{S,i}$ is a trace following the basic B_i-cycle $\mathcal{C}_{B_i}$ in $\mathcal{G}_{\mathcal{P}}$ and furthermore has the same value at both the instances of $\mathcal{R}(B_i)$ on $\mathcal{C}_{B_i}$. (Equivalently, $\tau_{S,i}$ is a trace following the unique path in $\mathcal{D}$ from $\mathcal{R}(B_i)$ to B_i and takes the same value at $\mathcal{R}(B_i)$ and B_i);

2. for all S, if $B_i, B_j \in \mathcal{B}_S$ and $B_j \leq_{\mathcal{P}} B_i$ then $\tau_{S,j}(\mathcal{R}(B_j)) = \tau_{S,i}(\mathcal{R}(B_j))$;

3. for every S, there is an i such that $\tau_{S,i}$ has at least one progress point.

The property of a pre-proof possessing a trace manifold is, in a sense, more manageable than the property of it possessing an infinitely progressing trace on some tail of every infinite path in its graph. Only a finite number of finite traces are required to specify a trace manifold, and for any given pre-proof one can algorithmically recover a trace manifold (if one exists). While the general proof condition can be checked using a (large) non-deterministic Büchi automaton (c.f. Proposition 5.1.10), it is reasonably easy to construct an ordinary non-deterministic finite automaton (on finite words) to decide whether a given pre-proof $\mathcal{P}$ has a trace manifold. First, computing the non-trivial strongly connected subgraphs of the pre-proof graph and the $\leq_{\mathcal{P}}$ relation on its buds is straightforward. Having done this, it is then obvious how to construct a deterministic finite automaton that checks whether a given set of traces satisfies the trace manifold condition. Then a non-deterministic finite automaton can check whether the pre-proof has a trace manifold by non-deterministically "guessing" a set of traces and then checking whether this set forms a trace manifold. We have not investigated whether such a construction yields a lower complexity bound than the general condition, but it would be interesting to do so.

For the purposes of illustration, we show how to construct a trace manifold for the CLKID$^\omega$ proof given in Example 5.2.3, whose soundness was justified informally in that example. (The actual CLKID$^\omega$ proof is also reproduced here for convenience.)

Example 7.2.2 ("The $P\&Q$ Example Revisited")**.** Let Φ_{NPQ} be the inductive definition set consisting of the usual productions for the "natural number" predicate N (c.f. Example 2.2.5) together with the following productions for the unary predicate P and binary predicate Q:

$$\frac{}{P0} \qquad \frac{Px \quad Q(x,sx)}{Psx} \qquad \frac{}{Q(x,0)} \qquad \frac{Q(x,y) \quad Px}{Q(x,sy)}$$

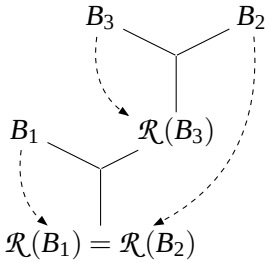
We wish to provide a CLKID$^\omega$ proof of the sequent $Nx, Ny \vdash Q(x,y)$, and begin as follows:

$$\dfrac{\dfrac{}{Nx \vdash Q(x,0)}\,(QR_1)}{Nx, y = 0 \vdash Q(x,y)}\,(=\!\mathrm{L}) \qquad \dfrac{\dfrac{\dfrac{Nx, Ny \vdash Q(x,y)\ (\dagger 1)}{Nx, Nz \vdash Q(x,z)}\,(\mathrm{Subst}) \qquad Nx \vdash Px\ (*)}{Nx, Nz \vdash Q(x,sz)}\,(QR_2)}{Nx, y = sz, Nz \vdash Q(x,y)}\,(=\!\mathrm{L})}{Nx, Ny \vdash Q(x,y)\ (\dagger)}\,(\text{Case } N)$$

We then continue on the rightmost branch of the proof $(*)$ as follows:

$$\dfrac{\dfrac{}{\vdash P0}\,(PR_1)}{x = 0 \vdash Px}\,(=\!\mathrm{L}) \qquad \dfrac{\dfrac{Nx \vdash Px\ (*)}{Nz \vdash Pz}\,(\mathrm{Subst}) \qquad \dfrac{Ny, Nx \vdash Q(x,y)\ (\dagger 2)}{Nsz, Nz \vdash Q(z,sz)}\,(\mathrm{Subst})}{Nsz, Nz \vdash Psz}\,(PR_2)}{Nx, x = sz, Nz \vdash Px}\,(=\!\mathrm{L})}{Nx, Nx \vdash Px}\,(\text{Case } N)$$
$$\dfrac{}{Nx \vdash Px\ (*)}\,(\mathrm{ContrL})$$

In the above, we shall write B_1 for the bud $(\dagger 1)$, B_2 for $(\dagger 2)$ and B_3 for $(*)$, so that the pre-proof may be schematically illustrated as follows (solid lines denote paths in the derivation tree and dashed arrows indicate the assignation of companions to buds):



Note that we have $B_2 \leq_{\mathcal{P}} B_1 \leq_{\mathcal{P}} B_1$ and $B_3 \leq_{\mathcal{P}} B_2$ (as well as $B_i \leq_{\mathcal{P}} B_i$ for $i \in \{1,2,3\}$), so the weakly $\leq_{\mathcal{P}}$-connected sets of buds in the pre-proof are the following:

$$\begin{array}{lcllcl} S_1 & = & \{B_1\} & S_4 & = & \{B_1, B_2\} \\ S_2 & = & \{B_2\} & S_5 & = & \{B_2, B_3\} \\ S_3 & = & \{B_3\} & S_6 & = & \{B_1, B_2, B_3\} \end{array}$$

To give a trace manifold we therefore require to give, for each S_i and each $B_j \in S_i$, a trace $\tau_{i,j}$ following the basic cycle $\mathcal{C}_{B_j}$ and taking the same value at both instances of $\mathcal{R}(B_j)$ on the cycle, as required in order to satisfy condition 1 in the definition of trace manifold above. Furthermore, the resulting set of traces must satisfy the further conditions 2 and 3 in the definition.

First, define $\tau_{1,1}$ to be the progressing trace (Ny, Nz, Nz, Ny) following $\mathcal{C}_{B_1}$ (i.e. the path in the pre-proof above from $(\dagger)$ to $(\dagger 1)$). Also define $\tau_{4,1}$ and $\tau_{6,1}$ to be the non-progressing trace (Nx, Nx, Nx, Nx) following $\mathcal{C}_{B_1}$. Next, for all i such that $B_2 \in S_i$, define $\tau_{i,2}$ to be the

progressing trace $(Nx, Nx, Nx, Nx, Nx, Nz, Nz, Nz, Nx)$ following $\mathcal{C}_{B_2}$. Finally, for all i such that $B_3 \in S_i$, define $\tau_{i,3}$ to be the progressing trace (Nx, Nx, Nz, Nz, Nz, Nx) following $\mathcal{C}_{B_3}$.

Note that each trace $\tau_{i,j}$ takes the same value at the start and end of the path it follows, as required by part 1 of the trace manifold condition. Also, it is easily seen that for each S_i there is a j such that $\tau_{i,j}$ is a progressing trace, as required by part 3 of the condition (e.g. $\tau_{1,1}$, $\tau_{2,2}$, $\tau_{3,3}$, $\tau_{4,2}$, $\tau_{5,2}$ and $\tau_{6,2}$ are all progressing traces).

Finally, we need to check that part 2 of the trace manifold condition holds, i.e. that if B_j, B_i in S_k for some k and $B_j \leq_{\mathcal{P}} B_i$ then $\tau_{k,j}(\mathcal{R}(B_j)) = \tau_{k,i}(\mathcal{R}(B_j))$. For $k \in \{1,2,3\}$ this is trivial. For the cases $k \in \{4,5,6\}$ it follows from the fact, firstly, that $\mathcal{R}(B_1) = \mathcal{R}(B_2)$ and $\tau_{i,1}(\mathcal{R}(B_1)) = \tau_{i,2}(\mathcal{R}(B_1)) = Nx$ for $i \in \{4,6\}$ and, secondly, that $\tau_{i,3}(\mathcal{R}(B_3)) = \tau_{i,2}(\mathcal{R}(B_3)) = Nx$ for $i \in \{5,6\}$. So indeed we have constructed a trace manifold as required.

We now establish formally that a trace manifold for a pre-proof does indeed contain sufficiently many trace segments to enable the construction of an infinitely progressing trace on every infinite path, as previously indicated.

Proposition 7.2.3. *Any CS^{ω} pre-proof (in cycle normal form) with a trace manifold is a CS^{ω} proof.*

Proof. Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be the pre-proof in question and consider any infinite path π through $\mathcal{G}_{\mathcal{P}}$. By Lemma 7.1.3, there is a tail π' of π and a non-trivial strongly connected subgraph $S \subseteq \mathcal{G}_{\mathcal{P}}$ such that π' is an infinite path through S that visits all vertices of S infinitely often. Furthermore, by Lemma 7.1.7, we have $S = \bigcup_{B \in \mathcal{B}_S} \mathcal{C}_B$ for some weakly $\leq_{\mathcal{P}}$-connected set $\mathcal{B}_S \subseteq \mathrm{Bud}(\mathcal{D})$. It suffices to construct an infinitely progressing trace on π', which we may assume without loss of generality to have the companion of some $B \in \mathcal{B}_S$ as its first vertex. By Lemma 7.1.8, π' is composed of finite paths in S of the form $\mathcal{R}(B_j) \ldots \mathcal{R}(B_k)$, where $B_j, B_k \in \mathcal{B}_S$ and $\mathcal{R}(B_j) \ldots \mathcal{R}(B_k)$ is a subpath of some basic cycle $\mathcal{C}_{B_i}$ where $B_i \in \mathcal{B}_S$. To each such path, we associate the portion of $\tau_{S,i}$ that follows $\mathcal{R}(B_j) \ldots \mathcal{R}(B_k)$.

To construct the required trace on π', we need to check that we can compose the traces associated with the component subpaths of π'. Suppose we compose two paths of the form above to obtain the path $\mathcal{R}(B_j) \ldots \mathcal{R}(B_k) \ldots \mathcal{R}(B_m)$, where $\mathcal{R}(B_j) \ldots \mathcal{R}(B_k)$ is a subpath of $\mathcal{C}_{B_1}$ and $\mathcal{R}(B_k) \ldots \mathcal{R}(B_m)$ is a subpath of $\mathcal{C}_{B_2}$. We have associated a portion τ_1 of $\tau_{S,1}$ to $\mathcal{R}(B_j) \ldots \mathcal{R}(B_k)$ and a portion τ_2 of $\tau_{S,2}$ to $\mathcal{R}(B_k) \ldots \mathcal{R}(B_m)$. To see that τ_1 and τ_2 compose, we note that $B_k \leq_{\mathcal{P}} B_1$ and $B_k \leq_{\mathcal{P}} B_2$ since $\mathcal{R}(B_k)$ occurs on both $\mathcal{C}_{B_1}$ and $\mathcal{C}_{B_2}$, and so by clause 2 in the definition of trace manifold we have $\tau_{S,1}(\mathcal{R}(B_k)) = \tau_{S,k}(\mathcal{R}(B_k)) = \tau_{S,2}(\mathcal{R}(B_k))$ as required.

The only remaining issue is to establish that the trace on π' obtained by composing the traces on the individual path segments has infinitely many progress points. We note that by clause 3 of the trace manifold definition, there is some B_i such that $\tau_{S,i}$ has a progress point on $\mathcal{C}_{B_i}$, and as every edge in S occurs infinitely often in π', we need only ensure that the relevant portion of $\tau_{S,i}$ is used infinitely often in our construction.

Suppose therefore that we are in the situation of assigning a trace portion to the path $\mathcal{R}(B_j)\dots\mathcal{R}(B_k)$, where $B_j, B_k \in \mathcal{B}_S$, and this path happens to be a subpath of both $\mathcal{C}_{B_1}$ and of $\mathcal{C}_{B_2}$ (where $B_1, B_2 \in \mathcal{B}_S$), so we are in the situation of assigning the relevant portion of either $\tau_{S,1}$ or $\tau_{S,2}$ to the path. Now observe that we have $B_j \leq_{\mathcal{P}} B_1$ and $B_j \leq_{\mathcal{P}} B_2$, so $\tau_{S,1}(\mathcal{R}(B_j)) = \tau_{S,j}(\mathcal{R}(B_j)) = \tau_{S,2}(\mathcal{R}(B_j))$. Similarly, $B_k \leq_{\mathcal{P}} B_1$ and $B_k \leq_{\mathcal{P}} B_2$, so $\tau_{S,1}(\mathcal{R}(B_k)) = \tau_{S,k}(\mathcal{R}(B_k)) = \tau_{S,2}(\mathcal{R}(B_k))$. Thus the considered portions of $\tau_{S,1}$ and $\tau_{S,2}$ both follow the path $\mathcal{R}(B_j)\dots\mathcal{R}(B_k)$ and agree at $\mathcal{R}(B_j)$ and $\mathcal{R}(B_k)$, so we may equally well assign either trace to the path. It is then clear that one can always assign a progressing segment to a path where there is a choice of segments, and since every edge of S occurs infinitely often on the path π', we can ensure that a progressing segment of $\tau_{S,i}$ is assigned to infinitely many path segment in π' as required. □

We remark that the existence of a trace manifold for a pre-proof is a stronger requirement than the general soundness condition. For example, it is easy to give a proof in CLKID$^\omega$ for which no trace manifold exists:

Example 7.2.4. Define the unary "natural number" predicate N by the usual productions given in Example 2.2.5 and define the binary predicate N' by the productions:

$$\frac{Ny}{N'(0,y)} \qquad \frac{N'(y,x)}{N'(sx,y)}$$

(The intended interpretation of $N'xy$ is "Nx and Ny", although the definition of N' above is not necessarily the most natural one.) The following is then a CLKID$^\omega$ proof of the sequent $Nx, Ny \vdash N'(x,y)$:

$$\dfrac{\dfrac{\dfrac{\dfrac{}{Ny \vdash Ny}\,(\mathrm{Ax})}{Ny \vdash N'(0,y)}\,(N'\mathrm{R}_1)}{Ny, x = 0 \vdash N'(x,y)}\,(=\mathrm{L}) \qquad \dfrac{\dfrac{\dfrac{Ny, Nx \vdash N'(x,y)\ \ (*)}{Nz, Ny \vdash N'(y,z)}\,(\mathrm{Subst})}{Nz, Ny \vdash N'(sz,y)}\,(N'\mathrm{R}_2)}{x = sz, Nz, Ny \vdash N'(x,y)}\,(=\mathrm{L})}{Nx, Ny \vdash N'(x,y)\ \ (*)}\,(\mathrm{Case}\ N)$$

We use $(*)$ to indicate the pairing of the bud in this proof with a suitable companion. To see that this is a CLKID$^\omega$ proof, notice that any infinite path in the pre-proof graph necessarily has a tail consisting of repetitions of the basic cycle $\mathcal{C}_{(*)}$ (i.e. the path in the proof from the companion to the bud). To construct the required infinitely progressing trace on this tail, we must compose copies of the progressing trace (Nx, Nz, Nz, Nz, Ny) on $\mathcal{C}_{(*)}$ alternately with copies of the (non-progressing) trace (Ny, Ny, Ny, Ny, Nx) on $\mathcal{C}_{(*)}$.

