Computational Content of Fixed Points

Submitted in partial fulfillment of the requirements
for Licentiate Degree in

LOGIC

Giacomo Barlucchi

Under the guidance of

Bahareh Afshari

Graham E. Leigh

Examiner
Fredrik Engström
University of Gothenburg

Opponent
Sebastian Enqvist
Stockholm University

Department of Philosophy, Linguistics and Theory of Science

20 December, 2022

UNIVERSITY OF GOTHENBURG

Sammanfattning

Vi studerar det beräkningsmässiga innehållet hos fixpunkter i förhållande till två logiska system med olika egenskaper. Gemensamt för båda studierna är en metod för att hantera fixpunkternas iterativa karaktär. I den första delen riktas intresset mot ett cykliskt system ICA för intuitionistisk aritmetik. Den formella definitionen av systemet följs av införandet av en typad λY-kalkyl, vars termer representerar den deduktiva processen för cykliska bevis. Här ges en metod för att producera rekursionsscheman från instanser av cykliska bevis. Resultatet är en grammatik vars språk består av λ-termer, som fångar det beräkningsmässiga innehållet som finns i det ursprungliga beviset. I den andra delen tittar vi på iteration av fixpunkter i termer av tillslutningsordinaltal för formler i den modala μ-kalkylen. Här presenteras en metod för att bestämma en övre gräns för tillslutningsordinaltal och den tillämpas på formler i fragment av Σ_1-klassen, där resultaten ligger i linje med redan existerande arbeten. De viktigaste verktygen för att fastställa en övre gräns är kommenterade strukturer, för att spåra hur modelländringar påverkar ordinaltalen, och en pumpteknik för dessa strukturer.

Abstract

We study the computational content of fixed points in relation to two logical systems with distinct characteristics. Common to both research strands is a method for dealing with the iterative nature of fixed points. In the first part the interest is directed to a cyclic system ICA for intuitionistic arithmetic. The formal definition of the system is followed by the introduction of typed λY-calculus, whose terms represent the deductive process of cyclic proofs. A method for producing recursion schemes from instances of cyclic proofs is given. The result is a grammar whose language consists of λ-terms, capturing the computational content implicit in the initial proof. In the second part, we look at the iteration of fixed points in terms of closure ordinals of formulas in the modal μ-calculus. A method for determining an upper bound on closure ordinals is presented and applied to formulas in fragments of the Σ_1 class, with results that are in line with the already existing works. Annotated structures, to track how model changes affect the ordinals, and a pumping technique for these structures are the main tools used to establish an upper bound.

Acknowledgements

I would like to thank Bahareh, my supervisor, for her guidance and support in these two years. I am also grateful for the positivity that she brings in every situation. Thanks to Graham, my second supervisor, for his constant help and, most importantly, his patience. Thanks to Tjeerd, fellow traveler in this adventure, and to Mattias and Dominik for motivating me. I owe my thanks to the Logic group at Gothenburg University, for accepting me so kindly into their company and creating an inspiring environment. Thanks to the Department of Philosophy, Linguistics, Theory of Science for the opportunity to work in this field. I wouldn't even be here if it wasn't for Chiara, so my final thanks to her.

Contents

Introduction

In the present work we investigate the computational content of fixed points. The fixed points of a function f are defined in general as those elements x such that $f(x) = x$. Intrinsic in their nature is a process of iteration given by the infinite series of substitutions $x = f(x) = f(f(x)) = \dots$. Among the many possible fixed points of a function, particular importance is attributed to the least and greatest of them, that share an intimate bond with induction and coinduction. It is an established fact that a monotone function f on a complete lattice has both a least and a greatest fixed point,[1] and that both can be obtained by iteration starting from the bottom or top elements respectively, through a series of self-applications. The possibility to define operators that pick out least and greatest fixed points, that is, to compute the fixed point of a given function by iteration, enriches the expressibility and deductive power of a logical system, hence it has been studied and implemented in a variety of contexts. In the present work, we focus on two specific systems, in which the process of iteration is studied from different perspectives.

In the first part we investigate the role of fixed points in cyclic proof systems, a relatively new formal method of proof that originates from Fermat's proof by infinite descent. The core of its success is the possibility to reduce an infinite deduction to a compact finite object. While the generality of the claim is preserved in the finite representation by the prospect of an infinite iteration of the argument, at the same time the soundness of the deduction is guaranteed by relying on an external well-founded structure, preventing an actual infinite regress. Not surprisingly, cyclic proofs systems have been implemented to include induction structurally, with interesting results for example with Peano and Heyting arithmetic. In this work we focus on one of these cyclic systems for intuitionistic arithmetic. To be able to talk about the process of computation that is expressed by an intuitionistic cyclic proof, an appropriate formally defined object is necessary. Typed λ-calculus is known to be apt for the role, thanks to the Curry–Howard correspondence. In that language the iteration process is achieved through the introduction of a combinator Y that defines fixed points. Terms of λY-calculus can then be used to describe the program expressed by a cyclic proof, the regular trees generated by the term corresponding to the derivation trees. An equivalent way of expressing terms of the λY-calculus is through higher-order recursion schemes, that is a series of term rewriting operations on a

[1] Thanks to the Knaster–Tarski's theorem.

typed language that reflects the computational steps of the reduction of a λ-term. As a consequence, the grammar so obtained generates λ-terms, capturing the computational content implicit in the initial proof.

In the second part, the process of fixed point iteration is studied in the context of modal μ-calculus. It is possible to compute the denotation of the least and greatest fixed points of a modal μ-formula in a Kripke structure with a transfinite series of applications of the formula to itself, starting from the empty set or from the whole domain, for the least or greatest fixed point respectively. Considering positive (i.e. monotone) formulas, there necessarily exists an ordinal that corresponds to the least number of iterations in which the fixed point is reached on a given model. The notion of closure ordinals is motivated by a generalisation of this property over all possible models. For a given formula, or even classes of formulas, it is possible that after a definite number of iterations the fixed point is always obtained. Not every formula has a closure ordinal, and there is no known uniform method for deciding whether a single formula possesses one. Despite this limitation, it is known that an upper bound can be given for some fragments of the calculus, meaning that either a closure ordinal exists below a certain threshold, or some formula does not have a closure ordinal. The value of a closure ordinal depends on both the structure of the formula and the form of its model. That is not surprising if we consider that the structure of the formula determines the possibilities for satisfaction, i.e., the conditions for a progress. The models, on the other hand, realise (or not) the conditions, ultimately determining the ordinal. Any small modification in the structure or valuation is capable of having a great effect in terms of the computation of fixed points. In order to keep a control over both these factors, we start with a simple class of formulas with no nesting of fixed points, and focus on developing a toolbox to perform modifications on models. The goal consists in the definition and test of these tools, with the prospective of a future extension to more complex formulas.

In summary, the two parts of this thesis examine the computational content of fixed points under two different lights. In the first part, the work proceeds from the existence of a cyclic proof to arrive at a formalism that computes fixed point iterations at a syntactic level. In the second part, the fixed point operators are already defined in the language, and the perspective is directed towards the way in which their iterative process is reflected in the semantics, that is, we focus on the fixed points of a formula in a model.

Part I

Cyclic Proofs

Introduction

In this first chapter we explore the computational content of cyclic proofs. In recent years the interest and research around cyclic proof systems has grown significantly. The possibility to represent potentially infinite proofs as finite objects has evident advantages in all those situations where induction and recursion are involved. The core idea behind cyclic proofs is to give a formal structure to the reasoning process that goes under the name of *proof by infinite descent*. Also known as Fermat's method of descent, the argument was already known by ancient Greeks[2] and follows a common-sense way of reasoning. The canonical example is the proof of the irrationality of $\sqrt{2}$, where from the assumed existence of a rational a/b equal to $\sqrt{2}$, the existence of another rational c/d is deduceed, with $c < a$ and $d < b$. Since an infinite decreasing list of natural number is impossible, the non-existence of the initial a/b is established. The key components of a proof by infinite descent are a deductive argument that returns to some previous step, opening the door to its potentially infinite repetition; and a condition that *guards* such a door, ensuring that the argument is indeed potentially infinite, and hence generally applicable, but never such when adopted in concrete instances.

In cyclic proof theory the two components correspond to the existence of a cyclic deduction, i.e., returning to a point in the argument already seen in the proof, and a well-ordered structure, on which a regression in every potentially infinite derivation guarantees soundness. It was only in the 2000s that the idea of defining proof systems to exploit the power of this argument was pursued. Among the first proof systems that explicitly use infinite descent were Santocanale [San02], and Dam and Gurov [DG02] and Sprenger and Dam [SD03] in the context of modal μ-calculus, where tableau proof systems were introduced with proof search in mind. In his doctoral thesis [Bro06],[3] and later with Simpson [BS11], Brotherston developed a theoretical framework for cyclic sequent calculi with inductive predicates, starting from the work of Martin-Löf on systems of natural deductions with inductive predicates [Mar71]. Simpson in [Sim17] defined a cyclic proof system equivalent to Peano Arithmetic. Berardi and Tatsuta in the same year presented an intuitionistic cyclic system CLJID^{ω} and proved its equivalence with Martin-Löf's inductive system LJID, under the assumption that Heyting Arithmetic is added to both

[2] The first known appearance is reported by [Wir04] to be in the works of Hippasus of Metaponto, V century b.C.

[3] The system was given already in [Bro05].

(see [BT17c]). Cyclic proof systems in the style of [Bro06] are not equivalent to inductive ones in general, as showed by Berardi and Tatsuta [BT17a], but they become such in the presence of arithmetic (see [BT17b]). In the next chapter an intuitionistic cyclic sequent calculus ICA for arithmetic is defined, which is a version of $\mathrm{CLJID}^{\omega} + \mathrm{HA}$ adjusted to the necessities of the present work.

A correspondence exists between proofs in sequent calculi and terms of typed λ-calculus, a relation that goes under the name of *Curry–Howard correspondence*. The relationship can be seen in different aspects: *proofs* can be transformed into terms of λ-calculus, that is *programs* or functions; *formulas*, on the other hand, can be seen as the *types* of the λ-terms, in a sense describing their behaviour and talking about their meaning. λ-terms are one possible way of actualising the notion of witnesses for proofs, that is the core of the Brower–Heyting–Kolmogorov (BHK) interpretation of constructive provability. According to their proposed interpretation, the meaning of a proof in an intuitionistic framework is an object that is able to be a witness of the argument, a realisation of the process of proving the desired statement. A proof of an implication, for example, is an object that, working as a function, transforms any witness of the antecedent into an argument for the consequent. Inside the philosophical standpoint on the nature of proofs, there is room for different ways of converting the vague notion of object-witness into a formally defined entity. One of the first attempts has been Kleene's realisability, that is the choice to use a coding to assign numbers to proofs, and interpret deductions as the application of numerical functions to the obtained witnesses (see [Tro98]). A closer approach to Curry–Howard correspondence in the context of intuitionistic arithmetic is Gödel's system T (see [AF98]). The process of deduction is translated into functionals, i.e., typed combinators: combinatory completeness and the addition of recursion make the representation of intuitionistic arithmetic possible. A cyclic version of Gödel's T has been recently given by Das in [Das20], together with an analysis of the complexity of the circular system CT with respect to the standard system T.

A well-defined representation of a proof carries in itself information about its content, and the formalism adopted can influence the kind of data displayed by a given witness. In the context of classical logic, an information of great interest is a set of terms t_i that satisfies Herbrand's theorem. It is a famous result from Herbrand that every prenex formula $\exists x_0 \dots x_n.\varphi$ of first order logic is equivalent to the disjunction of a series of quantifier-free instances $\varphi(t_0) \vee \dots \vee \varphi(t_n)$ called Herbrand disjunction. A direct way of computing the set of terms $t_0, \dots, t_n$ consists in passing through cut-elimination, a costly procedure that might produce distinct cut-free proofs, with different possible sets of terms. Attention has been devoted to alternative ways of producing such a set. Gerhardy and Kohlenbach [GK03] gave a first description of a method to extract Herbrand disjunction using Gödel's functional interpretation. The first step of their method consists in the production of a witness, a functional realiser of the proof, followed by the reduction to its β-normal form from which they extract the desired terms.

The idea of exploiting recursion schemes to extract computational content from proofs is presented in details in [Het12] and [AHL15], where the connection between grammars (recursion schemes) and proofs is motivated in the context of term extraction. Building on these ideas, Afshari, Hetzl and Leigh in [AHL20] proposed a method to extract the Herbrand set from a proof in classical one-sided sequent calculus using higher-order recursion schemes (HORS). From each proof a recursion scheme can be defined, where one production rule corresponds to each specific deductive step of the proof. The obtained non-deterministic rewrite system is proved to be invariant with respect to the process of cut elimination. As a result, the language of the rewrite system subsumes the Herbrand sets obtained from any classical process of extraction through cut-elimination, circumventing the problem of different Herbrand disjunctions.

The connection between recursion schemes and typed λ-calculus is well known and almost immediate. Reduction rules of λ-calculus can be seen as rewrite rules, and abstraction can be simulated by functionals (*non-terminals*). At the same time, the structure of higher-order recursion schemes allows for an almost direct correspondence between proofs and schemes. The possibility to establish a one-to-one relationship between the deduction steps of a proof and the rules in the corresponding HORS is an unquestionable advantage in terms of readability and manageability, with respect to the λ-terms of the Curry–Howard correspondence.

Overview

In the next two chapters we will present a method for defining higher-order recursion schemes from cyclic proofs of Heyting Arithmetic. Given the cyclic nature of our proof, we will first define a correspondence with terms of λY-calculus, that is, with the addition of a fixed point combinator Y. Recursion schemes present no limitation in treating cycles and it is already known that they can be seen as another syntax for λY-calculus.[4] This last connection closes the argument motivating the present work. In the framework of Curry–Howard correspondence, we show the relationship between a cyclic system ICA for Heyting Arithmetic and terms of λY-calculus. The close correspondence between this last formalism and higher-order recursion schemes will motivate the definition of a HORS $\mathcal{H}^\pi$ to extract witnesses from a given proof, in line with the work in [AHL20]. Once $\mathcal{H}^\pi$ has been defined, the rewriting process determines a grammar whose specification at the moment is left open for future analysis, but that corresponds, in a general perspective, to the production of terms-witnesses for the given initial proof, from which the extraction of relevant content can be implemented.

In Chapter 1 we introduce the language $\mathcal{L}^=$ of arithmetic, the standard sequent calculus for intuitionistic logic with equality, and present the cyclic system ICA, motivating its development from the works of Brotherston and Simpson and Berardi and Tatsuta. Once the notions of proof and pre-proof have been characterised, the λ-terms constituting the language Σ are defined together with their types. After a brief comment on the rules and type of the inductive predicate N, we proceed with the definition of cyclic terms thanks

[4]See [SW12].

to the introduction of the fixed point combinator Y. The last section is devoted to the definition of recursion schemes and their rewriting procedure.

Chapter 2 begins with the definition of the higher-order recursion scheme $\mathcal{H}^\pi$ built from a cyclic proof π. The language Σ^H of the terminals is defined, and the rules of the HORS are motivated. In Section 2.1 some properties of the schemes are showed, regarding type, substitution, cut-strategies and their order. In the last part a closer look is given to the process that extracts the language from a $\mathcal{H}^\pi$, with some result and comment on confluence, reduction strategies, normal forms and termination.

Chapter 1

Intuitionistic Cyclic Arithmetic

The language $\mathcal{L}^=$ is the language of arithmetic: the set of terms is defined by individual variables $x, y, \ldots$, a constant 0 and functors $+, \cdot, \mathbf{s}(-)$ for the usual arithmetic operations

$$t ::= x \mid 0 \mid t + t \mid \mathbf{s}(t) \mid t \cdot t$$

The set For of formulas of the language is defined by

$$\varphi ::= \bot \mid t = s \mid Nt \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \varphi \to \varphi \mid \exists x.\varphi \mid \forall x.\varphi$$

Atomic formulas are equalities and the unary predicate Nt, which is the predicate for natural numbers; the remaining formulas are defined using booleans and quantifiers. A constant symbol $\bot$ for falsum is used to define negation $\neg\varphi := \varphi \to \bot$. Sequents of the form $\Gamma \Rightarrow \varphi$ are interpreted as lists of formulas of $\mathcal{L}^=$, with φ the unique formula on the right-hand side. The intuitive meaning of a sequent $\Gamma \Rightarrow \chi$ is $\bigwedge \Gamma \to \chi$. The proof system ICA defined in this chapter is a cyclic sequent calculus for intuitionistic arithmetic. The proper definition of ICA will be given later in the chapter, after the aspects that differentiate it from a standard sequent calculus have been presented, starting with the introduction of inductive predicates.

1.1 LJID$^\omega$ and induction.

In [BS11][1] Brotherston and Simpson defined the system LKID, a sequent calculus for classical first-order logic with equality and inductive predicates, and an infinitary extension LKID$^\omega$ was given in the same paper. An intuitionistic version is defined here, similar to the system LJID$^\omega$ by Berardi and Tatsuta [BT17b] but with some minor variations. The rules of LJID$^\omega$ are the following:

[1]See table on page 1184.

Axioms

$$\frac{}{t = s, \Gamma \Rightarrow t = s}\ (\text{Ax}) \qquad \frac{}{\bot, \Gamma \Rightarrow t = s}\ (L\bot)$$

Structural rules

$$\frac{\Gamma \Rightarrow \varphi \qquad \varphi, \Delta \Rightarrow \chi}{\Gamma, \Delta \Rightarrow \chi}\ (\text{cut}) \qquad \frac{\Gamma \Rightarrow \chi}{\varphi, \Gamma \Rightarrow \chi}\ (\text{W}) \qquad \frac{\varphi, \varphi, \Gamma \Rightarrow \chi}{\varphi, \Gamma \Rightarrow \chi}\ (\text{C}) \qquad \frac{\Gamma \Rightarrow \chi}{\Gamma[\theta] \Rightarrow \chi[\theta]}\ (\text{Sub})$$

Logical rules

$$\frac{\Gamma \Rightarrow \varphi \qquad \psi, \Delta \Rightarrow \chi}{\varphi \to \psi, \Gamma, \Delta \Rightarrow \chi}\ (\text{L}\to) \qquad \frac{\varphi, \Gamma \Rightarrow \psi}{\Gamma \Rightarrow \varphi \to \psi}\ (\text{R}\to)$$

$$\frac{\varphi, \psi, \Gamma \Rightarrow \chi}{\varphi \wedge \psi, \Gamma \Rightarrow \chi}\ (\text{L}\wedge) \qquad \frac{\Gamma \Rightarrow \varphi \qquad \Delta \Rightarrow \psi}{\Gamma, \Delta \Rightarrow \varphi \wedge \psi}\ (\text{R}\wedge)$$

$$\frac{\varphi, \Gamma \Rightarrow \chi \qquad \psi, \Delta \Rightarrow \chi}{\varphi \vee \psi, \Gamma, \Delta \Rightarrow \chi}\ (\text{L}\vee) \qquad \frac{\Gamma \Rightarrow \varphi_i}{\Gamma \Rightarrow \varphi_0 \vee \varphi_1}\ (\text{R}\vee)$$

$$\frac{\varphi(z), \Gamma \Rightarrow \chi}{\exists y.\varphi(y), \Gamma \Rightarrow \chi}\ (\text{L}\exists) \qquad \frac{\Gamma \Rightarrow \varphi(t)}{\Gamma \Rightarrow \exists y.\varphi(y)}\ (\text{R}\exists) \qquad \frac{\varphi(t), \Gamma \Rightarrow \chi}{\forall y.\varphi(y), \Gamma \Rightarrow \chi}\ (\text{L}\forall) \qquad \frac{\Gamma \Rightarrow \varphi(z)}{\Gamma \Rightarrow \forall y.\varphi(y)}\ (\text{R}\forall)$$

The eigenvariable condition requires that $z \notin FV(\Gamma, \chi)$ in rules $(L\exists)$ and $(R\forall)$. Without loss of generality, we adopt the convention that the eigenvariables do not occur outside of the subproof above their respective existential or universal rule, an assumption that we call *regularity condition* from [AHL20]. Unlike [BT17b], here we opt for a multiplicative sequent calculus. The axiom rules (Ax) and $(L\bot)$ present only equalities $t = s$ as principal, in order to facilitate future inductive arguments. In the rest of the paper, however, we will treat sequents of the form $\varphi, \Gamma \Rightarrow \varphi$ and $\bot, \Gamma \Rightarrow \varphi$ as axioms for any formula φ, inaccurately but truthfully. It is provable, in fact, that both are always derivable in ICA.[2] The present version of LJID^ω has left contraction even if it is not present in [BT17b], because of a different interpretation of sequents as multisets instead of lists. Notice also that there is an explicit rule (Sub) for substitution, where $[\theta]$ stands for some substitution of free variables with terms. Usually, substitution is an operation that we perform at a meta-level: given a proof of a sequent $\Gamma \Rightarrow \chi$ we agree that it is always possible to instantiate the free variables with terms and have a proof of $\Gamma[\theta] \Rightarrow \chi[\theta]$. In ICA a formal

[2]An easy proof by induction, where the interesting case is the one with $\varphi \equiv Nt$. See Example 1.11.

rule is defined for that. Brotherston and Simpson, reflecting on the role of the substitution rule in [BS11], conjectured that it is essential for cyclic proofs in order to achieve perfect correspondence between sequents, necessary in the case of cyclic proofs. The conjecture appears to be more than reasonable, but they left the question open to future work. In all the rules above, and also in general, the principal formula is the one that is subject of the proof; the cut-formula is the φ in the example. Finally we have the rules for

Equality

$$\frac{}{\Gamma \Rightarrow t = t}\ (id\ ) \qquad \frac{\Gamma[t/x, s/y] \Rightarrow \chi[t/x, s/y]}{t = s, \Gamma[s/x, t/y] \Rightarrow \chi[s/x, t/y]}\ (\mathrm{L}{=})$$

where $(L =)$ gives the opportunity to switch terms in the whole sequent whenever they are known to be identical.

The system described so far is an intuitionistic first-order sequent calculus with equality and an explicit rule for substitution. The list of rules of LJID$^\omega$ is completed by the rules for the inductive predicates. For a finite set of inductive predicates $\{P_0, \dots, P_k\}$ in our language, we introduce a number of rules for each one of them. Given an inductive predicate P_j, its right rules have the form

$$\frac{\Gamma_0 \Rightarrow P_{j0} \qquad \dots \qquad \Gamma_n \Rightarrow P_{jk}}{\Gamma_0, \dots, \Gamma_n \Rightarrow P_j}\ (RP_j)$$

Where the P_{ji}'s are determined by the inductive definition of P_j. If P_j is the natural number predicate Nx, for example, the definition says that N holds always for 0, and for $\mathbf{s}(t)$ if is true of t. The left rule for P_j is more complicated to describe. In its general form[3] it is given in [BS11] as

$$\frac{\text{case distinctions}}{P_j t, \Gamma \Rightarrow \chi}\ (LP_j\ )$$

where the case distinctions are defined by the set of predicates from which P_j depends according to its inductive definition. We are not going to look into the details of the definition of minor premises, the gist being a finite series of sequents of the form

$$t = s, P_{j0}, \dots P_{jm}, Q_{j0}, \dots, Q_{jn}, \Gamma_i \Rightarrow \chi$$

with P's and Q's being respectively inductive and non-inductive predicates. The P_{ji}'s are also called *case descendant* of P_j. The result is that the inductive definition is decomposed into all the possible cases. In the language $\mathcal{L}^=$ just one inductive predicate Nx is defined, hence we have three rules according to the construction seen above:

[3]The rule below is already the one defined for the cyclic version of LKID$^\omega$: in the non-cyclic one (see p.1185) we consider also another predicate F constituting the induction hypothesis. There is no point in giving that formulation here since LJID is presented only as an introductory step.

Natural numbers

$$\frac{}{\Gamma \Rightarrow N0}\ (RN_0\ ) \qquad \frac{\Gamma \Rightarrow Nt}{\Gamma \Rightarrow N\mathbf{s}(t)}\ (RN_1\ )$$

$$\frac{t = 0, \Gamma \Rightarrow \chi \qquad t = \mathbf{s}(r), Nr, \Delta \Rightarrow \chi}{Nt, \Gamma, \Delta \Rightarrow \chi}\ (LN)$$

To (LN) the eigenvariable condition applies, preventing $r \in FV(\Gamma, \Delta, \chi)$. For each rule we call *principal formula* the resulting formula Nt and in the case of (LN) we also call Nr in the right premise the *case descendant* of Nt. From now on, we assume that the principal formula always appears as leftmost in the conclusion of each left rule. In order to have such a property syntactically, we should define a rule that formally takes care of adjusting the order of the list in the antecedent

$$\frac{\Gamma, \varphi, \psi, \Delta \Rightarrow \chi}{\Gamma, \psi, \varphi, \Delta \Rightarrow \chi}\ (ex\ )$$

Including such a rule in LJID$^\omega$ or later in ICA would have the effect of making the work and presentation more tedious, without any relevant addition. We decide, then, to don't include (ex) among the rules. Instead, we will assume that before any instance of each rule, the order of the formulas in the antecedent has been implicitly organised so that the rule is applicable, without any chance for misinterpretation or confusion, *as if* an analogous of the exchange rule was contained and operating in each formula.

1.1.1 Pre-proofs and proofs

Now that the rules have been presented, it is possible to look at what is an infinite proof in LJID$^\omega$. Deduction trees with possibly infinitely long branches are called *pre-proofs*:

Definition 1.1 (Pre-proof)**.** *A pre-proof of a sequent* $\Gamma \Rightarrow \chi$ *in* LJID$^\omega$ *is a possibly infinite labelled tree* $\mathcal{D}$ *such that* $\Gamma \Rightarrow \chi$ *is labelled at the root, and every child node is given according to the rules of* LJID$^\omega$*.*

It is easy to show that pre-proofs are not sound in general.

Example 1.2.

$$\dfrac{\dfrac{\overset{\vdots}{x = 1 \Rightarrow x = 2} \qquad x = 2 \Rightarrow x = 2}{x = 1 \Rightarrow x = 2} \qquad x = 2 \Rightarrow x = 2}{x = 1 \Rightarrow x = 2}$$

With an infinite trivial application of cut we can have a pre-proof of any formula.

In order to ensure soundness of derivations, transforming pre-proofs into proofs by infinite descent, we need to impose a control condition on infinite paths. The condition consists in the request that on every infinite path the inductive predicates are unfolded infinitely often. The infinite regression becomes then only potential, since the path is bound to the well-founded structure of the natural numbers. To track the evolution of the computation with respect to each formula the notion of a trace is necessary.

Definition 1.3 (Path). *A path $\mathbb{P}$ in a derivation tree $\mathcal{D}$ is a possibly infinite list of sequents $(\Gamma_i \Rightarrow \chi_i)$ such that the $i+1^{th}$ element of the list is a child node of the i^{th} element, for all $0 \leq i \in \mathbb{N}$.*

Definition 1.4 (Trace). [4] *A trace along a path $(\Gamma_i \Rightarrow \chi_i)_{i\geq 0}$ in a pre-proof $\mathcal{D}$ is a possibly infinite sequence of formulas $\tau_i = P_j t_i$ such that:*

1. $\tau_i \in \Gamma_i$
2. *if $\Gamma_i \Rightarrow \chi_i$ is the result of a substitution θ, then $\tau_{i+1}[\theta] = \tau_i$;*
3. *if $\Gamma_i \Rightarrow \chi_i$ is the result of a $(L =)$ for $t = s$ principal formula, then there is a formula φ and variables x, y such that $\tau_i = \varphi[t/x, s/y]$ and $\tau_{i+1} = \varphi[s/x, t/y]$;*
4. *if $\Gamma_i \Rightarrow \chi_i$ is the result of a (LP_j) and τ_i is principal, then τ_{i+1} is the case-descendant of τ_i;*
5. *if $\Gamma_i \Rightarrow \chi_i$ is the result of any other rule, then $\tau_{i+1} = \tau_i$*

As desired, the trace follows an inductive predicate along a path, adjusting to potential term changes due to (Sub) and $(L =)$, and progresses only when an inductive predicate P_j is the principal formula. The steps described by point 4 are called *progressions* of the derivation, and a trace with infinite progressions is an infinitely progressing trace.

Definition 1.5 (Global trace condition). *For every infinite path $\mathbb{P}$ there is an infinitely progressing trace following some tail of the path.*

We have all the elements now to define the notion of a proof in LJID^ω.

Definition 1.6 (Proof). *A pre-proof $\mathcal{D}$ is a proof if it satisfies the global trace condition.*

Theorem 1.7 (Soundness). *The system LJID^ω is sound: if there is a proof of $\Gamma \Rightarrow \chi$, then $\Gamma \Rightarrow \chi$ is valid with respect to standard models.*

Proof. In [Bro06] soundness of LKID^ω with respect to standard models[5] is proved by local soundness of the rules and the fact that any derivation of an unsatisfiable sequent contains an infinite trace generating an infinitely decreasing chain of ordinals, leading to a contradiction. The same argument can be made for proofs in LJID^ω. □

[4] See the Definitions in [BS11], p.1196.

[5] In which inductive predicates are interpreted as the fixed points of the denotation of their corresponding operator.

1.2 Cyclic proofs and Arithmetic

Systems like LKID^ω and LJID^ω are powerful and constitute good instruments to deal with induction in sequent calculus, but they are impractical to manage given their infinitary nature, the most evident problem being the fact that the global trace condition has to be checked on all infinite paths. An informal look at their structure, however, strongly suggests the existence of a finite method to represent such infinite arguments. There exist, in fact, systems CLKID^ω and CLJID^ω that consist of finite objects, corresponding to a relevant subset of the derivations of LKID^ω and LJID^ω, namely those derivations that are regular. A regular proof is a possibly infinite tree in which there are only finitely many distinct subtrees. In order to have cyclic proofs, it is necessary to define a relation between leaf-nodes s (called *buds*) and some internal nodes t, identified as their *companions.*

Definition 1.8 (Bud/companion relationship)**.** *Let $\mathcal{D}$ be a labelled tree and s a leaf that is not an instance of an axiom (bud).* $\mathtt{R}$ *is a function from bud nodes to internal nodes in $\mathcal{D}$ such that $\mathtt{R}(s) = t$ for some node t with the same labelled sequent as s.*

The two nodes must share the very same sequent for the relation to hold between them.[6] Notice, however, that it is not necessary for them to appear on the same path. With the definition of such relation we can define cyclic pre-proofs. The rules of CLJID^ω are the same of LJID^ω.

Definition 1.9 (Cyclic Pre-Proof)**.** *A pre-proof of a sequent $\Gamma \Rightarrow \chi$ is a pair $(\mathcal{D}, \mathtt{R})$ where $\mathcal{D}$ is a finite derivation of $\Gamma \Rightarrow \chi$ from the rules of* CLJID^ω*, and* $\mathtt{R}$ *is a function that assigns to each of the non-axiomatic leaves a node inside $\mathcal{D}$ with the same label.*

The definition of a path in $(\mathcal{D}, \mathtt{R})$ is like the one for LJID^ω adjusted to include bud nodes.

Definition 1.10 (Path)**.** *A path $\mathbb{P}$ in a derivation tree $\mathcal{D}$ is a possibly infinite list of sequents $(\Gamma_i \Rightarrow \chi_i)$ such that the $i+1^{th}$ element of the sequence is a child node of the i^{th} element, or its companion.*

The definition of trace, progressing trace and global trace condition are the same as in the infinite proof system.

Definition (Proof)**.** *A cyclic pre-proof $(\mathcal{D}, \mathtt{R})$ is a cyclic proof if it satisfies the global trace condition.*

Even though by definition the paths of a cyclic proof of CLJID^ω are potentially infinite, the object that constitutes a proof is a finite tree. It is decidable whether a pre-proof is a proof.[7] As we mentioned, the proofs of the cyclic system CLJID^ω correspond to a proper subset of the proofs of LJID^ω. As a consequence, we have that the soundness of the former can be inferred by the soundness of the latter.

[6] See comment on (Sub) rule above.

[7] For a formal proof of this fact using automata see Proposition 7.4 of [BS11].

As a first example of a cyclic proof, let's see the proof of the fact that $Nt, \Gamma \Rightarrow Nt$ is always provable

Example 1.11.

$$
\dfrac{
  (L=)\,\dfrac{(RN_0)\,\dfrac{}{\Gamma \Rightarrow N0}}{t = 0, \Gamma \Rightarrow Nt}
  \qquad
  \dfrac{\dfrac{\dfrac{Nt, \Gamma \Rightarrow Nt \quad (\star)}{Ns, \Gamma \Rightarrow Ns}\,(Sub)}{Ns, \Gamma \Rightarrow N\mathbf{s}(s)}\,(RN_1)}{t = \mathbf{s}(s), Ns, \Gamma \Rightarrow Nt}\,(L=)
}{
  \dfrac{Nt, \Gamma, \Gamma \Rightarrow Nt}{Nt, \Gamma \Rightarrow Nt \quad (\star)}
}\,(LN)
$$

In this example we can see the cycle generated by the right branch, where the bud node is labelled likewise the root, and the trace is coloured in blue, progressing every time in which the rule (LN) is applied.