Notice that there does not exist a trace following $\mathcal{C}_{(*)}$ that takes the same value at the companion and the bud. If we take Nx as the initial trace value at the companion then the only possible trace to the bud is (Nx, Nz, Nz, Nz, Ny) and so the trace value at the bud is Ny, and similarly if we take Ny as the initial trace value then the only possible trace is (Ny, Ny, Ny, Ny, Nx)

and so the trace value at the bud is Nx. However, the proof above can be unfolded into a larger CLKID$^\omega$ proof that does have a trace manifold. It is unclear to us whether this can be done in general.

The quantification over strongly connected subgraphs in the definition of a trace manifold is somewhat heavy-handed. A slightly more elegant version of the trace manifold condition can be obtained by formulating the manifold with respect to a so-called *induction order* on the buds occurring in a pre-proof (a notion introduced by Schöpp [57] and subsequently employed by Sprenger and Dam in their translation from circular proofs to finite proofs by explicit transfinite induction for the μ-calculus [63, 62]):

Definition 7.2.5 (Induction order)**.** Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a CS$^\omega$ pre-proof in cycle normal form. A (non-strict) partial order $\lhd$ on $\mathrm{Bud}(\mathcal{D})$ is said to be an *induction order for* $\mathcal{P}$ if:

- $B \lhd B_1$ and $B \lhd B_2$ implies $B_1 = B_2$ or $B_1 \lhd B_2$ or $B_2 \lhd B_1$ (i.e. $\lhd$ is *forest-like*);
- every weakly $\leq_{\mathcal{P}}$-connected set $\mathcal{B} \subseteq \mathrm{Bud}$ has a $\lhd$-greatest element, i.e. an element $B_{max} \in \mathcal{B}$ such that $B \lhd B_{max}$ for all $B \in \mathcal{B}$. (Note that, in particular, we then have $B_1 \leq_{\mathcal{P}} B_2$ implies $B_1 \lhd B_2$ or $B_2 \lhd B_1$.)

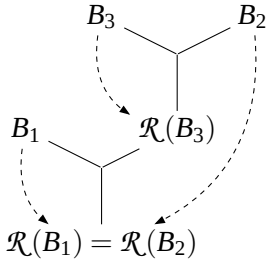
Definition 7.2.6 (Ordered trace manifold)**.** Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a pre-proof in cycle normal form and let $\lhd$ be an induction order for $\mathcal{P}$. An *ordered trace manifold* with respect to $\lhd$ is a set of traces:

$$\{\tau_{j,i} \mid B_j, B_i \in \mathrm{Bud}(\mathcal{D}), B_i \lhd B_j\}$$

satisfying all of the following:

1. for all i and j with $B_i \lhd B_j$, $\tau_{j,i}$ is a trace following the basic B_i-cycle $\mathcal{C}_{B_i}$ in $\mathcal{G}_{\mathcal{P}}$ and furthermore has the same value at both the instances of $\mathcal{R}(B_i)$ on $\mathcal{C}_{B_i}$. (Equivalently, $\tau_{j,i}$ is a trace following the unique path in $\mathcal{D}$ from $\mathcal{R}(B_i)$ to B_i and takes the same value at $\mathcal{R}(B_i)$ and B_i);
2. for all i, j, k, if $B_j, B_i \lhd B_k$ and $B_j \leq_{\mathcal{P}} B_i$ then $\tau_{k,j}(\mathcal{R}(B_j)) = \tau_{k,i}(\mathcal{R}(B_j))$;
3. for all i, $\tau_{i,i}$ has at least one progress point.

Example 7.2.7. Consider the CLKID$^\omega$ pre-proof given in Example 7.2.2, and recall that its cycle structure has the following form:



We give an induction order for the pre-proof. We define a partial order $\lhd$ on $\{B_1, B_2, B_3\}$ by $B_3 \lhd B_2$, $B_1 \lhd B_2$, and $B_i \lhd B_i$ for $i \in \{1,2,3\}$. One can easily check that $\lhd$ is a forest-like partial order. Furthermore, note that every weakly $\leq_{\mathcal{P}}$-connected subset of $\{B_1, B_2, B_3\}$ has a $\lhd$-greatest element. , so $\lhd$ is indeed an induction order.

We can now give an ordered trace manifold for the pre-proof with respect to $\lhd$. We require to specify, for each pair of buds B_i, B_j such that $B_i \lhd B_j$, a trace $\tau_{j,i}$ following the basic cycle $\mathcal{C}_{B_i}$ and taking the same value at both instances of $\mathcal{R}(B_i)$ on the cycle, as required in order to satisfy condition 1 in the definition of ordered trace manifold above. In other words, we need the following traces:

$$\tau_{1,1}, \tau_{2,1}, \tau_{2,2}, \tau_{2,3}, \tau_{3,3}$$

Furthermore, the resulting set of traces must satisfy the further conditions 2 and 3 in the definition.

First, define $\tau_{1,1}$ to be the progressing trace (Ny, Nz, Nz, Nz, Ny) following $\mathcal{C}_{B_1}$, and define $\tau_{2,1}$ to be the non-progressing trace (Nx, Nx, Nx, Nx, Nx) following $\mathcal{C}_{B_1}$. Next, define $\tau_{2,2}$ to be the progressing trace $(Nx, Nx, Nx, Nx, Nx, Nz, Nz, Nz, Nx)$ following $\mathcal{C}_{B_2}$. Finally, define both $\tau_{2,3}$ and $\tau_{3,3}$ to be the progressing trace (Nx, Nx, Nz, Nz, Nz, Nx) following $\mathcal{C}_{B_3}$.

Note that each trace $\tau_{i,j}$ takes the same value at the start and end of the path it follows, as required by part 1 of the ordered trace manifold condition. Also, as required by part 3 of the condition, $\tau_{i,i}$ is a progressing trace for each $i \in \{1,2,3\}$. Finally, we need to check that part 2 of the trace manifold condition holds, i.e. that if $B_j, B_i \lhd B_k$ and $B_j \leq_{\mathcal{P}} B_i$ then $\tau_{k,j}(\mathcal{R}(B_j)) = \tau_{k,i}(\mathcal{R}(B_j))$. This may easily be verified.

Lemma 7.2.8. *A pre-proof $\mathcal{P}$ (in cycle normal form) has a trace manifold if and only if there exists an induction order $\lhd$ for $\mathcal{P}$ and $\mathcal{P}$ has an ordered trace manifold with respect to $\lhd$.*

Proof. ($\Leftarrow$) Let $\lhd$ be an induction order for the pre-proof $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ and suppose $\mathcal{P}$ has an ordered trace manifold with respect to $\lhd$. Now let $S = \bigcup_{B \in \mathcal{B}_S} \mathcal{C}_B$ where $\mathcal{B}_S \subseteq \mathrm{Bud}(\mathcal{D})$ is weakly $\leq_{\mathcal{P}}$-connected. By the definition of induction order, $\mathcal{B}_S$ has a $\lhd$-greatest element, say B_k. We define $\tau_{S,i} = \tau_{k,i}$ for each i. We just need to check that this definition satisfies the properties of a (standard) trace manifold:

1. By clause 1 of the definition of ordered trace manifold, $\tau_{S,i} = \tau_{k,i}$ is a trace following the basic cycle $\mathcal{C}_{B_i}$, and takes the same value at both occurrences of $\mathcal{R}(B_i)$ on this cycle;

2. Suppose $B_j \leq_{\mathcal{P}} B_i$ where $B_j, B_i \in \mathcal{B}_S$. As B_k is the $\lhd$-greatest element of $\mathcal{B}_S$, we have $B_j, B_i \lhd B_k$ and thus $\tau_{k,j}(\mathcal{R}(B_j)) = \tau_{k,i}(\mathcal{R}(B_j))$ by clause 2 of the definition of order trace manifold, i.e. $\tau_{S,j}(\mathcal{R}(B_j)) = \tau_{S,i}(\mathcal{R}(B_j))$ as required;

3. By clause 3 of the definition of ordered trace manifold, the trace $\tau_{S,k} = \tau_{k,k}$ has at least one progress point as required.

($\Rightarrow$) Suppose $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ has a trace manifold. We show how to construct an induction order $\lhd$ for $\mathcal{P}$ and an ordered trace manifold with respect to $\lhd$.

Consider any maximal weakly $\leq_{\mathcal{P}}$-connected set $\mathcal{B} \subseteq \mathrm{Bud}(\mathcal{D})$ (so there is no weakly $\leq_{\mathcal{P}}$-connected subset of $\mathrm{Bud}(\mathcal{D})$ that is a strict superset of $\mathcal{B}$), and let $S = \bigcup_{B \in \mathcal{B}} \mathcal{C}_B$. By the definition of trace manifold, there is some $B_k \in \mathcal{B}$ such that $\tau_{S,k}$ progresses. (Of course, there may be more than one such bud, in which case we choose B_k to be any suitable one.) For each such maximal set $\mathcal{B}$, we define some suitable B_k to be the $\lhd$-greatest element of $\mathcal{B}$ (i.e. $B_i \lhd B_k$ for all $B_i \in \mathcal{B}$) and define $\tau_{k,i} = \tau_{S,i}$ for each $B_i \in \mathcal{B}$. Note that B_k is then also the $\lhd$-greatest element of any subset of $\mathcal{B}$ that contains B_k. We then consider all maximal weakly $\leq_{\mathcal{P}}$-connected subsets of $\mathrm{Bud}(\mathcal{D}) \setminus \{B_k\}$ and iterate the process of assigning $\lhd$-greatest elements to these sets until we obtain a $\lhd$-greatest element for every weakly $\leq_{\mathcal{P}}$-connected set. Furthermore, notice that in this way we never assign two different traces to $\tau_{k,i}$ (for any fixed i and k).

We check first that $\lhd$ as constructed is actually an induction order. It is obvious by construction that every weakly $\leq_{\mathcal{P}}$-connected set has a $\lhd$-greatest element, so we just need to check $\lhd$ is a forest-like partial order. $\lhd$ is reflexive by construction. To see $\lhd$ is transitive, observe that if $B_i \lhd B_j$ and $B_j \lhd B_k$, then there exist weakly $\leq_{\mathcal{P}}$-connected sets $\mathcal{B}_j, \mathcal{B}_k$ such that B_j, B_k are the $\lhd$-greatest elements of $\mathcal{B}_j, \mathcal{B}_k$ respectively and furthermore $\mathcal{B}_j \subseteq \mathcal{B}_k$. As $B_i \in \mathcal{B}_j$ (by construction since $B_i \lhd B_j$) we thus have $B_i \in \mathcal{B}_k$ and so $B_i \lhd B_k$ as required. For anti-symmetry, suppose $B_j \lhd B_k$ and $B_k \lhd B_j$. Then by construction there exist weakly $\leq_{\mathcal{P}}$-connected sets $\mathcal{B}_j, \mathcal{B}_k$ such that B_j, B_k are the (unique) $\lhd$-greatest elements of $\mathcal{B}_j, \mathcal{B}_k$ respectively, and we must have $\mathcal{B}_j \subseteq \mathcal{B}_k$ and $\mathcal{B}_k \subseteq \mathcal{B}_j$, so $\mathcal{B}_j = \mathcal{B}_k$ and $B_j = B_k$ as required. Lastly, we need to check $\lhd$ is forest-like. Suppose $B_i \lhd B_j$ and $B_i \lhd B_k$, so there exist weakly $\leq_{\mathcal{P}}$-connected sets $\mathcal{B}_j, \mathcal{B}_k$ such that B_j, B_k are the (unique) $\lhd$-greatest elements of $\mathcal{B}_j, \mathcal{B}_k$ respectively, and we have $B_i \in \mathcal{B}_j$ and $B_i \in \mathcal{B}_k$. By construction we must have either $\mathcal{B}_j \subseteq \mathcal{B}_k$ (in which case $B_j \lhd B_k$) or $\mathcal{B}_k \subseteq \mathcal{B}_j$ (in which case $B_k \lhd B_j$) or both (in which case $B_j = B_k$).