1.3 Σ-terms and types

Following the BHK interpretation, having a proof of a formula φ in intuitionistic arithmetic means for us to have an object that justifies the assertion of φ. A proof of a statement of the form $\psi_0 \wedge \psi_1$, for example, is an object consisting of a proof of ψ_0 and a proof of ψ_1. For $\psi_0 \to \psi_1$, we need an object that, like a function, takes a proof of ψ_0 and turns it into a proof of ψ_1. A proof of a disjunction is an object that, in accordance with the meaning of disjunction in intuitionistic logic, is already a proof of one of the disjuncts.

In the present paper we adopt the *proofs-as-programs* principle of the Curry–Howard correspondence, namely the fact that to an intuitionistic proof of a formula φ it is possible to assign a λ-term (the program) whose type is the type corresponding to φ. We use terms of the λ-calculus as proof-objects to describe our derivations. With this goal in mind, the set of types corresponding to formulas in our language $\mathcal{L}^{=}$ is defined, followed by the set of terms inhabiting those types, i.e. the term language Σ. We begin by introducing the set of types that are called basic

Definition 1.12 (Basic Types)**.**

- *ι is the type of individuals*
- *ϵ the unit type*
- *$\bot$ is the empty type;*
- *for σ, τ types*
  - *$\sigma \to \tau$ is the function type*
  - *$\sigma \times \tau$ is the product type*

– $\sigma + \tau$ *is the sum type*

The correspondence between terms or formulas of $\mathcal{L}^=$ and types is given in the following table:

Term or Formula	**Type**
t	ι
$t = s$	ϵ
$\bot$	$\bot$
$\varphi \to \psi$	$\sigma_\varphi \to \sigma_\psi$
$\varphi \wedge \psi$	$\sigma_\varphi \times \sigma_\psi$
$\varphi \vee \psi$	$\sigma_\varphi + \sigma_\psi$
$\forall x.\varphi$	$\iota \to \sigma_\varphi$
$\exists x.\varphi$	$\iota \times \sigma_\varphi$

Types ι and ϵ are called ground types. Quantified formulas correspond to dependent types, meaning that they represent a collection of types depending on the individual object they receive as input. A formula $\varphi(x)$ with a free variable needs to be read as a class of types given by all the formulas $\varphi(t)$ for t an individual term. We will often use greek letters $\rho, \sigma, \nu, \ldots$ as meta-variables for types. In the rest of the paper we will use formulas and their corresponding types as interchangeable, choosing one instead of the other for the sake of clarity. The expression $[\varphi]$ is to be read as "the type of the formula φ".

We introduce now the language Σ of λ-terms that inhabit the types defined above. For each type except $\bot$ and ground ones there are two operations to construct and de-construct a term of the given type. We adopt the usual notation $r : \sigma$ to indicate that the Σ-term r is of type σ. Since we need to be able to refer to individual terms of $\mathcal{L}^=$ in our Σ, we also agree that for each individual variable, term and functor in the alphabet of $\mathcal{L}^=$ we use in Σ a corresponding identical term. Formally speaking, however, the term 0 in Σ is a different object than the term 0 in $\mathcal{L}^=$, and the same is true for variables, successor and the other function symbols.

Definition 1.13 (Basic Σ-terms and types)**.** *The definition establishes the basic elements of Σ and the relation between terms and types:*

Type	***Terms***	***Constructors***	***De-constructors***
ι	$0 : \iota \quad \mathbf{s}() : \iota \to \iota \quad a : \iota$	$\mathbf{s}(t) : \iota$	
	$(+) : \iota \to \iota \to \iota$	$t + s : \iota$	
	$(\cdot) : \iota \to \iota \to \iota$	$t \cdot s : \iota$	
ϵ	$\langle\rangle : \epsilon$		
$\bot$	$*_\psi : \bot \to [\psi]$		$*_\psi\, p : \psi$
$\sigma \to \tau$		$\lambda p^\sigma.r^\tau : \sigma \to \tau$	$(q^{\sigma\to\tau} p^\sigma) : \tau$
$\sigma \times \tau$		$\langle p^\sigma, r^\tau\rangle : \sigma \times \tau$	$\pi_0 q^{\sigma\times\tau} : \sigma \quad \pi_1 q^{\sigma\times\tau} : \tau$
$\sigma + \tau$		$\kappa_0 p^\sigma : \sigma + \tau \quad \kappa_1 r^\tau : \sigma + \tau$	$Case^{\vee}_{p,r}(k_i q, n^\rho_p, n^\rho_r) : \rho$

In order to minimise confusion, from now on we adopt the following notational convention:

- greek letters like $\varphi, \psi, \chi \dots$ will be used as meta-variables for formulas, while
- greek letters like $\rho, \sigma, \nu, \dots$ will be meta-variables for types
- $a, b, c, \dots$ are used for individual variables (in both $\mathcal{L}^=$ and Σ)
- $t, r, s, \dots$ will be meta-variables for individual terms (in both $\mathcal{L}^=$ and Σ)
- $x, y, z, \dots$ will be used as variables for λ-terms in Σ
- $p, q, n, \dots$ are meta-variables for Σ-terms

We will try to carefully follow this distinction, explicitly mentioning the nature of the symbols when necessary.

We have presented so far the usual terms corresponding to the types of first order intuitionistic formulas. The correspondence is reflected by the following term-calculus,[8] where, to each rule of the first order fragment of ICA, corresponds a term of the type of the formula. To each sequent $\Gamma, \varphi \Rightarrow \psi$ obtained in ICA corresponds a sequent $\overline{x} : \Gamma, y : \varphi \Rightarrow p : \psi$. The λ-term p is a witness of the type $[\psi]$, while the whole sequent can be represented by the term $\lambda\overline{x}y.p$, reflecting the dependency relation between the set of assumptions and the term on the right. Define the following term-calculus, where w is a fresh variable (we omit the context Γ for readability)

$$\frac{}{\mathbf{y} : \epsilon \Rightarrow \mathbf{y} : \epsilon}\text{(Ax)} \qquad \frac{}{\mathbf{y} : \bot \Rightarrow \ast_\epsilon\, \mathbf{y} : \epsilon}(L\bot)$$

$$\frac{\Rightarrow \mathbf{n} : \varphi \qquad \mathbf{y} : \varphi \Rightarrow \mathbf{p} : \psi}{\Rightarrow \mathbf{p}[\mathbf{n}/\mathbf{y}] : \psi}\text{(cut)}$$

$$\frac{\Rightarrow \mathbf{p} : \psi}{\mathbf{w} : \varphi \Rightarrow \mathbf{p} : \psi}\text{(W)} \qquad \frac{\mathbf{y} : \varphi, \mathbf{x} : \varphi \Rightarrow \mathbf{p} : \psi}{\mathbf{y} : \varphi \Rightarrow \mathbf{p}[\mathbf{y}/\mathbf{x}] : \psi}\text{(C)} \qquad \frac{\Rightarrow \mathbf{p} : \psi}{\Rightarrow \mathbf{p}[\theta] : \psi}\text{(Sub)}$$

$$\frac{\Rightarrow \mathbf{p_0} : \varphi \qquad \mathbf{y} : \psi \Rightarrow \mathbf{p_1} : \chi}{\mathbf{w} : \varphi \to \psi \Rightarrow \mathbf{p_1}[(\mathbf{w}\mathbf{p_0})/\mathbf{y}] : \chi}\text{(L}\to\text{)} \qquad \frac{\mathbf{y} : \psi \Rightarrow \mathbf{p} : \varphi}{\Rightarrow \lambda\mathbf{y}.\mathbf{p} : \psi \to \varphi}\text{(R}\to\text{)}$$

$$\frac{\mathbf{y} : \psi, \mathbf{z} : \varphi \Rightarrow \mathbf{p} : \chi}{\mathbf{w} : \psi \times \varphi \Rightarrow \mathbf{p}[(\pi_\mathbf{0}\mathbf{w})/\mathbf{y}, (\pi_\mathbf{1}\mathbf{w})/\mathbf{z}] : \chi}\text{(L}\wedge\text{)} \qquad \frac{\Rightarrow \mathbf{p} : \psi \qquad \Rightarrow \mathbf{q} : \varphi}{\Rightarrow \langle \mathbf{p}, \mathbf{q} \rangle : \psi \times \varphi}\text{(R}\wedge\text{)}$$

$$\frac{\mathbf{y} : \psi \Rightarrow \mathbf{p_0} : \chi \qquad \mathbf{z} : \varphi \Rightarrow \mathbf{p_1} : \chi}{\mathbf{w} : \psi + \varphi \Rightarrow \mathbf{Case}^\vee_{\mathbf{y},\mathbf{z}}(\mathbf{w}, \mathbf{p_0}, \mathbf{p_1}) : \chi}\text{(L}\vee\text{)} \qquad \frac{\Rightarrow \mathbf{p} : \psi_i}{\Rightarrow \kappa_\mathbf{i}(\mathbf{p}) : \psi_0 + \psi_1}\text{(R}\vee\text{)}$$

[8]These rules in Troelstra-Schwittenberg [TS00] constitute the term-calculus t-G2i.

$$\frac{\mathbf{y} : \varphi \Rightarrow \mathbf{p} : \psi}{\mathbf{w} : \iota \times \varphi \Rightarrow \mathbf{p}[(\pi_\mathbf{0}\mathbf{w})/\mathbf{a}, (\pi_\mathbf{1}\mathbf{w})/\mathbf{y}] : \psi} \text{(L}\exists\text{)} \qquad \frac{\Rightarrow \mathbf{p} : \varphi}{\Rightarrow \langle \mathbf{t}, \mathbf{p} \rangle : \iota \times \varphi} \text{(R}\exists\text{)}$$

$$\frac{\mathbf{y} : \varphi \Rightarrow \mathbf{p} : \chi}{\mathbf{w} : \iota \to \varphi \Rightarrow \mathbf{p}[(\mathbf{wt})/\mathbf{y}] : \chi} \text{(L}\forall\text{)} \qquad \frac{\Rightarrow \mathbf{p} : \varphi}{\Rightarrow \lambda \mathbf{a}.\mathbf{p} : \iota \to \varphi} \text{(R}\forall\text{)}$$

$$\frac{}{\Rightarrow \langle\rangle : \epsilon} (id\ ) \qquad \frac{\Rightarrow \mathbf{p} : \psi}{\mathbf{w} : \epsilon \Rightarrow \mathbf{p} : \psi} \text{(L=)}$$

Instead of going through the meaning of each rule, let's just highlight some motivation for the term assignment above. The dependency relation between terms in the antecedent and consequent is visible in almost all the left rules, and clearly in the implication rules. $(R \to)$ produces a term $\lambda y.p$ from a term p and a context that includes y. Dependency can also be vacuous, as it appears evidently in the case of (W): the introduction of a fresh variable w doesn't affect the term on the right, in accordance with the intuitive meaning of the *weakening* rule. A remark on rule $(L =)$: dependency can be vacuous even when some connection between the new variable and the right term potentially exists. This should not be surprising, because from the point of view of dependent types $[\varphi(t)] = [\varphi(s)]$ under the assumption that $t = s$, hence if $p : [\varphi(t)]$ then $p : [\varphi(s)]$.

The role of right rules is to act as definitions. They combine the material in the antecedents to produce terms of the new type on the right. The case of $(R\vee)$ reflects the intuitionistic nature of the rule, by keeping track of the disjunct that witnesses the validity of the formula with κ_0 or κ_1. Left rules re-define dependency. To the principal formula in the context is assigned a fresh variable w of the correct type, while every possible occurrence of the old variable(s) on the right-hand side is substituted by a new term. The new term is usually a de-constructor on the new variable. For example: in the case of left conjunction the variable corresponding to the first element y occurring in p is replaced by the first projection $\pi_0 w$ of the new variable of pair type w. The $(L\vee)$ rule requires some explanation. An intuitionistic disjunction always comes from one specific disjunct. The fresh variable w stands for a term of the form $\kappa_i q$ but since it is a variable, we don't know the value of i. We are forced to consider both cases: $Case^{\vee}_{y,z}(w, p_0, p_1)$ tells us that "depending on w being of the type of y or z, the witness for the final formula is given by p_0 or p_1". The same convoluted principle holds for the inductive predicate case, as we will see. A few comments on the (cut) and quantifiers' rules. In a cut, the variable $y : \varphi$ in the right antecedent is substituted by the term $p : \varphi$ from the left, resulting in an instantiation of the variable. Looking at the term witnessing the whole sequent of the right premise, i.e., $\lambda y\overline{x}.q$, a cut corresponds to the β-reduction of the term $(\lambda y\overline{x}.q)p \to_\beta \lambda\overline{x}.q[p/y]$. The rules for quantifiers mimic faithfully the rules for the pairing and abstraction as seen in conjunction and implication, following the interpretation of dependent types. The only difference here is that the first object has a specific type, namely ι.

The term assignment using Σ-terms follows the assignment present in literature, see [TS00]. In order to define terms that fully represent the cyclic ICA we still need to define the type of N and address the cyclic behaviour. Before moving to that, we want to spend a few words on substitution. For reasons that will become clear later, we would like to have the chance to differentiate between *explicit* and *implicit* substitutions.[9] In λ-calculus we commonly use implicit substitutions, for example in the definition of β reduction $(\lambda x.M)N \to_\beta M[N/x]$ we write $M[N/x]$ to indicate the term obtained by a substitution of all the occurrences of x inside M with N. The substitution is written in square brackets to underline its effect and it is syntactically immediate: $M[N/x]$ is an actual term M' where N replaces x. Working with proof terms we might like to keep track of a substitution, but wait to evaluate its effect until some other internal process has finished. In this case we talk about explicit substitution, and for this purpose we introduce a term α of type ς called substitution stack. To Definition 1.13 we add the following lines:

Type	**Terms**	**Constructor**	**De-constructor**
ς		$[t \mapsto a]\alpha^\varsigma : \varsigma$	
ρ basic		$p \circ \alpha : \rho$	

The meaning of $[t \mapsto a]$ is precisely what one expects: the term t replaces the variable a. With $p \circ \alpha : \rho$ we indicate the Σ-term p together with the substitutions listed in α. The formal definition of an implicit substitution is given by

Definition 1.14 (Implicit substitution)**.** *For $p, q \in \Sigma$, the result of a substitution $[t/a]$ is given by the following definition:*

$$
\begin{array}{rlrl}
0[t/a] & \longrightarrow 0 & (\mathbf{s}(r))[t/a] & \longrightarrow \mathbf{s}(r[t/a]) \\
(s+r)[t/a] & \longrightarrow s[t/a] + r[t/a] & b[t/a] & \longrightarrow b \\
a[t/a] & \longrightarrow t & & \\
\langle\rangle[t/a] & \longrightarrow \langle\rangle & *_\psi q[t/a] & \longrightarrow *_\psi q \\
(\lambda a.q)[t/a] & \longrightarrow \lambda a.q & (\lambda y.q)[t/a] & \longrightarrow \lambda y.(q[t/a]) \\
(pq)[t/a] & \longrightarrow (p[t/a]q[t/a]) & \langle p, q\rangle & \longrightarrow \langle p[t/a], q[t/a]\rangle \\
(\pi_i q)[t/a] & \longrightarrow \pi_i(p[t/a]) & (\kappa_i p)[t/a] & \longrightarrow \kappa_i(p[t/a]) \\
([s \mapsto b]\beta)[t/a] & \longrightarrow [s[t/a] \mapsto b]\beta[t/a] & (b \circ \beta)[t/a] & \longrightarrow b[t/a] \circ \beta[t/a]
\end{array}
$$

The correspondence between implicit and explicit substitution at the term level is given by the process of evaluation, that is when α is actually enforced and the two notions become equivalent.

Definition 1.15 (Evaluation)**.** *Given a Σ term $(s \circ \beta)$ with $s \in \Sigma$ basic and substitution stack $\beta \equiv [t_0 \mapsto a_0] \ldots [t_n \mapsto a_n]$, the evaluation s^β of s relative to β is the Σ term given*

[9]This idea is taken from [AHL20].

by $s[t_0/a_0]\dots[t_n/a_n]$. The term resulting from the evaluation of all the substitutions in s is indicated with s°.

For example $\varphi(a,b) \circ [t_0 \mapsto a][t_1 \mapsto b]$ will result in $\varphi(t_0,t_1)$ after evaluation, that is precisely $\varphi(a,b)[t_0/a, t_1/b]$. In section 2.1 we will see that in our recursion schemes the two different kinds of substitution coincide under a few assumptions. We are ready now to talk about arithmetical axioms, λ-terms for natural number predicate and cyclic terms.

1.4 Heyting Arithmetic

Let's finally introduce arithmetic in both ICA and the language Σ. To the list of rules presented above, add the 6 axioms of arithmetic given in the form of zero-premise rules:

$$\frac{}{\mathbf{s}t = 0, \Gamma \Rightarrow}\,(\bot 0) \qquad \frac{}{\Gamma \Rightarrow t + 0 = t}\,(+0) \qquad \frac{}{\Gamma \Rightarrow r + \mathbf{s}t = \mathbf{s}(r+t)}\,(+\mathbf{s})$$

$$\frac{}{\Gamma \Rightarrow t \cdot 0 = 0}\,(\cdot 0) \qquad \frac{}{\Gamma \Rightarrow r \cdot \mathbf{s}t = (r \cdot t) + r}\,(\cdot \mathbf{s}) \qquad \frac{}{\Gamma, \mathbf{s}t = \mathbf{s}r \Rightarrow t = r}\,(= \mathbf{s})$$

Any term of unit type can be a witness for an axiom, since its validity is not depending on the context but on the arithmetic content. We then have

$$\frac{}{\mathbf{y} : \epsilon \Rightarrow \ast_\bot \mathbf{y} : \bot}\,(\bot 0) \qquad \frac{}{\mathbf{y} : \epsilon \Rightarrow \mathbf{y} : \epsilon}\,(= \mathbf{s}) \qquad \frac{}{\Rightarrow \mathbf{y} : \epsilon}\,(\ast\star)$$

where $\ast \in \{+, \cdot\}$ and $\star \in \{0, \mathbf{s}\}$.

Remember that it is not necessary to introduce an inductive scheme or rule, because induction is taken care already by the cyclic proof and the inductive predicates: it is possible to prove the inductive scheme for any formula φ as follows

$$\dfrac{\dfrac{\varphi(0), \forall x.\varphi x \to \varphi(\mathbf{s}(x)) \Rightarrow \varphi(0)}{y = 0, \varphi(0), \forall x.\varphi x \to \varphi(\mathbf{s}(x)) \Rightarrow \varphi(y)}\,(L=) \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{Ny, \varphi(0) \Rightarrow \varphi(y)}{Nz, \varphi(0), \Rightarrow \varphi(z)}\,(Sub) \qquad \varphi(\mathbf{s}(z)) \Rightarrow \varphi(\mathbf{s}(z))}{Nz, \varphi(0), \varphi(z) \to \varphi(\mathbf{s}(z)) \Rightarrow \varphi(\mathbf{s}(z))}\,(L\to)}{Nz, \varphi(0), \forall x.\varphi x \to \varphi(\mathbf{s}(x)) \Rightarrow \varphi(\mathbf{s}(z))}\,(L\forall)}{y = \mathbf{s}(z), Nz, \varphi(0), \forall x.\varphi x \to \varphi(\mathbf{s}(x)) \Rightarrow \varphi(y)}\,(L=)}{Ny, \varphi(0), \forall x.\varphi x \to \varphi(\mathbf{s}(x)) \Rightarrow \varphi(y)}\,(LN)$$

With some substitution, weakening and contraction, the open node is equivalent to the root node, hence this is a cyclic proof of $Ny, \varphi(0), \forall x.\varphi x \to \varphi(\mathbf{s}(x)) \Rightarrow \varphi(y)$.

1.4.1 The N predicate

Introducing inductive predicates in general, we defined the left and right rules for the atomic predicate N of ICA:

$$\frac{}{\Gamma \Rightarrow N0}\,(RN_0) \qquad \frac{\Gamma \Rightarrow Nt}{\Gamma \Rightarrow N\mathbf{s}(t)}\,(RN_1) \qquad \frac{t = 0, \Gamma \Rightarrow \chi \qquad t = \mathbf{s}(r), Nr, \Delta \Rightarrow \chi}{Nt, \Gamma, \Delta \Rightarrow \chi}\,(LN)$$

Before introducing the corresponding λ-terms in Σ, let's look at the type of N. A definition of the property of being a natural number of a term t can be given as $Nt := t = 0 \vee \exists y.t = \mathbf{s}(y) \wedge Ny$. Such a formula, translated directly into a type, results in

$$\omega = \iota \to (\epsilon + (\iota \times (\epsilon \times \omega)))$$

Building a term for Nt following the assignment already given would generate terms $\kappa_0\langle\rangle : N0$ and $\kappa_1(\langle t, \langle\langle\rangle, \alpha\rangle\rangle : N\mathbf{s}(t)$, with α being itself a term of type ω. As much as this is a faithful representation of the meaning of Nt, it is not reflecting the step indicated by the rules in our system (and it is also not optimal for its verbosity). Notice that from a literal reading of the definition we could have defined rules like

$$\frac{\Gamma \Rightarrow t = 0 \vee \exists y.t = \mathbf{s}(y) \wedge Ny}{\Gamma \Rightarrow Nt}\,(RN^*) \qquad \frac{t = 0 \vee \exists y.t = \mathbf{s}(y) \wedge Ny, \Gamma \Rightarrow \chi}{Nt, \Gamma \Rightarrow \chi}\,(LN^*)$$

while the rules of ICA are a condensed version of them. From the point of view of derivability, to change $(RN_i)/(LN)$ with $(RN^*)/(LN^*)$ does not make a difference, for example

Example 1.16.

$$\dfrac{\dfrac{t = 0 \Rightarrow \chi \qquad \dfrac{\dfrac{t = \mathbf{s}(y), Ny \Rightarrow \chi}{t = \mathbf{s}(y) \wedge Ny \Rightarrow \chi}}{\exists y.t = \mathbf{s}(y) \wedge Ny \Rightarrow \chi}}{t = 0 \vee \exists y.t = \mathbf{s}(y) \wedge Ny \Rightarrow \chi}}{Nt \Rightarrow \chi}\,(LN^*)$$

If we were to take the new rules we could define the type ω as the fixed point of the definition of natural number, with the rules being just the folding of the definition. While an additional positive side of those rules would be the perfect symmetry between left and right rules, we find ourselves with a system that excessively lingers on each step: we would have to undergo many troubles in dealing with them, since we would be bound with disjunctions and existential quantification to unravel each time, steps that at this point not necessarily occur right above the $(L/R\text{-}N^*)$ rule. It seems that the best solution is to take the original ICA rules, and accept a non-completely matching situation in terms of syntax between the typed terms and the rule. In the end, we are interested into collecting just some of the information available. We concisely define the type $\omega := \epsilon + (\iota \times \omega)$ and introduce two functional terms $f_0^N : \epsilon \to \omega$ and $f_1^N : (\iota \times \omega) \to \omega$ such that

Type	Terms	Constructor	Destructor
ω	$f_N^0 : \epsilon \to \omega$ $f_N^1 : (\iota \times \omega) \to \omega$	$f_N^0(\langle\rangle) : \omega$ $f_N^1(t,p) : \omega$	$Case_{q,(tp)}^N(f_N^i r, n_q^\rho, n_{tp}^\rho) : \rho$

where $Case_{q,(tp)}^N(f_N^i r, n_q^\rho, n_{tp}^\rho)$ has the same behaviour of the disjunctive case, with the difference that if $r \equiv q$ then the witness is n_q from the left branch, n_{tp} from the right branch if $r \equiv (tp)$. Note that the disjunctive step was inevitable also in the definitional extended approach. The assignment defines the following rules:

$$\frac{}{\Rightarrow \mathbf{f_N^0}(\langle\rangle) : N0}\ (RN_0\ ) \qquad \frac{\Rightarrow \mathbf{p} : Nt}{\Rightarrow \mathbf{f_N^1}(\mathbf{t},\mathbf{p}) : N\mathbf{s}(t)}\ (RN_1\ )$$

$$\frac{\mathbf{y} : \epsilon \Rightarrow \mathbf{p_0} : \varphi \qquad \mathbf{v} : \epsilon, \mathbf{z} : Nt \Rightarrow \mathbf{p_1} : \varphi}{\mathbf{w} : N\mathbf{s}(t) \Rightarrow \mathrm{Case}_{\mathbf{y},\mathbf{z}}^{\mathbf{N}}(\mathbf{w}, \mathbf{p_0}, \mathbf{p_1}) : \varphi}\ (LN)$$

Now that we have given a type to all formulas of $\mathcal{L}^=$, including the natural number predicate, it is possible to define the order of each type

Definition 1.17 (Order)**.** *The order of a type ρ is defined as follows:*

- $ord(\iota) = ord(\epsilon) = 0$
- $ord(\rho \to \sigma) = \max\{ord(\rho)+1, ord(\sigma)\}$
- $ord(\rho \times \sigma) = \max\{ord(\rho), ord(\sigma)\}$
- $ord(\rho + \sigma) = \max\{ord(\rho), ord(\sigma)\}$
- $ord(\varsigma) = 0$
- $ord(\omega) = 0$

The choice of 0 as the order for ι and ϵ is taken from [AHL20], where it is motivated by technicalities. We follow that choice, and assign the same value to the order of ω: if we consider it as being defined by $\omega := \epsilon + (\iota \times \omega)$ we have :

$$ord(Nt) = \max\{0, \max\{0, ord([Ny])\}\}$$

assuming $ord(N0) = 0$ for the base case we have

$$ord(\omega) = 0$$

1.5 Cyclic terms

The Σ-terms represent faithfully the initial part of a cyclic proof, up to the bud nodes. To be able to introduce cycles we need a new term in the language of λ-calculus. Not surprisingly, it is possible to represent iterations through fixed points. A series of new terms Y is then introduced into the language Σ to be the fixed point combinators, with type $(\rho \to \rho) \to \rho$ for all basic types ρ. The system resulting from the expansion of the simply typed λ-calculus with such combinators is the λY-calculus, where the reduction

$$Yp \to p(Yp)$$

defines the behaviour of Y. We add to Definition 1.13 also the combinator Y, obtaining finally a complete definition of the language Σ and the corresponding typing:

Definition 1.18 (Σ terms and types)**.** *The following table defines the set of Σ-terms and their type*

Type	***Terms***	***Constructor***	***Deconstructor***
ι	$0 : \iota \quad \mathbf{s}() : \iota \to \iota \quad a : \iota$	$\mathbf{s}(t) : \iota$	
	$(+) : \iota \to \iota \to \iota$	$t + s : \iota$	
	$(\cdot) : \iota \to \iota \to \iota$	$t \cdot s : \iota$	
ϵ	$\langle\rangle : \epsilon$		
$\perp$	$*_\psi : \perp \to [\psi]$		$*_\psi\, t : \psi$
$\sigma \to \tau$		$\lambda p^\sigma . r^\tau : \sigma \to \tau$	$(q^{\sigma \to \tau} p^\sigma) : \tau$
$\sigma \times \tau$		$\langle p^\sigma, r^\tau \rangle : \sigma \times \tau$	$\pi_0 q^{\sigma\times\tau} : \sigma \quad \pi_1 q^{\sigma\times\tau} : \tau$
$\sigma + \tau$		$\kappa_0 p^\sigma : \sigma + \tau \quad \kappa_1 r^\tau : \sigma + \tau$	$Case^{\vee}_{p,r}(k_i q, n^\rho_p, n^\rho_r) : \rho$
ω	$f^0_N : \epsilon \to \omega$	$f^0_N(\langle\rangle) : \omega$	
	$f^1_N : (\iota \times \omega) \to \omega$	$f^1_N(t, p) : \omega$	$Case^N_{q,(tp)}(f^i_N r, n^\rho_q, n^\rho_{tp}) : \rho$
ρ *basic*	$Y : (\rho \to \rho) \to \rho$	$Yp : \rho$	
ς		$[t \mapsto a]\alpha : \varsigma$	
ρ *basic*		$p \circ \alpha : \rho$	

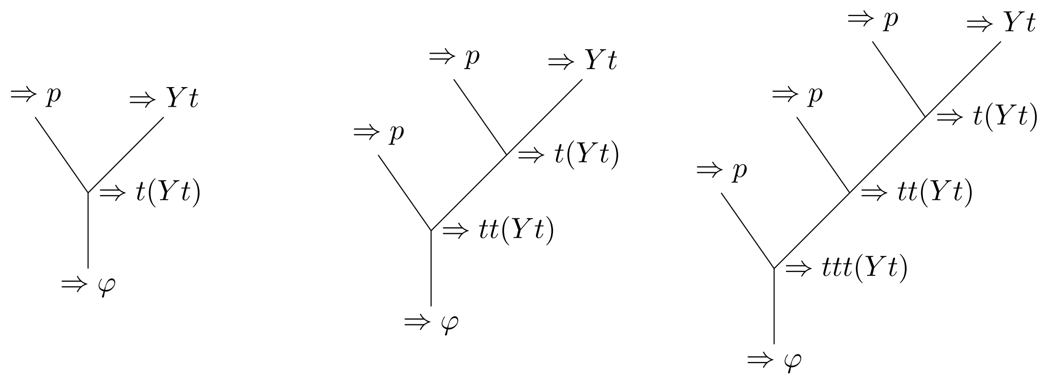
The Y combinator makes it possible to represent cyclic proofs as terms. For every cyclic proof π, there is a series of pairs of nodes according to the bud-companion relation $\mathtt{R}$. So far no Σ-term has been assigned to the formulas at bud nodes. We stipulate now that to the right-hand side of each bud sequent $\Gamma \Rightarrow \psi$ is assigned a fresh term variable z of the appropriate type $[\psi]$, that provisionally represents the term resulting from the process that occurred in a previous cycle above. The rest of the leaves have terms assigned according to the rules for axioms. Proceeding to the definition of λ-terms with the rules of our term assignment, whenever the companion node is reached we find a term t with z open variable inside. Notice, in fact, that z is never deleted as an effect of some rule, nor changed, given that it is fresh and so it doesn't appear on the left-hand side. Hence z is a variable occurring free in t. We also know that z and t are two witnesses for the same proof of ψ, so they not only have the same context by definition, but most importantly they are of the same type $[\psi]$. They are two terms representing the same sequent, hence we could in principle substitute one for the other: $z = t(z)$. The solution of the equivalence for z is

$$z = Y(\lambda x.t[x/z])$$

from which we have that

$$Y(\lambda x.t(x)) \to t(Y(\lambda x.t(x))) \to t(t(Y(\lambda x.t(x)))) \to \dots$$

The resulting term assignment is then $Y(\lambda x.t(x))$ for the bud node and $t(Y(\lambda x.t(x)))$ for the companion. Whenever we reach a bud node from below, the fixed point combinator Y generates a new term corresponding to the companion node, producing another cycle of the proof. In other words: by unfolding the fixed point we jump from the bud to the companion node in the path along the tree. The next figure gives an intuition of the correspondence with cyclic proofs:



As a result we have defined a set of rules that allow us to build terms in the λY-calculus, terms that represent the corresponding proofs in the cyclic system ICA. Whenever a potentially infinite computation is expressed by a finite deduction tree π according to the rules of ICA, we have a method to express such a computation with a single term in the typed λY-calculus. In the last section of this chapter we introduce the definition of recursion schemes: a term rewriting system in which we can compute terms of our language in an direct formalism.

1.6 Recursion Schemes

The final part of this chapter defines recursion schemes. The main reason for introducing the notation via λ-terms is that each intuitionistic proof is also a prescription, a function that is able to determine the components of the proof from a given input. For a proof with $\varphi \Rightarrow \psi$ at the root, the corresponding term assignment $x \Rightarrow p$ tells us that $\lambda x.p$ is the function that, for every input of type φ, the computation expressed by p will result in a proof of ψ. The process of computation follows the rules of β-reduction, that is a series of substitutions of equivalent subterms inside the main term, until no additional substitution is possible. In the case of recursion schemes the process is similar. We start from a list of equivalences, as in the case of β, with the difference that we don't have one unique kind like $(\lambda x.m)n = m[n/x]$ but a finite list of the form

$$\begin{aligned} n_0 n_1 \dots n_k &= p \\ n'_0 n'_1 \dots n'_{k'} &= p' \\ n''_0 n''_1 \dots n''_{k''} &= p'' \\ \dots\dots\dots \end{aligned}$$

By taking an oriented reading of the equivalences from left to right, and assigning one functional term called *non-terminal* to each one of them, the result is a series of rewriting rules that defines the computation. The easiest way of seeing the relation with λ-calculus is to take the closed terms S, K, I and express their β-reductions as rewrite rules:

$$\begin{aligned} Kab &\to a \\ Sabc &\to ac(bc) \\ Ia &\to a \end{aligned}$$

Recursion schemes have the same expressive power of λ-terms and even more, including the ability to mimic the Y combinator thanks to internal cycles.[10]

We give a general presentation of recursion schemes, referring to the next chapter for a detailed definition of schemes tailored for our necessity. A higher-order recursion scheme is given by an alphabet Λ of typed terms, a set of functions $\mathcal{F}_i$ called non-terminals and a set $\mathcal{R}$ of oriented equations $\mathcal{F}\overline{x} = p$ assuming the role of rewrite rules.

Definition 1.19 (Higher-Order Recursion Scheme)**.** *A HORS is a tuple* $\mathcal{H} = \langle \Lambda, \mathtt{F}, \mathcal{F}_\perp, \mathcal{R} \rangle$ *where*

- Λ *is a typed alphabet*
- $\mathtt{F}$ *is a set of non-terminals* $\mathcal{F}_i$ *each one of a given arity*
- $\mathcal{F}_\perp \in \mathtt{F}$ *is a starting symbol of ground type*
- $\mathcal{R}$ *is a set of production rules of the form*

$$\mathcal{F}_j x_0 \dots x_k \to p$$

one for each non-terminal $\mathcal{F}_j \in \mathtt{F}$*, with* p *a term of* $\Lambda \cup (\mathtt{F} \backslash \mathcal{F}_\perp) \cup \{x_0 \dots x_k\}$*.*

The set of $\mathcal{H}$-terms is given by the terms from Λ together with the non-terminals in $\mathtt{F}$. The language Λ is a typed language, and nonterminals are also typed: each $\mathcal{F}_j$ whose rule of production in $\mathcal{H}$ is $\mathcal{F}_j x_0 \dots x_k \to p$ has the type

$$[x_0] \to [x_1] \to \dots \to [p]$$

[10] See [SW12] for a direct translation from λY terms to HORS and vice versa.

As a consequence each $\mathcal{F}_i \in \mathtt{F}$ has an order according to Definition 1.17, and the rewrite system $\mathcal{H}$ has an order corresponding to the supremum of the orders of its non-terminals. Similarly to the β-reduction in λ-calculus, it is possible to define a relation of reduction $\to_{\mathcal{R}}$ (together with its reflexive transitive closure $\twoheadrightarrow_{\mathcal{R}}$) on $\mathcal{H}$-terms:

Definition 1.20 ($\to_{\mathcal{R}}, \twoheadrightarrow_{\mathcal{R}}$)**.** *Given a set of rewrite rules $\mathcal{R}$ and two $\mathcal{H}$-terms p, q, we say that $p \to_{\mathcal{R}} q$:*

1. *if $p \equiv \mathcal{F} p_0 \ldots p_n$ and there is a rule $\mathcal{F}\overline{x}_0 \ldots x_n \to q \in \mathcal{R}$ then $p \to_{\mathcal{R}} q[p_0/x_0, \ldots, p_n/x_n]$*
2. *if $p \to_{\mathcal{R}} q$ then $r(p) \to_{\mathcal{R}} r(q)$*

$\twoheadrightarrow_{\mathcal{R}}$ is the reflexive transitive closure of $\to_{\mathcal{R}}$.

Alternative systems with respect to the ones of Definition 1.19 can be defined where determinism is not requested, hence multiple rules might start with the same non-terminal, and so different computations can result from the same input. Another possible feature is the introduction of *pattern matching*. It consists in the possibility to specify the structure of some of the input-terms taken by the non-terminal, and subordinate the activation of the rule to the presence of an input of the given form. There is no direct relation between pattern-matching and non-determinism: a system can have one without the other, as we will see in the next chapter. We will make use of pattern-matching for our recursion schemes.

In the next chapter we will translate each sequent from a proof into a typed functional term, with the left-hand side as input and the right-term as output, similarly to what we did with $\lambda x.p$ above. A sequent like $\overline{x} : \Gamma \Rightarrow p : \psi$, for example, will become a term $F\overline{x} = p$. The way in which the reduction rules will be defined will make the computation climb the proof, building step-by-step the desired term-witness.

Chapter 2

A recursion scheme for ICA

In the introduction we stated that the goal is a process that is able to extract the computational content from an intuitionistic cyclic proof of an arithmetical statement. The rewrite process has the form of a higher-order recursion scheme as defined above. We will make use of the structure of the non-terminals and the possibility given by cycles in the recursion scheme to reduce the number of symbols necessary with respect to the language Σ above. We can exploit the function type of non-terminals to avoid using λ-abstraction: to a term of the form $\lambda p.\psi$ corresponds a rule $\mathcal{F}$ with type $\sigma_p \to \rho_\psi$. We still need the converse of abstraction, i.e. application. A similar consideration can be made for the Y combinator, since the unfolding of the fixed point can also be simulated by a series of rewrite rules. We also drop all the de-constructors (except application, of course) because the decomposition of complex terms is delegated to rules with pattern matching. Let's start by introducing the term language Σ^H specific for our recursion schemes, and then proceed to define the higher-order recursion scheme for cyclic arithmetic.

Definition 2.1 (Σ^H-Terms)**.** *The following definition sets the list of terms of Σ^H and their type:*

Type	***Terms***	***Constructor***	***Deconstructor***
ι	$0 : \iota \quad \mathbf{s}() : \iota \to \iota \quad a : \iota$	$\mathbf{s}(t) : \iota$	
	$(+) : \iota \to \iota \to \iota \quad (\cdot) : \iota \to \iota \to \iota$	$t + s : \iota \quad t \cdot s : \iota$	
ϵ	$\langle\rangle : \epsilon$		
$\bot$	$*_\psi : \bot \to [\psi]$		$*_\psi\, t : \psi$
$\sigma \to \tau$			$(qp) : \tau$
$\sigma \times \tau$		$\langle p, r\rangle : \sigma \times \tau$	
$\sigma + \tau$		$\kappa_0 p : \sigma + \tau \quad \kappa_1 r : \sigma + \tau$	
ω	$f_0^N : \epsilon \to \omega \quad f_1^N : (\iota \times \omega) \to \omega$	$f_0^N(\langle\rangle) : \omega \quad f_1^N(t, p) : \omega$	
ς		$[t \mapsto a]\alpha : \varsigma$	
ρ *basic*		$p \circ \alpha : \rho$	

Together with Σ^H-terms, the $\mathcal{H}$-language is completed by a set $\mathtt{F}$ of nonterminals.

Definition 2.2 ($\mathcal{H}^\pi$)**.** *Given an ICA proof π, $\mathcal{H}^\pi$ is a higher-order recursion scheme $\langle \Sigma, \mathtt{F}, \mathcal{F}_\perp, \mathcal{R} \rangle$ such that*

- *Σ is the typed alphabet Σ^H of Definition 2.1*
- *$\mathtt{F}$ is a set of non-terminals $\mathcal{F}_i$, one for each occurrence of a rule in π*[1]
- *$\mathcal{F}_\perp : \epsilon \in \mathtt{F}$ is the starting symbol*
- *$\mathcal{R}$ is a set of production rules of the form*

$$\mathcal{F}_j \alpha x_0 \dots x_k \to p$$

$\mathcal{F}_j \in \mathtt{F}$ is of type $\varsigma \to \sigma_{x_0} \to \dots \to \sigma_{x_k} \to [p]$, where $p \in \mathcal{H} \backslash \mathcal{F}_\perp$. Depending on the last rule R of each subproof π, we add to the list $\mathcal{R}$ a production rule $\mathcal{F}_\pi \alpha x_0 \dots x_k \to p$ as determined by the next table

In the table below $\mathcal{F}_\pi$ is the non-terminal corresponding to the last sequent of the proof π. By notational convention: when the rule is a one-premise, π_0 is the subproof above the last rule R; in case R is a two-premise rule, the two subproofs are called π_0 and π_1 as in the following scheme [2]

$$\frac{\Gamma' \Rightarrow_{\pi_0} \varphi'}{\Gamma \Rightarrow_\pi \varphi} \qquad \frac{\Gamma' \Rightarrow_{\pi_0} \varphi' \qquad \Gamma'' \Rightarrow_{\pi_1} \varphi''}{\Gamma', \Gamma'' \Rightarrow_\pi \varphi}$$

$$\frac{\overline{x}' \Rightarrow_{\pi_0} p'}{\overline{x} \Rightarrow_\pi p} \qquad \frac{\overline{x} \Rightarrow_{\pi_0} r \qquad \overline{y} \Rightarrow_{\pi_1} q}{\overline{x}, \overline{y} \Rightarrow_\pi p}$$

We also stipulate that $[\Gamma] = \overline{\gamma}$, while y is the term of the type of the principal formula when the rule has not pattern-matching. α is a substitution stack (possibly empty). Finally: we introduce one rule for each bud-node such that the subproof π above the bud-node is identified with the subproof π' above the companion node.

[1] In the case of $(L\vee)$ and (LN) we have two formally, see considerations on determinism below.

[2] The order of the premises is not fixed, but we take the order from left to right with respect to the graphical representation given in the first chapter.

R	Type of $\mathcal{F}_\pi$	Production Rule
(Ax)	$\varsigma \to \epsilon \to \bar{\gamma} \to \epsilon$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow y$
$(L\bot)$	$\varsigma \to \bot \to \bar{\gamma} \to \epsilon$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow *_\epsilon\, y$
(id)	$\varsigma \to \bar{\gamma} \to \epsilon$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \langle\rangle$
$(L =)$	$\varsigma \to \epsilon \to \bar{\gamma} \to \varphi$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow \mathcal{F}_{\pi_0} \alpha \overline{x}$
$(L \to)$	$\varsigma \to (\varphi \to \psi) \to \bar{\gamma} \to \chi$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow \mathcal{F}_{\pi_1} \alpha (y \mathcal{F}_{\pi_0} \alpha \overline{x}_0) \overline{x}_1$
$(R \to)$	$\varsigma \to \bar{\gamma} \to \varphi \to \psi$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \mathcal{F}_{\pi_0} \alpha \overline{x}$
$(L\wedge)$	$\varsigma \to (\varphi \times \psi) \to \bar{\gamma} \to \chi$	$\mathcal{F}_\pi \alpha \langle y, z\rangle \overline{x} \longrightarrow \mathcal{F}_{\pi_0} \alpha y z \overline{x}$
$(R\wedge)$	$\varsigma \to \bar{\gamma} \to (\varphi \times \psi)$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \langle \mathcal{F}_{\pi_0} \alpha \overline{x_0}, \mathcal{F}_{\pi_1} \alpha \overline{x_1} \rangle$
$(L\vee)$	$\varsigma \to (\psi_0 + \psi_1) \to \bar{\gamma} \to \chi$	$\mathcal{F}_\pi \alpha (k_i y) \overline{x} \longrightarrow \mathcal{F}_{\pi_i} \alpha y \overline{x}_i$
$(R\vee)$	$\varsigma \to \bar{\gamma} \to (\psi_0 + \psi_1)$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow k_i \mathcal{F}_{\pi_0} \alpha \overline{x}$
$(L\exists)$	$\varsigma \to (\iota \times \varphi) \to \bar{\gamma} \to \chi$	$\mathcal{F}_\pi \alpha \langle t, y\rangle \overline{x} \longrightarrow \mathcal{F}_{\pi_0} [t \mapsto a] \alpha y \overline{x}$
$(R\exists)$	$\varsigma \to \bar{\gamma} \to (\iota \times \varphi)$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \langle t \circ \alpha, \mathcal{F}_{\pi_0} \alpha \overline{x} \rangle$
$(L\forall)$	$\varsigma \to (\iota \to \varphi) \to \bar{\gamma} \to \chi$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow \mathcal{F}_{\pi_0} \alpha (y t) \overline{x}$
$(R\forall)$	$\varsigma \to \bar{\gamma} \to (\iota \to \varphi)$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \mathcal{F}_{\pi_0} \alpha \overline{x}$
(RN_0)	$\varsigma \to \bar{\gamma} \to \omega$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow f^0_N(\langle\rangle)$
(RN_1)	$\varsigma \to \bar{\gamma} \to \omega$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow f^1_N(t \circ \alpha, \mathcal{F}_{\pi_0} \alpha \overline{x})$
(LN)	$\varsigma \to \omega \to \bar{\gamma} \to \chi$	$\mathcal{F}_\pi \alpha f^0_N(\langle\rangle) \overline{x} \longrightarrow \mathcal{F}_{\pi_0} \alpha \langle\rangle \overline{x}$
(LN)	$\varsigma \to \omega \to \bar{\gamma} \to \chi$	$\mathcal{F}_\pi \alpha f^1_N(t, p) \overline{x} \longrightarrow \mathcal{F}_{\pi_1} [s \mapsto a] \alpha \langle\rangle p \overline{x}$
(cut)	$\varsigma \to \bar{\gamma} \to \chi$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \mathcal{F}_{\pi_1} \alpha (\mathcal{F}_{\pi_0} \alpha \overline{x_0}) \overline{x_1}$
(W)	$\varsigma \to \varphi \to \bar{\gamma} \to \chi$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow \mathcal{F}_{\pi_0} \alpha \overline{x}$
(C)	$\varsigma \to \varphi \to \bar{\gamma} \to \chi$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow \mathcal{F}_{\pi_0} \alpha y y \overline{x}$
(Sub)	$\varsigma \to \bar{\gamma} \to \chi$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \mathcal{F}_{\pi_0} [\theta] \alpha \overline{x}$
$(\bot 0)$	$\varsigma \to \bar{\gamma} \to \epsilon \to \bot$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow *_\bot\, y$
$(+0)$	$\varsigma \to \bar{\gamma} \to \epsilon$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \langle\rangle$
$(+\mathbf{s})$	$\varsigma \to \bar{\gamma} \to \epsilon$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \langle\rangle$
$(\cdot 0)$	$\varsigma \to \bar{\gamma} \to \epsilon$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \langle\rangle$
$(\cdot \mathbf{s})$	$\varsigma \to \bar{\gamma} \to \epsilon$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \langle\rangle$
$(= \mathbf{s})$	$\varsigma \to \epsilon \to \bar{\gamma} \to \epsilon$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow y$
$Start$	ϵ	$\mathcal{F}_\bot \longrightarrow \mathcal{F}_\pi$
$Cycle$	ϵ	$\mathcal{F}_\pi \longrightarrow \mathcal{F}_{\pi'}$

We proceed with a closer look at the rules and give some explanation.

Axioms and equality

(Ax)	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow y$	$(L\bot)$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow *_\epsilon y$
(id)	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \langle\rangle$	$(L =)$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow \mathcal{F}_{\pi_0} \alpha \overline{x}$
$(\bot 0)$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow *_\bot y$	$(\star *)$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \langle\rangle$
$(= \mathbf{s})$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow y$		

where $\star \in \{+, \cdot\}$ and $* \in \{0, \mathbf{s}\}$. Whenever the axiom is independent from the context, it returns directly $\langle\rangle$ or $f_N^0\langle\rangle$. In the remaining cases, the input term from which the right-hand side depends, and that is witnessing the axiom, is returned, while the rest of the input is deleted since there is nothing left to consider. In the case of $(L =)$ the rule removes the input term of type ϵ, the justification being that the substitution occurs between two terms whose equivalence is an arithmetical fact and not a logical one, hence from our perspective there is no change in the computational content. Finally: in the cases of $(L\bot)$ and $(\bot 0)$ the term obtained is a function from an input of type $\bot$ to an object of type ϵ or $\bot$ respectively.

Implication

$(L \to)$	$\mathcal{F}_\pi \alpha y \overline{x} \longrightarrow \mathcal{F}_{\pi_1} \alpha (y \mathcal{F}_{\pi_0} \alpha \overline{x_0}) \overline{x_1}$	$(R \to)$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \mathcal{F}_{\pi_0} \alpha \overline{x}$

$(L \to)$ is branching, so its rewrite rule takes the input $y : \varphi \to \psi$ and applies to it the subproof of the branch with type $\mathcal{F}_{\pi_0} \alpha \overline{x_0} : \varphi$. The result is that the type of the right branch $\mathcal{F}_{\pi_1}$ is matched: $\varsigma \to \overline{\gamma} \to \psi \to \chi$. Notice that, despite the fact that the proof splits into two branches, we still have just one term with two non-terminals. The term obtained by $(R \to)$ has the same function type as before: we have $\mathcal{F}_{\pi_0} : \varsigma \to \overline{\gamma} \to (\varphi \to \psi)$ hence $\mathcal{F}_{\pi_0} \alpha \overline{x} : \varphi \to \psi$ with an argument for φ that needs to be given as input to produce a term of type ψ. The motivation for this choice will become clear when cut reduction is analysed.

Conjunction and disjunction

$(L\wedge)$	$\mathcal{F}_\pi \alpha \langle y, z\rangle \overline{x} \longrightarrow \mathcal{F}_{\pi_0} \alpha y z \overline{x}$	$(R\wedge)$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow \langle \mathcal{F}_{\pi_0} \alpha \overline{x_0}, \mathcal{F}_{\pi_1} \alpha \overline{x_1} \rangle$
$(L\vee)$	$\mathcal{F}_\pi \alpha (k_i y) \overline{x} \longrightarrow \mathcal{F}_{\pi_i} \alpha y \overline{x_i}$	$(R\vee)$	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow k_i \mathcal{F}_{\pi_0} \alpha \overline{x}$

There is not much to say about these logical rules, except pointing out how the left rules have pattern-matching, de-constructing the term of a specific form. The right rules, on the other hand, move the computation inside, re-constructing the term outside of the non-terminal. Note that in the disjunction case we are talking about two distinct rules with pattern matching: depending on the value of $i \in \{0, 1\}$ we have two different left production rules. It is a good point here to consider what happens when pattern matching fails. From the point of view of the computation, if instead of $k_i p$ we have an unspecified term r of sum type, it means that we still need some additional information as input in order to proceed with the reduction. Maybe r is waiting to be constructed by another non-terminal: the intended purpose of pattern matching is precisely to stop the computation until more detailed data is available. When the term is constructed and the information is complete, the specific content of the input forces a decision about the correct branch. However, since we are giving a general definition here and we are working with meta-variables, it is sometimes useful to act *as if* we had a specific term, but considering both possibilities. In that case, we will write, for example

$$\mathcal{F}_{\pi}\alpha(k_i y)\overline{x} \longrightarrow \mathcal{F}_{\pi_0}\alpha y\overline{x_0} \mid \mathcal{F}_{\pi_1}\alpha y\overline{x_1}$$

to be able to follow both branches.

Quantifiers

$$\begin{array}{llll} (L\exists) & \mathcal{F}_{\pi}\alpha\langle t, y\rangle\bar{x} \longrightarrow \mathcal{F}_{\pi_0}[t \mapsto a]\alpha y\bar{x} & (R\exists) & \mathcal{F}_{\pi}\alpha\overline{x} \longrightarrow \langle t \circ \alpha, \mathcal{F}_{\pi_0}\alpha\overline{x}\rangle \\ (L\forall) & \mathcal{F}_{\pi}\alpha y\overline{x} \longrightarrow \mathcal{F}_{\pi_1}\alpha(yt)\overline{x} & (R\forall) & \mathcal{F}_{\pi}\alpha\overline{x} \longrightarrow \mathcal{F}_{\pi_0}\alpha\overline{x} \end{array}$$

As in the case of implication, we don't need to use the λ-abstractor to work with $\forall$, it is sufficient to take advantage of the functional type of the non-terminal. The left rule behaves like in the case $(L \rightarrow)$, with the difference that t is an individual term, while the right rule leaves everything unchanged and gives a term that waits for an object of type ι in accordance with the dependent type of open formulas. Existential quantifier is worth a closer look, since it presents the most peculiar behaviour. Similarly to the conjunctive case the right rule produces a pair of objects. In this case, the first element is an individual term t, together with the stack of substitutions α that has been produced at that moment. The choice for the stack to be copied together with the term comes from the fact that any future substitution results from a rule above in the proof, hence doesn't involves the term. The expansion of the substitution stack results from left existential rule, where the pattern-matching demands an input of the right form, and an individual term can be extracted from the pair. The substitution for the eigenvariable a is inserted on top of the stack. That is because a substitution $[t \mapsto a]$ informally corresponds to an instantiation of the eigenvariable a with the term t in the rest of the above subproof.[3] The following example clarifies the process

$$\begin{array}{ccc} \pi & \mathcal{D}[t \mapsto a] & \pi[t/a] \\ \vdots & \vdots & \vdots \\ \dfrac{\psi(a) \Rightarrow \chi}{\exists x.\psi(x) \Rightarrow \chi} & \dfrac{w[t \mapsto a] \Rightarrow p}{\langle t, w\rangle \Rightarrow p} & \dfrac{\psi(a)[t/a] \Rightarrow \chi}{\exists x.\psi(x) \Rightarrow \chi} \\ (1) & (2) & (3) \end{array}$$

[3]See Lemma 2.8 below.

If we consider a pair $\langle t, w\rangle$ as witnessing the existential claim, we have already a candidate for the eigenvariable a in (1). On the other hand, the reduction proceeds ignoring the information, so in the recursion scheme the term $w : [A(a)]$ is considered. The rule adds the substitution to the stack and continues with the subproof, that is what happens in the case (3). Unsurprisingly, left and right existential rules are thought to be complementary. An example is the following

$$\dfrac{\dfrac{\Rightarrow_B \varphi(t)}{\Rightarrow_A \exists y.\varphi(y)} \qquad \dfrac{\dfrac{\varphi(a) \Rightarrow_4 \psi(\mathbf{s}(a))}{\varphi(a) \Rightarrow_3 \exists y.\psi(y)}}{\exists y.\varphi(y) \Rightarrow_2 \exists y.\psi(y)}}{\Rightarrow_1 \exists y.\psi(y)}$$

A suitable candidate to witness the existential claim, hence to instantiate the variable a, can be found on the left branch of the proof via a cut. The rewriting process is then

$$\begin{aligned}
\mathcal{F}_\perp &\longrightarrow \mathcal{F}_1 \\
&\longrightarrow \mathcal{F}_2(\mathcal{F}_A) \\
&\longrightarrow \mathcal{F}_2\langle t, \mathcal{F}_B\rangle \\
&\longrightarrow \mathcal{F}_3[t \mapsto a]\mathcal{F}_B \\
&\longrightarrow \langle \mathbf{s}(a) \circ [t \mapsto a], \mathcal{F}_4[t \mapsto a]\mathcal{F}_B\rangle \\
&= \langle \mathbf{s}(t), \mathcal{F}_4[t \mapsto a]\mathcal{F}_B\rangle
\end{aligned}$$

The final term is a pair with the individual term $\mathbf{s}(t)$ and $\mathcal{F}_B : \varphi(a)$ witnessing $\exists y.\varphi(y)$.

N predicate

(RN_0)	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow f_N^0(\langle\rangle)$	(RN_1)	$\mathcal{F}_\pi \alpha \overline{x} \longrightarrow f_N^1(t \circ \alpha, \mathcal{F}_{\pi_0} \alpha \overline{x})$
(LN)	$\mathcal{F}_\pi \alpha f_N^0(\langle\rangle) \overline{x} \longrightarrow \mathcal{F}_{\pi_0} \alpha \langle\rangle \overline{x}$	(LN)	$\mathcal{F}_\pi \alpha f_N^1(t, p) \overline{x} \longrightarrow \mathcal{F}_{\pi_1} \alpha \langle\rangle p \overline{x}$

As already seen, the right rules push the computation inside and build the term outside, preserving the final type. Clearly (RN_0) does not introduce a new non-terminal, since the computation ends for 0; (RN_1) constructs a term of type $N\mathbf{s}(t)$ by giving as argument to f_N^1 the predecessor t together with a proof of Nt. Once again, this is intended to match the left rules. Depending on the presence of f_N^0 or f_N^1 as input we have two pattern-matching rules, like in the case of disjunction. As we pointed out in Section 1.4.1, what these rules are doing is condensing a series of implicit steps corresponding to the unravelling of the definition of N. In the first scenario, (LN) substitutes Nt with its justification $t = 0$, hence to the term $f_0^N\langle\rangle$ of type $N0$ corresponds a term $\langle\rangle : \epsilon$. In the second case, the individual term t is forgotten, because the focus shifted onto Nt. An extra term $\langle\rangle$ is necessary to match the additional assumption $t = \mathbf{s}(r)$. Pattern-matching is again essential to determine the correct branch of the derivation. As in the previous case of $\exists$, consider the following derivation in which after a (LN) rule, the formula Nt is cut:

$$R\ \dfrac{\dfrac{\Gamma \Rightarrow_{00} (?)}{\Gamma \Rightarrow_0 Nt} \qquad \dfrac{t = 0, \Delta' \Rightarrow_{10} \chi \qquad t = \mathbf{s}(r), Nr, \Delta'' \Rightarrow_{11} \chi}{Nt, \Delta \Rightarrow_1 \chi}}{\Gamma, \Delta \Rightarrow \chi}$$

The subproof on the left gives a construction for Nt to be used as premise in the right branch. Depending on the value of t, the possible computations are the following

$$\begin{array}{llllllll}
\mathcal{F}_{\perp}\overline{xy} & \longrightarrow & & & \mathcal{F}\overline{xy} & & & \\
 & \longrightarrow & & & \mathcal{F}_1(\mathcal{F}_0\overline{y})\overline{x} & (cut) & & \\
 & \longrightarrow & \mathcal{F}_1 f_N^0(\langle\rangle)\overline{x} & (RN_0) & \mid & & \mathcal{F}_1 f_N^1(r, \mathcal{F}_{00}\overline{y})\overline{x} & (RN_1) \\
 & \longrightarrow & \mathcal{F}_{10}\langle\rangle\overline{x_0} & (LN) & \mid & & \mathcal{F}_{11}\langle\rangle(\mathcal{F}_{00}\overline{y})\overline{x_1} & (LN)
\end{array}$$

Structural rules

$$\begin{array}{llll}
(cut) & \mathcal{F}_\pi\alpha\overline{x} \longrightarrow \mathcal{F}_{\pi_1}\alpha(\mathcal{F}_{\pi_0}\alpha\overline{x_0})\overline{x_1} & (W) & \mathcal{F}_\pi\alpha y\overline{x} \longrightarrow \mathcal{F}_{\pi_0}\alpha\overline{x} \\
(C) & \mathcal{F}_\pi\alpha y\overline{x} \longrightarrow \mathcal{F}_{\pi_0}\alpha yy\overline{x} & (Sub) & \mathcal{F}_\pi\alpha\overline{x} \longrightarrow \mathcal{F}_{\pi_0}[\theta]\alpha\overline{x}
\end{array}$$

Weakening and contraction can be viewed as simple adjustments of the premises, both affecting the left-hand side of the sequent. Notice that contraction is the only rule that duplicates input-terms. The substitution rule is reported in the stack, for the same motivation given in the existential case. Even if the term-value of the variable is known, the subproof might need to keep working with the variable. Once written on the stack, however, the substitution is performed in the case of an individual term extracted by $R\exists$ or RN_1. The rule for cut introduces the term resulting from the left branch of the proof in the context of the right branch, as we already discussed above. Note that despite the branching, the rule produces just one $\mathcal{H}$-term with two different non-terminals inside for the two branches, as in the case of implication.

2.1 Properties of $\mathcal{H}^\pi$

In this section we show some of the characteristics of the recursion schemes defined above. The first straightforward property is that the rules of any system $\mathcal{H}^\pi$ are type preserving. This fact is evident by inspection of the rules, and as a consequence we have that any $\mathcal{H}$-term obtained by a series of reduction rules of a recursion scheme $\mathcal{H}^\pi$ from a proof π of $\Gamma \Rightarrow \varphi$ are of type $[\varphi]$.

Lemma 2.3 (Type preservation)**.** *Given a recursion scheme $\mathcal{H}^\pi$ from a proof π, for any m, n terms obtained by a series of reduction rules from the start symbol, if $m \twoheadrightarrow_{\mathcal{R}} n$ then $[m] = [n]$.*

Proof. By induction on the length of $m \twoheadrightarrow_{\mathcal{R}} n$. The base case is trivial, while the induction step holds by inspection of the rules, see Definition 2.2. □

Corollary 2.4. *Given a proof π of a sequent $\Gamma \Rightarrow \varphi$, any $\mathcal{H}$-term obtained by a series of reduction rules from $\mathcal{F}_\perp$ is of type $[\varphi]$.*

Introducing higher-order recursion schemes in Section 1.6 we mentioned the existence of non-deterministic schemes. If with 'deterministic' we refer to the presence of only one rule in $\mathcal{R}$ for each non-terminal, then our schemes $\mathcal{H}^\pi$ should be considered non-deterministic, because the rules $(L\vee)$ and (LN) give two different outcomes based on the kind of input-term they receive. However, thanks to pattern-matching we know that there is never confusion about which rule is to be applied. The non-terminal either needs to wait for some term of the proper form, or there is only one rule that can be applied: the system is then deterministic.

Directly connected with the type of recursion schemes is its order. In [AHL20] a bound of n on the order of the recursion scheme was established, for cut-formulas all in Π_n or Σ_n and in prenex-form. In the present context we can determine an easy correspondence between the complexity of the cut-formulas and the order. Recall that the order of a non-terminal $\mathcal{F}^\pi$ was defined as the one of its type according to Definition 1.17, and the order of a recursion scheme is the supremum of the orders of its non-terminals. Let's indicate with $o(\varphi)$ the order associated to the type of φ. From a superficial look at the definitions, we see that for any non-terminal $\mathcal{F}^\pi : \varsigma \to \gamma_0 \to \cdots \to \gamma_m \to \rho$ it can be defined $o(\mathcal{F}^\pi) := \max\{o(\rho), 1 + o(\gamma_i) : i \leq m\}$. This is because by Definition 1.17 the order of an implication $o(\varphi \to \psi) := \max\{o(\psi), 1 + o(\varphi)\}$ and $o(\varsigma) = 0$. We also know that the reduction rules are type preserving, hence the value $o(\rho)$ can either be the same of the previous non-terminal, or it can be decreased after a right rule. The same happens with left rules, where terms are de-constructed and reduced to lower complexity, hence the order of γ_i will either be the same or decreased. The only rule that can increase the order of a non-terminal with respect to the previous one is cut. That determines a value for the order of recursion schemes in terms of the maximal order of the (types of) all cut formulas.

Theorem 2.5 (Order of $\mathcal{H}^\pi$). *Given a cyclic proof π, for $\varphi_0, \dots, \varphi_m$ being all the cut formulas in π, and $\mathcal{F}^\pi$: the first non-terminal, the order of $\mathcal{H}^\pi$ is*

$$o(\mathcal{H}^\pi) = \max\{o(\mathcal{F}^\pi), o([\varphi_i]) + 1 : i \leq m\}$$

Proof. By inspection of the rules we see that the order of $\mathcal{F}^\pi$ and any following non-terminal can be increased only by cut. If there are no cuts, or the maximum order of a cut is lower than $o(\mathcal{F}^\pi)$, the order of the recursion scheme is the same as the one of its initial non-terminal. If m is at least the same value of $o(\mathcal{F}^\pi)$, since the cut-formulas occur as input of some non-terminal, its order is by definition $m + 1$, as it is the final order of the higher-order recursion scheme. □

The bound given here is as straightforward as it is uninformative. It is true that the proof π is a finite object and so we know for sure the order of its cut-formulas, but there is not much more that can be add here, since the notion of order of a type corresponds to the functional complexity of the formula, and we have no way of restricting the complexity of the cut formulas under a known threshold.

Let's focus on the properties of a higher-order recursion scheme with respect to the terms that can be obtained as a result of the reduction rules. In particular, we are interested in the language that a given $\mathcal{H}^\pi$ generates, assuming that the inputs are well-typed.

Definition 2.6 (Language)**.** *The language of a recursion scheme $\mathcal{H}^\pi$ with $\mathcal{F}_\perp$ as starting symbol is*

$$\mathcal{L}(\mathcal{H}^\pi) := \{t \in \Sigma^H \mid \mathcal{F}_\perp \overline{s} \twoheadrightarrow_\mathcal{R} t\}$$

for $\overline{s}$ any series of closed input terms of the right type.