It remains to check that the construction yields an ordered trace manifold with respect to $\lhd$. We note that by construction, $\tau_{j,i}$ exists whenever $B_i \lhd B_j$, and check the required properties:

1. By clause 1 of the definition of trace manifold, $\tau_{j,i} = \tau_{S,i}$ (for some S) is a trace following the basic cycle $\mathcal{C}_{B_i}$ in $\mathcal{G}_{\mathcal{P}}$, and takes the same value at both the instances of $\mathcal{R}(B_i)$ on this cycle;

2. Suppose $B_j, B_i \lhd B_k$ and $B_j \leq_{\mathcal{P}} B_i$. Then there is some weakly $\leq_{\mathcal{P}}$-connected set $\mathcal{B}$ such that B_k is the $\lhd$-greatest element of $\mathcal{B}$ and $B_i, B_j \in \mathcal{B}$, so that $\tau_{S,j}(\mathcal{R}(B_j)) = \tau_{S,i}(\mathcal{R}(B_j))$, i.e. $\tau_{k,j}(\mathcal{R}(B_j)) = \tau_{k,i}(\mathcal{R}(B_j))$ as required;

3. By construction, $\tau_{k,k} = \tau_{S,k}$ is always a progressing trace.

This completes the proof. □

7.3 Translation of LKID into CLKID$^\omega$

We now return to the setting of our proof systems for $\mathrm{FOL}_{\mathrm{ID}}$. Recall that the proof system LKID then contains, for each inductive predicate in the language, a rule which embodies the principle of induction over the definition of the predicate in the inductive definition set. Proofs in LKID are the usual finite trees of sequents and thus may straightforwardly be seen as capturing the usual notion of proof by (explicit) induction. On the other hand, in the cyclic system CLKID$^\omega$, the induction rule for an inductive predicate is replaced by a weaker case distinction rule and proofs in the system may be viewed as regular infinite trees (as discussed in Chapter 5). The soundness condition imposed on CLKID$^\omega$ proofs (c.f. Definition 5.1.6) ensures that, if any sequent occurring on an infinite branch were false, an infinitely decreasing chain of ordinals (in fact, natural numbers) would exist, which is a contradiction. Proofs in CLKID$^\omega$ can thus be seen as embodying infinite descent arguments. (Of course, proofs in the general infinitary system LKID$^\omega$ likewise embody infinite descent arguments, but these proofs cannot be given a useful representation in general.)

The use of infinite descent in mathematical proofs (which are implicitly understood as finite) is regarded in some quarters as essentially interchangeable with the explicit use of induction, notwithstanding that one method or the other may be more suited to the structure of a particular problem. If this is true in the setting of $\mathrm{FOL}_{\mathrm{ID}}$ then one would expect that an arbitrary sequent is provable in LKID just in case it is provable in CLKID$^\omega$. Moreover, if the two methods are really interchangeable, then there ought to exist a procedure for transforming an arbitrary LKID proof of a sequent into a CLKID$^\omega$ proof of the same sequent and vice versa.

In this section we begin to address the problem of establishing the equivalence of induction and infinite descent for $\mathrm{FOL}_{\mathrm{ID}}$, by giving a translation from LKID into CLKID$^\omega$, and explaining why giving a translation from CLKID$^\omega$ into LKID appears hard. We leave the equivalence between LKID and CLKID$^\omega$ as the main open conjecture of this thesis.

In what follows, we shall consider a fixed first-order language Σ with inductive predicates $\{P_1,\ldots,P_n\}$, and a fixed inductive definition set Φ for Σ.

Lemma 7.3.1. *Any instance of the LKID induction rule (Ind P_j) for an inductive predicate P_j is derivable in CLKID$^\omega$.*

Proof. We recall the construction of the induction rule for an inductive predicate given in Section 3.1. Let $j \in \{1,\ldots,n\}$; we show how to derive an arbitrary instance of the induction rule (Ind P_j) in which the induction hypothesis F_i and the induction variables $\mathbf{z_i}$ have been associated to the inductive predicate P_i for each $i \in \{1,\ldots,n\}$:

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

Now define $\mathcal{M}$ to be the smallest set of formulas satisfying: for each minor premise of the considered instance of (Ind P_j), say:

$$\Gamma, Q_1\mathbf{u_1}(\mathbf{x}), \ldots, Q_h\mathbf{u_h}(\mathbf{x}), G_{j_1}\mathbf{t_1}(\mathbf{x}), \ldots, G_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash F_i\mathbf{t}(\mathbf{x}), \Delta$$

where $j_1, \ldots, j_m \in \{1, \ldots, n\}$ and $Q_1, \ldots, Q_h$ are ordinary predicates, the formula:

$$\forall \mathbf{x}.(Q_1\mathbf{u_1}(\mathbf{x}) \wedge \ldots \wedge Q_h\mathbf{u_h}(\mathbf{x}) \wedge G_{j_1}\mathbf{t_1}(\mathbf{x}) \wedge \ldots \wedge G_{j_m}\mathbf{t_m}(\mathbf{x}) \rightarrow F_i\mathbf{t}(\mathbf{x}))$$

is in $\mathcal{M}$. (Recall that the formula G_i is defined for each $i \in \{1, \ldots, n\}$ by:

$$G_i = \begin{cases} F_i & \text{if } P_i \text{ and } P_j \text{ are mutually dependent} \\ P_i\mathbf{z_i} & \text{otherwise} \end{cases}$$

and $G_i\mathbf{t}$ is written to mean $G_i[\mathbf{t}/\mathbf{z_i}]$). Now, consider the following derivation in CLKID$^\omega$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\mathcal{M}, P_j\mathbf{y} \vdash F_j\mathbf{y}}{\mathcal{M}, P_j\mathbf{t} \vdash F_j\mathbf{t}}\,(\text{Subst})}{\vdots}\,(\wedge\text{L})}{\bigwedge \mathcal{M}, P_j\mathbf{t} \vdash F_j\mathbf{t}}\,(\wedge\text{L})}{P_j\mathbf{t} \vdash \bigwedge \mathcal{M} \rightarrow F_j\mathbf{t}}\,(\rightarrow\text{R}) \qquad \dfrac{\dfrac{\text{minor premises}}{\vdots}}{\{\Gamma \vdash M, \Delta \mid M \in \mathcal{M}\}}\,(\wedge\text{R})}{\dfrac{\dfrac{\vdots}{\Gamma \vdash \bigwedge \mathcal{M}, \Delta}\,(\wedge\text{R}) \qquad \Gamma, F_j\mathbf{t} \vdash \Delta}{\Gamma, \bigwedge \mathcal{M} \rightarrow F_j\mathbf{t} \vdash \Delta}\,(\rightarrow\text{L})}\,(\text{Cut})$$

$$\Gamma, P_j\mathbf{t} \vdash \Delta$$

where $\mathbf{y}$ is a tuple of appropriately many variables. We remark that obtaining each of the minor premises of the considered instance of (Ind P_j) from the sequents $\{\Gamma \vdash M, \Delta \mid M \in \mathcal{M}\}$ is simply a matter of decomposing each M using the rules ($\forall$R),($\rightarrow$R) and ($\wedge$L). It thus suffices to provide a CLKID$^\omega$ proof of the sequent $\mathcal{M}, P\mathbf{y} \vdash F\mathbf{y}$, and we continue by showing how to do so.

First, we apply the CLKID$^\omega$ casesplit rule (Case P_j) to the sequent $\mathcal{M}, P_j\mathbf{y} \vdash F_j\mathbf{y}$, thus generating a case for each production which has P_j occurring in its conclusion. We show how to treat a case arising from an arbitrary production, say:

$$\frac{Q_1\mathbf{u_1}(\mathbf{x}) \ldots Q_h\mathbf{u_h}(\mathbf{x}) \quad P_{j_1}\mathbf{t_1}(\mathbf{x}) \ldots P_{j_m}\mathbf{t_m}(\mathbf{x})}{P_j\mathbf{t}(\mathbf{x})} \qquad j_1, \ldots, j_m, j \in \{1, \ldots, n\}$$

We abbreviate the set of formulas $Q_1\mathbf{u_1}(\mathbf{x}), \ldots, Q_h\mathbf{u_h}(\mathbf{x})$ by $\mathcal{Q}$. Now, as there is a minor premise $\Gamma, \mathcal{Q}, G_{j_1}\mathbf{t_1}(\mathbf{x}), \ldots, G_{j_m}\mathbf{t_m}(\mathbf{x}) \vdash F_j\mathbf{t}(\mathbf{x}), \Delta$ corresponding to this production in the considered instance of (Ind P_j), the formula $\forall \mathbf{x}(\bigwedge \mathcal{Q} \wedge G_{j_1}\mathbf{t_1}(\mathbf{x}) \wedge \ldots \wedge G_{j_m}\mathbf{t_m}(\mathbf{x}) \rightarrow F_j\mathbf{t}(\mathbf{x}))$ is in the set $\mathcal{M}$. This formula is the subject of the contraction in the following CLKID$^\omega$ derivation:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\{Q\vdash Q_i \mid Q_i\in Q\}}\,(\mathrm{Ax}) \quad \dfrac{\{\mathcal{M},P_{j_i}\mathbf{y}\vdash G_{j_i}\mathbf{y}\mid i\in\{1,\dots,m\}\}\ (\dagger 2)}{\{\mathcal{M},P_{j_i}\mathbf{t_i}(\mathbf{x})\vdash G_{j_i}\mathbf{t_i}(\mathbf{x})\mid i\in\{1,\dots,m\}\}}\,(\mathrm{Subst})}{}\,(\wedge\mathrm{R})}{\vdots}}{\dfrac{}{F_j\mathbf{t}(\mathbf{x})\vdash F_j\mathbf{t}(\mathbf{x})}\,(\mathrm{Ax}) \quad \dfrac{}{\mathcal{M},Q,P_{j_1}\mathbf{t_1}(\mathbf{x}),\dots,P_{j_m}\mathbf{t_m}(\mathbf{x})\vdash \bigwedge Q\wedge G_{j_1}\mathbf{t_1}(\mathbf{x})\wedge\ldots\wedge G_{j_m}\mathbf{t_m}(\mathbf{x})}\,(\wedge\mathrm{R})}\,(\to\mathrm{L})}{\mathcal{M},\bigwedge Q\wedge G_{j_1}\mathbf{t_1}(\mathbf{x})\wedge\ldots\wedge G_{j_m}\mathbf{t_m}(\mathbf{x})\to F_j\mathbf{t}(\mathbf{x}),Q,P_{j_1}\mathbf{t_1}(\mathbf{x}),\dots,P_{j_m}\mathbf{t_m}(\mathbf{x})\vdash F_j\mathbf{t}(\mathbf{x})}\,(\forall\mathrm{L})}{\vdots}}{\mathcal{M},\forall\mathbf{x}.(\bigwedge Q\wedge G_{j_1}\mathbf{t_1}(\mathbf{x})\wedge\ldots\wedge G_{j_m}\mathbf{t_m}(\mathbf{x})\to F_j\mathbf{t}(\mathbf{x})),Q,P_{j_1}\mathbf{t_1}(\mathbf{x}),\dots,P_{j_m}\mathbf{t_m}(\mathbf{x})\vdash F_j\mathbf{t}(\mathbf{x})}\,(\forall\mathrm{L})}{\mathcal{M},Q,P_{j_1}\mathbf{t_1}(\mathbf{x}),\dots,P_{j_m}\mathbf{t_m}(\mathbf{x})\vdash F_j\mathbf{t}(\mathbf{x})}\,(\mathrm{ContrL})}{\text{(other cases)}\ldots \quad \mathcal{M},\mathbf{y}=\mathbf{t}(\mathbf{x}),Q,P_{j_1}\mathbf{t_1}(\mathbf{x}),\dots,P_{j_m}\mathbf{t_m}(\mathbf{x})\vdash F_j\mathbf{y}}\,(=\mathrm{L})}{\mathcal{M},P_j\mathbf{y}\vdash F_j\mathbf{y}\ (\dagger 1)}\,(\text{Case } P_j)$$

where $x\notin FV(\mathcal{M}\cup\{P_j\mathbf{y}\}\cup\{F_j\mathbf{y}\})$ for all $x\in\mathbf{x}$. We have thus far obtained a CLKID$^\omega$ derivation with root sequent $\mathcal{M},P_j\mathbf{y}\vdash F_j\mathbf{y}$ $(\dagger 1)$ and bud nodes $\{\mathcal{M},P_{j_i}\mathbf{y}\vdash G_{j_i}\mathbf{y}\mid i\in\{1,\dots,m\}\}$ $(\dagger 2)$, and we observe that for each $i\in\{1,\dots,m\}$ there is a progressing trace:

$$(P_j\mathbf{y},P_{j_i}\mathbf{t_i}(\mathbf{x}),\dots,P_{j_i}\mathbf{t_i}(\mathbf{x}),P_{j_i}\mathbf{y})$$

following the path in this derivation from the root sequent $(\dagger 1)$ to the bud $\mathcal{M},P_{j_i}\mathbf{y}\vdash G_{j_i}\mathbf{y}$. Now note that, for each $i\in\{1,\dots,m\}$, if the predicates P_{j_i} and P_j are not mutually dependent, then $G_{j_i}=P_{j_i}$, and so we may apply the rule (Ax) to the bud node $\mathcal{M},P_{j_i}\mathbf{y}\vdash G_{j_i}\mathbf{y}$. Thus we need to consider only the bud nodes $\mathcal{M},P_{j_i}\mathbf{y}\vdash G_{j_i}\mathbf{y}$ such that P_{j_i} and P_j *are* mutually dependent, and are thus of the form $\mathcal{M},P_{j_i}\mathbf{y}\vdash F_{j_i}\mathbf{y}$. We treat these as follows:

- if $P_{j_i}=P_j$, then the bud node is identical to the root sequent $(\dagger 1)$, and we set the companion of the bud to be $(\dagger 1)$.
- if $P_{j_i}\neq P_j$, then note that as P_{j_i} and P_j are mutually dependent, there is a minor premise (and corresponding formula in $\mathcal{M}$) for every production which has P_{j_i} occurring in its conclusion. We thus can repeat the derivation above for the bud node under consideration to obtain new bud nodes $(\dagger 3)$, to which we may assign $(\dagger 1)$ or any ancestor node of the form $(\dagger 2)$ as a companion.

We iterate this process as often as required, successively generating bud nodes of the form $(\dagger 3),(\dagger 4),\ldots$, noting that any bud node of the form $(\dagger k)$ may potentially be assigned an ancestral companion of the form $(\dagger k')$ for any $k'<k$, and that bud nodes are always assigned ancestors as companions. This iteration is possible because $\mathcal{M}$ contains a formula corresponding to each production having in its conclusion a predicate that is mutually dependent with P_j and, since mutual dependency between predicates is transitive, the predicate P_{j_i} occurring on

the left of any bud node $(\dagger k)$ is always mutually dependent with P_j. Also, we observe that the iteration process never produces bud nodes of the form $(\dagger n+2)$, because there are at most n inductive predicates that are mutually dependent with P_j.

We thus obtain a CLKID$^\omega$ derivation tree $\mathcal{D}$ with root sequent $\mathcal{M}, P_j\mathbf{y} \vdash F_j\mathbf{y}$ and a repeat function $\mathcal{R}$ that assigns to every bud node of $\mathcal{D}$ an ancestor of the bud as companion, i.e. $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ is a CLKID$^\omega$ pre-proof in cycle normal form. Furthermore, for each bud node B_i in the tree, there is a trace τ_i following the unique path in $\mathcal{D}$ from $\mathcal{R}(B)$ to B that takes the same value at B and $\mathcal{R}(B)$.

We claim that there exists a trace manifold for $\mathcal{P}$. For any $S = \bigcup_{B\in\mathcal{B}_S} \mathcal{C}_B$ where $\mathcal{B}_S \subseteq \mathrm{Bud}(\mathcal{D})$ is a weakly $\leq_{\mathcal{P}}$-connected set, we define $\tau_{S,i} = \tau_i$ (so, in fact, $\tau_{S,i}$ is independent of S). Note that τ_i is a trace following the basic cycle $\mathcal{C}_{B_i}$ in $\mathcal{G}_{\mathcal{P}}$ taking the same value at both instances of $\mathcal{R}(B_i)$ on $\mathcal{C}_{B_i}$, as required by clause 1 of the definition of a trace manifold. Also notice that clause 3 of the definition is immediately satisfied since τ_i is a progressing trace. Finally, suppose $B_1 \leq_{\mathcal{P}} B_2$ where $B_1, B_2 \in \mathcal{B}_S$, i.e. $\mathcal{R}(B_1)$ appears on the cycle $\mathcal{C}_{B_2}$. To satisfy clause 2 of the trace manifold definition we need to show that $\tau_1(\mathcal{R}(B_1)) = \tau_2(\mathcal{R}(B_1))$. Note that by construction, $\mathcal{R}(B_1)$ is of the form $(\dagger k)$ (for some $k \in \{1,\ldots,n\}$) and thus is labelled with a sequent of the form $\mathcal{M}, P_{j_i}\mathbf{y} \vdash F_{j_i}\mathbf{y}$. It is then clear that any trace passing through $\mathcal{R}(B_1)$ takes the value $P_{j_i}\mathbf{y}$ at $\mathcal{R}(B_1)$, so in particular τ_2 takes this value as $\mathcal{R}(B_1)$ is a vertex of $\mathcal{C}_{B_2}$. This completes the proof. □

Theorem 7.3.2. *Every LKID proof of $\Gamma \vdash \Delta$ can be transformed into a CLKID$^\omega$ proof of $\Gamma \vdash \Delta$. Furthermore, this CLKID$^\omega$ proof has a trace manifold.*

Proof. Given any LKID proof $\mathcal{D}$ of $\Gamma \vdash \Delta$ we can obtain a CLKID$^\omega$ pre-proof $\mathcal{P}$ of $\Gamma \vdash \Delta$ by replacing every instance of an induction rule in $\mathcal{D}$ with the corresponding CLKID$^\omega$ derivation constructed in Lemma 7.3.1. Furthermore, by inspection it is clear that the set of strongly connected subgraphs of $\mathcal{P}$ is the (disjoint) union of the sets of strongly connected subgraphs of each inserted CLKID$^\omega$ derivation. The union of the trace manifolds constructed for each of the inserted derivations is thus a trace manifold for $\mathcal{P}$, whence $\mathcal{P}$ is a CLKID$^\omega$ proof by Proposition 7.2.3. □

Theorem 7.3.2 shows how to convert a proof by induction to a proof by infinite descent in the setting of $\mathrm{FOL}_{\mathrm{ID}}$. Essentially, as shown in Lemma 7.3.1, any use of induction over an inductive formula $P\mathbf{t}$ can be replaced by a cut on a formula which states that the combined minor premises imply the induction hypothesis $F\mathbf{t}$ associated with $P\mathbf{t}$, together with an infinite descent proof of this formula. Unfortunately, transforming a CLKID$^\omega$ proof into a LKID proof appears to be a much harder problem, and we state it here as our conjecture:

Conjecture 7.3.3. *If there is a CLKID$^\omega$ proof of a sequent $\Gamma \vdash \Delta$ then there is an LKID proof of $\Gamma \vdash \Delta$.*

Conjecture 7.3.3 is stated in a relatively weak form. As remarked earlier, if proof by induction and proof by infinite descent are really equivalent proof techniques for FOL_{ID} then there should exist a procedure for transforming an arbitrary CLKID^{ω} proof into a LKID proof of the same sequent. (A semantic proof of the conjecture would also be of interest, but we have no idea how to obtain one.) We are aware of two main related developments in the literature, the first being Sprenger and Dam's equivalence result for cyclic and non-cyclic proof systems for μ-calculus with explicit approximations [63], and the second being Walukiewicz' completeness result for the μ-calculus [82]. We examine these works in more detail in subsections 7.3.1 and 7.3.2 below, respectively.

We have observed that any LKID proof can be transformed into a CLKID^{ω} proof with a trace manifold, i.e. LKID provability $\Rightarrow$ trace manifold provability $\Rightarrow$ CLKID^{ω} provability. One approach to a proof of Conjecture 7.3.3 would thus be to establish that both the implications above hold in reverse, that is, to establish the two conjectures:

- if there is a CLKID^{ω} proof of a sequent, then there is a CLKID^{ω} proof of the same sequent with a trace manifold;
- if there is a CLKID^{ω} proof of a sequent with a trace manifold, then there is an LKID proof of the same sequent.

A proof of either of these statements would be of clear independent interest, but both appear hard. The second is a weaker version of Conjecture 7.3.3 and appears more approachable, as the existence of a trace manifold for a CLKID^{ω} proof gives much more structural information about the proof than the general soundness condition. A proof would presumably have some aspects in common with Sprenger and Dam's transformation from cyclic proofs in the μ-calculus to non-cyclic proofs by transfinite induction development (see below), but there are significant complications in our FOL_{ID} setting entailed by the complex form of our induction rules, our use of a trace-based soundness condition and our restriction to standard syntax. One could also notice that our proof of Theorem 7.3.2 produces CLKID^{ω} proofs with very constrained trace manifolds, and it may be easier to translate proofs with similarly constrained manifolds into LKID proofs. A proof of the first of the two statements above would establish that we can restrict our attention to CLKID^{ω} proofs with trace manifolds (at the possible expense of needing larger proofs), and could be seen as an analogue for traces of our cycle-normalisation results in Chapter 6. The main problem presented by this conjecture is that, as the infinitely progressing traces on any two infinite paths in the pre-proof graph can behave entirely differently despite potential overlap between the paths, it is not obvious that a manifold need exist. We speculate that it may be possible to use the proof machinery of Chapter 6 in conjunction with a combinatorial argument about trace values to unfold a proof into one with a trace manifold. However, this and other possible approaches to establishing the status of Conjecture 7.3.3 will for now

have to be left as the main avenue for potential future work stemming from this thesis.

7.3.1 The Sprenger-Dam translation from global to local proof in the μ-calculus

Sprenger and Dam established effective translations between a cyclic proof system for the μ-calculus and a non-cyclic system equipped with a local transfinite induction rule for the ordinals. The syntax of formulas ϕ in the μ-calculus with *explicit approximants* is given by the following definition:

$$\phi ::= t = u \mid \kappa' < \kappa \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \exists x.\phi \mid \exists\kappa.\phi \mid X \mid \mu X(\mathbf{x}).\phi \mid \mu^\kappa X(\mathbf{x}).\phi$$

where κ, κ' range over *ordinal variables* (i.e. variables whose interpretation ranges over the ordinals), t, u range over standard first-order terms, and X ranges over a set of predicate variables. The μ operator indicates least fixed points: the intended interpretation of the formula $\mu X(\mathbf{x}).\phi$ is (roughly) "$\mathbf{x}$ is in the least predicate X such that ϕ". and the least fixed points given by the μ operator can be approached by ordinal-indexed approximants in much the same way as the least fixed points of our monotone operators for inductive definitions (c.f. Definition 2.2.8), whence the operator μ^κ indicates the κth approximant of the least fixed point under consideration.

Sprenger and Dam define two systems S_{loc} and S_{glob} for the μ-calculus with explicit approximants. The systems are presented as Gentzen-style sequent calculi employing sequents of the form $\Gamma \vdash_O \Delta$, where Γ and Δ are finite multisets of formulas and O is a finite set of ordinal variables which acts as a kind of context for the system. The two systems share the same basic rules, which are essentially the usual structural and first-order rules from $\mathrm{LK_e}$ together with rules governing the behaviour of ordinal variables, and unfolding rules for (approximated) least fixed point formulas on the left and right of sequents. The rules for ordinal variables are as follows:

$$\frac{\Gamma, \phi \vdash_{O,\kappa} \Delta}{\Gamma, \exists\kappa.\phi \vdash_O \Delta}\ \kappa \notin O\ (\exists\kappa\,\mathrm{L}) \qquad \frac{\Gamma \vdash_O \phi[\iota/\kappa], \Delta}{\Gamma \vdash_O \exists\kappa.\phi, \Delta}\ \iota \in O\ (\exists\kappa\,\mathrm{R})$$

$$\frac{}{\Gamma, \kappa < \kappa \vdash_O \Delta}\ (<\mathrm{L}) \qquad \frac{\Gamma \vdash_O \kappa_1 < \kappa_2, \Delta \quad \Gamma \vdash_O \kappa_2 < \kappa_3, \Delta}{\Gamma \vdash_O \kappa_1 < \kappa_3, \Delta}\ (<\mathrm{R})$$

The rules for unfolding approximated μ-formulas are as follows, and can be seen as analogues of our right-unfolding and casesplit rules for inductive predicates:

$$\frac{\Gamma, \exists\kappa' < \kappa.\phi[\mu^{\kappa'} X(\mathbf{x}).\phi/X, \mathbf{t}/\mathbf{x}] \vdash_O \Delta}{\Gamma, (\mu^\kappa X(\mathbf{x}).\phi)(\mathbf{t}) \vdash_O \Delta}\ (\mu^\kappa\,\mathrm{L}) \qquad \frac{\Gamma \vdash_O \exists\kappa' < \kappa.\phi[\mu^{\kappa'} X(\mathbf{x}).\phi/X, \mathbf{t}/\mathbf{x}], \Delta}{\Gamma \vdash_O (\mu^\kappa X(\mathbf{x}).\phi)(\mathbf{t}), \Delta}\ (\mu^\kappa\,\mathrm{R})$$

A cyclic proof in the system S_{glob} is then a pre-proof, constructed analogously to a CLKID^ω pre-proof (but always in cycle normal form), satisfying an appropriate global proof condition ensuring soundness. Instead of directly employing a notion of trace as we do, Sprenger and

Dam consider an alternative condition involving the derivability of ordering statements on ordinal variables along each basic cycle in the pre-proof. Roughly speaking, an ordinal variable κ is said to be *preserved* by a path if it is present in the context of every sequent on the path and whenever the substitution rule (Subst) is applied along the path with substitution θ then the statement $\theta(\kappa) \leq \kappa$ is derivable, and is said to *progress* on a path if it is preserved by the path and there is a point on the path at which (Subst) is applied and the statement $\theta(\kappa) < \kappa$ is derivable. The proof condition then states that for every strongly connected subgraph of the pre-proof there is an ordinal variable that is preserved everywhere on the graph and progresses somewhere on the graph. This condition is then restated with respect to an induction order (c.f. Definition 7.2.5), and can be seen as somewhat similar to our ordered trace manifold condition.

Sprenger and Dam show that any proof in S_{glob} can be translated into a proof in the finitary proof system S_{loc} which adds the following transfinite induction rule to the rules of S_{glob}:

$$\frac{\Gamma, \phi \vdash_{O,\kappa} \exists \kappa' < \kappa.\phi[\kappa'/\kappa], \Delta}{\Gamma, \exists \kappa.\phi \vdash_{O} \Delta} \; \kappa \notin O \;\; \text{(Ind L)}$$

The analogue of our Theorem 7.3.2 is straightforward to prove; one shows that uses of (Ind L) in an S_{loc} derivation can be replaced by S_{glob} derivations and that the derivation so obtained is an S_{glob} proof. Translating from S_{glob} proofs to S_{loc} proofs is much harder. First of all, Sprenger and Dam show how to unfold a cyclic S_{glob} proof into one that is "tree-compatible" — i.e. one in which the induction order on buds matches their structural ordering given by the relation $\leq_{\mathcal{P}}$. They then demonstrate how to directly translate a tree-compatible S_{glob} proof into an S_{loc} proof. Essentially, for each basic cycle in the proof an instance of the induction rule (Ind L) is inserted at the companion node, using an internalisation of the sequent itself — guarded by a bounded existential quantifier on the progressing ordinal variable associated with the cycle — as the induction hypothesis. (The transformation is done bottom-up, so that induction hypotheses occurring further down the tree are included as part of induction hypotheses occurring higher up.) The fact that the induction variable progresses is then used to give a direct S_{loc} proof of the (modified) sequent occurring at the bud on the cycle. Crucially, given an induction hypothesis $H(\kappa)$ produced at some stage at the translation, and a derivable statement $\kappa' \leq \kappa$ one can produce the "regenerated" hypothesis $H(\kappa')$. This property, which relies on the explicit use of the $<$ relation on ordinal variables, is needed in order to successfully exploit the induction hypotheses in the S_{loc} proof.