The language is given by all the terms without non-terminals that can be obtained from a reduction sequence. Since the system has pattern-matching, the input must be not only of the right type, but also detailed enough to let the computation progress. If we have a non-terminal of the type $\mathcal{F} : \varsigma \to [\varphi] \to [\psi]$ and a variable $x : [\varphi]$ there is a high probability that the computation will not be completed due to failed pattern-matching. Looking at the Definition 2.1 of the Σ^H terms, however, we notice that variables can only be of type ι. It is a characteristic of every scheme $\mathcal{H}^\pi$ that every term of each type is formed by (a.) a constant term, like 0 and $\langle\rangle$ for the ground types, or an individual variable $a : \iota$; (b.) the unique constructor of each type, like $k_i p$ or $f^1_N(t,p)$ from typed subterms; or (c.) it is given by a functional $\mathcal{F}\overline{p} : [\psi]$ or $*_\psi p$ with $[\psi] = \epsilon$. It follows that the only free variables in a well-typed Σ^H-term can be individual variables. When no free variable occurs in a term p we say that p is *closed*.

Since $\mathcal{L}(\mathcal{H}^\pi)$ is given by the end-terms of a computation, we would like to know if the recursion scheme has the property of termination. Unfortunately, this is not the case for recursion schemes extracted from generic cyclic trees. In fact, even if at the level of proofs we have a global trace condition, we don't have an equivalent notion in the corresponding recursion schemes. We will come back to this issue in Section 2.2, but notice already that there is no structural difference between the recursion schemes that can be extracted from a proof and from a pre-proof. It is possible to define a recursion scheme from the unsound pre-proof of Example 1.2, obtaining a series of reductions that compute the same steps $\mathcal{F}_\pi\langle\rangle \to \mathcal{F}_{\pi'}(\mathcal{F}_{\pi''}\langle\rangle) \to \mathcal{F}_{\pi'}\langle\rangle \to \mathcal{F}_\pi\langle\rangle \ldots$ infinitely often. Moreover, we cannot appeal to some property of λY-calculus or cyclic recursion schemes in general, since both don't have the normalisation property. We leave this issue aside for the present section, accepting the possibility of non-terminating reductions and focusing on the characteristics of the language resulting from those that do terminate eventually.

So far we have stated without motivation that there is a correspondence between implicit and explicit substitution. If that is immediate at the individual term level, given the way in which the implicit substitution was defined in Section 1.3, we need to prove that the same extends to the non-terminals of $\mathcal{H}$. The kind of correspondence that we are interested into is with respect to the language $\mathcal{L}(\mathcal{H})$, i.e., we want to be sure that the presence of one kind of substitution instead of the other will have no effect in the language.

Let's introduce, then, a symbol for such a relation

Definition 2.7 ($\sim$)**.** *Given two $\mathcal{H}$-terms p, q, we say that $p \succeq q$ iff given any $\mathcal{H}$-term $m(x)$, if $m(p) \twoheadrightarrow_{\mathcal{R}} u \in \Sigma^H$ then there is a $v \in \Sigma^H$ such that $m(q) \twoheadrightarrow_{\mathcal{R}} v$ and $u^\circ = v^\circ$. Whenever $p \succeq q$ and $q \succeq p$ then the two are said to be* equivalent*: $q \sim p$.*

Notice already that by Definitions 1.14 and 1.15, it is always true that $\langle\rangle \circ [t \mapsto a] = \langle\rangle[t/a] \sim \langle\rangle$, and as a consequence the same holds for $f^0_N(\langle\rangle)$. It is also true that $*_\psi\, y \circ [t \mapsto a] = (*_\psi\, y)[t/a] \sim *_\psi\, y$, since y is a term of type $\perp$.

The next lemma ensures that the use of explicit or implicit substitution does not affect the language of the recursion scheme $\mathcal{H}^\pi$. To be able to prove it, however, we need to assume that in the proof π not only there is no eigenvariable outside the subproof above its rule application, as it is already by regularity, but also that the terms in the substitution stack of $\mathcal{F}^\pi$ contain none of the other eigenvariables of π. This request is not implausible, as it was assumed and proved to be sufficiently general already in [AHL20], where this is one of the conditions that determines the so called *normal terms.*[4] Informally we can argue that the only way in which a term of $\alpha : \varsigma$ can contain an eigenvariable is through the left existential rule or a substitution rule. In both cases there is no problem in opting for a different term, since the two occurrences of the variable are not logically related.

Lemma 2.8 (Substitution equivalence)**.** *Given a proof π, terms $t, b : \iota$ and a substitution stack α. $\mathcal{F}^\pi_i[t \mapsto a]\alpha\overline{x}$ is such that: if for some m, $m(\mathcal{F}^\pi_i[t \mapsto a]\alpha\overline{x}) \twoheadrightarrow_{\mathcal{R}} p$ and $p \in \Sigma^H$, then there is a $q \in \Sigma^H$ such that $m(\mathcal{F}^{\pi[t/a]}_i\alpha\overline{x}) \twoheadrightarrow_{\mathcal{R}} q$ and $p^\circ = q^\circ$. That is*

$$\mathcal{F}^\pi_i[t \mapsto a]\alpha\overline{x} \sim \mathcal{F}^{\pi[t/a]}_i\alpha\overline{x}$$

Proof. Assuming that $m(\mathcal{F}^\pi_i[t \mapsto a]\alpha\overline{x}) \twoheadrightarrow p$, by co-induction on the distance from the final term p we prove the lemma. We distinguish between two main possibilities: (1.) the next step reduces a non-terminal in m different than $\mathcal{F}^\pi_i$, or (2.) the next step is a reduction of $\mathcal{F}^\pi_i$.

1. If the redex is another non-terminal in m we have three sub-cases:
   (a) the reduction does not affect $\mathcal{F}^\pi_i$, since it involves only the context. In this case we have that

   $$m(\mathcal{F}^\pi_i[t \mapsto a]\alpha\overline{x}) \to m'(\mathcal{F}^\pi_i[t \mapsto a]\alpha\overline{x}) \twoheadrightarrow p$$

   and since $m(\mathcal{F}^{\pi[t/a]}_i\alpha\overline{x}) \to m'(\mathcal{F}^{\pi[t/a]}_i\alpha\overline{x})$ we conclude by hypothesis

   $$m(\mathcal{F}^{\pi[t/a]}_i\alpha\overline{x}) \to m'(\mathcal{F}^{\pi[t/a]}_i\alpha\overline{x}) \twoheadrightarrow q$$

   for $p \sim q$.
   (b) the reduction does not affect $\mathcal{F}^\pi_i$ but occurs in one of its arguments. The conclusion is analogous to (1a).

[4]See Definition 6.3 §1, p.28

(c) $\mathcal{F}_i^\pi$ is affected by the reduction. That is possible only if the rule is $(L =)$ or (W) and $\mathcal{F}_i^\pi$ is principal, or if it is not principal in any axiom. In any case the whole term $(\mathcal{F}_i^\pi[t \mapsto a]\alpha\overline{x})$ is cancelled, so we have

$$m(\mathcal{F}_i^\pi[t \mapsto a]\alpha\overline{x}) \to m' \twoheadrightarrow p$$

and

$$m(\mathcal{F}_i^{\pi[t/a]}\alpha\overline{x}) \to m' \twoheadrightarrow p$$

2. The redex is $\mathcal{F}_i^\pi$. We proceed by cases on the rule. In the majority of possibilities the reduction produces new non-terminal(s) without modifying the substitution stack. As a consequence a simple inductive step is sufficient. We give just two potentially interesting examples

- with (cut) we have

$$m(\mathcal{F}_i^\pi[t \mapsto a]\alpha\overline{x}) \to m(\mathcal{F}_{j1}^\pi[t \mapsto a]\alpha(\mathcal{F}_{j0}^\pi[t \mapsto a]\alpha\overline{x}_0)\overline{x}_1) \twoheadrightarrow p$$

  and also $m(\mathcal{F}_i^{\pi[t/a]}\alpha\overline{x}) \to m(\mathcal{F}_{j1}^{\pi[t/a]}\alpha(\mathcal{F}_{j0}^{\pi[t/a]}\alpha\overline{x}_0)\overline{x}_1)$. By hypothesis for both the non-terminals obtained

$$m(\mathcal{F}_i^{\pi[t/a]}\alpha\overline{x}) \to m(\mathcal{F}_{j1}^{\pi[t/a]}\alpha(\mathcal{F}_{j0}^{\pi[t/a]}\alpha\overline{x}_0)\overline{x}_1) \twoheadrightarrow q$$

- even in the case $(L\forall)$ of a universal quantifier there is no change in the substitution stack α

$$m(\mathcal{F}_i^\pi[t \mapsto a]\alpha w\overline{x}) \to m(\mathcal{F}_j^\pi[t \mapsto a]\alpha(ws)\overline{x}) \twoheadrightarrow p$$

  and also $m(\mathcal{F}_i^{\pi[t/a]}\alpha w\overline{x}) \to m(\mathcal{F}_j^{\pi[t/a]}\alpha(ws)\overline{x})$. By hypothesis

$$m(\mathcal{F}_i^{\pi[t/a]}\alpha w\overline{x}) \to m(\mathcal{F}_j^{\pi[t/a]}\alpha(ws)\overline{x}) \twoheadrightarrow q$$

The relevant cases are the ones that extract the stack α, or determine a different one after the first reduction, that is $(L\exists)$, $(R\exists)$, (RN_1), (LN) and (Sub). We focus here on the two existential rules, the remaining cases being analogous.

- $(R\exists)$ the reduction rule extracts the term together with the substitution stack. We know by definition that explicit and implicit substitutions coincide at the level of individual terms. We have that

$$m(\mathcal{F}_i^\pi[t \mapsto a]\alpha\overline{x}) \to m(\langle s \circ [t \mapsto a], \mathcal{F}_j^\pi[t \mapsto a]\alpha\overline{x}\rangle) \twoheadrightarrow p$$

  and also $m(\mathcal{F}_i^{\pi[t/a]}\alpha\overline{x}) \to m(\langle s[t/a], \mathcal{F}_j^{\pi[t/a]}\alpha\overline{x}\rangle)$. Since by definition $(s \circ [t \mapsto a])^\circ = s^{[t \mapsto a]} = s[t/a]$, it follows by hypothesis

$$m(\mathcal{F}_i^{\pi[t/a]}\alpha\overline{x}) \to m(\langle s[t/a], \mathcal{F}_j^{\pi[t/a]}\alpha\overline{x}\rangle) \twoheadrightarrow q$$

- the case (RN_1) is analogous in the sense that it also extracts an individual term with the stack. The conclusion follows from the same argument.
- $(L\exists)$. In this case we need to invoke the assumed property that no eigenvariable appears in some individual term in the substitution stack. That is because from the assumptions we have

$$m(\mathcal{F}_i^\pi[t \mapsto a]\alpha\langle s, y\rangle\overline{x}) \to m(\mathcal{F}_j^\pi[s \mapsto b][t \mapsto a]\alpha y\overline{x}) \twoheadrightarrow p$$

and $m(\mathcal{F}_i^{\pi[t/a]}\alpha\langle s, y\rangle\overline{x}) \to m(\mathcal{F}_j^{\pi[t/a]}[s \mapsto b]\alpha\overline{x})$ from which we cannot proceed further, because the necessary hypothesis is $m(\mathcal{F}_j^\pi[t \mapsto a][s \mapsto b]\alpha y\overline{x})$, and in general it is not true that

$$\mathcal{F}_j^\pi[t \mapsto a][s \mapsto b]\alpha\overline{x} \sim \mathcal{F}_j^\pi[s \mapsto b][t \mapsto a]\alpha\overline{x}$$

However, thanks to the condition on eigenvariables we have that $\pi[t/a][s/b] = \pi[s/b][t/a]$, so we can conclude that

$$m(\mathcal{F}_i^{\pi[t/a]}\alpha\langle s, y\rangle\overline{x}) \to m(\mathcal{F}_j^{\pi[t/a]}[s \mapsto b]\alpha\overline{x}) \twoheadrightarrow q$$

- the cases (Sub) and (LN) are analogous to the latter, since they also introduce a new substitution on top of the stack. By the same argument we can conclude the desired reduction.

A final remark for the axiom cases of $\mathcal{F}_i^\pi[t \mapsto a]\alpha\overline{x}$. In many instances the term obtained is $\langle\rangle$, hence no stack or substitution has any effect. The same for the cases (RN_0) with $f_N^0(\langle\rangle)$, and $(L\bot)$ with $*_\epsilon\, y$. For (Ax) and $(\mathbf{s}(=))$, when the resulting term of type ϵ is not $\langle\rangle$, then it must be the case that $p[t/a] \sim p$. As we pointed out above, in Σ^H the only possible terms of such a type are $\langle\rangle$, $*_\epsilon\, y$ or an application $(\mathcal{F}_k r)$. We just discussed the first and second terms. If it is the case that $(\mathcal{F}_k r)$, by hypothesis we have the conclusion.

□

Theorem 2.9 (Cut reduction invariance)**.** *Given a cyclic proof π, if π' is the proof that results from the application of one step of the cut reduction strategy, then $\mathcal{F}_\bot^\pi \sim \mathcal{F}_\bot^{\pi'}$.*

Proof. We proceed by cases considering the last rules applied before the cut that is being reduced. Let's call $\langle R_1, R_2\rangle$ the pair of last rules respectively on the left and right branches above the cut. The structure of the argument consists of four main cases, with some subcases:

A. R_1 is an axiom

B. R_2 is an axiom

C. the cut-formula φ is principal in both R_1, R_2

D. φ is not principal in at least one branch

For each scenario, we will highlight the reasons that make the recursion scheme invariant with respect to the application of a permutation/reduction rule.

(A). R_1 is an instance of a rule (Ax), $(L\bot)$, (id) or arithmetic axiom (the case of $R_1 = (RN_0)$ is different and it is included in (C) and (D) below). $\varphi \equiv t = s$ is principal in both, so depending on the nature of R_2 there are only two possible scenarios: (i) the resulting sequent is an instance of an axiom itself, or (ii) the final sequent can be obtained via weakening from the right branch. Since the second case is also covered by (C), we show only the first case here. An example of (i) is $\langle Ax, L\bot\rangle$:

$$\dfrac{\dfrac{}{t = s, \Gamma \Rightarrow_0 t = s}\,(Ax) \qquad \dfrac{}{t = s, \bot, \Delta \Rightarrow_1 r = q}\,(L\bot)}{t = s, \bot, \Gamma, \Delta \Rightarrow r = q} \qquad\qquad \dfrac{}{t = s, \bot, \Gamma, \Delta \Rightarrow r = q}\,(L\bot)$$

$$\begin{array}{lll} \mathcal{F}_\bot wz\overline{xy} & \longrightarrow & \mathcal{F}wz\overline{xy} \\ & \longrightarrow_{cut} & \mathcal{F}_1(\mathcal{F}_0 wz\overline{x})\overline{y} \\ & \longrightarrow_{Ax} & \mathcal{F}_0 wz\overline{x} \\ & \longrightarrow_{L\bot} & *_\epsilon z \end{array} \qquad \begin{array}{lll} \mathcal{F}_\bot wz\overline{xy} & \longrightarrow & \mathcal{F}w\bot\overline{xy} \\ & \longrightarrow_{L\bot} & *_\epsilon z \end{array}$$

The rest of the combinations with R_1 an axiom are analogous.

(B) R_2 is an instance of a rule (Ax), $(L\bot)$, (id), (RN_0) or arithmetical axiom. We have again two possible scenarios. In the first the cut-formula is in the context of R_2 but it is irrelevant for the reduction, so we always have that the final sequent is an axiom itself like in the following example $\langle R_1, RN_0\rangle$:

$$\dfrac{\dfrac{\Gamma' \Rightarrow_0 \varphi'}{\Gamma \Rightarrow_1 \varphi}\,(R_1) \qquad \dfrac{}{\varphi, \Delta \Rightarrow_2 N0}\,(RN_0)}{\Gamma, \Delta \Rightarrow N0} \qquad\qquad \dfrac{}{\Gamma, \Delta \Rightarrow_1 N0}\,(RN_0)$$

$$\begin{array}{lll} \mathcal{F}_\bot\overline{xy} & \longrightarrow & \mathcal{F}\overline{xy} \\ & \longrightarrow_{cut} & \mathcal{F}_1(\mathcal{F}_0\overline{x})\overline{y} \\ & \longrightarrow_{RN_0} & f_N^0(\langle\rangle) \end{array} \qquad \begin{array}{lll} \mathcal{F}_\bot\overline{xy} & \longrightarrow & \mathcal{F}\overline{xy} \\ & \longrightarrow_{RN_0} & f_N^0(\langle\rangle) \end{array}$$

In the second scenario there is a dependency between the input term and the result, that is for $R_2 = (Ax)$ or $(= \mathbf{s})$. We have for example $\langle R_1, Ax\rangle$

$$\dfrac{\Gamma \Rightarrow_0 t = s \qquad \dfrac{}{t = s, \Delta \Rightarrow_1 t = s}\,(Ax)}{\Gamma, \Delta \Rightarrow t = s} \qquad \dfrac{\Gamma \Rightarrow_0 t = s}{\Gamma, \Delta \Rightarrow_1 t = s}\,(W)$$

$$\begin{array}{lll} \mathcal{F}_\bot \overline{x}\overline{y} & \longrightarrow & \mathcal{F}\overline{x}\overline{y} \\ & \longrightarrow_{cut} & \mathcal{F}_1(\mathcal{F}_0\overline{x})\overline{y} \\ & \longrightarrow_{Ax} & \mathcal{F}_0\overline{x} \end{array} \qquad \begin{array}{lll} \mathcal{F}_\bot \overline{x}\overline{y} & \longrightarrow & \mathcal{F}\overline{x}\overline{y} \\ & \longrightarrow_{W} & \mathcal{F}_0\overline{x} \end{array}$$

(C) in the third case the cut formula φ is principal in both R_1 and R_2. The strategy for cut reduction consists in producing cuts of a lower complexity. The way in which the rewrite rules are defined takes care of this strategy. We will give here the cases of disjunction, existential quantifier and N predicate, because they present the most interesting situations.

- $\varphi \equiv \psi_0 \vee \psi_1$

$$\dfrac{(R\vee)\,\dfrac{\Delta \Rightarrow_{00} \psi_i}{\Delta \Rightarrow_0 \psi_0 \vee \psi_1} \qquad \dfrac{\psi_0, \Gamma' \Rightarrow_{10} \chi \qquad \psi_1, \Gamma'' \Rightarrow_{11} \chi}{\psi_0 \vee \psi_1, \Gamma', \Gamma'' \Rightarrow_1 \chi}\,(L\vee)}{\Gamma', \Gamma'', \Delta \Rightarrow \chi} \qquad \dfrac{\dfrac{\Delta \Rightarrow_{00} \psi_i \qquad \psi_i, \Gamma^i \Rightarrow_{1i} \chi}{\Gamma^i, \Delta \Rightarrow_B \chi}\,(cut)}{\Gamma', \Gamma'', \Delta \Rightarrow_A \chi}\,(W)$$

$$\begin{array}{lll} \mathcal{F}_\bot \overline{x_0}\overline{x_1}\overline{y} & \longrightarrow & \mathcal{F}\overline{x_0}\overline{x_1}\overline{y} \\ & \longrightarrow_{cut} & \mathcal{F}_1(\mathcal{F}_0\overline{y})\overline{x_0}\overline{x_1} \\ & \longrightarrow_{R\vee} & \mathcal{F}_1\kappa_i(\mathcal{F}_{00}\overline{y})\overline{x_0}\overline{x_1} \\ & \longrightarrow_{L\vee} & \mathcal{F}_{1i}(\mathcal{F}_{00}\overline{y})\overline{x_i} \end{array} \qquad \begin{array}{lll} \mathcal{F}_\bot \overline{x_0}\overline{x_1}\overline{y} & \longrightarrow & \mathcal{F}_A\overline{x_0}\overline{x_1}\overline{y} \\ & \longrightarrow_{W} & \mathcal{F}_B\overline{x_i}\overline{y} \\ & \longrightarrow_{cut} & \mathcal{F}_{1i}(\mathcal{F}_{00}\overline{y})\overline{x_i} \end{array}$$

This example shows that the computation does not need to inspect all the branches, once the input in the context indicate the one that it is to pursue.

- $\varphi \equiv \exists z.\psi(z)$

$$\dfrac{(R\exists)\,\dfrac{\Delta \Rightarrow_{00} \psi(s)}{\Delta \Rightarrow_0 \exists z.\psi(z)} \qquad \dfrac{\psi(v), \Gamma \Rightarrow_{11} \chi}{\exists z.\psi(z), \Gamma \Rightarrow_1 \chi}\,(L\exists)}{\Gamma, \Delta \Rightarrow \chi} \qquad \dfrac{\Delta \Rightarrow_{00} \psi(s) \qquad \psi(v)[s/v], \Gamma \Rightarrow_{11} \chi}{\Gamma, \Delta \Rightarrow \chi}\,(cut)$$

$$\begin{array}{lll} \mathcal{F}_\bot & \longrightarrow & \mathcal{F}\overline{x}\overline{y} \\ & \longrightarrow_{cut} & \mathcal{F}_1(\mathcal{F}_0\overline{y})\overline{x} \\ & \longrightarrow_{R\exists} & \mathcal{F}_1\langle s, \mathcal{F}_{00}\overline{y}\rangle\overline{x} \\ & \longrightarrow_{L\exists} & \mathcal{F}_{11}[s \mapsto v](\mathcal{F}_{00}\overline{y})\overline{x} \end{array} \qquad \begin{array}{lll} \mathcal{F}_\bot & \longrightarrow & \mathcal{F}\overline{x}\overline{y} \\ & \longrightarrow_{cut} & \mathcal{F}_{11}^{[s/v]}(\mathcal{F}_{00}\overline{y})\overline{x} \end{array}$$

In this case we use Lemma 2.8 to be sure that $\mathcal{F}_1[s \mapsto v](\mathcal{F}_0\overline{y})\overline{x} \sim \mathcal{F}_1^{[s/v]}(\mathcal{F}_0\overline{y})\overline{x}$ are equivalent with respect to the final language.

- $\varphi \equiv Nt$. We need to distinguish between two cases: (a) $t = 0$,[5] and (b) $t = \mathbf{s}(s)$. In both cases there is not a direct reduction strategy for the system we have defined. We can, however, build two reductions by using (id) and the knowledge of whether (a) or (b).

  (a) In the first case we can reduce the cut to an atomic one by taking the axiom $\Delta \Rightarrow 0 = 0$ instead of $\Delta \Rightarrow N0$. If $N0$ is the cut-formula, in fact, we have that $t \equiv 0$ also on the right branch.

$$\dfrac{\Delta \Rightarrow_0 N0 \qquad \dfrac{0=0,\Gamma' \Rightarrow_{10} \chi \qquad 0=\mathbf{s}(s), Ns, \Gamma'' \Rightarrow_{11} \chi}{N0, \Gamma', \Gamma'' \Rightarrow_1 \chi}\,(LN)}{\Gamma', \Gamma'', \Delta \Rightarrow \chi} \qquad \dfrac{\dfrac{\Rightarrow_{10} 0=0 \qquad 0=0,\Gamma' \Rightarrow_{11} \chi}{\Gamma' \Rightarrow_A \chi}}{\Gamma', \Gamma'', \Delta \Rightarrow \chi}$$

$$\begin{array}{lll@{\qquad}lll}
\mathcal{F}_\perp\overline{x_0x_1y} & \longrightarrow & \mathcal{F}\overline{x_0x_1y} & \mathcal{F}_\perp\overline{x_0x_1y} & \longrightarrow & \mathcal{F}\overline{x_0x_1y} \\
 & \longrightarrow_{cut} & \mathcal{F}_1(\mathcal{F}_0\overline{y})\overline{x_0x_1} & & \longrightarrow_W & \mathcal{F}\overline{x_0} \\
 & \longrightarrow_{RN_0} & \mathcal{F}_1 f_N^0(\langle\rangle)\overline{x_0x_1} & & \longrightarrow_{cut} & \mathcal{F}_{11}(\mathcal{F}_{00})\overline{x_0} \\
 & \longrightarrow_{LN} & \mathcal{F}_{11}\langle\rangle\overline{x_0} & & \longrightarrow_{id} & \mathcal{F}_{11}\langle\rangle\overline{x_0}
\end{array}$$

  (b) In the second case we have again to consider the fact that $t \equiv \mathbf{s}(s)$ and use the axiom $\Rightarrow \mathbf{s}(s) = \mathbf{s}(s)$. We can then perform the following

$$\dfrac{\dfrac{\Delta \Rightarrow_{00} Ns}{\Delta \Rightarrow N\mathbf{s}(s)} \qquad \dfrac{\mathbf{s}(s) = 0, \Gamma' \Rightarrow \chi \qquad \mathbf{s}(s) = \mathbf{s}(a), Na, \Gamma'' \Rightarrow_{11} \chi}{N\mathbf{s}(s), \Gamma', \Gamma'' \Rightarrow \chi}}{\Gamma', \Gamma'', \Delta \Rightarrow \chi}$$

becomes

$$\dfrac{\dfrac{\Delta \Rightarrow_{00} Ns \qquad \dfrac{\Rightarrow \mathbf{s}(s) = \mathbf{s}(s) \qquad \mathbf{s}(s) = \mathbf{s}(a)[s/a], Na[s/a], \Gamma'' \Rightarrow_{11} \chi}{Na[s/a], \Gamma'' \Rightarrow \chi}}{\Gamma'', \Delta \Rightarrow \chi}}{\Gamma', \Gamma'', \Delta \Rightarrow \chi}$$

[5] In this case we consider $R_1 = RN_0$.

$$
\begin{array}{llllll}
\mathcal{F}_\perp\overline{x_0x_1y} & \longrightarrow & \mathcal{F}\overline{x_0x_1y} & \mathcal{F}_\perp\overline{x_0x_1y} & \longrightarrow & \mathcal{F}\overline{x_0x_1y} \\
 & \longrightarrow_{cut} & \mathcal{F}_1(\mathcal{F}_0\overline{y})\overline{x_0x_1} & \mathcal{F}_\perp & \longrightarrow_{W} & \mathcal{F}\overline{x_1y} \\
 & \longrightarrow_{RN_1} & \mathcal{F}_1 f^1(s,\mathcal{F}_{00}\overline{y})\overline{x_0x_1} & & \longrightarrow_{cut} & \mathcal{F}_1^{[s/a]}(\mathcal{F}_{00}\overline{y})\overline{x_1} \\
 & \longrightarrow_{LN} & \mathcal{F}_{11}[s\mapsto a]\langle\rangle(\mathcal{F}_{00}\overline{y})\overline{x_1} & & \longrightarrow_{cut} & \mathcal{F}_{11}^{[s/a]}(\mathcal{F}_{10})(\mathcal{F}_{00}\overline{y})\overline{x_1} \\
 & & & & \longrightarrow_{id} & \mathcal{F}_{11}^{[s/a]}\langle\rangle(\mathcal{F}_{00}\overline{y})\overline{x_1}
\end{array}
$$

By Lemma 2.8

$$\mathcal{F}_{11}[s\mapsto a]\langle\rangle(\mathcal{F}_{00}\overline{y})\overline{x_1} \sim \mathcal{F}_{11}^{[s/a]}\langle\rangle(\mathcal{F}_{00}\overline{y})\overline{x_1}$$

Note that C covers also the cases of φ principal with R_2 being (W) or (C). In the first case we simply make use of weakening on the right premise to fix the context, while in the second case we have that

$$
\dfrac{\Delta\Rightarrow_0\varphi \qquad \dfrac{\varphi,\varphi,\Gamma\Rightarrow_1\chi}{\varphi,\Gamma\Rightarrow_2\chi}}{\Gamma,\Delta\Rightarrow\chi}
\qquad
\dfrac{\Delta\Rightarrow_0\varphi \qquad \dfrac{\Delta\Rightarrow_{10}\varphi \qquad \varphi,\varphi,\Gamma\Rightarrow_{11}\chi}{\varphi,\Gamma\Rightarrow_1\chi}}{\Gamma,\Delta\Rightarrow\chi}
$$

and as a result we have the computations

$$
\begin{array}{llllll}
\mathcal{F}_\perp\overline{xy} & \longrightarrow & \mathcal{F}\overline{xy} & \mathcal{F}_\perp\overline{xy} & \longrightarrow & \mathcal{F}\overline{xy} \\
 & \longrightarrow_{cut} & \mathcal{F}_1(\mathcal{F}_0\overline{y})\overline{x} & & \longrightarrow_{cut} & \mathcal{F}_1(\mathcal{F}_0\overline{y})\overline{x} \\
 & \longrightarrow_{C} & \mathcal{F}_1(\mathcal{F}_0\overline{y})(\mathcal{F}_0\overline{y})\overline{x} & & \longrightarrow_{cut} & \mathcal{F}_{11}(\mathcal{F}_0\overline{y})(\mathcal{F}_{10}\overline{y})\overline{x}
\end{array}
$$

with $\mathcal{F}_{10}=\mathcal{F}_0$.

(D) φ is not principal in at least one of the branches. This generates four possibilities depending on the number of premises of the rule under scrutiny, plus the case of one of the rules being a cut. Let's see the case of φ non-principal in a rule R_1 with one premise:

$$
\dfrac{(R_1)\,\dfrac{\Delta'\Rightarrow_{00}\varphi}{\Delta\Rightarrow_0\varphi} \qquad \varphi,\Gamma\Rightarrow_1\chi}{\Gamma,\Delta\Rightarrow\chi}
\qquad
(R_1)\,\dfrac{\dfrac{\Delta'\Rightarrow_{00}\varphi \qquad \varphi,\Gamma\Rightarrow_1\chi}{\Gamma,\Delta'\Rightarrow_A\chi}}{\Gamma,\Delta\Rightarrow\chi}
$$

$$
\begin{array}{llllll}
\mathcal{F}_\perp\overline{xy} & \longrightarrow & \mathcal{F}\overline{xy} & \mathcal{F}_\perp\overline{xy} & \longrightarrow & \mathcal{F}\overline{xy} \\
 & \longrightarrow_{cut} & \mathcal{F}_1(\mathcal{F}_0\overline{y})\overline{x} & & \longrightarrow_{R_1} & \mathcal{F}_A\overline{xy_0} \\
 & \longrightarrow_{R_1} & \mathcal{F}_1(\mathcal{F}_{00}\overline{y_0})\overline{x} & & \longrightarrow_{cut} & \mathcal{F}_1(\mathcal{F}_{00}\overline{y_0})\overline{x}
\end{array}
$$

The rules such that R_1 has two premises are $(L\vee)$ and (LN) only. We already saw how their pattern-matching restricts the reduction to just one of the two branches, hence they are similar to the case of a one premise rule. Say φ is not principal in R_2 with two premises.

$$\dfrac{\Delta \Rightarrow_0 \varphi \qquad \dfrac{\varphi, \Gamma' \Rightarrow_{10} \chi' \qquad \Gamma'' \Rightarrow_{11} \chi''}{\varphi, \Gamma \Rightarrow_1 \chi}}{\Delta, \Gamma \Rightarrow \chi} \qquad \dfrac{\dfrac{\Delta \Rightarrow_0 \varphi \qquad \varphi, \Gamma' \Rightarrow_{10} \chi'}{\Delta, \Gamma' \Rightarrow_A \chi'} \qquad \Gamma'' \Rightarrow_{11} \chi''}{\Delta, \Gamma \Rightarrow \chi}$$

$$\begin{array}{lll} \mathcal{F}_\bot\overline{xy} & \longrightarrow & \mathcal{F}\overline{xy} \\ & \longrightarrow_{cut} & \mathcal{F}_1(\mathcal{F}_0\overline{y})\overline{x} \\ & \longrightarrow_{R_2} & \mathcal{F}_{11}(\mathcal{F}_{10}\overline{x_1}(\mathcal{F}_0\overline{y}))\overline{x_0} \end{array} \qquad \begin{array}{lll} \mathcal{F}_\bot\overline{xy} & \longrightarrow & \mathcal{F}\overline{xy} \\ & \longrightarrow_{R_2} & \mathcal{F}_{11}(\mathcal{F}_A\overline{x_1 y})\overline{x_0} \\ & \longrightarrow_{cut} & \mathcal{F}_{11}(\mathcal{F}_{10}\overline{x_1}(\mathcal{F}_0\overline{y}))\overline{x_0} \end{array}$$

The final structure of $\mathcal{F}_{11}(\mathcal{F}_{10}\overline{x_1}(\mathcal{F}_0\overline{y}))\overline{x_0}$ depends on the nature of R_2, but we know that the combination preserves the two distinct subterms, hence the result. The case of a one-premise rule is easier and not given here. It remains to show the case of two consecutive cuts that are inverted, which is similar to what we just saw. Say that R_2 is a cut

$$\dfrac{\Delta \Rightarrow_0 \varphi \qquad \dfrac{\varphi, \Gamma' \Rightarrow_{10} \psi \qquad \psi, \Gamma'' \Rightarrow_{11} \chi}{\varphi, \Gamma \Rightarrow_1 \chi}}{\Delta, \Gamma \Rightarrow \chi} \qquad \dfrac{\dfrac{\Delta \Rightarrow^0 \varphi \qquad \varphi, \Gamma' \Rightarrow^{10} \psi}{\Delta, \Gamma' \Rightarrow^A \psi} \qquad \psi, \Gamma'' \Rightarrow^{11} \chi}{\Delta, \Gamma \Rightarrow \chi}$$

and as a computation

$$\begin{array}{lll} \mathcal{F}_\bot\overline{xy} & \longrightarrow & \mathcal{F}\overline{xy} \\ & \longrightarrow_{cut} & \mathcal{F}_1(\mathcal{F}_0\overline{y})\overline{x} \\ & \longrightarrow_{cut} & \mathcal{F}_{11}(\mathcal{F}_{10}(\mathcal{F}_0\overline{y})\overline{x_0})\overline{x_1} \end{array} \qquad \begin{array}{lll} \mathcal{F}_\bot\overline{xy} & \longrightarrow & \mathcal{F}\overline{xy} \\ & \longrightarrow_{cut} & \mathcal{F}_{11}(\mathcal{F}_A\overline{y x_0})\overline{x_1} \\ & \longrightarrow_{cut} & \mathcal{F}_{11}(\mathcal{F}_{10}(\mathcal{F}_0\overline{y})\overline{x_0})\overline{x_1} \end{array}$$

The case $R_1 = (cut)$ is analogous.