Unfortunately, the problem of reducing cyclic reasoning on the ordinary syntax of first-order logic (CLKID$^{\omega}$) to induction over predicate definitions (LKID) seems significantly harder than the transformation of cyclic reasoning on a syntax extended by ordinal variables (S_{glob}) to the explicit use of ordinal transfinite induction (S_{loc}). This is partly because of our restriction to standard syntax and thus the unavailability of $<$ on ordinals (and thus the loss of the regen-

eration property outlined above), but is also due to the more complex forms of our casesplit and induction rules, and the complexity inherent in our proof condition, even when we restrict to CLKID$^\omega$ proofs having trace manifolds.

7.3.2 Connections with Walukiewicz' completeness for the μ-calculus

Figure 7.2 shows the main aspects of Walukiewicz' demonstration of the completeness of Kozen's axiomatisation for the propositional modal μ-calculus. The syntax for formulas of this calculus is given by the following definition:

$$\phi ::= \top \mid \bot \mid p \mid X \mid \neg\phi \mid \phi_1 \wedge \phi_2 \mid \phi_1 \vee \phi_2 \mid \langle a \rangle \phi \mid [a]\phi \mid \mu X.\phi(X)$$

where p ranges over a set of *propositional constants*, X ranges over a set of *propositional variables* and a ranges over a set of *actions*. Judgements in the system are of the form $\phi_1 \leq \phi_2$ with the intended meaning that the formulas $\phi_1 \wedge \phi_2$ and ϕ_1 are semantically equivalent. Kozen's axiomatisation [40] then adds the following rules concerning the least fixed point operator μ to the standard axiomatisation of propositional modal logic K:

$$\frac{}{\phi(\mu X.\phi(X)) \leq \mu X.\phi(X)} \qquad \frac{\phi(\psi) \leq \psi}{\mu X.\phi(X) \leq \psi}$$

The second rule above is often called *Park induction*. Niwinski and Walukiewicz first showed that a formula of the calculus is unsatisfiable iff it has a (regular) *refutation*, and that every formula with a refutation also has a regular refutation [52]. Then Walukiewicz showed that a formula is unprovable from Kozen's axiomatisation iff it has a refutation, from which the completeness of Kozen's axiomatisation immediately follows [82].

Our investigations into proof in FOL$_{\mathrm{ID}}$ can be seen as having aspects in common with the μ-calculus completeness proof, as we attempt to show in Figure 7.3. Firstly, the system LKID can be seen as an analogue for FOL$_{\mathrm{ID}}$ of Kozen's axiomatisation of the μ-calculus, which adds an induction rule and an axiom governing the μ operator to the axiomatisation of the modal logic K [40], as LKID adds rules for induction to the complete "axiomatisation" LK$_{\mathrm{e}}$ of first-order logic with equality. Of course, LKID is not complete with respect to the standard semantics of FOL$_{\mathrm{ID}}$, but it is (cut-free) complete with respect to the Henkin semantics. On the other hand, Niwinski and Walukiewicz' refutations are similar to proofs in the infinitary system LKID$^\omega$; they are trees, possibly containing infinite branches and satisfying an analogous trace condition, and both the system of refutations and LKID$^\omega$ are complete for the standard semantics of their respective languages. However, in contrast to the situation with μ-calculus refutations, not every LKID$^\omega$-provable sequent is also provable in its restriction to regular trees, CLKID$^\omega$. The problem of deciding whether LKID and CLKID$^\omega$ are equivalent thus seems somewhat reminiscent of the difficulties involved in Walukiewicz' proof that every μ-calculus formula

with a regular refutation is unprovable in Kozen's axiomatisation. However, it is far from clear whether his methods are applicable in our setting.

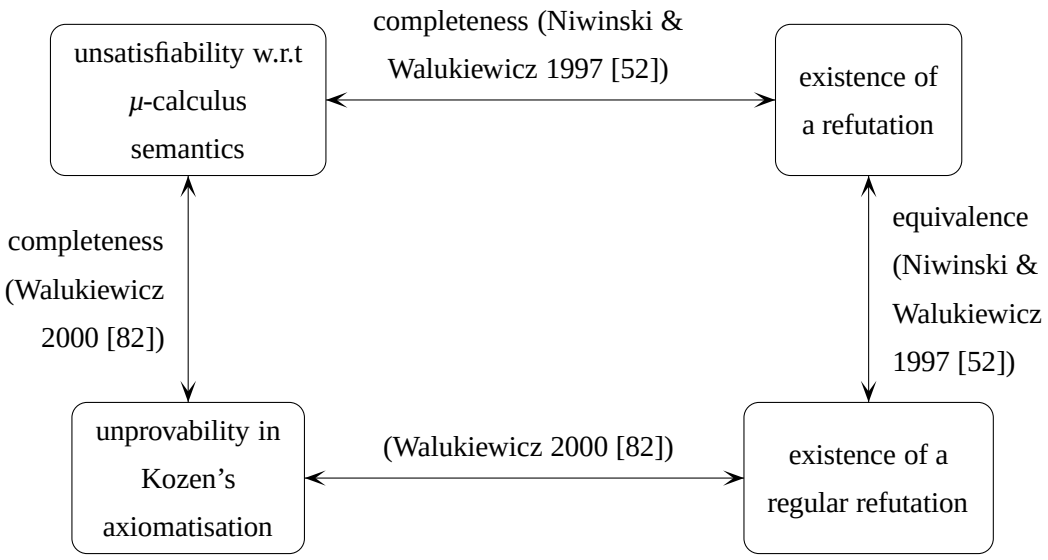


Figure 7.2: A diagrammatic representation of the work of Niwinski and Walukiewicz ultimately showing completeness of Kozen's axiomatisation of the propositional modal μ-calculus. All arrows represent implications.

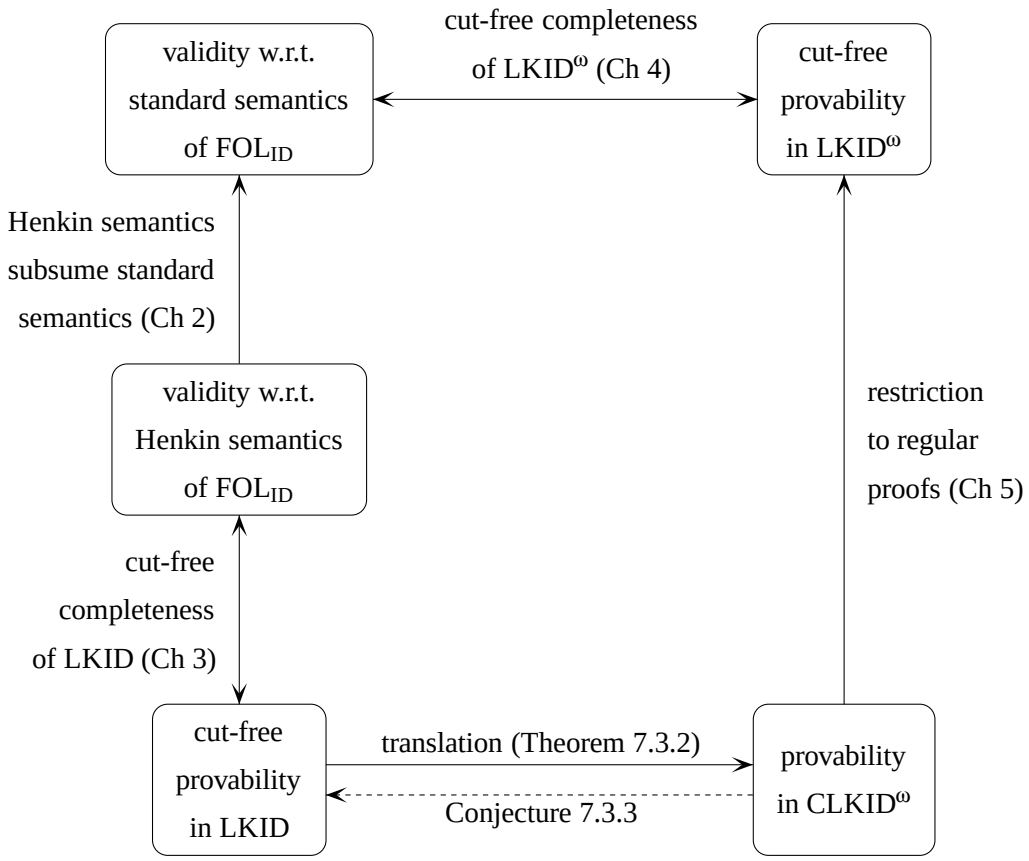


Figure 7.3: A diagrammatic representation of (some of) our developments in the setting of $\mathrm{FOL}_{\mathrm{ID}}$. The solid arrows represent implications, and the dashed arrow indicates the conjectured implication.

Chapter 8

Conclusions

To conclude this thesis, we first summarise our main contributions in Section 8.1, and then outline what appear to be the main promising directions for future research stemming from the thesis in Section 8.2.

8.1 Summary of our contributions

In this thesis we address (some of) the foundational issues concerning the well-known methods of proof by induction and proof by infinite descent in the setting of first order logic with ordinary (mutual) inductive definitions, $\mathrm{FOL}_{\mathrm{ID}}$.

Firstly, we give a classical sequent calculus LKID that supports proof by induction in $\mathrm{FOL}_{\mathrm{ID}}$, and for which we have established cut-free completeness with respect to a natural class of Henkin models. LKID can thus be viewed as being a canonical proof system for $\mathrm{FOL}_{\mathrm{ID}}$ in much the same way that the second-order functional calculus is a canonical system for second-order logic. As well as being of independent interest, the completeness result yields the eliminability of cut in LKID as a corollary. We believe this to be the first cut-elimination result for a classical system incorporating induction rules, although it seems plausible that this is because it previously appeared that such a result would be (a) relatively unsurprising in the light of the cut-elimination properties of related intuitionistic systems, and also (b) rather difficult to prove; as we have shown, there is no elementary proof of cut-eliminability in LKID, and a syntactic proof is likely to be very difficult. Our proof thus fills a fairly long-standing gap in the literature.

Secondly, we give an infinitary proof system LKID^{ω} that uses non-well-founded proofs to model proof by infinite descent in $\mathrm{FOL}_{\mathrm{ID}}$. The system uses only the standard syntax of first-order logic, in contrast to proof systems for cyclic or infinitary reasoning in other logics, in which the logical syntax is typically extended by, e.g., ordinal variables or ordering annotations. Soundness of the system is guaranteed by a trace condition that ensures that some inductive

definition is unfolded infinitely often along every branch. The infinite branches in a proof thus represent the cases of the proof which can be disregarded by the infinite descent principle. As for the system LKID, we establish cut-free completeness — this time with respect to standard models — and cut-eliminability for the system LKID^ω. Because the system is in a sense "too powerful", and so it is impossible to recursively enumerate a complete set of LKID^ω proofs, we consider the natural restriction CLKID^ω of the system to regular trees, in which it is decidable whether a given pre-proof satisfies the soundness condition. Unfortunately, it seems likely that cut is not eliminable in the restricted system. Nevertheless, CLKID^ω arises as a simple restriction of a well-behaved infinitary system, and is thus a highly natural system in its own right, providing a powerful framework for (regular) proof by infinite descent in FOL_{ID}.

Thirdly, we show that CLKID^ω subsumes LKID, giving rise to the important conjecture that the two systems are in fact equivalent. This conjecture can be seen as a formal positioning of the claim that proof by induction is equivalent to regular proof by infinite descent. Unfortunately, to provide a proof of the conjecture appears hard. We provide proof machinery for analysing the structure of proofs in CLKID^ω (and related systems) and the soundness condition imposed upon them, which has direct relevance for the machine implementation of such systems and which may also be of help in eventually establishing the status of the conjecture.

Finally, this thesis collects together a substantial amount of material on sequent calculus for inductive definitions, and we hope that it may serve as a useful reference for researchers working in inductive theorem proving or concerned with inductive definitions more generally.

8.2 Future work

This thesis opens up several potential avenues for future work. Perhaps the most obvious gap is to provide answers to the following presently open questions:

- Is every CLKID^ω-provable sequent also LKID-provable? (Conjecture 7.3.3 states that this is so.)
- Is every sequent that has a CLKID^ω proof with a trace manifold also LKID-provable?
- Does every CLKID^ω-provable sequent have a CLKID^ω proof with a trace manifold?
- Is cut eliminable in the system CLKID^ω? (Conjecture 5.2.4 states that it is not.)

It would also be interesting to generalise the proof systems, techniques and results of the thesis to include more general forms of inductive definition (e.g. iterated inductive definitions [44]), and coinductive definitions. In our opinion, the main technical issues arise seem to arise already with ordinary inductive definitions as we consider them here, so one would hope that extending the work to more general inductive schemas would not be too difficult.

One could also look at more liberal subsystems of LKID^ω that are nevertheless still useful for formal reasoning, e.g. proofs representable by a regular grammar or by some type of automata. It would be particularly interesting to find whether there is some class of such proofs that is closed under cut-elimination. Similarly, one could examine alternative proof conditions for CLKID^ω (and other general cyclic systems) and the relationship between them; finding conditions whose computational complexity is strictly better than the complexity of the general proof condition would be of special interest. Our trace manifold condition, or more restrictive variants thereof, may be suitable candidates for the latter.

A strand of proof theory for inductive definitions that we have not covered at all in this thesis is the ordinal analysis of systems of inductive definitions, as covered by e.g. Buchholz, Feferman, Pohlers and Sieg [11]. It is not clear to us what the connections are between this work and our own; for example, it is not obvious how to assign ordinals to LKID^ω proofs.

We have not yet said a great deal about inductive theorem proving. However, it would be interesting to see an actual implementation of CLKID^ω built in, or on top of an existing proof assistant. An extremely desirable feature of such an implementation would be a checking procedure that, given a pre-proof, either confirms that it is a proof or returns some finite representation of an infinite path in the pre-proof on which no infinitely progressing trace exists. Such an algorithm would provide much more feedback to users about the reasons for the failure of a particular proof attempt than a simple "yes/no" black box procedure. In the most ideal of cases, a positive answer to Conjecture 7.3.3 that actually gives a translation from CLKID^ω proofs to LKID proofs would result in the situation where meta-level proof search could be performed in some implementation of CLKID^ω and the resulting cyclic proof could then be transformed into a traditional (finitary) inductive proof at the object level. However, this appears to be a very challenging objective that would necessarily depend on a positive answer to the conjecture (or at least a partial answer, giving a translation into LKID for some class of cyclic proofs) as a first step. We have also given a general format for cyclic proof systems CS^ω, of which CLKID^ω is one instance, and it seems plausible that further variants could possibly yield useful formal systems for, e.g., program verification or model checking.

Appendix A

Decidability of proof in trace-based cyclic proof systems

In this appendix we give, in detail, a directly implementable construction based on Büchi automata for deciding whether a pre-proof is a proof in a cyclic proof system CS^{ω} equipped with an appropriate notion of trace (c.f. Chapter 5 and Section 4.2.1). We first recall the definition of a Büchi automaton from the literature. Note that we write Σ^{ω} for the set of all infinite words over the finite alphabet Σ.

Definition A.1 (Büchi automaton)**.** A (non-deterministic) *Büchi automaton* is a tuple $\mathcal{A} = (\Sigma, Q, q_0, \Delta, F)$, where:

- Σ is a finite alphabet;
- Q is a set of states;
- $q_0 \in Q$ is the distinguished initial state;
- $\Delta \subseteq Q \times \Sigma \times Q$ is the transition relation;
- $F \subseteq Q$ is a set of final (or accepting) states.

Given an infinite word $\alpha = \alpha_0\alpha_1\alpha_2 \ldots \in \Sigma^{\omega}$, a *run* of $\mathcal{A}$ on α is a sequence of states $\sigma = \sigma_0\sigma_1\sigma_2 \ldots$ such that $\sigma_0 = q_0$ and $\Delta(\sigma_i, \alpha_i, \sigma_{i+1})$ for all $i \geq 0$. Define the set of states occurring infinitely often in σ, $inf(\sigma)$, by:

$$inf(\sigma) = \{q \mid \exists \text{ infinitely many } i.\ \sigma_i = q\}$$

Then a run σ of $\mathcal{A}$ on α is said to be *accepting* if $inf(\sigma) \cap F \neq \emptyset$, i.e. if some final state occurs infinitely often on the run. The *language accepted by* $\mathcal{A}$, written $\mathcal{L}(A)$, is defined by:

$$\mathcal{L}(A) = \{\alpha \in \Sigma^{\omega} \mid \text{there is an accepting run of } \mathcal{A} \text{ on } \alpha\}$$

In their investigation into proof conditions for cyclic proof in the μ-calculus [62], Sprenger and Dam express the problem of deciding whether a pre-proof is a cyclic proof (with respect to a similar, albeit less general trace condition) as a problem of whether a language inclusion $\mathcal{L}(B_1) \subseteq \mathcal{L}(B_2)$ holds between two Büchi automata $\mathcal{B}_1$, $\mathcal{B}_2$. Such problems are known to be decidable [71]. In their construction, $\mathcal{B}_1$ accepts strings that correspond to rooted infinite paths in a (fixed) pre-proof graph and $\mathcal{B}_2$ accepts strings over which infinitely progressing traces can be found. Thus the inclusion $\mathcal{L}(\mathcal{B}_2) \subseteq \mathcal{L}(\mathcal{B}_2)$ holds exactly when the pre-proof under consideration is a proof. The standard method of checking such an inclusion is to build an automaton accepting the language $\mathcal{L}(\mathcal{B}_1) \cap \overline{\mathcal{L}(\mathcal{B}_2)}$ and check that the language of this automaton is empty (which can be done by e.g. depth-first search). However, this construction is rather complex due to the well-known difficulties involved in complementing Büchi automata [56, 71, 72, 41, 24].

Here we shall spell out the details of building a Büchi automaton from a CS^ω pre-proof $\mathcal{P}$ whose accepted language is empty if and only if $\mathcal{P}$ is a CS^ω proof. This gives rise to a directly mechanisable way of deciding whether pre-proofs are proofs in CS^ω.

Our approach is as follows. Given a CS^ω pre-proof $\mathcal{P}$, we first construct a Büchi automaton *Trace* such that $\mathcal{L}(\mathit{Trace})$ is the set of strings of vertices of $\mathcal{G}_\mathcal{P}$ such that an infinitely progressing trace can be found on a suffix of the string (irrespective of whether the string is actually a valid *path* in $\mathcal{G}_\mathcal{P}$ or not). Then, following the complementation method for Büchi automata given by Kupferman and Vardi [41], we build an automaton $\overline{\mathit{Trace}}$ whose accepted language is the complement of $\mathcal{L}(\mathit{Trace})$, i.e. the set of strings of vertices of $\mathcal{G}_\mathcal{P}$ such that no infinitely progressing trace exists on any suffix of the string. Finally, from $\overline{\mathit{Trace}}$ we build another automaton PrfDec accepting only those strings that are in $\mathcal{L}(\overline{\mathit{Trace}})$ and that also are valid paths in $\mathcal{G}_\mathcal{P}$. One can then easily see that $\mathcal{P}$ is a CS^ω proof if and only if $\mathcal{L}(\mathit{PrfDec}) = \emptyset$.

Definition A.2 (Trace automaton)**.** Let $\mathcal{P}$ be a CS^ω pre-proof and let V be the (finite) set of vertices of the pre-proof graph $\mathcal{G}_\mathcal{P}$. Also let *TVal* be the trace value relation for the system CS^ω as given in Definition 4.2.7. (Recall that $\mathit{TVal}(\tau, S)$ holds iff τ is a valid possible trace value for the sequent S.) Then the *trace automaton* corresponding to $\mathcal{P}$ is defined by $\mathit{Trace} = (V, Q, q_0, \Delta, F)$, where:

- $Q = \{q_0\} \cup \{(v, \tau, p) \mid v \in V, \mathit{TVal}(\tau, s(v)), p \in \{1, 2\}\}$;
- $F = \{(v, \tau, 2) \mid v \in V, \mathit{TVal}(\tau, s(v))\}$;
- the transition relation Δ is defined by:

$\Delta(q_0, v, q_0)$	
$\Delta(q_0, v, (v, \tau, 1))$	where $\mathit{TVal}(\tau, s(v))$
$\Delta((v, \tau, p), v', (v', \tau', 1))$	if (τ, τ') is a trace pair on (v, v')
$\Delta((v, \tau, p), v', (v', \tau', 2))$	if (τ, τ') is a progressing trace pair on (v, v')

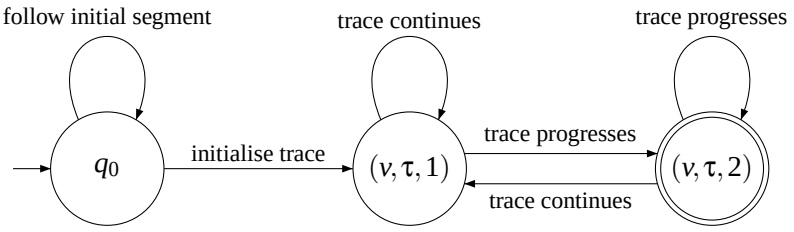


Figure A.1: Schematic representation of the trace automaton for a pre-proof. The accepting states are indicated by a double-circle. Note that the "meta-states" $(v,\tau,1)$ and $(v,\tau,2)$ correspond to multiple states in the automaton.

Note that the state set Q is finite since V is finite and there are only finitely many τ satisfying $TVal(\tau, s(v))$ for any $v \in V$. We also note that whether or not (τ,τ') is a valid (progressing) trace pair on (v,v') is decidable since the trace pair function for CS^ω is required to be computable. This property is not necessarily dependent on (v,v') actually being an edge of $\mathcal{G}_\mathcal{P}$, and so we do not insist that this is the case, although doing so would not significantly affect our construction.

Figure A.1 shows a schematic representation of the trace automaton for a pre-proof. Informally, it proceeds on an input $w \in V^\omega$ by non-deterministically "guessing" an initial segment of w on which it remains in the initial state, and then guessing an infinitely progressing trace following the remaining part of w. We make this description precise by the following claim:

Proposition A.3. *For any $w \in V^\omega$, Trace accepts w if and only if there is an infinitely progressing trace following some suffix of w (irrespective of whether w is a path in $\mathcal{G}_\mathcal{P}$).*

Proof. $(\Leftarrow)$ Suppose there is a suffix of w, say $v_i v_{i+1} v_{i+2} \ldots$, such that there is an infinitely progressing trace $\tau = \tau_i \tau_{i+1} \tau_{i+2} \ldots$ following the suffix. Now define a sequence $(\sigma_j)_{j \geq 0}$ by:

$$\sigma_j = \begin{cases} q_0 & \text{if } 0 \leq j \leq i-1 \\ (v_j, \tau_j, 1) & \text{if } j \geq i \text{ and } j \text{ is not a progress point of } \tau \\ (v_j, \tau_j, 2) & \text{if } j \geq i \text{ and } j \text{ is a progress point of } \tau \end{cases}$$

It is easy to see that $\sigma = \sigma_0 \sigma_1 \sigma_2 \ldots$ is a run of *Trace* on w, and moreover, as τ has infinitely many progress points, some final state (of the form $(v,\tau,2)$) must occur infinitely often in σ, i.e. σ is an accepting run of *Trace* on w.

$(\Rightarrow)$ Let σ be an accepting run of *Trace* on $w = v_0 v_1 v_2 \ldots$. By the definition of accepting run, some state of the form $(v,\tau,2)$ occurs infinitely often in σ. As the state q_0 is not reachable from

any such state, there is a suffix of σ, say $\sigma_i\sigma_{i+1}\sigma_{i+2}\ldots$, in which the state q_0 does not occur, so for all $j \geq i$, we have $\sigma_j = (v_j, \tau_j, s_j)$ (where $s_j \in \{1,2\}$). It is then easy to see that the sequence $\tau = \tau_i\tau_{i+1}\tau_{i+2}\ldots$ is a trace following the suffix $v_iv_{i+1}v_{i+2}\ldots$ of w. Moreover, as some state $(v,\tau,2)$ occurs infinitely often in σ, this trace progresses infinitely often. □

From now on, we shall work with reference to a fixed CS^ω pre-proof $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ with trace automaton $\mathit{Trace} = (V, Q, q_0, \Delta, F)$. Although the remaining definitions and results are stated with respect to this automaton, they in fact do not depend on any special property of Trace and apply equally to arbitrary Büchi automata. Our interest, however, lies specifically in constructing an automaton for deciding the property of being a cyclic proof and we shall proceed accordingly.

From Proposition A.3 we have:

$$\mathcal{L}(\mathit{Trace}) = \{w \in V^\omega \mid \exists \text{ infinitely progressing trace following some suffix of } w\}$$

We now show how to construct the complemented automaton $\overline{\mathit{Trace}}$ such that $\mathcal{L}(\overline{\mathit{Trace}}) = \overline{\mathcal{L}(\mathit{Trace})} (= V^\omega \setminus \mathcal{L}(\mathit{Trace}))$. The construction here follows the one given by Kupferman and Vardi [41] which, though fairly involved, offers a considerable simplification over previous complementation procedures (see e.g. [56, 39]). We first define, for a particular input w, a DAG that represents all possible runs of the automaton Trace on input w:

Definition A.4 (Run DAG)**.** For a fixed input $w \in V^\omega$, we define the *run DAG* of the automaton Trace on w by $G^w_{\mathit{run}} = (V_{\mathit{run}}, E_{\mathit{run}})$, where $V_{\mathit{run}} \subseteq Q \times \mathbb{N}$ and $E_{\mathit{run}} \subseteq V_{\mathit{run}} \times V_{\mathit{run}}$ are the smallest sets closed under the rules:

- $\langle q_0, 0\rangle \in V_{\mathit{run}}$;
- if $\langle q, i\rangle \in V_{\mathit{run}}$ and $\Delta(q, v_i, q')$, then $\langle q', i+1\rangle \in V_{\mathit{run}}$ and $E_{\mathit{run}}(\langle q, i\rangle, \langle q', i+1\rangle)$.

Note that G^w_{run} is infinite. We say that a vertex $\langle q, i\rangle$ of G^w_{run} is an *F-vertex* iff $q \in F$.

Proposition A.5. *For all $v \in V^\omega$, Trace accepts w if and only if there exists a path in G^w_{run} in which infinitely many F-vertices occur.*

Proof. Given an accepting run $\sigma = \sigma_0\sigma_1\sigma_2\ldots$ of Trace on w, it is easy to see that $(\langle \sigma_i, i\rangle)_{i\geq 0}$ is a path in G^w_{run}, and as some state in F occurs infinitely often in σ, there are infinitely many F-vertices on this path. Conversely, given a path $(\langle q_i, i\rangle)_{i\geq 0}$ in G^w_{run} in which infinitely many F-vertices occur, it is easy to see that $(q_i)_{i\geq 0}$ is an accepting run of Trace. □

We now introduce the concept of a *ranking* for the run DAG of Trace on w. A ranking is an assignment of values to vertices of G^w_{run} such that F-vertices are assigned only even values and the values along every path in G^w_{run} are monotonically decreasing. An *odd ranking* is a ranking in which every path eventually becomes "stuck" in an odd rank:

Definition A.6 (Ranking / Odd ranking)**.** A *ranking* for G^w_{run} is a function $f : V_{run} \to \{0,\ldots,2n\}$, where n is the number of vertices of the automaton $Trace$ (i.e. $n = |V|$), satisfying:

- if $f(\langle q,i\rangle)$ is odd then $q \notin F$;
- if $\langle q,i\rangle, \langle q',i+1\rangle \in E_{run}$ then $f(\langle q',i+1\rangle) \leq f(\langle q,i\rangle)$.