This concludes the analysis of the possible cases of cut-reduction. It remains to remark that the case of (Sub) has not been analysed, but it sufficient to point out that the strategy in that case consists in a simple substitution of terms for variables. As for the existential rule, we obtain the desired final term by noticing that $(\mathcal{F}\alpha\overline{x})[t/x] \sim \mathcal{F}^{[t/x]}\alpha\overline{x}$, as resulting from Lemma 2.8.

□

2.2 The language $\mathcal{L}(\mathcal{H}^\pi)$

Now that we have shown some preliminary properties of $\mathcal{H}^\pi$, we want to conclude this work by looking at the language obtained. In this section we focus on the language $\mathcal{L}(\mathcal{H}^\pi)$ and the process of reduction in general. First of all, we prove that $\twoheadrightarrow_\mathcal{R}$ is confluent, that is for $t \twoheadrightarrow_\mathcal{R} p_0$ and $t \twoheadrightarrow_\mathcal{R} p_1$ there is a term p such that $p_0 \twoheadrightarrow_\mathcal{R} p$ and $p_1 \twoheadrightarrow_\mathcal{R} p$.

Theorem 2.10 ($\to_\mathcal{R}$ confluence)**.** *The reduction relation $\to_\mathcal{R}$ is confluent, i.e., for all t, p_0, p_1, if $t \twoheadrightarrow_\mathcal{R} p_0$ and $t \twoheadrightarrow_\mathcal{R} p_1$ there is a term p such that $p_0 \twoheadrightarrow_\mathcal{R} p$ and $p_1 \twoheadrightarrow_\mathcal{R} p$.*

Proof. For $\to_\mathcal{R}^+$ the reflexive closure of $\to_\mathcal{R}$ we prove that if $t \to_\mathcal{R}^+ p_0$ and $t \twoheadrightarrow_\mathcal{R} p_1$ there is a term p such that $p_0 \twoheadrightarrow_\mathcal{R} p$ and $p_1 \to_\mathcal{R}^+ p$.[6] Let $\mathcal{F}\overline{x}$ be the redex in t that is reduced to obtain p_0, that is $t(\mathcal{F}\overline{x}) \to_\mathcal{R} t(\mathcal{F}'\overline{x}') \equiv p_0$. There are four possibilities for the series of reductions $t \twoheadrightarrow_\mathcal{R} p_1$

1. $\mathcal{F}\overline{x}$ still occurs in $p_1 \equiv t'(\mathcal{F}\overline{x})$ untouched, meaning that all the reductions involved redexes outside of the term $\mathcal{F}\overline{x}$. Clearly we have $t(\mathcal{F}'\overline{x}') \twoheadrightarrow_\mathcal{R} t'(\mathcal{F}'\overline{x}')$ and $t'(\mathcal{F}\overline{x}) \to_\mathcal{R}^+ t'(\mathcal{F}'\overline{x}')$
2. $\mathcal{F}$ still occurs in $p_1 \equiv t'(\mathcal{F}\overline{x}'')$ but some reduction changed some of the terms x_i. Since all the rules are type preserving it is still possible to reduce $\mathcal{F}\overline{x}'$ and obtain $t'(\mathcal{F}\overline{x}'') \to_\mathcal{R}^+ t'(\mathcal{F}'\overline{y})$, while clearly $t(\mathcal{F}'\overline{x}') \twoheadrightarrow_\mathcal{R} t'(\mathcal{F}'\overline{y})$.[7]
3. $\mathcal{F}\overline{x}$ has been deleted by some reduction in $t(\mathcal{F}\overline{x}) \twoheadrightarrow_\mathcal{R} p_1$. Then we have $t(\mathcal{F}'\overline{x}') \twoheadrightarrow_\mathcal{R} p_1$ and the conclusion for $p \equiv p_1$
4. $\mathcal{F}\overline{x}$ has been reduced in the process $t(\mathcal{F}\overline{x}) \twoheadrightarrow_\mathcal{R} t'(\mathcal{F}\overline{x}) \to_\mathcal{R} t'(\mathcal{F}'\overline{x}') \twoheadrightarrow_\mathcal{R} p_1$. We have that $t(\mathcal{F}'\overline{x}') \twoheadrightarrow_\mathcal{R} t'(\mathcal{F}'\overline{x}') \twoheadrightarrow_\mathcal{R} p_1$.

A direct application of the property showed proves confluence. □

As a consequence, whenever a term has a normal form, that is unique. The next property of $\twoheadrightarrow_\mathcal{R}$ is the leftmost reduction property, that is: if there is a normal form, then a strategy that reduces always the leftmost possible redex terminates. Before we are able to prove that, it is convenient to define normal forms of $\mathcal{H}$-term and characterise them by showing that they don't contain non-terminals.

Definition 2.11 (Normal form)**.** *A term $q \in \mathcal{H}$ is in normal form iff there is no term $q' \in \mathcal{H}$ such that $q \to_\mathcal{R} q'$.*

Given a recursion scheme $\mathcal{H}^\pi$, let assume that $\overline{p}$ is a series of input terms of the right type for the initial non-terminal $\mathcal{F}_\perp$, and also that no individual variable occurs in $\overline{p}$, i.e. it is a closed term. We prove that

[6] This is called Strip Lemma in [Bar84], p.282. A version for rewrite systems can be found in [Klo92], p.72 as Parallel Moves Lemma, whose argument we follow here.

[7] The possibility is a consequence of the fact that no pattern matching is blocked by a reduction on an argument term.

Lemma 2.12 (Normal form)**.** *Given a recursion scheme $\mathcal{H}^\pi$, a closed input $\overline{p} : [\Gamma]$ and an initial non-terminal $\mathcal{F}_\perp : [\Gamma] \to [\psi]$, if $\mathcal{F}_\perp \overline{p} \twoheadrightarrow_\mathcal{R} q$ and q is in normal form, then $q \in \Sigma^H$ does not contain non-terminals.*

Proof. Assume $\mathcal{F}_\perp \overline{p} \twoheadrightarrow_\mathcal{R} q$ and q is in normal form. Either $q \in \Sigma^H$, then we are fine, or there is a non-terminal $\mathcal{F}_i$ such that $\mathcal{F}_i \overline{x}$ cannot be reduced. That is the case if either some input-term is not of the right type, but that is impossible for the assumption on $\overline{p}$ and Lemma 2.3, or because a pattern-matching rule cannot be reduced. As we pointed out already, for every type $[\psi]$ there are in $\mathcal{H}$ three possible kinds of terms by definition: a term obtained by a constructor from well-typed subterms, a term $*_\epsilon\, y : \epsilon$ of unit type, or a functional term $\mathcal{F}_j \overline{y} : [\psi]$. Every rule with pattern matching can be reduced in the first case, while no pattern matching exists for a term of type ϵ, hence the situation is necessarily $\mathcal{F}_i(\mathcal{F}_j \overline{y})\overline{x}$. The reason why the initial $\mathcal{F}_i$ is stuck is because another irreducible non-terminal occurs as argument. However, since our term is finite, we can find an innermost non-terminal $\mathcal{F}_k$ that cannot be reduced and doesn't have non-terminals as input. Since this is impossible, we have that no term in normal form has non-terminals. □

Now that we are sure that non-terminals don't appear in normal forms, we can prove that if a term in normal form exists, the leftmost reduction strategy terminates with that term. Remember that the leftmost possible reduction is not necessarily on the outermost non-terminal, in this context for pattern-matching.

Definition 2.13 (Leftmost strategy)**.** *We say that a reduction $p \to^l_\mathcal{R} p'$ is a leftmost reduction iff the reduction is operated on the first possible redex read from the left.*

Lemma 2.14 (Leftmost reduction)**.** *Given a recursion scheme $\mathcal{H}^\pi$ on a closed input $\overline{p}$, if $\mathcal{F}_\perp \overline{p} \twoheadrightarrow_\mathcal{R} q$ for q in normal form, then q can be computed from $\mathcal{F}_\perp \overline{p}$ by reducing at every step the leftmost possible redex.*

Proof. Assume that $\mathcal{F}_\perp \overline{p} \twoheadrightarrow_\mathcal{R} q$ for q in normal form. By Lemma 2.12 q does not contain non-terminals, hence the leftmost redexes have been reduced at some point during the computation. Say that $\mathcal{F}_\perp \overline{p} \to_\mathcal{R} p_0 \to_\mathcal{R} \ldots \to_\mathcal{R} p_m \to_\mathcal{R} q$ and notice that necessarily $p_n \to^l_\mathcal{R} q$. Since all the leftmost redexes during the reduction are computed eventually, we just need to re-order the series of reductions so that the result is in the desired order. Take the first such that $p_i(\mathcal{F}_0, \mathcal{F}_1) \to_\mathcal{R} p_{i+1}(\mathcal{F}_0, \mathcal{F}'_1) \twoheadrightarrow^l_\mathcal{R} q$, where $(\mathcal{F}_0, \mathcal{F}_1)$ are the leftmost reduction and the one computed instead, respectively. Since the remaining are all leftmost reductions, we know that $p_{i+1}(\mathcal{F}_0, \mathcal{F}'_1) \to^l_\mathcal{R} p_{i+2}(\mathcal{F}'_0, \mathcal{F}'_1) \twoheadrightarrow^l_\mathcal{R} q$. Depending on the rule for $\mathcal{F}_0$ that starts the leftmost reduction sequence, we can have three possible situations:

1. $\mathcal{F}'_1$ is copied
2. $\mathcal{F}'_1$ is cancelled
3. $\mathcal{F}'_1$ is duplicated, if $\mathcal{F}_0$ is the (C) rule for contraction.

In the first case, by reducing $\mathcal{F}_0$ instead of $\mathcal{F}_1$ we can postpone its reduction until it is the leftmost (if it is not cancelled before). We have that $p_i(\mathcal{F}_0, \mathcal{F}_1) \to^l_{\mathcal{R}} p_{i+1}(\mathcal{F}'_0, \mathcal{F}_1) \twoheadrightarrow^l_{\mathcal{R}} q$. In the second case, then we simply jump one step and have $p_i(\mathcal{F}_0, \mathcal{F}_1) \to_{\mathcal{R}} p_{i+2}(\mathcal{F}'_0) \twoheadrightarrow^l_{\mathcal{R}} q$. In the third case we have that $p_i(\mathcal{F}_0, \mathcal{F}_1) \to_{\mathcal{R}} p_{i+1}(\mathcal{F}'_0, \mathcal{F}_1, \mathcal{F}_1)$. As in the first case, we can wait until each occurrence of $\mathcal{F}_1$ is either cancelled or is the leftmost. We know that these are the only possibilities because in the initial computation we have $p_{i+1}(\mathcal{F}_0, \mathcal{F}'_1) \to^l_{\mathcal{R}} p_{i+2}(\mathcal{F}'_0, \mathcal{F}'_1, \mathcal{F}'_1) \twoheadrightarrow^l_{\mathcal{R}} q$. Applying this strategy to the whole reduction sequence guarantees that the leftmost reduction strategy eventually terminates. □

2.2.1 On termination

The reduction relation is confluent, the normal form is unique (when exists) and we have a strategy that terminates whenever there is a normal form. These results are not surprising given that the same holds for λY-terms. The remaining question is about termination. On the one hand, we know that λY-calculus does not have the property of termination, due to the behaviour of the fixed point combinator Y:

$$Yp = p(Yp) = p(p(Yp)) \dots$$

The simple correspondence with term-witnesses is not enough for us to believe that the process of term rewriting might terminate. On the other hand, at the very core of the motivation for cyclic proofs is the certainty that to every instantiation of formulas with closed terms corresponds a finite computation, the infinite regression being possible only at the limit. From this fact and the faithful correspondence between HORS and cyclic proofs we would reasonably expect that the reduction starting on closed Σ^H-terms can terminate, that is, has a normal form.

As we anticipated above, these two positions are compatible together if we consider that the realisation via λY-terms covers all cyclic trees including pre-proofs, while the argument for termination holds in the case of proofs only. It is perfectly reasonable, then, to expect non-termination in general, and at the same time that there is a finite reduction for every $\mathcal{H}^\pi$ defined from a proof π that satisfies the global trace condition. The key point is the absence in the definition given of $\mathcal{H}^\pi$ of a counterpart for the global trace condition. Notice, in fact, that even if the rewrite rules are determined by the deduction rules of π, the reduction steps don't follow a specific path, but insist on multiple points of the derivation at the same time, hence insist on multiple paths. Whenever a rewrite rule branches into two non-terminals, we have that the computation splits, and from the next step considers the two branches of the subproof. In the framework of Definition 2.2 we cannot claim that we are able to track the progress of an inductive formula in a single trace. In order to be able to do so, we need to expand further our toolbox of definitions, a goal that we defer to future work. Nonetheless, since we believe that the processes on closed terms really terminate, we want to conclude this first part with an informal description of infinite reductions, giving an insight on the tools required to formally prove termination.

Following the argument for the soundness of cyclic proofs, to prove termination we assume the existence of infinite reductions for $\mathcal{H}^\pi$ recursion schemes where π is a proof, and on closed inputs. Then we want to derive the existence of an infinitely decreasing chain of natural numbers to conclude its impossibility. As a first step we connect the infinite reduction in $\mathcal{H}^\pi$ to the progression on an infinite path in π, to be able to refer to the global trace condition. Strictly speaking, every non-terminal is different from the non-terminals generated by its reduction, so we need to define some notion of position in the term of $\mathcal{H}^\pi$ that is fixed along the reduction process. Assume that we have defined the notion of position in a $\mathcal{H}^\pi$-term. Between the position in the latter and the points of π there is a correspondence: for every reduction step, the term at a given position corresponds to the sequence one step up along a path. We can then argue by cases:

1. there is a position in the term where the non-terminals are reduced infinitely often
2. every position is reduced finitely many times

Every reduction step of a non-terminal corresponds to a step of a path in the proof. As a consequence, to the non-terminal progressing along the proof infinitely many times in the first case corresponds already an infinite path in π. Notice that when it reaches an axiom, a path is ended, but not necessarily the computation. It might be the case that a non-terminal was witnessing the principal formula on the left of (Ax)[8] and that the rewrite rule for the axiom gives

$$m(\mathcal{F}_0(\mathcal{F}_1\overline{y})\overline{x}) \to_{\mathcal{R}} m(\mathcal{F}_1\overline{y})$$

corresponding to a jump in the proof tree to another path:

$\mathcal{F}_1$ now occupies the position of $\mathcal{F}_0$. In the informal correspondence between reductions and paths, that represents the continuation of the computation on another path, (we can consider the previous segment as a detour). If there is a position that is reduced infinitely often, then we have a correspondence with an infinite path in π. If instead every position is reduced finitely many times while the reduction is infinite, that is possible only because every non-terminal at some point is stuck by pattern-matching, waiting for an input of the proper form to be produced by some internal computation.[9] That is, at some point the computation is pushed inside infinitely many times. Similar to the case of the axiom, a non-terminal inside is created by an occurrence of a cut at some lower level. To continue the computation on the newly generated non-terminal inside corresponds to jumping to the other path of a cut rule. Since this process is infinite while our cyclic proof is only finitely branching, it follows that the computation progresses along an infinite path in π.

The second and final component of an argument for termination is a proof of the fact that from the global trace condition of π it follows an impossibility for such an infinite path to exist on closed inputs. That is where our notation comes short, since we don't have yet

[8] An identical situation occurs for the rest of axioms where a left argument is the result of the reduction.

[9] See proof ofs Lemma 2.12.

a method to track the progression of a single term t over an infinite reduction. We know that by construction of $\mathcal{H}^\pi$ with respect to π, any infinite reduction that corresponds to an infinite path must have infinitely many steps of the form $\mathcal{F}_i \alpha f^1_N(t_i, p)\overline{x} \to^{LN} \mathcal{F}_j \alpha p \overline{x}$. Assuming that nothing happened to p that might have changed the value of the individual term in it, we can conclude that an infinite reduction entails an infinite descending chain of natural numbers t, hence it is never the case.

2.3 Conclusion

In the previous chapter a sequent calculus ICA for cyclic Heyting Arithmetic was presented. A corresponding λY-calculus was defined in the spirit of Curry–Howard correspondence, in order to obtain a faithful representation of the computation expressed by a given proof. Thanks to the close correspondence between lambda terms and recursion schemes, in the second chapter a method to define higher-order recursion schemes was given, such that the reduction steps correspond to the rules of the initial proof π. The recursion scheme so obtained preserves the typing constraints and generates terms in the language of λ-calculus that witness the proof of the final statement. The language obtained by the reduction procedure is not affected by substitutions being expressed internally in the λ-term or directly on the proof, nor it is influenced by some modification in the proof driven by a cut-reduction strategy. Since there are no constraints on the cut-formula complexity, the order of the recursion scheme is dependant on it, in addition to the obvious connection with formulas in the final sequent. The computation induced by the recursion scheme has the expected property of confluence, and we showed that a leftmost reduction strategy guarantees the reachability of the normal form whenever a normal form exists. Unfortunately, the formalism defined is not able to conclude termination, a property of the system that we expect for recursion schemes generated by actual proofs. The impossibility to track formally the evolution of progressing traces forced us to give just an informal argument for termination.

Any potential future work originating from the present one must start by addressing termination. That is not only because of its intrinsic importance, but also because the informal argument reveals how the present correspondence between cyclic proofs and recursion schemes is not complete, an equivalent notion to the one of progressing trace still missing. Once we are sure that from any sound proof π we can compute Σ^H-terms, the analysis on the information that it is possible and desirable to extract from it can be pursued. At the present stage we can already imagine that the recursion schemes might be expanded with additional reductions to extract relevant data, even from terms not in normal form. For example, assuming a cyclic proof of a Σ_1 prenex formula, from the behaviour of the rules we know that all the individual term-witnesses can be extracted already from a term whose leftmost non-terminal has reached the first leaf, the rest of the computation to the normal form being only the definition of a λ-term for the quantifier-free formula. We can imagine an extended recursion scheme where rules are introduced to extract the desired informative terms without the necessity of a full reduction. The HORS defined here can be seen as the basis for implementation in specific contexts.

We didn't stress in this work the advantages of having an arithmetic language, nor the benefit received by the intuitionistic logic. Having to deal with just one inductive predicate, whose definition is as simple as can be, made the definition of the recursion schemes easier than it could have been in a more general setting. An interesting direction for subsequent work is represented by the generalisation to a system with a list of inductive predicates.

The choice to work with intuitionistic logic represented an advantage in terms of an obvious correspondence with λ-terms, but also for the possibility of a deterministic recursion scheme. In the present definition of the system, each rule matches the functional character of sequents: at every step there is a clear set of input-terms and only one output. In [AHL20], the choice of a classical environment resulted in a non-deterministic system, due to the fact that that direct correspondence is lost. It would be interesting to work in the direction of cyclic Peano Arithmetic, and classical cyclic proofs in general.

Part II

Closure ordinals

Introduction

We now turn to look at the semantic content of another system with fixed points by investigating the notion of closure ordinal for the modal μ-calculus (ML_μ). The system is obtained by the addition of greatest and least fixed point quantifiers (ν and μ) to propositional modal logic. To work with modal μ-calculus means looking at the crossroad of many different but contiguous areas of research: modal logic, automata theory, game theory and program verification. Its central position, together with its extremely powerful language, contributes to make ML_μ an interesting and challenging topic.

Any general overview of the origins of ML_μ retraces the different intersections with the already mentioned fields. The interests around fixed points in modal logic began in the 1970-80s, in parallel with the development of modal temporal logics and logics of programs. It immediately was entwined with automata theory and game theory, a bond from which mutual benefits resulted. There is general consensus now in appointing the work of Kozen [Koz83] as the place where modal μ-calculus is defined for what it is known today.

μ-calculus can be presented as Hennessy–Milner logic (HML) with fixed points[10] referring to its connections with dynamic logics and logics for programming. ML_μ is capable of talking about properties of labelled transition systems and tree models, hence it has been used to reason about the computation tree of given programs. Expressions can be formalised that are fundamental for talking about a program, like *safety*: “something bad will never happen” or *liveness*: “something good will eventually happen”, that cannot be stated in a simple modal logic nor in HML.

The popularity that modal μ-calculus gained is not surprising, given that the presence of fixed point operators increases substantially the expressivity of the system with respect to the rest of temporal logics. Already in the aforementioned [Koz83] it was proved that ML_μ is strictly more expressive than propositional dynamic logic PDL.[11] In the following years other temporal logics have been proved to be strictly included in ML_μ, like for example computation tree logic CTL and linear temporal logic LTL.[12] Fixed

[10]It is the case, for example, of [BS07], but also [Jun10].

[11]PDL was defined by Fischer Ladner [FL79]. Its syntax includes atomic terms for programs and ways of composing them, its purpose being to reason about their properties.

[12]The two are temporal logics on branching trees and linear models (respectively), and both have a

points can be useful or necessary also in the context of other modal logics. For example in modal epistemic logic, the definition of common knowledge involves a potentially infinite iteration of the concept of shared knowledge, and that can be achieved via a fixed point operator, see for example [Bar88].

Great expressive power comes not only from the presence of fixed points but also from the possibility of alternating and nesting them. Intuitively, infinitely many levels Π_n and Σ_n can be defined, corresponding to n levels of alternation similar to the quantifier case, having a ν or μ as outermost quantifier respectively. There are multiple definitions of hierarchy for fragments of modal μ-calculus, characterising the interactions between different fixed points in slightly different ways. In the present work the definition from Niwiński [Niw86] is chosen (see Definition 3.14). The hierarchy was proven to be strict by Bradfield [Bra98], and it is also known that the aforementioned temporal logic are expressible already at the low levels.[13] As a consequence, most of the everyday tasks of ML_μ can be performed with relatively easy level of complexity, which is a positive fact if we agree with the common saying that no one really understands μ-formulas with more than two nested fixed points. Despite our human incapability of treating nesting, another reason for the success of modal μ-calculus is due to the fact that great expressivity comes with a quite simple complexity. ML_μ is known to have the finite model property,[14] and both the model checking and the satisfiability problems are decidable, i.e., whether a formula φ holds in a given model, and the existence of a model for φ, respectively.[15]

Even though they are not a part of the present work, automata theory and game semantics gave an undeniable contribution to the study of modal μ-calculus. Many of the results mentioned above have been obtained using concepts and methods of automata and game theory. Examples of this include the satisfiability problem of μ-calculus reduced to the emptiness problem for finite automata by Street and Emerson [SE89], or the proof that ML_μ corresponds to the bisimulation invariant fragment of monadic second order logic given by Janin and Walukiewicz [JW96], as well as the model checking problem for ML_μ proved equivalent to a parity game on finite graphs [EJS93]. An axiomatisation of modal μ-calculus was given in [Koz83] and in an infinitary version in [Koz88]. The system was proved to be sound and complete in [Wal00].

Background

The present work focuses on the closure ordinals for formulas of the modal μ-calculus. The concept of closure ordinal comes directly from the approximation interpretation of fixed points formulas in connection to a Kripke models. Fixed points are usually defined

syntax that includes temporal operators to talk about paths of the model. They are known to be not equivalent, despite the fact that both are subsumed by another temporal logic: CTL*. The inclusion of the latter in ML_μ is proven in [Dam94].

[13] See the introduction of [KV03].

[14] Proved by Kozen in [Koz88]. A small model theorem was already known as a corollary of the decision procedure proposed by Street and Emerson [SE84].

[15] See [KP84].

as the solutions to equations of the form $X = F(X)$. In the context of Kripke models, positive formulas with free variables determine monotone operators on the powerset of the domain, as a result of the interpretation of the free variables. We have that a fixed point of $\varphi(x)$ is a set of states of the model such that $\|x\| = \|\varphi(x)\|$.

Given a Kripke model and a formula $\varphi(x)$, its least and greatest fixed points can be seen as the results of a process of iteration, that approximates the interpretation of the free variable starting with the empty set (μ) or the whole domain (ν). It is a process that is potentially transfinite and most importantly can be counted, and that necessarily has a point in which it stabilises: for some ordinal α the denotation of the α^{th} iteration corresponds to the denotation of the $\alpha + 1^{th}$. The value α is what we call closure ordinal of $\varphi(x)$ with respect to the given model. The notion of closure ordinal, in fact, not only depends on the formula, but it is clearly bound by the size of the model. The notion of closure ordinal in a model can naturally be extended to a more general idea of upper bound, resulting from the consideration of all possible models that satisfy the formula. Given a ML_μ formula φ, we call its closure ordinal the least ordinal that is an upper bound with respect to all possible models for φ, if such an ordinal exists. It is this last notion of closure ordinal that we will investigate.

Closure ordinals have been a topic of interest since the past 20 years. Fontaine in [Fon08] and [Fon10] investigated the relationship between continuous and constructive formulas, that is, formulas whose fixed point is always reached in a finite number of steps, or at most ω respectively. It is known that continuity entails constructivity, while the converse doesn't hold. Fontaine and Venema [FV18] provided a syntactic characterisation to several semantic properties of modal μ-formulas, from finite width or depth models to continuity. At the same time the question of the possibility of a syntactic presentation of constructive formulas is left open. In 2010 Czarnecki [Cza10] showed that each ordinal $\alpha < \omega^2$ is the closure ordinal of a Σ_1 formula in disjunctive form. On top of the importance of the result in itself, that paper has had a great relevance in furnishing a standard way of defining formulas whose approximation interpretation is bigger than ω, called *Czarnecki's formulas*, and in making almost canonical the folklore's method for building models associated with ordinals. The method used by Czarnecki to prove the existence of such formulas is extremely effective: a formula in disjunctive form is built, and each disjunct plays the role of a *fuse*. Each disjunct, in fact, has a point where it starts to hold in the model, that is where the fuse is lighted, and a point where it ceases to be satisfied, that is when the fuse is exhausted. The specific syntactic structure of the formula ensures that each fuse cannot be reused once it has been exhausted. Depending on its structure, each disjunct corresponds to the successor or the limit step. We will see how much this idea has influenced the present work.

A particular class of formulas called *primary*, with a structure similar to Czarnecki's formulas, inspired the work of Afshari and Leigh [AL13]. After having showed that such a class of formulas has a closure ordinal bounded by ω^2, with a semantic argument involving the use of a tableaux system they were able to prove that the whole alternation-free fragment of modal μ-calculus has an upper bound of ω^2 for its closure ordinals.[16] The structure of the argument can be summarised as follows. A minimal order α can be associated to each tableau of a formula φ in the alternation-free fragment of ML_μ . The order corresponds to the closure ordinal of the formula in the model constructed by the tableau. Given a formula and assumed the existence of a model with closure ordinal greater or equal to ω^2, by performing a series of substitution on top of the tableau it is possible to obtain a new model where the minimal order is increased. That being a contradiction with the assumed existence of an upper bound on the order, it follows that there is no closure ordinal greater or equal than ω^2. The structure of the argument heavily reminds the reader of the pumping lemma for regular languages (see [RS59]). The possibility to consider a tableau-proof of a formula as directly providing a model, and hence the possibility to quickly *change the lens* from a syntactic to a semantic perspective, makes the tableaux method extremely powerful. At the same time, the progression of the argument in [AL13] shows that in order to employ such a method, the details given by tableaux are not necessary: the same argument could be carried on directly on the model.

Kozen in the already mentioned [Koz88] gave a proof of the finite model theorem using a tableau-like method. The main tool used to show the existence of a finite model for every satisfiable formula is the definition of *well-annotations*. By annotating the formulas satisfied at each state of the model and saturating the corresponding sets, he was able to show the existence of a well-quasi-order between those sets of formulas. As a consequence, it is always possible to assert the existence of a finite model thanks to the properties of well-quasi-orders, and some *cut-and-paste* operation on the annotated models. The good functionality of well-annotations in dealing with substitution of sub-models was already recognised by Kozen: "the following definition of well-annotation gives local syntactic conditions that insure that states of an annotated model satisfy their labels. [...] This is useful in *performing surgery on models*, because in practice it is easily checked that these local conditions are preserved by certain cutting and pasting operations."[17]

Overview

The present work is heavily based on the ideas and methods from [AL13] and [Koz88]. In the next two chapters we will explore the possibility to replicate the argumentative structure from [AL13] using directly well-annotations to refer to models. The nature of the cut-and-paste operations that need to be performed by a pumping argument will reveal the kind of conditions that are necessary to conclude the non-existence of closure ordinals for given fragments of ML_μ . With this plan in mind, the remaining part of the work is structured as follows.

In chapter 3 we introduce the syntax and semantics of modal μ-calculus, in the language

[16] The alternation-free fragment of ML_μ corresponds to $\Sigma_2 \cap \Pi_2$ in the Niwiński hierarchy mentioned above.

[17] [Koz88] pp. 236-237, the italic is not from the original text.

with explicit ordinal annotation for the least fixed point quantifiers.[18] The definition of well-annotation is given, together with a property called *conservativity*, that corresponds to a minimality condition. The models we will be using are all conservative well-annotations, so in the central part of the chapter we establish the necessary correspondence between models and annotations. In the final part of the chapter we introduce properly the notion of closure ordinal with respect to a model and in general, and conclude with two lemmas that express the relationship between the existence of closure ordinals and conservative well-annotations.

Chapter 4 is where the defined concepts are tested and refined. A first result is obtained: a bound of ω^2 on the closure ordinal for primary formulas. The result was already given in [AL13], but this time it is given via well-annotations. This preliminary step is useful to test the usability of a pumping-like argument in the current framework, at least on a very restricted fragment. In section 4.2 there is the account of the attempt to apply the same argumentative structure to a more general level, namely disjunctive Σ_1 formulas. The attempt proved to be harder than expected, but we decided to present it anyway to motivate the changes made in the next part. Section 4.3 contains the most important part of this work: it is where we refine and apply the formalism to a more general fragment. We prove that ω^2 is a bound for formulas in the defined Σ_1^{ML} fragment, namely those formulas with only one μ-quantifier. The result does not cover the whole Σ_1, but it is still fundamental because it represents the major step toward a generalisation of the argument. An insight of the way in which the result can be obtained for formulas with multiple non-nesting least fixed points is also given. In the final part of the chapter we present the direction for future work: some final consideration about the whole Σ_1 fragment, together with some notes on the difficulties that the extension of our framework to multiple and different fixed points could bring.

[18] This language is also taken from [Koz88].

Chapter 3

ML_μ, conservativity and closure ordinals.

3.1 Syntax and semantics

We start by presenting the syntax and semantics of the modal μ-calculus. The version defined here is already the one with explicit approximants instead of μ, following Kozen's method. The formulas of the language $\mathcal{L}_\mu^+$ are defined as follows:

$$\varphi := p \mid \neg p \mid x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \alpha x.\varphi \mid \nu x.\varphi$$

VAR is a countable set of variables $x, y, \ldots$; PR is a finite set of propositional constants $p_1, p_2, p_3, \ldots$; negation is applied to propositional constants only. We define $\bot := p \wedge \neg p$. ν represents the greatest fixed point quantifier, while α is an ordinal. The formula $\alpha x.\varphi$ stands for the α^{th} approximation of the least fixed point of $\varphi(x)$. Formulas of language $\mathcal{L}_\mu$ are obtained by replacing all the occurrences of ordinals in a formula $\varphi \in \mathcal{L}_\mu^+$ with μ. Formulas are interpreted into labelled transition systems:

Definition. *A labelled transition system $T := (S, \rightarrow, \lambda)$ consists of*

- *S: a set of nodes (states);*
- *$\rightarrow$: $S \times S$ a binary relation on S;*
- *$\lambda : S \mapsto \mathcal{P}(\mathrm{PR})$ a labelling function from states to propositional constants*

In particular we restrict our interest to tree transition systems, namely transition systems such that for any node v there exists a unique u such that $u \rightarrow v$, except for a root node ρ such that $u \not\rightarrow \rho$ for any $u \in S$. The semantics for the formulas is the following: given a model T and a valuation function $\mathcal{V} : \mathrm{VAR} \to \mathcal{P}(S)$

$$\begin{array}{ll}
\|p\|^T_{\mathcal{V}} = \{v \mid p \in \lambda(v)\} & \|\neg p\|^T_{\mathcal{V}} = \{v \mid p \notin \lambda(v)\} \\
\|x\|^T_{\mathcal{V}} = \mathcal{V}(x) & \|\varphi \wedge \psi\|^T_{\mathcal{V}} = \|\varphi\|^T_{\mathcal{V}} \cap \|\psi\|^T_{\mathcal{V}} \\
\|\varphi \vee \psi\|^T_{\mathcal{V}} = \|\varphi\|^T_{\mathcal{V}} \cup \|\psi\|^T_{\mathcal{V}} & \|\Box\varphi\|^T_{\mathcal{V}} = \{s \in S \mid \forall t.s \to t \Rightarrow t \in \|\varphi\|^T_{\mathcal{V}})\} \\
\|\Diamond\varphi\|^T_{\mathcal{V}} = \{s \in S \mid \exists t.s \to t \wedge t \in \|\varphi\|^T_{\mathcal{V}})\} & \|\nu x.\varphi\|^T_{\mathcal{V}} = \bigcup\{U \subseteq S \mid U \subseteq \|\varphi\|^T_{\mathcal{V}[x \mapsto U]}\}
\end{array}$$

Together with these standard clauses, in the language $\mathcal{L}_\mu$ the denotation of a least fixed point formula is usually defined as $\|\mu x.\varphi\|^T_{\mathcal{V}} = \bigcap\{U \subseteq S | \|\varphi\|^T_{\mathcal{V}[x\mapsto U]} \subseteq U\}$. Often such definition is followed by the inductive definition of approximation

$$\begin{array}{ll}
\|0x.\varphi\|^T_{\mathcal{V}} = \emptyset & \|(\alpha+1)x.\varphi\|^T_{\mathcal{V}} = \|\varphi(\alpha x.\varphi)\|^T_{\mathcal{V}} \\
\|\lambda x.\varphi\|^T_{\mathcal{V}} = \bigcup_{\beta<\lambda} \|\varphi(\beta x.\varphi)\|^T_{\mathcal{V}} & \lambda \textit{ limit ordinal}
\end{array}$$

In $\mathcal{L}^+_\mu$ we work with explicit approximants, hence we use directly the above definition in the unified version

$$\|\alpha x.\varphi\|^T_{\mathcal{V}} = \bigcup_{\beta<\alpha} \|\varphi(\beta x.\varphi)\|^T_{\mathcal{V}}$$

and define at the meta-level $\|\mu x.\varphi\| = \bigcup_\alpha \|\alpha x.\varphi\|$ for all ordinals α. In the rest of the paper we will want to refer to a formula ψ that has some ordinal quantifier in it, e.g. $\psi = \chi \wedge \alpha x.\varphi$, by showing a superscript on ψ. The intended meaning of an expression of the form ψ^α is the formula ψ in which the ordinal quantifier is α. Since the meaning of closure ordinal for multiple occurrences of the μ-quantifier will be discussed only in the last section, we begin here by focusing on formulas of $\mathcal{L}^+_\mu$ with at most one least fixed point variable x, and the ordinal superscript will indicate the ordinal that is binding it inside the formula.

Definition 3.1. *The meaning of φ^α where $\varphi \in \mathcal{L}_\mu$ is*

$$\begin{array}{ll}
p^\alpha = p & (\neg p)^\alpha = \neg p \\
(\varphi \wedge \psi)^\alpha = \varphi^\alpha \wedge \psi^\alpha & (\varphi \vee \psi)^\alpha = \varphi^\alpha \vee \psi^\alpha \\
(\Diamond\varphi)^\alpha = \Diamond\varphi^\alpha & (\Box\varphi)^\alpha = \Box\varphi^\alpha \\
(\nu y.\varphi)^\alpha = \nu y.\varphi^\alpha & (\mu x.\varphi)^\alpha = \alpha x.\varphi
\end{array}$$

Sometimes it will be useful to refer to the structure of a formula $\varphi \in \mathcal{L}^+_\mu$ regardless of the ordinals. In those situation we use the notation φ^- to refer to the *template* formula of φ, that is, to the formula in the language $\mathcal{L}_\mu$ corresponding to $\varphi \in \mathcal{L}^+_\mu$. With templates we can easily compare two instances φ^α and φ^β of the same formula φ^- that differ in the ordinal:

Definition 3.2 ($\preceq$)**.** *We say that $\varphi \preceq \psi$ if $\varphi^- = \psi^-$ and the ordinal in φ is not greater than the ordinal in ψ. When Γ, Δ are sets of formulas, we say that $\Delta \preceq \Gamma$ if for all $\psi \in \Gamma$*

then there is a $\psi' \in \Delta$ such that $\psi' \preceq \psi$. The smallest $\Gamma' \subseteq \Gamma$ such that $\Gamma' \preceq \Gamma$ is called kernel of Γ: $Ker(\Gamma)$

When talking about subformulas of a formula $\varphi \in \mathcal{L}_\mu$ we use the Fischer-Ladner closure as common in literature.

Definition 3.3 (Fischer-Ladner closure)**.** *The Fischer Ladner closure of a formula φ in the language $\mathcal{L}_\mu$ is the smallest set such that:*

1. $\varphi \in FL(\varphi)$
2. *if* $\psi_0 \circ \psi_1 \in FL(\varphi)$ *where* $\circ \in \{\wedge, \vee\}$ *then* $\psi_0, \psi_1 \in FL(\varphi)$
3. *if* $\nabla\psi \in FL(\varphi)$ *where* $\nabla \in \{\Box, \Diamond\}$ *then* $\psi \in FL(\varphi)$
4. *if* $\sigma x.\psi \in FL(\varphi)$ *where* $\sigma \in \{\nu, \mu\}$ *then* $\psi(\sigma x.\psi) \in FL(\varphi)$

In the case of $\varphi \in \mathcal{L}_\mu^+$, that is when the language expects explicit ordinals, the definition of Fischer-Ladner closure[1] has the extra condition

5. *if* $\alpha x.\psi \in FL(\varphi)$ *then* $\psi[\beta x.\psi/x] \in FL(\varphi)$ *for all* $\beta < \alpha$.

Sometimes we may commit an abuse of terminology and refer to the formulas ψ^β such that $\psi^- \in FL(\varphi^-)$ as *subformulas of* $\varphi^\alpha \in \mathcal{L}_\mu^+$ with $\alpha \geq \beta$. We also use the condensed notation $\varphi^{\alpha \leq \beta}$ to express the fact that *for some* $\alpha \leq \beta$ *we consider* φ^α.

Finally one last assumption: all the formulas in our language are closed, and guarded in the sense of the following definition:

Definition 3.4 (Guarded)**.** *A formula φ is guarded if in every subformula $\sigma x.\psi$ of φ where $\sigma \in \{\nu, \mu\}$, every occurrence of the bound variable x occurs under the scope of some modal operator.*

It is an established fact that this does not constitute a limitation, since every formula of $\mathcal{L}_\mu$ (and hence of $\mathcal{L}_\mu^+$) is known to be equivalent to a guarded one ([NW96]).

3.2 Conservative well-annotations

To be able to talk about sets of formulas satisfied at some state and their relations we will use as a main instrument the notion of *well-annotation* as defined by Kozen [Koz88], with a modification to include the appended ordinals. Well-annotations, we said, were fundamental in the proof of the finite model theorem. However in this context we can't just consider satisfiability, we also need to ensure that the formulas with the least possible ordinal are present in the annotation. For that purpose we then refine the definition of well-annotation and introduce the notion of conservative well-annotations.

[1] This is called *strong closure* for example in [AJL19].

Definition 3.5 (Well-annotation)**.** *Given a model T, an annotation Θ is a function associating to each state s of T a set of formulas Θ_s of the language $\mathcal{L}_{\mu}^{+}$. $\hat{\Theta}$ is called a well-annotation if the following conditions hold:*

1. *if $p^{\alpha} \in \Theta_s$ ($(\neg p)^{\alpha} \in \Theta_s$), then $s \models p$ ($s \models \neg p$);*
2. *if $(\varphi \vee \psi)^{\alpha} \in \Theta_s$, then either $\varphi^{\alpha} \in \Theta_s$ or $\psi^{\alpha} \in \Theta_s$;*
3. *if $(\varphi \wedge \psi)^{\alpha} \in \Theta_s$, then both $\varphi^{\alpha} \in \Theta_s$ and $\psi^{\alpha} \in \Theta_s$;*
4. *if $(\Diamond\varphi)^{\alpha} \in \Theta_s$, then $\exists t.s \rightarrow t$ and $\exists \beta \leq \alpha.\varphi^{\beta} \in \Theta_t$;*
5. *if $(\Box\varphi)^{\alpha} \in \Theta_s$, then $\forall t.s \rightarrow t\ \exists \beta \leq \alpha.\varphi^{\beta} \in \Theta_t$;*
6. *if $(\nu x.\varphi)^{\alpha} \in \Theta_s$, then $(\varphi(\nu x.\varphi))^{\alpha} \in \Theta_s$;*
7. *if $\alpha x.\varphi \in \Theta_s$, then $\varphi(\beta x.\varphi) \in \Theta_s$ for some $\beta < \alpha$;*

It is an immediate consequence of the definition that well-annotations respect satisfaction, and that any annotation can be extended to a well-annotation.

Theorem 3.6 (Kozen)**.** *Given a model T*

a. *If Θ is a well-annotation, then $T, s \models \Theta_s$ for all $s \in S$.*

b. *If Θ is an annotation s.t. $T, s \models \Theta_s$ for all s, then Θ can be extended to a well-annotation such that $\bigcup_s \Theta_s^{-} \subseteq \bigcup_{\varphi \in \Theta_t, t \in T} FL(\varphi^{-})$*

The proof of the first part is by induction on $\varphi \in \Theta_s$ and transfinite induction on α. The second part is shown by cases. See for both [Koz88] (the proofs can be easily modified to match with the current version with appended ordinals). From now on, a well-annotation of φ is a well-annotation Θ such that $\bigcup_s \Theta_s^{-} \subseteq FL(\varphi)$. Extending the definition we could characterise the well-annotation of a set of formulas Γ. A well-annotation of φ is called *conservative* when, for every ψ^{α} that appears in some Θ_s, there is a ψ^{β} that is the least approximation of ψ that is satisfied at s. The proper definition is the following

Definition 3.7 (Conservative well-annotation)**.** *Given a model T, a well-annotation Θ of φ is conservative if for any Θ' well-annotation of φ and $s \in T$, $\Theta_s \preceq \Theta'_s$.*

Note that such definition determines another property of conservative well-annotations: for any formula $\psi^{-} \in FL(\varphi)$ that is satisfied at s, an instance of ψ^{β} has to occur in Θ_s for some β. If that is not the case, in fact, there exists another well-annotation of φ that is not comparable according to Definition 3.2, hence Definition 3.7 is not satisfied. Given the possibility of the simultaneous presence of more than one formula with the same template in the same set, e.g. $\psi^{\alpha}, \psi^{\beta}$, from now on we adopt the convention that when we talk about a formula $\psi^{\alpha} \in \Theta_s$ we are implicitly saying $\psi^{\alpha} \in Ker(\Theta_s)$, i.e. taking the occurrence of such formula in the set with the least ordinal, unless stated otherwise. A not surprising but important fact is that, just like an annotation can always be extended to a well-annotation, the existence of a well-annotation entails the existence of a conservative version:

Lemma 3.8. *For any Θ well-annotation of φ^α, there exists a conservative well-annotation $\hat{\Theta} \preceq \Theta$ of φ^γ for some $\gamma \leq \alpha$.*

Proof. In order to prove the existence of a conservative annotation of φ^γ define a new annotation $\hat{\Theta}$ such that $\hat{\Theta}_s := \{\psi^\beta | \psi^- \in FL(\varphi)$ and ψ^β the least s.t. $s \vDash \psi^\beta\}$. The new annotation is a well-annotation, as can be checked by confronting Definition 3.3 of FL and Definition 3.5. It is also conservative of φ: given any other well-annotation Θ' of φ, every formula $\psi^\delta \in \Theta'_s$ is such that $\psi^- \in FL(\varphi)$ and $s \vDash \psi^\delta$ by Theorem 3.6. By definition $\hat{\Theta}_s$ contains the least such, hence $\hat{\Theta}_s \preceq \Theta'_s$. Moreover, from the existence of a well-annotation of φ^α that ensures that $\varphi^\alpha \in \Theta_\rho$ for ρ the root state, and the fact that $\hat{\Theta}_\rho \preceq \Theta_\rho$ we know that there is $\varphi^\gamma \in \hat{\Theta}_\rho$ and $\gamma \leq \alpha$. □

Definition 3.7 above captures in a concise way the intended property of conservativity of a well-annotation, but is unfortunately not detailed enough about the specific characteristics of such annotations. For future purposes, it will be useful to have at hand the following lemmas.

Lemma 3.9. *Given Θ a conservative well-annotation and φ^α the least annotation for φ at Θ_s*

1. *if $p^\alpha(\neg p^\alpha) \in \Theta_s$ then $s \vDash p(\neg p)$* ***and*** *$\alpha = 0$;*
2. *if $(\psi_0 \vee \psi_1)^\alpha \in \Theta_s$ then $\psi_i^{\alpha_i} \in \Theta_s$ $(i \in \{0,1\})$* ***and*** *$\alpha = \min\{\alpha_0, \alpha_1\}$;*
3. *if $(\psi_0 \wedge \psi_1)^\alpha \in \Theta_s$ then there are $\psi_0^{\alpha_0} \in \Theta_s$ and $\psi_1^{\alpha_1} \in \Theta_s$* ***and*** *$\alpha = \max\{\alpha_0, \alpha_1\}$;*
4. *if $(\Diamond\varphi)^\alpha \in \Theta_s$ then there is a t such that $s \to t$ and $\varphi^\alpha \in \Theta_t$* ***and*** *$\forall t_j.(s \to t_j)$ if $t_j \vDash \varphi^{\beta_j}$ then $\beta_j \geq \alpha$;*
5. *if $(\Box\varphi)^\alpha \in \Theta_s$ then $\forall(s \to t_i).\varphi^{\beta_i} \in \Theta_{t_i}$ for some $\beta_i \leq \alpha$* ***and*** *$\alpha = \sup(\bigcup_i \beta_i)$;*
6. *if $(\nu x.\varphi)^\alpha \in \Theta_s$ then $(\varphi(\nu x.\varphi))^\alpha \in \Theta_s$;*
7. *if $\alpha x.\varphi \in \Theta_s$ then $\varphi(\beta x.\varphi) \in \Theta_s$* ***and*** *$\alpha = \beta + 1$;*

Proof. By cases:

1. if $p^\alpha(\neg p^\alpha) \in \Theta_s$ then also $p^0(\neg p^0) \in \Theta_s$, otherwise it would be possible to have a smaller annotation by adding p^0 to Θ_s;
2. if $(\psi_0 \vee \psi_1)^\alpha \in \Theta_s$ then $\psi_i^{\alpha_i} \in \Theta_s$ for i either 0 or 1, and α cannot be less than both the α_i by Definition 3.5. Assume that $\alpha > \min\{\alpha_0, \alpha_1\}$. Since $s \vDash (\psi_0 \vee \psi_1)^{\alpha_i}$ then $\Theta_s \not\preceq \Theta'_s = \Theta_s \cup \{(\psi_0 \vee \psi_1)^{\alpha_i}\}$, contradicting conservativity;
3. if $(\psi_0 \wedge \psi_1)^\alpha \in \Theta_s$ then there are $\psi_0^{\alpha_0} \in \Theta_s$ and $\psi_1^{\alpha_1} \in \Theta_s$ and α cannot be less than each of the α_i by Definition 3.5. It cannot be greater, otherwise for $\Theta'_s = \Theta_s \cup \{(\psi_0 \wedge \psi_1)^{\max\{\alpha_0,\alpha_1\}}\}$ we'd have $\Theta_s \not\preceq \Theta'_s$. Hence $\alpha = \max\{\alpha_0, \alpha_1\}$

4. if $(\Diamond\varphi)^\alpha \in \Theta_s$ then there is a t such that $s \to t$ and $\varphi^{\beta\leq\alpha} \in \Theta_t$, by Definition 3.5. For all t_j seen by s, if $t_j \vDash \varphi^{\beta_j}$ then we know that $\alpha \leq \beta_j$, otherwise we would have a $\Theta_s \not\preceq \Theta'_s = \Theta_s \cup \{(\Diamond\varphi)^{\beta_j}\}$ contradicting the conservativity of Θ. That also means that $\beta = \alpha$.

5. if $(\Box\varphi)^\alpha \in \Theta_s$, then $\forall(s \to t_i)$ there exists $\varphi^{\beta_i} \preceq \varphi^\alpha$ such that $\varphi^{\beta_i} \in \Theta_{t_i}$ by Definition 3.5. Moreover, $\alpha = \sup(\bigcup_i \beta_i)$, otherwise there exists a $\Theta'_s = \Theta_s \cup \{(\Box\varphi)^\gamma\}$ with $\gamma = \sup(\bigcup_i \beta_i) < \alpha$ that contradicts conservativity;

6. if $(\nu x.\varphi)^\alpha \in \Theta_s$ by Definition 3.5 $(\varphi(\nu x.\varphi))^\alpha \in \Theta_s$;

7. if $\alpha x.\varphi \in \Theta_s$, then $\varphi(\beta x.\varphi) \in \Theta_s$ by Definition 3.5 . The same argument seen above holds for $\Theta'_s = \Theta_s \cup \{(\beta+1)x.\varphi\}$ if $\alpha > \beta + 1$, hence the conclusion.

□

Note that in the case of disjunction (2.) the definition of conservative well-annotation ensures that, if the other disjunct $s \vDash \psi_{1-i}^{\alpha_{1-i}}$, then also $\psi_{1-i}^{\alpha_{1-i}} \in \Theta_s$. If that was not the case, in fact, we could define a $\Theta'_s = \Theta_s \cup \{\psi_{1-i}^{\alpha_{1-i}}1\}$ and have that $\Theta_s \not\preceq \Theta'_s$, which contradicts the conservativity of Θ. In case (4.) of a formula $\Diamond\varphi \in \Theta_s$, the same argument makes it necessary that $\varphi^{\beta_j} \in \Theta_{t_j}$ for all $s \to t_j$ such that $t_j \vDash \varphi^{\beta_j}$. Finally, note that (7.) has as a consequence that formulas like $\lambda x.\varphi$ -lambda a limit ordinal- cannot be the least in any well-annotated set Θ_s. Limit ordinals are introduced only when an infinite number of successor states with increasing ordinals forces $\alpha = \sup(\bigcup_i \beta_i) = \lambda$. When that happens, we can only have a formula $\Box\varphi^\lambda$ in Θ_s that becomes $(\lambda+1)x.\varphi \in \Theta_s$.

Corollary 3.10. *In any conservative Θ there are no set Θ_s, formula φ and limit ordinal λ such that $\lambda x.\varphi \in Ker(\Theta_s)$.*

The following lemma confirms that what we defined is in fact a conservative well-annotation in the sense that it contains the least satisfiable occurrence of each template formula.

Lemma 3.11 (Truth Lemma). *Given Θ conservative well-annotation, for all $s \in T$ and $\varphi^\alpha \in Ker(\Theta_s)$, then $s \nvDash \varphi^\beta$ for all $\beta < \alpha$.*

Proof. By induction on the ordinal and the formula φ^α. For $\alpha = 0$ it is vacuously true. For $\alpha > 0$

1. $\varphi \equiv p$: not possible since $p^\alpha = p^0$ by Lemma 3.9 (1.);

2. $\varphi \equiv \psi_0 \vee \psi_1$: say $\psi_0^\alpha \in \Theta_s$ and by IH $s \nvDash \psi_0^\beta$. Now either $s \nvDash \psi_1^\alpha$ and we are done, or $\psi_1^\alpha \in \Theta_s$ by Lemma 3.9 (2.) and conservativity, hence by IH $s \nvDash \psi_1^\beta$ (an analogous argument works starting with $\psi_1^\alpha \in \Theta_s$);

3. $\varphi \equiv \psi_0 \wedge \psi_1$: there are both $\psi_i^{\alpha_i} \in \Theta_s$. By Lemma 3.9(3.) $\alpha_i \leq \alpha$, and by IH $s \nvDash \psi_i^\beta$;

4. $\varphi \equiv \Diamond\psi$: there are some $\psi^\alpha \in \Theta_{t_j}$ for $s \to t_j$, and by Lemma 3.9 (4.) α is the least, meaning that none of the other successor states' annotation contains $\psi^{\beta<\alpha}$. By IH $t_j \not\models \psi^{\beta_j}$ for all $\beta_j < \alpha$, hence $s \not\models (\Diamond\psi)^\beta$

5. $\varphi \equiv \Box\psi$: given $\alpha > 1$ and conservativity, there is a number of $\psi^{\beta_i} \in \Theta_{t_i}$ for $s \to t_i$ and by Lemma 3.9 (5.) $\alpha = \sup(\bigcup_i \beta_i)$. Either there are $\varphi^\alpha \in \Theta_{t_j}$ such that $\Theta_{t_j} \not\models \varphi^\beta$ by induction hypothesis, or for any $\beta_l < \alpha$ there is a bigger β_j such that $\psi^{\beta_j} \in \Theta_{t_j}$ and by IH $t_j \not\models \psi^{\beta_l}$. In both cases $s \not\models (\Box\psi)^{\beta_l}$ for all $\beta_l < \alpha$.

6. $\varphi \equiv \nu x.\varphi$: by definition $\varphi^\alpha(\nu x.\varphi) \in \Theta_s$ and by IH $s \not\models \varphi^\beta(\nu x.\varphi)$;

7. $\varphi \equiv \alpha x.\psi$: by Lemma 3.9 (7.) $\psi(\beta x.\psi) \in \Theta_s$ and $\alpha = \beta + 1$. By induction hypotesis on the ordinal β, for all $\gamma < \beta$: $s \not\models \psi(\gamma x.\psi) \in \Theta_s$, hence $s \not\models \beta x.\psi$.

□

Before introducing closure ordinals, let's give some final definitions.

Definition 3.12. *An annotated model T is a model, together with an annotation Θ. A well-annotated model is an annotated model where Θ is a well-annotation. A conservative model is a well annotated model where Θ is conservative.*

From now on we will use Θ_s as both the set of formulas annotated at s and the state s itself (there is no risk of confusion). The final instrument that we want to define here is that of a *path* in an annotated model.

Definition 3.13 (Path)**.** *A path P trough an annotated model Θ is a sequence of states such that:*

1. *$\Theta_\rho \in P$ (Θ_ρ is the root)*
2. *if $\Theta_s \in P$ and $s \neq \rho$, then $\Theta_t \in P$ for $\Theta_t \to \Theta_s$*
3. *if $\Theta_s \in P$ either Θ_s is a leaf or there is exactly one Θ_t such that $\Theta_s \to \Theta_t \in P$*

In the next chapter we will work inside the Σ_1 fragment in the Niwiński hierarchy, hence we will not consider formulas with the ν quantifier.

Definition 3.14 (Niwiński hierarchy)**.** *A formula φ with no fixed points is in Π_0 and Σ_0. Σ_{n+1} and Π_{n+1} are defined as the closure of $\Sigma_n \cup \Pi_n$ under the following conditions (respectively):*

1. *if $\varphi, \psi \in \Sigma_{n+1}(\Pi_{n+1})$ then $\varphi \wedge \psi, \varphi \vee \psi, \Box\varphi, \Diamond\varphi \in \Sigma_{n+1}(\Pi_{n+1})$*
2. *if $\varphi \in \Sigma_{n+1}(\Pi_{n+1})$ then $\mu x.\varphi \in \Sigma_{n+1}(\nu x.\varphi \in \Pi_{n+1})$*
3. *if $\varphi, \psi \in \Sigma_{n+1}(\Pi_{n+1})$ then $\varphi(\psi) \in \Sigma_{n+1}(\Pi_{n+1})$*

in the last case we require that no capture of free variable of ψ occurs in the substitution.

3.3 Closure ordinals

We define now the notion of closure ordinal of a formula in a model. As we said in the introduction, the closure ordinal expresses the number of steps necessary to reach the fixed point in a given model

Definition 3.15 (Closure Ordinal in a model)**.** *Given a model T, for every formula φ there exists a least ordinal κ such that $\|\varphi^\kappa\|^T = \|\varphi^{\kappa+1}\|^T$. We call $\kappa = CO_T(\varphi)$ the closure ordinal of φ in T.*

When the perspective is extended to all possible models we have the general definition of closure ordinal of a formula

Definition 3.16 (Closure Ordinal)**.** *Given a formula φ, if there is a least ordinal κ such that $CO_T(\varphi) \leq \kappa$ for all possible models T then κ is the closure ordinal of φ: $CO(\varphi) = \kappa$.*

The way in which we defined conservative well-annotations is intended to match the desired relation of annotated formulas with closure ordinals, a relationship that is determined by the next important lemmas. Before turning to them, however, we need to address the case of conservativity for limit-ordinal formulas. In fact, if we can think of a model with a conservative well-annotation of $\varphi^{\kappa+1}$ simply by picturing one where $\rho \vDash \varphi^{\kappa+1}$ and $\varphi^{\kappa+1} \in \Theta_\rho$, the same is not always possible with φ^κ and κ is a limit ordinal (Corollary 3.10). Being $\kappa = \sup_i\{\beta_i\}$ a limit ordinal we know that for all $\beta_i < \kappa$ there is a conservative well-annotated model T_{β_i+1} with φ^{β_i+1} at the root, but no state satisfies φ^κ itself alone. We consider a conservative well-annotated model of a limit formula φ^κ the model Θ^* made of the disjoint union of the conservative models T_{β_i}, with an extra root element $\rho^* \to \rho_{T_{\beta_i}}$. In this model $\rho^* \vDash \Box\varphi^\kappa$, and the well-annotation such that $\varphi^{\beta_i} \in \Theta_{\rho_{\beta_i}}$ and $\Box\varphi^\kappa \in \Theta_{\rho^*}$ constitutes -for our purposes- a suitable conservative well-annotation for φ^κ. With such convention set, we can turn ourselves to the lemmas.

Lemma 3.17. *If $CO(\varphi) = \alpha$ then there exists a conservative well-annotation of φ^α.*

Proof. By transfinite induction on α. The base case of $\alpha = 0$ is trivial. If $\alpha = \kappa + 1$, by definition of CO there exists a model T such that at the root $\rho \vDash \varphi^\alpha$ and $\rho \nvDash \varphi^\beta$ for all $\beta < \alpha$. Define $\hat{\Theta}$ as $\hat{\Theta}_\rho = \{\varphi^\alpha\}$ and $\hat{\Theta}_{s_i} = \emptyset$ for any $s_i \neq \rho$. By Theorem 3.6 (b.) there exists a well-annotation of φ^α, such that all annotated formulas $\psi^- \in FL(\varphi)$. By Lemma 3.8 there is a conservative well-annotation $\hat{\Theta}^*$ of φ^γ for some $\gamma \leq \alpha$. We know that $\gamma = \alpha$ from the assumption that $\rho \nvDash \varphi^\beta$ for all $\beta < \alpha$ and part (a.) of Theorem 3.6. $\hat{\Theta}^*$ is a conservative well-annotation of φ^α.

If α is a limit and the closure ordinal of φ, the same argument holds except when it is impossible by Corollary 3.10. However, we stipulated that in those cases the conservative model is such that $\Box\varphi^\alpha \in \Theta_\rho$. By definition of CO, there exists a denumerable list of ordinals β_i such that $\sup_i\{\beta_i\} = \alpha$, and models T_{β_i+1} such that $\rho_{\beta_i+1} \vDash \varphi^{\beta_i+1}$. A new model T^* is obtained by taking the disjunct union of all the T_{β_i+1} plus a root element ρ such that $\rho \to \rho_{\beta_i+1}$. In the same way than the previous case, build a conservative well-annotation for each subtree T_{β_i+1} by setting $\varphi^{\beta_i+1} \in \Theta_{\rho_{\beta_i+1}}$ for all ρ_{β_i+1}, and then using Theorem 3.6 (b.) and Lemma 3.8. By construction $\rho \vDash \Box\varphi^{\kappa}$. With $\Box\varphi^{\kappa} \in \Theta_\rho$ we have a conservative well-annotation of φ^{κ}. □

Lemma 3.18. *If there exists a conservative well-annotation of φ^α then $CO(\varphi) \geq \alpha$.*

Proof. Let Θ be the conservative well-annotation of φ^α from the hypothesis, the model being T. $\varphi^\alpha \in Ker(\Theta_\rho)$ or $\Box\varphi^\lambda \in Ker(\Theta_\rho)$) (for some $\alpha = \lambda$). By Theorem 3.6 (a.) we know that $\rho \vDash \varphi^\alpha$ (or $\rho \vDash \Box\varphi^\lambda$). Assume $CO(\varphi) = \gamma < \alpha$. By Definitions 3.15 and 3.16 of closure ordinal $\|\varphi^\gamma\|_T{=}\|\varphi^\alpha\|_T$, so we know that $\rho \vDash \varphi^\gamma$ (or $\rho \vDash \Box\varphi^\gamma$). Define $\hat{\Theta}_\rho = \Theta_\rho \cup \{\varphi^\gamma\}$ (or $\hat{\Theta}_\rho = \Theta_\rho \cup \{\Box\varphi^\gamma\}$) and $\hat{\Theta}_{s_i} = \Theta_{s_i}$ for $s_i \neq \rho$. Extend the annotation so obtained to have a well-annotation $\hat{\Theta}'$ (Theorem 3.6 (b.)). We have that $\Theta_\rho \not\preceq \hat{\Theta}'_\rho$ contradicting the conservativity of Θ. We conclude that $CO(\varphi) \geq \alpha$.

□

The two lemmas that close the chapter will be fundamental in the final step of both proofs for primary and Σ_1^{ML} formulas. In fact, even if we don't have a perfect correspondence between well-annotations and closure ordinals, Lemma 3.17 gives us the motivation for talking about conservative models once the existence of a closure ordinal has been assumed. Lemma 3.18, on the other hand, will close the argument by giving the desired contradiction, because from the existence of a conservative model with a greater ordinal we can invalidate the initial assumption on the existence of a closure ordinal. For their fundamental role played in the general structure of the argument, they were both proved here before the specification of the fragment for which the rest of the tools are defined. We expect to make use of them again every time in which we extend the fragment. In the next chapter we begin by introducing the first fragment on which we test the possibilities of our method.

Chapter 4

A bound on closure ordinals

In this chapter we present the proof of the existence of an upper bound on closure ordinals for two fragments of modal μ calculus. The first one consists of primary formulas, that represent an ideal candidate for a first test of the concepts involved given their peculiar disjunctive structure. This structure facilitates the construction of a new model that satisfies the desired formula, limiting the possibility of a trace shifting. Trace shifting is the main topic of the second section, where we give a presentation of the problem with formulas with a general disjunctive form. The third section is the one where the set of working tools is expanded to include all the notions necessary to prove that ω^2 is the bound for Σ_1^{ML} formulas, that is the fragment with just one lest fixed point quantifier. The final sections contain a description of the case of Σ_1^W with multiple non-interactive least fixed points, and the conclusion.

4.1 Closure ordinal of primary formulas

We begin the journey towards Σ_1 formulas with a particular fragment of formulas of $\mathcal{L}_\mu^+$ called *primary formulas*. Similarly to the the case of Czarnecki's formulas, each disjunct of a primary formula can be seen as a fuse in the process of approximation of the ordinal from below.

Definition 4.1 (Primary formulas). *A formula of $\mathcal{L}_\mu$ is primary if it is of the form*

$$\varphi := \mu x.(P_1 \wedge \Box P_1' \wedge \triangledown_1 x) \vee \cdots \vee (P_n \wedge \Box P_n' \wedge \triangledown_n x) \vee \Box\bot$$

where $P_1, P_1', \dots P_n, P_n'$ are finite conjunctions of elements of $\mathrm{PR} \cup \overline{\mathrm{PR}}$, and $\triangledown_{i \leq n} \in \{\Box, \Diamond\}$. In the rest of the chapter φ stands for a primary formula as in Definition 4.1. Before moving to the main theorem, let's highlight some useful definitions that are necessary to capture the peculiar relations and properties of models and primary formulas. In particular, the notion of traces and conservative traces play a fundamental role in the proof.