A ranking f is said to be an *odd ranking* iff for any path $\langle q_0,0\rangle\langle q_1,1\rangle\langle q_2,2\rangle\ldots$ in G^w_{run} there is a $j \geq 0$ such that $f(\langle q_j,j\rangle)$ is odd and $f(\langle q_k,k\rangle) = f(\langle q_j,j\rangle)$ for all $k \geq j$.

Lemma A.7. *The automaton Trace rejects input w if and only if there exists an odd ranking for the run DAG G^w_{run} of Trace on w.*

Proof. ($\Leftarrow$) Let f be an odd ranking for G^w_{run}. It is clear that there is no path in G^w_{run} in which infinitely many F-vertices occur, as F-vertices can be assigned only even ranks by the definition of ranking, and f assigns even ranks to only finitely many vertices on any path by the definition of odd ranking. *Trace* thus rejects *w* by Proposition A.5.

($\Rightarrow$) As *Trace* rejects *w*, there is no path in G^w_{run} in which infinitely many F-vertices occur, by Proposition A.5. In that case, one can construct an odd ranking from G^w_{run}. A full proof is given in [41]. □

As a consequence of Lemma A.7, an automaton that complements *Trace* can operate on input *w* by non-deterministically "guessing" an odd ranking for G^w_{run}. This can be accomplished in stages using the concept of *level rankings*:

Definition A.8 (Level ranking)**.** A *level ranking* for *Trace* is a function $g : Q \to \{0,\ldots,2n\} \cup \{\bot\}$ such that if $g(q)$ is odd then $q \notin F$. We denote the set of all level rankings for *Trace* by $\mathcal{LR}$.

Definition A.9 (Complement trace automaton)**.** The complement automaton of *Trace* is $\overline{Trace} = (V, \mathcal{LR} \times \mathrm{Pow}(Q), q'_0, \Delta', \mathcal{LR} \times \emptyset)$ where:

- $q'_0 = \langle g_0, \emptyset\rangle$, where g_0 is defined by $g_0(q_0) = 2n$ and $g_0(q) = \bot$ for all $q \neq q_0$;
- the transition relation Δ' is defined by: $\Delta'(\langle g,P\rangle, v, \langle g',P'\rangle)$ iff all of the following hold:
  - for all $q,q' \in Q$. $g(q) \geq 0$ and $\Delta(q,v,q')$ implies $g'(q') \leq g(q)$;
  - if $P \neq \emptyset$ then $P' = \{q' \in Q \mid \exists q \in P.\Delta(q,v,q')$ and $g'(q')$is even$\}$
  - if $P = \emptyset$ then $P' = \{q' \in Q \mid g'(q')$ is even$\}$

Lemma A.10. $\mathcal{L}(\overline{Trace}) = \overline{\mathcal{L}(Trace)} (= V^\omega \setminus \mathcal{L}(Trace))$.

Proof. We must show that for any $w = v_0v_1v_2\ldots \in V^\omega$, $\overline{Trace}$ accepts w iff $Trace$ rejects w. By Lemma A.7, it suffices to show that $\overline{Trace}$ accepts w if and only if there exists an odd ranking for the run DAG G^w_{run} of $Trace$ on w.

($\Leftarrow$) Let f be an odd ranking for G^w_{run}. We define a sequence $(\sigma_i)_{i\geq 0}$ by $\sigma_0 = q'_0$ and $\sigma_i = \langle g_i, P_i\rangle$ for $i > 0$, where g_i is a level ranking defined by:

$$g_i(q) = \begin{cases} f(\langle q,i\rangle) & \text{if } \langle q,i\rangle \in V_{run} \\ \bot & \text{otherwise} \end{cases}$$

and $P_i \subseteq Q$ is defined inductively on i by:

$$\begin{array}{rcl} P_0 & = & \emptyset \\ P_i \neq \emptyset & \Rightarrow & P_{i+1} = \{q' \in Q \mid \exists q \in P_i.\, \Delta(q,v,q') \text{ and } g_{i+1}(q') \text{is even}\} \\ P_i = \emptyset & \Rightarrow & P_{i+1} = \{q' \in Q \mid g_{i+1}(q') \text{ is even}\} \end{array}$$

We claim that $\sigma = \sigma_0\sigma_1\sigma_2\ldots$ is an accepting run of $\overline{Trace}$ on w. To see that σ is a run, we show that $\Delta'(\langle g_i,P_i\rangle, v_i, \langle g_{i+1},P_{i+1}\rangle)$ holds for all $i \geq 0$, which just involves showing that the three restrictions on Δ' given in Definition A.9 hold. The latter two restrictions obviously hold by definition of P_i. To see that the first restriction holds, assume $g_i(q) \geq 0$ and $\Delta(q,v_i,q')$. Now since $g_i(q) \geq 0$, either $g_i(q) = g_0(q_0) = 2n$ or $g_i(q) = f(\langle q,i\rangle)$. Further, since $\Delta(q,v_i,q')$ holds, $(\langle q,i\rangle, \langle q',i+1\rangle) \in E_{run}$ and so $f(\langle q',i+1\rangle)$ is defined and $g_{i+1}(q') = f(\langle q',i+1\rangle)$. Since f is a ranking, we thus have $0 \leq f(\langle q',i+1\rangle) \leq f(\langle q,i\rangle) \leq 2n$ as required.

σ is thus a run of $\overline{Trace}$ on w. To see that it is accepting, we require to prove that there are infinitely many i for which $P_i = \emptyset$. It suffices to show $\forall i.\exists j \geq i.\, P_j = \emptyset$. If $P_i = \emptyset$ we are of course done. So assume $P_i \neq \emptyset$ and assume for contradiction that $P_j \neq \emptyset$ for all $j > i$. But then, by definition of the P_j, there would be an infinite path $\langle q_i,i\rangle\langle q_{i+1},i+1\rangle\langle q_{i+2},i+2\rangle\ldots$ in G^w_{run} such that $g_{i+1}(q_{i+1}), g_{i+2}(q_{i+2}), \ldots$ all are even. By definition of the g_i, this would mean that $f(\langle q_{i+1},i+1\rangle), f(\langle q_{i+2},i+2\rangle)\ldots$ all are even, which is a contradiction as f is an odd ranking. So there must be some $j > i$ such that $P_j = \emptyset$, which establishes that σ is accepting.

($\Rightarrow$) Suppose $\overline{Trace}$ accepts w and let $\sigma = \sigma_0\sigma_1\sigma_2\ldots$ be an accepting run of $\overline{Trace}$ on w. We write $\langle g_i,P_i\rangle$ for σ_i and define a function f by $f(\langle q,i\rangle) = g_i(q)$. We require to prove that f is an odd ranking for G^w_{run}.

To establish that f is a ranking, we first prove by induction on i that $f(\langle q,i\rangle) \in \{0,\ldots,2n\}$, i.e. that $g_i(q) \geq 0$ for all $\langle q,i\rangle \in V_{run}$. For the base case, we just need to check that $g_0(q_0) \geq 0$ which is the case since $g_0(q_0) = 2n$ by definition. For the step case, let $\langle q',k+1\rangle \in V_{run}$, so there exists $\langle q,k\rangle \in V_{run}$ such that $(\langle q,k\rangle, \langle q',k+1\rangle) \in E_{run}$, i.e. $\Delta(q,v_k,q')$ holds. Now as σ is a run of $\overline{Trace}$ on w, there is a transition $\Delta'(\langle g_k,P_k\rangle, v_k, \langle g_{k+1},P_{k+1}\rangle)$, and as $g_k(q) \geq 0$ by induction hypothesis, we have by the first of the restrictions on Δ' in Definition A.9

that $0 \leq g_{k+1}(q') \leq g_k(q) \leq 2n$ as required. The step case argument also establishes that $(\langle q,i\rangle, \langle q',i+1\rangle) \in E_{run}$ implies $f(\langle q',i+1\rangle) \leq f(\langle q,i\rangle)$. So f is a ranking.

To see that f is an odd ranking, let $\langle q_0,0\rangle\langle q_1,1\rangle\langle q_2,2\rangle\ldots$ be a path in G^w_{run}. As f is a ranking we have $2n \geq f(\langle q_0,0\rangle) \geq f(\langle q_1,1\rangle) \geq f(\langle q_2,2\rangle)\ldots \geq 0$, so there must be a "limit" $j \geq 0$ such that for all $k \geq j$, $f(\langle q_k,k\rangle) = f(\langle q_j,j\rangle)$. Now suppose that $f(\langle q_j,j\rangle)$ is even, i.e. $g_j(q_j)$ is even. Then by definition of the P_i, we would have P_k nonempty for all $k \geq j$, which would contradict the fact that σ is accepting. Every path in G^w_{run} must therefore get stuck in an odd ranking, and we conclude that f is an odd ranking as required. This completes the proof. □

The final part of our construction is relatively straightforward and just involves a modification of the automaton $\overline{Trace}$ to accept only those strings of V^ω that are recognised by $\overline{Trace}$ and that also are paths in $\mathcal{G}_\mathcal{P}$:

Definition A.11 (Proof decision automaton)**.** The *proof decision automaton* for the pre-proof $\mathcal{P}$ is $PrfDec = (V, V \times (\mathcal{LR} \times \mathrm{Pow}(Q)), q''_0, \Delta'', V \times (\mathcal{LR} \times \emptyset))$, where:

- $q''_0 = (root(\mathcal{D}), q'_0)$;
- $\Delta''(\langle v, \langle g,P\rangle\rangle, v', \langle v', \langle g',P'\rangle\rangle)$ iff $\Delta'(\langle g,P\rangle, v', \langle g',P'\rangle)$ and (v,v') is an edge of $\mathcal{G}_\mathcal{P}$.

Proposition A.12. *Define* $\mathcal{L}_{Path} = \{w \in V^\omega \mid w \text{ is a rooted path in } \mathcal{G}_\mathcal{P}\}$. *Then* $\mathcal{L}(PrfDec) = \mathcal{L}_{Path} \cap \mathcal{L}(\overline{Trace})$.

Proof. We establish the reverse inclusion first. Let $w = v_0v_1v_2\ldots$ be a rooted path in $\mathcal{G}_\mathcal{P}$ that is accepted by $\overline{Trace}$ via the run $\sigma = \sigma_0\sigma_1\sigma_2\ldots$ Then it is clear that $\langle v_0,\sigma_0\rangle\langle v_1,\sigma_1\rangle\langle v_2,\sigma_2\rangle\ldots$ is an accepting run of $PrfDec$ on w.

Conversely, let $w = v_0v_1v_2\ldots$ be a string accepted by $PrfDec$ via the run $\delta = \delta_0\delta_1\delta_2\ldots$ and write $\langle v_i,\sigma_i\rangle$ for δ_i. It is clear that $\sigma = \sigma_0\sigma_1\sigma_2\ldots$ is then an accepting run of $\overline{Trace}$ on w. As $\delta_0 = q''_0$, we have $v_0 = root(\mathcal{D})$, and the restrictions placed on Δ'' ensure that for all $i \geq 0$, $(v_i, v_{i+1}$ is an edge of $\mathcal{G}_\mathcal{P}$, and w is thus also a rooted (infinite) path in $\mathcal{G}_\mathcal{P}$ as required. □

Theorem A.13. *The pre-proof* $\mathcal{P}$ *is a* CS^ω *proof if and only if* $\mathcal{L}(PrfDec) = \emptyset$. *Moreover, this property is decidable.*

Proof. We have:

$$\begin{array}{lcll}
\mathcal{P} \text{ is a } \mathrm{CS}^\omega \text{ proof} & \Leftrightarrow & \mathcal{L}_{Path} \subseteq \mathcal{L}(Trace) & \text{by Proposition A.3} \\
& \Leftrightarrow & \mathcal{L}_{Path} \cap \overline{\mathcal{L}(Trace)} = \emptyset & \\
& \Leftrightarrow & \mathcal{L}_{Path} \cap \mathcal{L}(\overline{Trace}) = \emptyset & \text{by Lemma A.10} \\
& \Leftrightarrow & \mathcal{L}(PrfDec) = \emptyset & \text{by Proposition A.12}
\end{array}$$

Checking whether or not the language of $PrfDec$ is empty amounts to searching for a final state q such that q is reachable from the initial state and q is (non-trivially) reachable from itself. If no such q exists, then $\mathcal{L}(PrfDec) = \emptyset$. This check is easily mechanisable and runs in time linear in the size of the transition function Δ'' of $PrfDec$. □

Bibliography

[1] Peter Aczel. An introduction to inductive definitions. In Jon Barwise, editor, *Handbook of Mathematical Logic*, pages 739–782. North-Holland, 1977.

[2] Mortimer J. Adler, William Benton, and Charles E. Swanson, editors. *Encyclopaedia Britannica*. Encyclopaedia Britannica Inc, 15 edition, 1985.

[3] Serge Autexier, Dieter Hutter, Heiko Mantel, and Axel Schairer. System description: Inka 5.0 - a logic voyager. In *Proceedings of CADE-16*, LNAI. Springer-Verlag, July 1999.

[4] Jürgen Avenhaus, Ulrich Kühler, Tobias Schmidt-Samoa, and Claus-Peter Wirth. How to prove inductive theorems? QuodLibet! In Franz Baader, editor, *Proceedings of CADE-19*, number 2741 in LNAI, pages 328–333. Springer, 2003.

[5] Franco Barbanera and Stefano Berardi. A symmetric lambda-calculus for classical program extraction. *Information and Computation*, 125(2):103–117, 1996.

[6] Jon Barwise. An introduction to first-order logic. In Jon Barwise, editor, *Handbook of Mathematical Logic*, pages 5–46. North-Holland, 1977.

[7] Robert S. Boyer and J.S. Moore. *A Computational Logic*. Academic Press, 1979.