Along a path we want to focus on the dependency relation between subformulas, and also between ordinals. We trace the first aspect by defining

Definition 4.2 (Trace)**.** *A trace $\mathcal{T}$ in an annotated model Θ is a sequence $(\psi_1^{\alpha_1}, \Theta_1), (\psi_2^{\alpha_2}, \Theta_2) \dots$ of pairs consisting of annotated elements of* $\mathrm{FL}(\varphi)$*, and states of a path P, such that $\psi_n^{\alpha_n} \in \Theta_n$ and, given a pair $(\psi_n^{\alpha_n}, \Theta_n)$*

1. *if $\psi_n = p$ or $\psi_n = \Box\varphi^0$, then there is no $(\psi_{n+1}^{\alpha_{n+1}}, \Theta_{n+1})$;*
2. *if $\psi_n = (\chi_0 \circ \chi_1)$ then $(\psi_{n+1}^{\alpha_{n+1}}, \Theta_{n+1}) = (\chi_i^{\beta \leq \alpha_n}, \Theta_n)$ $(i \in \{0, 1\}$ and $\circ \in \{\wedge, \vee\})$;*
3. *if $\psi_n = (\nabla\chi)$ then $(\psi_{n+1}^{\alpha_{n+1}}, \Theta_{n+1}) = (\chi^{\beta \leq \alpha_n}, \Theta_{n+1})$ and $\Theta_n \to \Theta_{n+1}$ $(\nabla \in \{\Box, \Diamond\})$;*
4. *if $\psi_n = \alpha_n x.\chi$ then $(\psi_{n+1}^{\alpha_{n+1}}, \Theta_{n+1}) = (\chi(\beta x.\chi), \Theta_n)$ and $(\beta < \alpha_n)$.*

A trace is called principal *if $\Theta_1 = \Theta_\rho$.*

To keep track of the ordinal relation, we define a specific kind of traces:

Definition 4.3 (Conservative Trace)**.** *A trace $\mathcal{T}$ in a conservative Θ is a* conservative trace *if given the first element $(\psi_1^{\alpha_1}, \Theta_1)$ of the sequence $\psi_1^{\alpha_1} \in Ker(\Theta_1)$, and for any pair $(\psi_n^{\alpha_n}, \Theta_n)$:*

1. *if $\psi_n = p$ or $\psi_n = \Box\varphi^0$, then there is no $(\psi_{n+1}^{\alpha_{n+1}}, \Theta_{n+1})$;*
2. *if $\psi_n = (\chi_0 \circ \chi_1)$ then $(\psi_{n+1}^{\alpha_{n+1}}, \Theta_{n+1}) = (\chi_i^{\alpha_n}, \Theta_n)$, $(i \in \{0, 1\}$ and $\circ \in \{\wedge, \vee\})$*
3. *if $\psi_n = (\Diamond\chi)$ then $(\psi_{n+1}^{\alpha_{n+1}}, \Theta_{n+1}) = (\chi^{\alpha_n}, \Theta_{n+1})$ and $\Theta_n \to \Theta_{n+1}$;*
4. *if $\psi_n = (\Box\chi)$ then $(\psi_{n+1}^{\alpha_{n+1}}, \Theta_{n+1}) = (\chi^{\beta \leq \alpha_n}, \Theta_{n+1})$, $\Theta_n \to \Theta_{n+1}$ with $\beta < \alpha_n$ only if $\alpha_n = \lambda$ (limit ordinal);*
5. *if $\psi_n = \alpha_n x.\chi$ then $(\psi_{n+1}^{\alpha_{n+1}}, \Theta_{n+1}) = (\chi(\beta x.\chi), \Theta_n)$ and $\alpha_n = \beta + 1$.*

Thanks to Lemma 3.9, we know that given a trace $\mathcal{T}$ in a conservative model Θ it is always possible to define a conservative trace $\hat{\mathcal{T}}$. Moreover, comparing this Definition with Lemma 3.9 and Lemma 3.11 it is straightforward that a conservative trace tracks only formulas in the kernel of each Θ_s. It is worth notice from this definition that at each step the next element of the trace is *almost* forced. It is not completely forced because a choice has to be made in the case (2) when both the subformulas have the same min/max ordinal, in cases (3) and (4) when more than one successor has the minimal/maximal formula χ^{α_n}. At this point the necessity of choosing doesn't constitute a problem, so we postpone any further comment about that to the next section. Interesting is the following lemma, that guarantees the existence of conservative traces:

Lemma 4.4. *If Θ is a conservative model of a primary φ, then for all $\varphi^\gamma \in \Theta_s$ and all $\beta < \gamma$ there is a conservative trace $\mathcal{T}$ from Θ_s such that $(\varphi^\beta, \Theta_t) \in \mathcal{T}$.*

Proof. By induction on γ and the formula $\psi \in FL(\varphi)$:

1. $\gamma = 0$ trivial;

2. $\gamma = \beta+1$. We know that $\varphi^\gamma \in \Theta_s$. The trace starts with $(\varphi^\gamma, \Theta_s)$. Now, depending on $\psi \in FL(\varphi)$:

   (a) $\psi^\gamma \equiv (\psi_0 \vee \psi_1)^\gamma$ by (2) of Lemma 3.9 there is at least one $\psi_i^{\gamma_i}$ annotated at Θ_s such that $\gamma_i = \gamma$. The next step in $\mathcal{T}$ is $(\psi_i^\gamma, \Theta_s)$;

   (b) $\psi^\gamma \equiv (\psi_0 \wedge \psi_1)^\gamma$ by (3) of Lemma 3.9 there are $\psi_i^{\gamma_i}$ annotated at Θ_s, and at least in one case $\gamma_i = \gamma$. The next step in $\mathcal{T}$ is $(\psi_i^\gamma, \Theta_s)$;

   (c) $\psi^\gamma \equiv (\Diamond\psi_0)^\gamma$ by (4) of Lemma 3.9 there is a Θ_t such that $\psi_0^\gamma \in \Theta_t$. The next step in $\mathcal{T}$ is $(\psi_0^\gamma, \Theta_t)$;

   (d) $\psi^\gamma \equiv (\Box\psi_0)^\gamma$ by (5) of Lemma 3.9 there are Θ_{t_j} such that $\psi_0^{\beta_j \leq \gamma} \in \Theta_{t_j}$. From the fact that $\gamma = \beta + 1$ we know that there is at least one Θ_{t_k} such that $\beta_k = \gamma$. The next step in $\mathcal{T}$ is $(\psi_0^\gamma, \Theta_{t_k})$;

   (e) $\psi^\gamma \equiv \gamma x.\psi_0$ by (7) of Lemma 3.9 there is $\psi_0(\beta x.\psi_0) = \psi_0^\beta \in \Theta_s$. The next step in $\mathcal{T}$ is (ψ_0^β, Θ_s);

   From the fact that at each step (a)-(d) the formula is reduced in complexity but not in the ordinal, and given the assumption that φ is primary, necessarily at some point case (e) occurs, since Θ is conservative and $\gamma > 0$. The trace from $(\varphi^\gamma, \Theta_s)$ to $(\varphi^\beta, \Theta_t)$ is conservative, and by induction hypothesis for all $\delta < \beta$ there is a conservative trace to the pair $(\varphi^\delta, \Theta_d)$.

3. $\gamma = \lambda$. (a')-(c') are identical to the successor case. In case of a box-formula now there are two possibilities: either there is a successor with the same ordinal, hence a next step like in (d) is taken, or the ordinal is decreased. This is the new clause for box:

   (d') $\psi^\gamma = (\Box\psi_0)^\gamma$ by (5) of Lemma 3.9 there are Θ_{t_j} such that $\psi_0^{\beta_j \leq \gamma} \in \Theta_{t_j}$. If there is at least one Θ_{t_k} such that $\beta_k = \gamma$, then the next step in $\mathcal{T}$ is $(\psi_0^\gamma, \Theta_{t_k})$. If instead there are infinitely many Θ_{t_j} such that $\psi_0^{\beta_j} \in \Theta_{t_j}$ and $\lambda = \sup\{\beta_j\}$, then for any β_j there is a $\beta_j \leq \beta < \lambda$ such that $\psi_0^\beta \in \Theta_{t_k}$. Any step $(\psi_0^\beta, \Theta_{t_k})$ continues the trace.

   (e') $\psi^\gamma = \lambda x.\varphi$ is not possible by Corollary 3.10.

   Once again the formula is reduced in complexity while the ordinal is untouched in steps (a')-(c'). (e') is not possible with a limit ordinal, but still an unfolding is necessary to proceed, otherwise conservativity fails. Hence at some point (d') with infinite successors is the case. Each possible trace has the ordinal decreased to some ordinal β_k, giving us $(\psi^{\beta_k}, \Theta_k)$. Given the primary structure of φ, at each state the trace produces one instance of φ. The trace from $(\varphi^\gamma, \Theta_s)$ to $(\varphi^{\beta_k}, \Theta_t)$ is conservative. The rest follows by induction hypothesis and the fact that for each $\delta < \gamma$ there is a $\beta_k > \delta$.

Note that each trace defined by the proof is indeed a conservative trace according to Definition 4.16. □

Corollary 4.5. *Assume* Θ *is a conservative model and* $\mathcal{T}$ *a conservative trace. If there are two occurrences of the same formula* φ *at different states* $(\varphi^{\alpha_s}, \Theta_s) \dots (\varphi^{\alpha_t}, \Theta_t)$ *then* $\alpha_s > \alpha_t$.

Proof. By definition of trace the ordinal never increases, hence $\alpha_s \geq \alpha_t$. As we have seen in the proof of the previous lemma, conservativity together with the assumed guardedness of formulas entails that at each step the formula in the trace is reduced in complexity with the same ordinal (cases $(a.) - (d.)$, $(a'.) - (c'.)$ and first case of $(d'.)$), reduced in the ordinal (second case of $(d.')$), or unfolded to the previous ordinal ($(e.)$). Either $(d'.)$ or $(e.)$ is always present between two occurrences of the same formula that appear on a conservative trace, hence the result. □

To reach our goal we want to isolate a characteristic that guarantees the possibility of a series of substitutions to expand the model. The existence of a trace is not enough to certify such a sequence of substitutions (not even a conservative one). First of all, for a substitution to be effective we need that the two states satisfy the very same set of formulas (modulo the ordinals). Secondly, we notice that it is not sufficient to have a finite difference between those ordinals, because that change could become irrelevant once a limit step is reached. There are particular traces that satisfy these conditions: we call them *repetition traces.*

Definition 4.6 (Repetition trace)**.** *A conservative trace* $\mathcal{T}$ *of* φ *is a* repetition trace *if there are* $(\varphi^{\alpha_r}, \Theta_r)$ *and* $(\varphi^{\alpha_s}, \Theta_s)$ *such that:*

- $(a.)$ $\Theta_s^- = \Theta_r^-$*;*
- $(b.)$ $\alpha_r > \alpha_s + n$ *for all* $0 \leq n < \omega$*;*

For practical reasons we focus our attention to

Definition 4.7 (Principal repetition trace)**.** *A repetition trace is principal if, in addition to Definition 4.6,*

- $(c.)$ Θ_r *is the root of* T

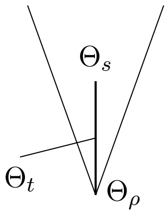
With all of this established, we can finally proceed to prove the bound for primary formulas. The main lemma of this part is the following, that guarantees an increase in the final ordinal -the first step of the process of pumping- at any time in which a conservative model has a principal repetition trace.

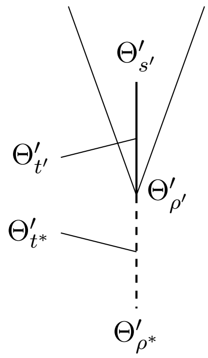
Lemma 4.8 (Increase)**.** *If there is a conservative model* Θ *for* φ^κ *that has a* principal repetition trace*, it is possible to build a conservative model for some* $\varphi^{\kappa' > \kappa}$*.*

Proof. We give a method to build a bigger model from the initial one.

1. Assume the existence of a conservative model with principal repetition trace $\mathcal{T}$, such that $\Theta_s^- = \Theta_\rho^-$, $\varphi^{\alpha_s} \in \Theta_s$ and $\alpha_s + n < \kappa$ for any natural number n.

2. Build a new model Θ' from Θ by adding to the root of the model a copy of the path-segment $\Theta_\rho \to \cdots \to \Theta_s$ from the repetition trace $\mathcal{T}$.





Notationally, in the new Θ' we call $\Theta'_{t'}$ any state above $\Theta'_{\rho'}$ corresponding to an initial Θ_t; while Θ'_{t*} is the state appended below $\Theta'_{\rho'}$ corresponding to a copy of Θ_t. The new annotated sets have the following content: $\forall \Theta'_{t'} \in \Theta'$, $\Theta'_{t'} = \Theta_t$; for all the sets Θ'_{t*}in the subpath $\Theta'_{\rho*} \to \cdots \to \Theta'_{\rho'}$ we take $\Theta'_{t*} = \Theta_t \cap \mathrm{PR}$.

3. Given the fact that the valuation function in Θ' assigns the same propositional constants to the copied states as in Θ, and given the existence of a conservative trace in the original model Θ, we know that each state along the new path satisfies at least one disjunct $(P_i \wedge \Box P'_i \wedge \nabla_i \varphi^\gamma)$. Given the particular structure of the primary formula φ, this means that the whole formula is satisfied for some ordinal γ: this is an annotation that matches the condition of Theorem 3.6 (b.), namely $\forall \chi^\beta \in \Theta'_r \Rightarrow r \vDash \chi^\beta$. We can extend Θ' to make it a well-annotation such that $\varphi^\alpha \in \Theta'_{\rho*}$. Thanks to Lemma 3.8 we define a conservative model $\hat{\Theta}$ for $\varphi^{\kappa' \leq \alpha}$. It remains to show that $\kappa' > \kappa$.

4. In order to prove it we invoke Corollary 4.5 after having showed that the trace from $\hat{\Theta}_{\rho*}$ to $\hat{\Theta}_{\rho'}$ is conservative. $\hat{\Theta}$ is conservative, and since Θ was itself conservative for $\varphi^\kappa \in \Theta_\rho$, we have in the new model that $\varphi^\kappa \in \hat{\Theta}_{\rho'}$. To check that $(\varphi^{\kappa'}, \hat{\Theta}_{\rho*}) \ldots (\varphi^\kappa, \hat{\Theta}_{\rho'})$ is conservative it is sufficient to note that each state satisfies the same propositions and (at least one of) the same disjunct(s) with respect to the original conservative trace (as seen in the previous point). It remains only to check that at each change of state the successor respects the ordinal as given by the definition of conservative trace:

   (a) for $((\Diamond\varphi)^{\alpha_j}, \hat{\Theta}_{t_j})$ we know by conservativity of $\hat{\Theta}$ that there is one successor such that $\varphi^{\alpha_j} \in \hat{\Theta}_{t_j}$, and by construction of the model we know that there is only one.

   (b) for $((\Box\varphi)^{\alpha_j}, \hat{\Theta}_{t_j})$ and $\alpha_j \neq \lambda$ it is the same.

(c) $((\Box\varphi)^{\alpha_j}, \hat{\Theta}_{t_j})$ and $\alpha_j = \lambda$ is not possible, given the construction of the model that forces just one successor, and conservativity.

5. We conclude that the path $\hat{\Theta}_{\rho^*} \to^* \hat{\Theta}_{\rho'}$ has a conservative trace, hence by Corollary 4.5 that $\kappa' > \kappa$, as desired.

□

One might think about the case of κ being a limit ordinal, and if that process works as well when the model Θ ends with a root satisfying $\Box\varphi^\lambda$, as in the intended meaning of conservative model for φ^λ. Unfortunately, the initial conditions of Lemma 4.8 exclude the case of $\kappa = \lambda$, because that would entail $\Theta_s = \{\Box\varphi^{\alpha_s}\}$ and so condition (a) of Definition 4.6 is impossible to meet. Lemma 4.8, then, can only refers to the cases in which κ is a successor ordinal. This is not a problem, because Corollary 4.10 will cover the limit case when a repetition trace is not principal (and if there is not a repetition trace then λ is the actual closure ordinal, and no pumping is possible).

The conservative model built in the proof above has not a repetition trace itself, since the ordinal κ has been increased by just a finite number, as can be seen from $4(c)$ in the proof. We know, however, that an iteration of the same process is always possible, with the consequence that a new model can be built that has a repetition trace. Before proving this fact, let's point out that in the original principal repetition trace there was (at least) one limit ordinal step, i.e. a segment $\ldots(\Box\varphi^\lambda, \Theta_p), (\varphi^{\beta_k}, \Theta_q)\ldots.$ By construction, that means that in each principal trace $\mathcal{T}$ of the new model Θ' there is a step $\ldots(\Box\varphi^\kappa, \Theta'_{p^*})(\varphi^\kappa, \Theta'_{q^*})\ldots.$ Let's call this step the *jump point* of the trace.

Lemma 4.9 (Pumping). *If there is a conservative model Θ for φ^κ that has a* principal repetition trace*, it is possible to build a conservative model for some φ^η with $\eta > \kappa + n$ for any $n < \omega$.*

Proof. Given a conservative model Θ for φ^κ and a principal repetition trace, let Θ^1 be the conservative model for $\varphi^{\kappa'}$ obtained with the process described in the proof of Lemma 4.8. The resulting Θ^1 does not have a repetition trace, because clause (b.) in Definition 4.6 is not satisfied. Note, however, that we haven't used that clause at all in the proof of the previous lemma. That will be necessary only later in this proof to induce a step to the limit. Since $(\Theta^1_{\rho'})^- = (\Theta^1_{\rho^*})^-$ we can repeat the procedure and add a copy of the path $\Theta^1_{\rho^*} \to \cdots \to \Theta^1_{\rho'}$ to the root and obtain a new model Θ^2 for some $\varphi^{\kappa''>\kappa'}$. What we obtain with Θ^1 and Θ^2 is just an increasing in the ordinal by some finite number. Define Θ^n to be the conservative model obtained after n iterations of the increasing process. Each one of them is conservative for φ with an increasing ordinal between κ' and the next limit ordinal λ.
In all models Θ^m the initial segments of the principal conservative trace are identical to the principal repetition trace of Θ. This means that each Θ^m has a first *jump point* $\ldots(\Box\varphi^{\beta_{km}}, \Theta^m_p), (\varphi^{\beta_{km}}, \Theta^m_q)\ldots$ on the principal trace, with $\beta_{km} < \beta_{kn}$ for any $m < n$. Take all the submodels $(\Theta^m \upharpoonright \Theta^m_q)$ -i.e. the submodel of Θ^m whose root state is Θ^m_q-

and build a new $\hat{\Theta}$ by adding a root state $\hat{\Theta}_r$ such that $\hat{\Theta}_r \to \Theta_q^m$ for all m. $\hat{\Theta}_r \vDash \Box\varphi^\lambda$ with $\lambda = \sup\{\beta_{km}\} > \kappa$ the next limit ordinal. Moreover: $\hat{\Theta}_r \vDash \Box P_i'$, since all the states Θ_q^m satisfy the propositions under the box of the i^{th} disjunct of φ, while $\hat{\Theta}_r \vDash P_i$ can be stipulated by construction. We conclude that $\varphi^{\lambda+1} \in \hat{\Theta}_r$. We can continue with one copy of the rest of the trace segment that was cut off from each model, and thanks to Theorem 3.6 and Lemma 3.8 we obtain a new model that is conservative for a formula φ^η, for $\eta > \kappa + \omega$. □

The last model built in the proof of the Lemma has itself a repetition trace between $\hat{\Theta}_\rho$ and each one of the Θ_s^m that started the pumping process, this means that the process can be iterated starting from φ^η, as stated by the next Corollary, until the next limit ordinal of the form ω^n. Moreover, to generalise the result we remove the constraint that the repetition trace has to be principal.

Corollary 4.10. *If there is a conservative model Θ for φ^κ with $\omega^{n-1} < \kappa < \omega^n$, that has a* repetition trace *of φ, it is possible to build a conservative model for φ^η with $\eta > \lambda > \kappa$ for any limit ordinal $\lambda < \omega^n$.*

Proof. Assume that there is a repetition trace such that $\Theta_s \to \cdots \to \Theta_t$ and $\Theta_t^- = \Theta_s^-$. It is sufficient to point out that such repetition trace is principal in the submodel $\hat{\Theta} = (\Theta \upharpoonright \Theta_s)$. Apply Lemma 4.9 to the submodel $\hat{\Theta}$ and obtain a new model $\hat{\Theta}^1$ with ordinal strictly bigger than the original, but also greater than the next limit ordinal. If that is not enough to reach an ordinal bigger than λ, the process can be iterated, as the new model is itself conservative and has a repetition trace,[1] reaching every time a conservative model $\hat{\Theta}^n$ for a bigger ordinal. We keep iterating the process, and stop only when we reach a conservative model where $\varphi^{\eta>\lambda} \in \hat{\Theta}_s^m$. □

With this additional step we removed the condition for Θ_s to be the root of the model, but we had to bound η to be lesser than ω^n. That is because any finite iteration of the process described by Lemma 4.9 gives a finite progression with respect to limit ordinals, never surpassing the limit represented by ω^n. However, the case is covered by the next Lemma.

Lemma 4.11. *If there is a conservative model Θ for $\varphi^{\kappa<\omega^n}$ that has a* repetition trace*, it is possible to build a conservative model for some φ^ξ with $\xi > \omega^n$.*

Proof. The model that is built in the proof of Corollary 4.10 has itself a repetition trace (countably many). Apply the procedure of Corollary 4.10 and build a model with ordinal η greater than the next limit greater than κ. The model so obtained has - again- a repetition trace. By iterating the process, for each limit ordinal λ between κ and ω^n we are able to build a conservative model whose annotated formula at the root has to be greater than λ. Identify in each of those models the first *jump point* and proceed to find

[1] Thanks to Lemma 4.9 we now know that condition (b.) of Definition 4.26 is satisfied after each application.

suitable submodels as in the case of Lemma 4.9: add a root to that infinite list of models. With the same process of the proof above, we have obtained a conservative model for $\varphi^{(\omega^n+1)}$. $\square$

We have reached the point in which the existence of a repetition trace in a model conservative for a primary formula φ entails the existence of conservative models of φ indefinitely bigger. It remains to define the condition that guarantees the existence of a repetition trace in a conservative model of φ.

Lemma 4.12. *Given a primary formula φ, if there is a conservative model of φ^α with $\alpha \geq \omega^2$, then Θ has a* repetition trace*.*

Proof. Let Θ be a conservative model of φ^α. By definition each Θ_s^- is subset of $\mathcal{P}(\mathrm{FL}(\varphi))$, hence there are at most $2^{|\mathrm{FL}(\varphi)|}$ sets such that $\Theta_t^- \neq \Theta_s^-$. φ is primary and by Lemma 4.4 each $\varphi^{\gamma<\alpha}$ occurs on a principal trace $\mathcal{T}$ from the root. Since $\alpha \geq \omega^2$ we know that there is a trace with more than $2^{|\mathrm{FL}(\varphi)|}$ limit formulas φ^{λ_i}. On that trace, necessarily there are $(\varphi^{\gamma_r}, \Theta_r)$ and $(\varphi^{\gamma_p}, \Theta_p)$ such that $\Theta_p^- = \Theta_r^-$ and $\gamma_p < \lambda < \gamma_r$. We found a repetition trace. $\square$

Notice that we haven't used in a strict way the fact that $\alpha \geq \omega^2$, since a bound of $\omega \cdot 2^{|\mathrm{FL}(\varphi)|}$ is enough. That brings us to the following corollary:

Corollary 4.13. *Given a primary formula φ, if there is a conservative model of φ^α with $\alpha > \omega \cdot 2^{|\mathrm{FL}(\mu x.\varphi)|}$, then Θ has a* repetition trace*.*

Finally we can prove the goal theorem:

Theorem 4.14. *The closure ordinal for any primary formula φ, if it exists, is $\alpha \leq \omega \cdot 2^{|\mathrm{FL}(\varphi)|}$.*

Proof. Define $N = \omega \cdot 2^{|\mathrm{FL}(\varphi)|}$. In searching for a contradiction say that $CO(\varphi) = \beta > N$. Lemma 3.17 ensures the existence of a conservative model of φ^β. By Corollary 4.13 we know that such conservative model has a repetition trace of φ, hence by Lemma 4.10 there is a conservative model $\hat{\Theta}$ of φ^η with $\eta > \beta$. The proof is concluded by Lemma 3.18: we know that $CO(\varphi) \geq \eta > \beta$, contradicting our initial hypothesis. $\square$

As a direct consequence of Corollary 4.13 and the fact that $|\mathrm{FL}(\varphi)|$ is always finite, we have that ω^2 is an upper bound on the closure ordinal for all primary formulas, as we expected. Despite the fact that the progression of lemmas has been tailored on the specificities of primary formulas, we could expect in principle that a simple adjustment of the definition to broader fragments would be able to give us the same result in an almost straightforward way. In the next section we will see why this is not the case, due to the problem of trace shifting after substitution.

4.2 An attempt with disjunctive formulas

This section contains an account of the attempt made to extend the argument for primary formulas to disjunctive Σ_1 formulas. Even if the completion of the task resulted to be harder than expected, leading to the decision to change the approach for a more general framework, we decided to include this account to justify the necessity of a refinement of the notions involved. It is also the opportunity to give an intuition of the problem with trace shifting, that was avoided in the case of primary formulas but couldn't be in this framework. The initial motivations for an attempt with the disjunctive fragment was the idea that the advantages given by the disjunctive form of primary formulas could be kept at a broader level. Unfortunately, that expectation was not matched, the reason being the persistence of the problem with trace shifting.

4.2.1 Trace shifting

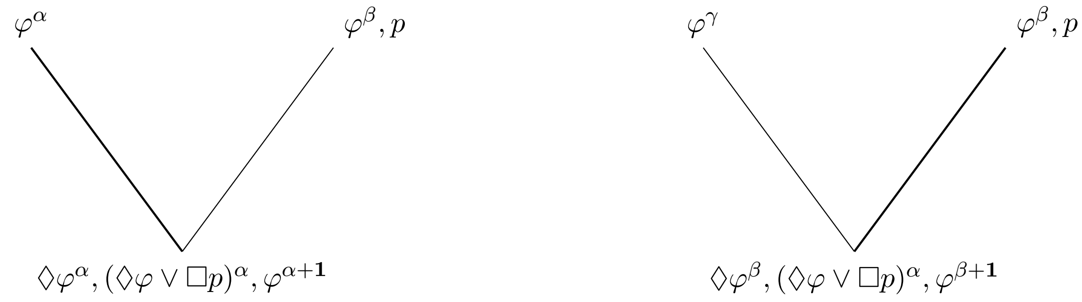
Computing the least ordinal of a formula at a given state is relatively easy, but predicting its value after a modification of the structure is a delicate matter. The ordinal, in fact, is defined on the values of the subformulas. As we pointed out in the previous section, not all subformulas nor branches are directly involved in the definition of the ordinal of a given φ. However, all of them are relevant indirectly. For example, consider

$$\varphi := \mu x.\Diamond x \vee \Box p$$

in the following situation where the model is modified by an addition of a propositional constant at one successor state:

The ordinal of φ at the root goes from $\alpha+1$ to just 1 because now $\Box p$ holds. In addition, the trace in the new model[2] could change its path and also take the right branch, since the subformula determining the ordinal is $\Box p$, and not $\Diamond\varphi$ anymore. We face two problems when we modify even a small detail of the model. The first is that a different situation about the satisfaction of subformulas could determine a different final ordinal, even when it is not affecting the satisfaction of φ. Moreover, even if we ignore the changes in the sets Θ^{-}, a second problem is that a different model could determine a different trace. Consider the same formula of the previous example and the following situations, with $\gamma > \beta > \alpha$:

[2] Signalled in the drawing with a thicker line.



Now the increase on the left from α to γ did not result in a final ordinal $\gamma + 1$ because the condition for conservativity of $\Diamond$-formulas was satisfied by the lesser β. The desired increment is not obtained, and the trace is now the one on the right. Both these possibilities obviously produce some complication when traces and substitutions are the main ingredients of our argument. In the case of primary formulas, we eliminated these possibilities by copying just the path segment from the conservative trace, cutting any possible branch so that the initial situation was freezed. We could do that because of the disjunctive structure of primary formulas. Clearly we cannot expect to replicate the very same thing with general formulas, but we might try and look at a fragment of modal μ-formulas that by definition imposes some condition to the successor states: disjunctive formulas.

4.2.2 Disjunctive formulas

Instead of working with $\mathcal{L}^+_\mu$ as before, we change the language to one with a unique modal operator called *cover modality* in place of the standard $\Box, \Diamond$, defining the new language

$$\varphi := p \mid \neg p \mid x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \alpha x.\varphi \mid \nu x.\varphi \mid \nabla\Gamma$$

where Γ is a set of formulas. The intended meaning of a formula $\nabla\{\psi_0, \ldots, \psi_n\}$ is $\bigwedge_i(\Diamond\psi_i) \wedge \Box(\bigvee_i \psi_i)$, in words corresponding to "every formula in the set is satisfied by some successor, every successor satisfies some formula of the set". The set of formulas of $\mathcal{L}^+_{\mu\nabla}$ is defined

Definition 4.15 (Σ_1 Disjunctive Formulas)**.**

- *$p, \neg p, x$ are DF;*
- *if ψ_0, ψ_1 are DF, then $\psi_0 \vee \psi_1$ is a DF;*
- *if $\psi(x)$ is a DF then $\alpha x.\psi$ is a DF;*
- *if $P \subseteq \mathrm{PR} \cup \overline{\mathrm{PR}}$ (P possibly empty) and Γ is a set of disjunctive formulas, then $\bigwedge P$ and $\bigwedge P \wedge \nabla\Gamma$ are DF.*

It is an established fact ([JW95]) that every modal formula φ can be translated into an equivalent formula with the ∇ operator, thanks to the following correspondences: $\Box p := \nabla\{p\} \vee \nabla\{\emptyset\}$ and $\Diamond p := \nabla\{p, \top\}$, where we set $\bigwedge(\Diamond\emptyset) \equiv \top$ and $\Box(\bigvee\emptyset) \equiv \Box\bot$.