[8] Julian Bradfield and Colin Stirling. Local model checking for infinite state spaces. *Theoretical Computer Science*, 96:157–174, 1992.

[9] James Brotherston. Cyclic proofs for first-order logic with inductive definitions. In *Automated Reasoning with Analytic Tableaux and Related Methods: Proceedings of TABLEAUX 2005*, Lecture Notes in Artificial Intelligence. Springer, 2005.

[10] James Brotherston and Alex Simpson. Classical sequent calculi for induction and infinite descent. Forthcoming, 2006.

[11] Wilfried Buchholz, Solomon Feferman, Wolfram Pohlers, and Wilfried Sieg. *Iterated Inductive Definitions and Subsystems of Analysis: Recent Proof-Theoretical Studies*, volume 897 of *Lecture Notes in Mathematics*. Springer-Verlag, 1981.

[12] Alan Bundy. The automation of proof by mathematical induction. In Alan Robinson and Andrei Voronkov, editors, *Handbook of Automated Reasoning*, volume I, chapter 13, pages 845–911. Elsevier Science, 2001.

[13] Alan Bundy, Andrew Stevens, Frank van Harmelen, Andrew Ireland, and Alan Smaill. Rippling: A heuristic for guiding inductive proofs. *Artificial Intelligence*, 62(2):185–253, 1993.

[14] Alan Bundy, F. van Harmelen, C. Horn, and Alan Smaill. The Oyster-Clam system. In M. E. Stickel, editor, *10th International Conference on Automated Deduction*, volume 449 of *LNAI*, pages 647–648. Springer, 1990.

[15] Samuel R. Buss. *Handbook of Proof Theory*. Elsevier Science Publishers B.V., 1998.

[16] A. Colmerauer. Prolog and infinite trees. *Logic Programming*, 16:231–251, 1982.

[17] Thierry Coquand. Infinite objects in type theory. In H. Barendregt and T. Nipkow, editors, *Types for Proofs and Programs*, pages 62–78. Springer, 1993.

[18] Bruno Courcelle. Fundamental properties of infinite trees. *Theoretical Computer Science*, 25, 1983.

[19] Mads Dam and Dilian Gurov. μ-calculus with explicit points and approximations. *Journal of Logic and Computation*, 12(2):255–269, April 2002.

[20] Augustus de Morgan. Induction (mathematics). In *Penny Cyclopaedia of the Society for the Diffusion of Useful Knowledge*, volume XII, pages 465–466. Charles Knight & Co., 1838.

[21] Lucas Dixon. *A Proof Planning Framework for Isabelle*. PhD thesis, University of Edinburgh, 2005.

[22] Lucas Dixon and Jacques Fleuriot. Higher order rippling in isaplanner. In *Theorem Proving in Higher Order Logics '04*, volume 3223 of *LNCS*. Springer, 2004.

[23] Solomon Feferman et al., editor. *Kurt Gödel: Collected Works, Vol. II: Publications 1937–1944*. Oxford University Press, 1990.

[24] Ehud Friedgut, Orna Kupferman, and Moshe Y. Vardi. Büchi complementation made tighter. In *Proceedings of the 2nd International Symposium on Automated Technology for Verification and Analysis*, 2004.

[25] Gerhard Gentzen. Untersuchungen über das Logische Schliessen. *Mathematische Zeitschrift*, 39:176–210 and 405–431, 1935. Reproduced with English translation in [66], 68–131.

[26] Gerhard Gentzen. Die Widerspruchfreiheit der reinen Zahlentheorie. *Mathematische Annalen*, 112:493–565, 1936. Reproduced with English translation in [66].

[27] Eduardo Giménez. *A Calculus of Infinite Constructions and its application to the verification of communicating systems*. PhD thesis, Ecole Normale Supérieure de Lyon, 1996.

[28] Jean-Yves Girard. Une extension de l'interprétation de Gödel à l'analyse et son application à l'élimination des coupures dans l'analyse et la théorie des types. In *Proceedings of the Second Scandinavian Logic Symposium*, volume 63 of *Studies in Logic and the Foundations of Mathematics*, pages 63–92. North-Holland, 1971.

[29] Jean-Yves Girard. *Proof Theory and Logical Complexity*, volume 1. Bibliopolis, 1987.

[30] Kurt Gödel. Über eine bisher noch nicht benützte Erweiterung des finiten standpunktes. *Dialectica*, 12:280–287, 1958. Reproduced with English translation in [23],240–251.

[31] Jeremy Gow. *The Dynamic Creation of Induction Rules Using Proof Planning*. PhD thesis, University of Edinburgh, 2004.

[32] Geoff Hamilton. Poítin: Distilling theorems from conjectures. In *Proceedings of the 12th Symposium on the Integration of Symbolic Computation and Mechanized Reasoning*, July 2005.

[33] Leon Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15:81–91, 1950.

[34] W.A. Howard. Assignment of ordinals to terms for primitive recursive functionals of finite type. In A. King, J. Myhill, and R.E. Vesley, editors, *Intuitionism and Proof Theory*, pages 443–458. North-Holland, 1970.

[35] H.R. Jervell. A normal form for first-order arithmetic. In J.E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 93–108. North-Holland, 1971.

[36] Stig Kanger. A simplified proof method for elementary logic. In P. Braffort and D. Hirschberg, editors, *Computer Programming and Formal Systems*, pages 87–94. North-Holland, 1963.

[37] Matt Kaufmann, Panagiotis Manolios, and J Strother Moore. *Computer-Aided Reasoning: An Approach*. Kluwer Academic Publishers, June 2000.

[38] Richard Kaye. *Models of Peano Arithmetic*. Oxford University Press, 1991.

[39] Nils Klarlund. Progress measures for complementation of ω-automata with applications to temporal logic. In *Proceedings of the 32nd IEEE Symposium on Foundations of Computer Science*, pages 358–367, San Juan, 1991.

[40] D. Kozen. Results on the propositional mu-calculus. *Theoretical Computer Science*, 27:333–354, 1983.

[41] Orna Kupferman and Moshe Y. Vardi. Weak alternating automata are not that weak. *ACM Trans. on Computational Logic*, pages 408–429, July 2001.

[42] Chin Soon Lee, Neil D. Jones, and Amir M. Ben-Amram. The size-change principle for program termination. *ACM SIGPLAN Notices*, 36(3):81–92, 2001.

[43] Michael Sean Mahoney. *The Mathematical Career of Pierre de Fermat 1601–1665*. Princeton University Press, 2 edition, 1994.

[44] Per Martin-Löf. Haupstatz for the intuitionistic theory of iterated inductive definitions. In J.E. Fenstad, editor, *Proceedings of the Second Scandinavian Logic Symposium*, pages 179–216. North-Holland, 1971.

[45] Per Martin-Löf. *Intuitionistic type theory : Notes by Giovanni Sambin of a series of lectures given in Padua, June 1980*. Studies in proof theory. Bibliopolis, 1984.

[46] Laurent Mauborgne. An incremental unique representation for regular trees. *Nordic Journal of Computing*, 7(4):290–311, 2000.

[47] Raymond McDowell. *Reasoning In A Logic With Definitions and Induction*. PhD thesis, University of Pennsylvania, 1997.

[48] Raymond McDowell and Dale Miller. Cut-elimination for a logic with definitions and induction. *Theoretical Computer Science*, 232:91–119, 2000.

[49] Alberto Momigliano and Alwen Tiu. Induction and co-induction in sequent calculus. In *Proceedings of TYPES 2003*, volume 3085 of *LNCS*, pages 293 – 308. Springer-Verlag, 2003.

[50] Yiannis N. Moschovakis. *Elementary Induction on Abstract Structures*, volume 77 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 1974.

[51] Sara Negri and Jan von Plato. *Structural Proof Theory*. Cambridge University Press, 2001.

[52] Damian Niwiński and Igor Walukiewicz. Games for the μ-calculus. *Theoretical Computer Science*, 163:99–116, 1997.

[53] Michel Parigot. Proofs of strong normalisation for second order classical natural deduction. *Journal of Symbolic Logic*, 62(4):1461–1479, December 1997.

[54] Martin Protzen. Lazy generation of inductive hypotheses. In *CADE-12*, volume 814 of *LNAI*, pages 42–56. Springer, 1994.

[55] H. Rogers. *Theory of recursive functions and effective computability*. McGraw-Hill, New York, 1967.

[56] Shmuel Safra. On the complexity of ω-automata. In *Proceedings of the 29th IEEE Symposium on Foundations of Computer Science*, pages 319–327, White Plains, 1988.

[57] Ulrich Schöpp. Formal verification of processes. Master's thesis, University of Edinburgh, 2001.

[58] Ulrich Schöpp and Alex Simpson. Verifying temporal properties using explicit approximants: Completeness for context-free processes. In *Foundations of Software Science and Computation Structure: Proceedings of FoSSaCS 2002*, volume 2303 of *Lecture Notes in Computer Science*, pages 372–386. Springer-Verlag, 2002.

[59] Peter Schroeder-Heister. Cut-elimination in logics with definitional reflection. In D. Pearce and H. Wansing, editors, *Nonclassical Logics and Information Processing*, volume 619 of *LNAI*, pages 146–171. Springer, 1992.

[60] Peter Schroeder-Heister. Definitional reflection and the completion. In R. Dyckhoff, editor, *Extensions of Logic Programming*, volume 798 of *LNAI*, pages 333–347. Springer, 1994.

[61] Carsten Schürmann. *Automating the Meta-Theory of Deductive Systems*. PhD thesis, Carnegie-Mellon University, 2000.

[62] Christoph Sprenger and Mads Dam. A note on global induction mechanisms in a μ-calculus with explicit approximations. *Theoretical Informatics and Applications*, July 2003. Full version of FICS '02 paper.

[63] Christoph Sprenger and Mads Dam. On the structure of inductive reasoning: circular and tree-shaped proofs in the μ-calculus. In *Proceedings of FOSSACS 2003*, volume 2620 of *Lecture Notes in Computer Science*, pages 425–440, 2003.

[64] A. Stevens. A rational reconstruction of Boyer & Moore's technique for constructing induction formulas. In Y. Kodratoff, editor, *Proceedings of ECAI-88*, pages 565–570, 1988.

[65] C. Stirling and D. Walker. Local model checking in the modal μ-calculus. *Theoretical Computer Science*, 89:161–177, 1991.

[66] M.E. Szabo, editor. *The Collected Papers of Gerhard Gentzen*. North-Holland, 1969.

[67] W.W. Tait. A nonconstructive proof of Gentzen's Hauptsatz for second order predicate logic. *Bulletin of the American Mathematical Society*, 72:980–983, 1966.

[68] W.W. Tait. Intentional interpretations of functionals of finite type. *Journal of Symbolic Logic*, 32:198–212, 1967.

[69] Gaisi Takeuti. On a generalized logic calculus. *Japanese Journal of Mathematics*, 23:39–96, 1953. Errata: ibid, vol. 24 (1954), 149–156.

[70] Gaisi Takeuti. *Proof Theory*. Studies in Logic and the Foundations of Mathematics. North-Holland, 2nd edition, 1987.

[71] Wolfgang Thomas. Automata on infinite objects. In J. van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B: Formal Models and Semantics, pages 133–192. Elsevier Science Publishers, 1990.

[72] Wolfgang Thomas. Complementation of Büchi automata revisited. In J. Karhumäki, H.A. Maurer, G. Paun, and G. Rozenberg, editors, *Jewels are Forever, Contributions on Theoretical Computer Science in Honor of Arto Salomaa*, pages 109–120. Springer, 1999.

[73] Alwen Tiu. *A Logical Framework For Reasoning About Logical Specifications*. PhD thesis, Pennsylvania State University, 2004.

[74] Alwen Tiu and Dale Miller. A proof search specification of the π-calculus. In *Proceedings of the 3rd workshop in Foundations of Global Ubiquitous Computing 2004*, volume 138 of *ENTCS*, pages 79–101. Elsevier, 2005.

[75] A.S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. Cambridge University Press, 2 edition, 2000.

[76] Franklyn Turbak and J.B. Wells. Cycle therapy: A prescription for fold and unfold on regular trees. In *Proceedings of the 3rd International Conference on Principles and Practice of Declarative Programming*, pages 137–149, 2001.

[77] Valentin Turchin. The concept of a supercompiler. *ACM Transactions on Programming Languages and Systems*, 8:90–121, 1986.

[78] Christian Urban and Gavin Bierman. Strong normalisation of cut-elimination in classical logic. *Fundamenta Informaticae*, 45:123–165, January 2001.

[79] M.Y. Vardi and P. Wolper. An automata-theoretic approach to automatic program verification. *Logic in Computer Science, LICS '86*, pages 322–331, 1986.

[80] S. Wainer and L. Wallen. Basic proof theory. In Peter Aczel, Harold Simmons, and Stanley Wainer, editors, *Proof Theory*, pages 1–26. Cambridge University Press, 1992.

[81] Christoph Walther. Computing induction axioms. In Andrei Voronkov, editor, *Proceedings of LPAR-1992*, volume 624 of *LNAI*, pages 381–392. Springer-Verlag, 1992.

[82] Igor Walukiewicz. Completeness of Kozen's axiomatisation of the propositional μ-calculus. *Information and Computation*, 157:142–182, 2000.

[83] H. Wang. Towards mechanical mathematics. *IBM Journal of Research and Development*, 4:2–22, 1960.

[84] Andreas Weiermann. How is it that infinitary methods can be applied to finitary mathematics? Gödel's T: a case study. *Journal of Symbolic Logic*, 63:1348–1370, 1998.

[85] Claus-Peter Wirth. Descente infinie + deduction. *Logic Journal of the IGPL*, 12(1):1–96, 2004.

[86] Claus-Peter Wirth. Progress in computer-assisted inductive theorem proving by human-orientedness and descente infinie? Technical Report SWP-2006-01, Universität des Saarlandes, 2006. SEKI working paper.