The semantic interpretation of the new formulas in the usual transition systems is given by

$$\|\nabla\Gamma\|_{\mathcal{V}}^{T} = \{s \in S | \forall\varphi \in \Gamma \exists t(s \to t \wedge t \in \|\varphi\|_{\mathcal{V}}^{T}) \wedge \forall t(s \to t \Rightarrow \exists\psi \in \Gamma(t \in \|\psi\|_{\mathcal{V}}^{T}))\}$$

while for our conventional ordinal notation we stipulate that

$$(\nabla\Gamma)^{\alpha} = \nabla\Gamma^{\alpha}$$

where $\Gamma^{\alpha} := \{\psi_0^{\alpha}, \dots, \psi_n^{\alpha}\}$. Comments on the ordinal assignment to multiple formulas are postponed till the end of the chapter, but for the time being let's say that α is taken to be big enough to satisfy all the formulas in the set. It is quite straightforward to adapt all the definitions concerning conservativity to the new syntax, and prove the same theorems that we showed in chapter 3 for the new language, but we will not include them here. We limit ourselves to the definition of conservative trace for disjunctive formulas because it can be helpful in understanding the next examples:

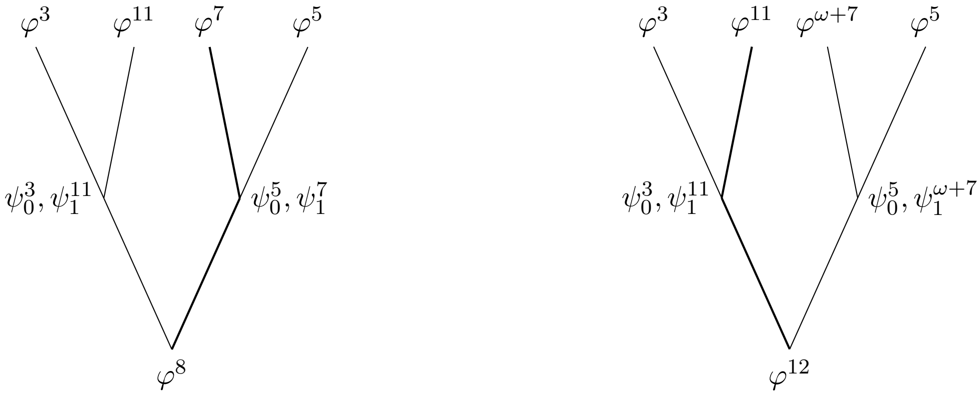
Definition 4.16 (DF Conservative Trace)**.** *A trace $\mathcal{T}$ in a conservative Θ is a* conservative trace *if given the first element $(\psi_1^{\alpha_1}, \Theta_1)$ of the sequence $\psi_1^{\alpha_1} \in Ker(\Theta_1)$, and given a pair $(\psi_n^{\alpha_n}, \Theta_n)$:*

1. *if $\psi_n = p$ or $\psi_n = \nabla\emptyset$, then there is no $(\psi_{n+1}^{\alpha_{n+1}}, \Theta_{n+1})$;*
2. *if $\psi_n = (\chi_0 \circ \chi_1)$ then $(\psi_{n+1}^{\alpha_{n+1}}, \Theta_{n+1}) = (\chi_i^{\alpha_n}, \Theta_n)$, $(i \in \{0, 1\}$ and $\circ \in \{\wedge, \vee\})$;*
3. *if $\psi_n = \alpha_n x.\chi$ then $(\psi_{n+1}^{\alpha_{n+1}}, \Theta_{n+1}) = (\chi(\beta x.\chi), \Theta_n)$ and $\alpha_n = \beta + 1$;*
4. *if $\psi_n = (\nabla\Gamma)$ then $(\psi_{n+1}^{\alpha_{n+1}}, \Theta_{n+1}) = (\chi^{\beta \leq \alpha_n}, \Theta_{n+1})$ and $\Theta_n \to \Theta_{n+1}$, $\chi \in \Gamma$ and $\beta < \alpha_n$ only if α_n is a limit ordinal.*

The definitions of repetition trace and principal repetition trace are not changed. Looking at how smoothly the new language fits in the old structure, one could expect that the same happens for the rest of the argument. That is, unfortunately, not the case. The way in which we defined disjunctive formulas, in fact, shares with primary formulas the issue of multiple disjuncts being satisfied at some successor states, with the consequent possibility that the conservative trace is changed by some modification in the above subtree. The reason why the very same procedure doesn't work is that we cannot freely cut all the branches that are not our repetition trace: because of the existential component of ∇ we need to have at least one instance of each formula in Γ that is satisfied. As a consequence, we cannot easily limit trace shifting. Here is an example: consider the formula

$$\varphi := \nabla\{\nabla\{x, \top\}, \nabla\{x\} \vee \nabla\emptyset\} \vee p$$

whose $\mathcal{L}_{\mu}^{+}$ equivalent formula is $\Diamond\Diamond x \wedge \Diamond\Box x \wedge \Box(\Diamond x \vee \Box x)$. Say $\psi_0 := \nabla\{x, \top\}$ and $\psi_1 := \nabla\{x\} \vee \nabla\emptyset$ (the trace is expressed by thicker lines in the drawings)



In the starting model we have that the formula responsible for the final ordinal (8) is ψ_1^7 on the right branch, while after the substitution, the conservativity condition for $\Diamond$ determines a trace shifting to ψ_1^{11} in the left branch. We have that the ordinal is increased, but the process cannot be automatically iterated because the trace has changed.[3]

Since a surgical modification of the model like in the primary case is not possible, we tried to translate disjunctive formulas φ into disjunctive formulas $\hat{\varphi}$, in which all logically independent formulas in the scope of a ∇ are also mutually inconsistent. In this setting, in fact, we have that each successor state satisfies one and only one of the formulas in the scope of the modality, so we would be able to respect the existential condition on the one hand, and eliminate the possibility of a trace shifting on the other. Unfortunately, we couldn't rule out the possibility that $\Theta_s \vDash \hat{\varphi}^\alpha$ and $\Theta_s \vDash \varphi^\beta$ for some $\beta < \alpha$. This is problematic because any claim we could make about the ordinal of the mutually-inconsistent disjunctive formula $\hat{\varphi}$ would have no measurable implication on the ordinal of the original formula φ. Most importantly: the reason for that, it turned out, is again the possibility of a trace shifting.

At this point we decided that the effort to establish a bound for such a peculiar fragment was exceeding the benefit of a possible success, since the method would be, again, *ad hoc* and not replicable. After this attempt, it seems that the problem of trace shifting cannot be easily tackled by completely removing the possibility of it. With the perspective of building a tool to facilitate the generalisation of the results, we then opt for a different approach. Instead of trying to remove the trace shifting, we will try to control them by restricting our attention to all the paths and nodes that we might encounter when trace shiftings occur. By ensuring an increment on all those paths, we can obtain the certainty of an increment in the final ordinal.

[3]In this simple case, we could cut the left branch and solve the problem, but we need to be sure that a solution is always possible.

4.3 Σ_1^{ML} formulas

Since it seems extremely cumbersome –if not impossible– to eliminate trace shifting we need to consider it a possible outcome of a substitution. We leave then the ∇ notation and return to the initial language. We focus now on Σ_1^{ML}, that is the fragment of Σ_1 with only one fixed point quantifier.

Definition 4.17 (Σ_1^{ML} formulas)**.** *The set of formulas of Σ_1^{ML} is defined by*

$$\varphi := p \mid \neg p \mid x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \alpha x.\psi$$

where ψ is a formula of modal logic.

If we have to accept the possibility of trace shifting after a substitution, we need to find a way to track all the possible traces that a series of substitution might determine. In order to find a set of states in the model that are sufficient to pump in order to produce an ordinal increment at the root, we need to establish a dependency relation between states in a model. Let's introduce the definition of a bar over a state in the model. Given a model $\Theta = (S, \to, \lambda)$ and a state s, we define a bar T:

Definition 4.18 (Bar)**.** *A* bar *over s is a set T of pairwise incomparable elements of S (wrt $\to$) such that*

1. $\forall t \in T\ t \not\to s$
2. *every infinite path from s intersects T*

We denote with $\text{Bar}_\Theta(s)$ the set of bars over a state s in Θ (from now on omit the model when it is not necessary). Note that by definition we have that $\{s\} \in \text{Bar}_\Theta(s)$ for all models Θ, and no other bar contains the state s. In a model with no infinite paths, like the ones we are interested at the present time, the set of all leaves is also in $\text{Bar}_\Theta(s)$ (as well as any set of incomparable elements that are not below s). On the other hand, there is no restriction on the size of T, that might also be infinite. Later on we will restrict our interest to a specific subset of $\text{Bar}_\Theta(s)$ to perform the substitutions that are necessary to increase the final ordinal. The purpose of a bar is the ability to restrict, or *prune*, the model below a certain limit. Once we have defined a bar T over a state s in Θ we can focus on the subtree below the bar, indicated by Θ_T. In particular we are interested in the dependency relation that the ordinals at a given state s share with the ordinals assigned to formulas at states in T.

Let's define a set of ordinal assignments $\mathcal{A} := \{\bar{\mathsf{a}} \mid \bar{\mathsf{a}} : (\Theta \times \text{Form}) \to \text{Ord}\}$ as a set of functions that for each state and formula of a model produce an ordinal. We define then $\mathcal{A}_T$ for any T set of states as the set of ordinal assignments relative to it: $\mathcal{A}_T := \{\bar{\mathsf{a}} \mid \bar{\mathsf{a}} : T \to \mathcal{A}\}$. Finally, we use $\mathsf{a}_t(x) : \Theta_t^- \to \text{Ord}$ to indicate a formula assignment at a specific state t. With this notation we can express as a function the dependency between bars and ordinal annotations.

Definition 4.19. *Let the function $g : \mathcal{A}_X \to \mathcal{A}$ take an assignment from any set X of states and produce a new assignment $\mathcal{A}$. For every set T and single state r, the function $g_{r,T} : \mathcal{A} \to \mathsf{a}_r$ is an ordinal assignment at state r that depends on the assignment at T. The function is defined as follows for every $\bar{\mathsf{a}} \in \mathcal{A}_T$ and ψ in Θ_r:*

1. *if $r \in T$ then $g_{r,T}(\bar{\mathsf{a}}, \psi) = \bar{\mathsf{a}}(r, \psi) = \mathsf{a}_r(\psi)$;*
2. *if $r \notin T$ and there is no s such that $r \to s$, by induction on ψ*

$$g_{r,T}(\bar{\mathsf{a}}, p) := 0 \qquad g_{r,T}(\bar{\mathsf{a}}, \Box\psi) := 0$$
$$g_{r,T}(\bar{\mathsf{a}}, \psi_0 \vee \psi_1) := \min(g_{r,T}(\bar{\mathsf{a}}, \psi_0), g_{r,T}(\bar{\mathsf{a}}, \psi_1)) \quad g_{r,T}(\bar{\mathsf{a}}, \sigma x.\psi) := g_{r,T}(\bar{\mathsf{a}}, \psi) + 1$$
$$g_{r,T}(\bar{\mathsf{a}}, \psi_0 \wedge \psi_1) := \sup(g_{r,T}(\bar{\mathsf{a}}, \psi_0), g_{r,T}(\bar{\mathsf{a}}, \psi_1))$$

while $g_{r,T}(\bar{\mathsf{a}}, \Diamond\psi)$ is not defined at leaf states since $\Theta_r \not\models \Diamond\psi$.

3. *if $r \notin T$ and r has $r_0, r_1, \ldots$ successors*

$$g_{r,T}(\bar{\mathsf{a}}, p) := 0 \qquad g_{r,T}(\bar{\mathsf{a}}, \Diamond\psi) := \min(g_{r_j,T}(\bar{\mathsf{a}}, \psi))$$
$$g_{r,T}(\bar{\mathsf{a}}, \psi_0 \vee \psi_1) := \min(g_{r,T}(\bar{\mathsf{a}}, \psi_0), g_{r,T}(\bar{\mathsf{a}}, \psi_1)) \quad g_{r,T}(\bar{\mathsf{a}}, \Box\psi) := \sup(g_{r_j,T}(\bar{\mathsf{a}}, \psi))$$
$$g_{r,T}(\bar{\mathsf{a}}, \psi_0 \wedge \psi_1) := \sup(g_{r,T}(\bar{\mathsf{a}}, \psi_0), g_{r,T}(\bar{\mathsf{a}}, \psi_1)) \quad g_{r,T}(\bar{\mathsf{a}}, \sigma x.\psi) := g_{r,T}(\bar{\mathsf{a}}, \psi) + 1$$

the r_j ranging over all the successor states of r.

If we choose a set T to be a bar with $\bar{\mathsf{a}} \in \mathcal{A}_T$, we have that the function $g_{r,T}$ defines the assignment to every formula ψ across the model depending on the actual values of $\bar{\mathsf{a}}$. Clearly we expect the assignment $g_{s,T}(\bar{\mathsf{a}}, \varphi)$ to correspond to the least ordinal satisfying φ at s in Θ whenever $\bar{\mathsf{a}}$ matches the conservative annotation at the bar states.

Lemma 4.20. *Given Θ a conservative well-annotation, for any $s \in \Theta$ and $T \in Bar(s)$, if we take $\bar{\mathsf{a}} \in \mathcal{A}_T$ such that $\bar{\mathsf{a}}(r, \varphi) = \beta$ iff $\varphi^\beta \in \mathrm{Ker}\Theta_r$ for all $r \in T$, then*

$$g_{s,T}(\bar{\mathsf{a}}, \varphi) = \beta \iff \varphi^\beta \in \mathrm{Ker}\Theta_s$$

for all formulas φ.

Proof. Inductively on the distance[4] of s from T and the complexity of φ. The base step $T = \{s\}$ is trivial by point (1.) of Definition 4.19 and the assumption on $\bar{\mathsf{a}}$. The induction steps follow by looking at the clauses of Definition 4.19 and the corresponding clauses in the Lemma 3.9 defining conservative well-annotations, that is, the conditions for $\varphi^\beta \in \mathrm{Ker}\Theta_s$ in a conservative annotation. In the case of $\varphi \equiv \Box\psi$, for example, we have by definition $g_{s,T}(\bar{\mathsf{a}}, \Box\psi) = \sup(g_{s_j,T}(\bar{\mathsf{a}}, \psi))$. By induction hypothesis $g_{s_j,T}(\bar{\mathsf{a}}, \psi) = \beta_j \iff \psi^{\beta_j} \in \Theta_{s_j}$, and by Lemma 3.9 (5) $\sup(\beta_j) = \beta$ iff $\psi^\beta \in \mathrm{Ker}\Theta_s$, hence

$$g_{s,T}(\bar{\mathsf{a}}, \Box\psi) = \beta \iff (\Box\psi)^\beta \in \mathrm{Ker}\Theta_s$$

[4]We haven't properly defined the notion of distance from a bar, but since there are not infinite paths we just use an informal notion, that could be for example the sum of the length of all the paths ending in a state in the bar T.

$\square$

We know, then, that the function works as desired: it gives the same least ordinal that satisfies a formula φ, like the conservative annotation does, whenever the assignment at the bar does the same, so it faithfully represents the dependency relation between bars and the rest of states, in terms of ordinals. Some useful properties of the function $g_{s,T}$ are expressed in the next lemma, but first let's define the relation $\bar{\mathsf{a}} \preceq \bar{\mathsf{b}}$ between two assignments over the same bar T if and only if for all states t and formulas φ

$$\bar{\mathsf{a}}(t, \varphi) \leq \bar{\mathsf{b}}(t, \varphi)$$

Lemma 4.21. *Given a conservative model Θ, for every state s and every bar T:*

a. *if there is not a path from s to $t \in T$ then for every $\bar{\mathsf{a}} \in \mathcal{A}_T$:*

$$g_{s,T}(\bar{\mathsf{a}}) = g_{s,T\backslash\{t\}}(\bar{\mathsf{a}} \backslash \mathsf{a}_t)$$

b. *$g_{s,T}$ is monotone wrt $\preceq$*

c. *$g_{s,T}$ is closed under composition: assume $T \in \mathrm{Bar}(s)$, $t \in T$, $T' \in \mathrm{Bar}(t)$ and $\hat{T} = T\backslash\{t\} \cup T'$. Denote with $\bar{\mathsf{a}}$ the assignments for $T\backslash\{t\}$, with $\bar{\mathsf{a}}'$ for T' and $\hat{\bar{\mathsf{a}}}$ for $\hat{T}$ such that they all agree on the assignments in the shared states.*

$$g_{s,\hat{T}}(\hat{\bar{\mathsf{a}}}) = g_{s,T}(\bar{\mathsf{a}}, g_{t,T'}(\bar{\mathsf{a}}'))$$

Proof. a. by induction on the Definition 4.19. The base cases are straightforward, the situation (1.) where $s \in T$ being

$$g_{s,T}(\bar{\mathsf{a}})(\psi) = \bar{\mathsf{a}}(s, \psi) = \bar{\mathsf{a}} \backslash \mathsf{a}_t(s, \psi) = g_{s,T\backslash\{t\}}(\bar{\mathsf{a}} \backslash \mathsf{a}_t, \psi)$$

Case (2.) instead is independent of T, hence trivially true. Case (3.) is also direct from the induction hypotheses, for example:

$$\begin{aligned} g_{s,T}(\bar{\mathsf{a}}, \psi_0 \vee \psi_1) &=_{df} \min(g_{s,T}(\bar{\mathsf{a}}, \psi_0), g_{s,T}(\bar{\mathsf{a}}, \psi_1)) \\ &=_{IH} \min(g_{s,T\backslash\{t\}}(\bar{\mathsf{a}} \backslash \mathsf{a}_t, \psi_0), g_{s,T\backslash\{t\}}(\bar{\mathsf{a}} \backslash \mathsf{a}_t, \psi_1)) \\ &=_{df} g_{s,T\backslash\{t\}}(\bar{\mathsf{a}} \backslash \mathsf{a}_t, \psi_0 \vee \psi_1) \end{aligned}$$

In the case of a modality, it is sufficient to point out that if $s \not\twoheadrightarrow t$ then also any successor state $s_j \not\twoheadrightarrow t$, and then proceed by induction.

b. Assume a bar T and $\bar{\mathsf{a}} \preceq \bar{\mathsf{b}}$. By induction on Definition 4.19 we can prove that for all s and ψ

$$g_{s,T}(\bar{\mathsf{a}}, \psi) \preceq g_{s,T}(\bar{\mathsf{b}}, \psi)$$

as follows:

1. for $s \in T$ we have $g_{s,T}(\bar{\mathsf{a}}, \psi) =_{df} \bar{\mathsf{a}}(s, \psi) \preceq \bar{\mathsf{b}}(s, \psi) =_{df} g_{s,T}(\bar{\mathsf{b}}, \psi)$
2. for $s \notin T$ a leaf, the assignment is independent from the bar, hence $g_{s,T}(\bar{\mathsf{a}}) =_{df} g_{s,T}(\bar{\mathsf{b}})$
3. the other cases by induction hypothesis. As an example the modalities:

$$
\begin{aligned}
g_{s,T}(\bar{\mathsf{a}}, \Diamond\psi) &=_{df} \min(g_{s_j,T}(\bar{\mathsf{a}}, \psi)) & g_{s,T}(\bar{\mathsf{a}}, \Box\psi) &=_{df} \sup(g_{s_j,T}(\bar{\mathsf{a}}, \psi)) \\
&\preceq_{IH} \min(g_{s_j,T}(\bar{\mathsf{b}}, \psi)) & &\preceq_{IH} \sup(g_{s_j,T}(\bar{\mathsf{b}}, \psi)) \\
&=_{df} g_{s,T}(\bar{\mathsf{b}}, \Diamond\psi) & &=_{df} g_{s,T}(\bar{\mathsf{b}}, \Box\psi)
\end{aligned}
$$

The rest of the cases conclude that proof of the claim.

c. By induction on the definition of $g_{s,T}$:

1. we have that $s = t$, so $T\backslash\{s\} = \emptyset$. By definition $\hat{T} = T'$ and $\hat{\bar{\mathsf{a}}} = \bar{\mathsf{a}}'$, hence $g_{s,\hat{T}}(\hat{\bar{\mathsf{a}}}) = g_{t,T'}(\bar{\mathsf{a}}') = g_{s,T}(g_{t,T'}(\bar{\mathsf{a}}'))$, the second equivalence holds since by definition $g_{s,T}(\bar{\mathsf{a}}, \psi) = \bar{\mathsf{a}}(s, \psi)$ for $s \in T$;
2. if s is a leaf, then also by definition $g_{s,\hat{T}}(\hat{\bar{\mathsf{a}}}) = g_{s,T}(\bar{\mathsf{a}}, g_{t,T'}(\bar{\mathsf{a}}'))$ since any function $g_{s,T}/g_{s,\hat{T}}$ is locally defined independently from the bar assignment;
3. the inductive step we relies on the hypothesis for the subformulas, with a help in the case of modalities from part (1) of this Lemma already proven above, for example:
$$g_{s,\hat{T}}(\hat{\bar{\mathsf{a}}}, \Diamond\psi) =_{df} \min(g_{s_j,\hat{T}}(\hat{\bar{\mathsf{a}}}, \psi))$$
   - if $s_j \not\to t$ then by (1) above we can remove any reference to T' and have $\min(g_{s_j,\hat{T}}(\hat{\bar{\mathsf{a}}}, \psi)) = \min(g_{s_j,T}(\bar{\mathsf{a}}, \psi)) = g_{s,T}(\bar{\mathsf{a}}, \Diamond\psi)$. By the same (1) we also know that $g_{s,T}(\bar{\mathsf{a}}, \Diamond\psi) = g_{s,T}(\bar{\mathsf{a}}, g_{s,T'}(\bar{\mathsf{a}}'), \Diamond\psi)$;
   - if $s_j \to t$ then by IH $g_{s_j,\hat{T}}(\hat{\bar{\mathsf{a}}}, \psi) = g_{s_j,T}(\bar{\mathsf{a}}, g_{t,T'}(\bar{\mathsf{a}}'), \psi)$

   hence the result. The same argument works for $g_{s_j,\hat{T}}(\hat{\bar{\mathsf{a}}}, \Box\psi)$.

□

Now that we have defined $g_{s,T}$ we can look at the model. We already know that, in order to have a working definition of a repetition condition, we cannot restrict ourselves to traces nor conservative traces: these might change after a substitution and we don't have a method to avoid that. On the other hand, we want to identify the cases in which a substitution is considered (and later known) to produce some effective increase in the ordinal. What is necessary is to find an intermediate level, one that isolates precisely that subset of states $\Phi \subseteq \Theta$ that contains all the formulas potentially sufficient for an ordinal increment at our desired state. Before doing that, let's define the *ordinal neighbourhood* (or level) of a formula ψ^β as the set of ordinals between (and including) the first limit ordinal λ smaller than β and the next one λ':

$$[\beta) := \{\gamma \mid \beta = \gamma + n \text{ or } \gamma = \beta + n \text{ for some } 0 \leq n < \omega\}$$

The definition of the structure Φ requires a detailed presentation, so it is worth to give some preliminary justification. We will be extracting all the relevant traces from the end formula that we want to pump. Each state Φ_s will be a set of formulas $\Phi_s \subseteq \Theta_s$, and the transition relations of Φ will be determined by the structure of Θ and the formulas in each Φ_s. In order to keep under control the ordinal neighbourhood of each Φ_s, we will begin the definition of each state in Φ by determining some initial sets $\Phi^\star_s$ of formulas all of the same ordinal level. After the decomposition into subformulas, we will cut all the formulas with an ordinal not in the initial neighbourhood. Since in some particular cases we will need to select multiple initial states for the same Φ_s in order to control the ordinal level, the final Φ_s will be the union of all the $\Phi^\star_s$. Once Φ_s has been finally determined, from the modal formulas in the set we proceed to define the initial set(s) of the next states, according to the model Θ.

Definition 4.22 (Structure Φ)**.** *From a conservative Θ and a formula φ^α at the root, we can define a tree structure Φ that has an accessibility relation taken from Θ, such that $\Phi_s \to \Phi_t$ only if $\Theta_s \to \Theta_t$, and the Φ_s extracted from Θ with the following method:*

1. *Starting from the root of the model, take the singleton $\{\varphi^\alpha\} = \Phi^\star_\rho$ as the initial set at the root.*
2. *from each initial set $\Phi^\star_s$ proceed with the decomposition of the formulas according to the definition of the FL closure, and include all the subformulas satisfied at Θ_s in $\Phi^\star_s$*
3. *remove from each $\Phi^\star_s$ all those formulas whose ordinal is not in the same initial neighbourhood, which was unique. Then finally define the set $\Phi_s := \bigcup \Phi^\star_s$*
4. *determine the initial set of the states visible from Φ_s among the visible states of Θ_s by looking at the set of modal formulas in Φ_s in the following order:*
   (a) *for every formula $\Diamond\psi_i^{\alpha_i} \in \Phi_s$ and reachable state $\Theta_t \vDash \psi_i^{\beta_i}$ with $[\alpha_i) = [\beta_i)$, let $\psi_i^{\beta_i} \in \Phi^\star_t$ if there already exists a state Φ_t, otherwise define Φ_t such that $\Phi_s \to \Phi_t$ and $\psi_i^{\beta_i} \in \Phi^\star_t$;*
   (b) *for every formula $\Box\psi_i^{\alpha_i} \in \Phi_s$ and reachable state $\Theta_t \vDash \psi_i^{\alpha_i}$ let $\psi_i^{\alpha_i} \in \Phi^\star_t$ if Φ_t exists already, otherwise define it as in (a);*
   (c) *for every formula $\Box\psi_i^{\alpha_i} \in \Phi_s$ such that no reachable state $\Theta_t \vDash \psi_i^{\alpha_i}$, for all states s.t. $\Theta_r \vDash \psi_i^{\beta_i}$ define a new Φ_r such that $\Phi_s \to \Phi_r$ and $\psi_i^{\beta_i} \in \Phi^\star_r$;*
5. *repeat the procedure from (2.) with the initial states just defined.*

A few comments are necessary at this point, before we proceed to prove some properties of Φ. The purpose of this procedure is to restrict the focus to those paths that *at the given moment* determine the level of the final ordinal. The goal of the definition of Φ is to keep only those paths that either determine the final ordinal, or that can potentially do that after a substitution has been performed. In other words: we are isolating those paths whose ordinal is *sufficient* to increase if we want to be sure that the increment propagates down to the root. Starting at the root with φ, at each state we define the sets of formulas that we are tracking -the *initial formulas*- and the corresponding ordinal level, and proceed with the backtracking of the dependency. Before moving to the next state, we get rid of all the subformulas obtained such that their level is of an interval higher or lower that the initial one. The motivation for this is the following:

- if the level is lower, then it is the result of the decomposition of a conjunction. Since for a conjunction we will always take the maximum ordinal of its subformulas, it is enough to increase the other conjunct, hence we discard the smaller formulas;
- conversely, if the level is higher, we know that the subformula comes from a disjunction. An increase in that term would produce no effect if the other disjunct is not increased at least to the same level. Hence we ignore that path at the moment.

Once we have restricted the set of formulas to those of the appropriate level, we need to determine which of the visible states are relevant, and for each one of them we list the sets of initial formulas. In order to keep control over the ordinal progression, we need to ensure that each of the initial sets has a unique level, otherwise step 3 cannot proceed. That is the reason for the case distinction of step 4. In order to ensure that all the next initial $\Phi_t^\star$ have a definite ordinal level, we (a) define one Φ_t for each state Θ_t that satisfies some diamond formula(s). The next step (b) consists of adding those formulas in the scope of a box that are not limit cases[5] by first considering the possibility of an already existing set, then creating one if that is not the case. Finally the limit case (c) defines a new set for each visible state. It is possible, then, that we have two sets $\Phi_s^\star$ and $\Phi_{s'}^\star$ both corresponding to the same Θ_s. That happens when there is a set Φ_s defined at 4 (a) or (b) that is duplicated by (c). The situation is not problematic: the two $\Phi_s^\star$ and $\Phi_{s'}^\star$ have different ordinal levels by construction. When step 3 is performed again, after each set has been restricted to the proper initial level, the two are merged again into Φ_s, from which the modal formulas are taken. For future convenience, we agree that

Definition 4.23. *Any state $\Phi_s \in \Phi$ has a unique ordinal level that is the maximum level of its formulas. The ordinal level of a state in Φ is called its order.*

An important remark is necessary at this point. For how it is built, Φ is not necessarily a model for the initial formula φ, and that is not its purpose. The structure Φ is a tree structure for which the definitions of $\mathrm{Bar}(s)$ applies, and with some additional care also the function $g_{s,T}$. The following lemma guarantees that the ordinal assignment given by

[5] Note that it might be the case that the ordinal α_i is a limit ordinal, but the step is not a limit step because some other modality must be considered first.

the function g in Φ corresponds to those in Θ when the bars and assignment share the same values on the common states.

Lemma 4.24. *Given a model Θ, let Φ be defined as in Def. 4.22. For every bar $\hat{T}$ and bar assignment $\hat{\bar{\mathsf{a}}}$ in Φ, let T be a bar in Θ such that $\Phi_s \in \hat{T} \Rightarrow \Theta_s \in T$, and $\bar{\mathsf{a}}$ an assignment such that $\hat{\bar{\mathsf{a}}}(\Phi_r, \psi) = \bar{\mathsf{a}}(\Theta_r, \psi)$ for all $\Phi_r \in \hat{T}$. Then*

$$g_{s,T}(\bar{\mathsf{a}}, \psi) = g_{\Phi_s,\hat{T}}(\hat{\bar{\mathsf{a}}}, \psi)$$

for all $\Phi_s \in \Phi$ and ψ.

Proof. By induction on the definition of g. The base cases are trivial. For the induction step, here an example with $\vee$ and one with $\Box$.

- $\psi \equiv \psi_0 \vee \psi_1$

$$\begin{aligned} g_{\Phi_s,\hat{T}}(\hat{\bar{\mathsf{a}}}, \psi_0 \vee \psi_1) &=_{df} \min(g_{\Phi_s,\hat{T}}(\hat{\bar{\mathsf{a}}}, \psi_i)) \\ &=_{IH} \min(g_{s,T}(\bar{\mathsf{a}}, \psi_i)) \\ &=_{df} g_{s,T}(\bar{\mathsf{a}}, \psi_0 \vee \psi_1) \end{aligned}$$

- $\psi \equiv \Box\varphi$

$$g_{\Phi_s,\hat{T}}(\hat{\bar{\mathsf{a}}}, \Box\varphi) =_{df} \sup(g_{\Phi_{s_j},\hat{T}}(\hat{\bar{\mathsf{a}}}, \varphi))$$

  By Definition 4.22 there exist $\Phi_{s_j} \vDash \varphi$, and by induction hypothesis $g_{\Phi_{s_j},\hat{T}}(\hat{\bar{\mathsf{a}}}, \varphi) = g_{s_j,T}(\bar{\mathsf{a}}, \Box\varphi)$. Since all the existing visible states Θ_{s_k} satisfying φ either have a Φ_{s_k} or a lesser ordinal by construction of Φ, we conclude that

$$\sup(g_{\Phi_{s_j},\hat{T}}(\hat{\bar{\mathsf{a}}}, \varphi)) = \sup(g_{s_j,T}(\bar{\mathsf{a}}, \varphi)) =_{df} g_{s,T}(\bar{\mathsf{a}}, \varphi))$$

□

The lemma certifies that the structure Φ captures the ordinal dependency relation between formulas as expressed by the function g. The next corollary highlights precisely this fact: the assignments at the root correspond in both Θ and Φ, with respect to T and $\hat{T}$.

Corollary 4.25. *Given a conservative annotation Θ of φ, $g_{\rho,T}(\bar{\mathsf{a}}, \varphi) = g_{\Phi_\rho,\hat{T}}(\hat{\bar{\mathsf{a}}}, \varphi)$ for any $T, \hat{T}$ and $\bar{\mathsf{a}}, \hat{\bar{\mathsf{a}}}$ as in Lemma 4.24.*

Proof. From Lemma 4.24 and the fact that $\varphi \in \Phi_\rho$ by construction. □

If it is true that Φ captures the essential structure for pumping, we can characterise the condition for a good repetition using Φ:

Definition 4.26 (Repetition condition)**.** *Given a conservative model* Θ*, two states* $\Theta_r \twoheadrightarrow \Theta_s$ *on a path* P *are* repetition states *if:*

(a.) $\Theta_r^- = \Theta_s^-$*;*

(b.) $\Phi_r^- = \Phi_s^-$ *and the first has a higher order than the second.*

The first condition is the same of the primary formulas, and it is necessary to ensure that the substitution will not be problematic with respect to satisfaction of formulas. The second condition ensures that after the substitution all the relevant formulas of Θ_s will be actually increased in their order. Note that the mere facts that (a.) and that there exists a corresponding path in Φ between the two states is not a guarantee that the two sets of relevant formulas coincide.

Now that we have defined the condition for having a series of fruitful substitutions, the argument proceeds as follows: we take a conservative model that is big enough and define a $T \in \mathrm{Bar}(\rho)$ from the root that ensures that a substitution is possible on all the necessary paths. Such a T is given by a number N of limit steps encountered on a path, that entails the presence of a pair of repetition states. We then prove that the substitutions occurring at the repetition stats ensure the increment of the order in Φ_ρ, hence the existence of a model with bigger closure ordinal.

Definition 4.27. *Given a conservative annotation* Θ *with root* ρ*, define a* $T \in \mathrm{Bar}(\rho)$ *by taking from all paths*

- *the first* Θ_{t_j} *such that for some* $\Phi_{s_j} \to \Phi_{t_j}$*,* $(\Theta_{s_j}, \Theta_{t_j})$ *is a pair of repetition states, or if there isn't one*
- *take the first state* Θ_{t_k} *after* N *limit steps from the root, or*
- *the leaf of the path, if the path is shorter.*

N is given by the size of the closure of φ as in the case of primary formulas, but this time we need a bigger limit to ensure that both conditions (a.) and (b.) of Definition 4.26 are met, so $N = 2^{2\cdot|FL(\varphi)|}$. Pruning the model above T determines a new tree where all the leaves Θ_t are in one of the following situations:

1. Θ_t is in a repetition pair with a corresponding state below
2. there is no Φ_t in Φ for Θ_t
3. Θ_t was already a leaf in Θ.

as a consequence of Definition 4.27. Notice that in the second and third case we don't expect the substitution to be possible, nor it is necessary for the final result. Let's proceed with the theorem that gives us an increase in the final order. Given a model Θ we call Θ_T the model pruned at T.

Lemma 4.28 (Increase)**.** *Let* N *be as above. Given a conservative model* Θ *for a formula* φ^α*, if* $\alpha > \omega \cdot N$ *then there is a conservative model* Θ' *for* $\varphi^{\alpha' > \alpha}$*. Moreover,* $\alpha' \geq \alpha + \omega$*.*

Proof. The fact that $\alpha > \omega \cdot N$ guarantees that we can find a bar $T \in \mathrm{Bar}(\rho)$ as in Definition 4.27 and define the pruned tree Θ_T. Any repetition pair by definition has a corresponding pair in Φ and by Lemma 4.24 the closure ordinal α of φ in Θ_T depends on the initial assignment $\bar{\mathsf{a}}$ at the nodes $t_j \in \Phi$. We can define a model in which each subtree above a repetition pair in Θ is replaced by the subtree generated by the companion. We call the new assignment deriving from this substitution $\bar{\mathsf{b}}$. We can then restrict our focus on those paths in Φ that present a repetition pair. Let's call $\hat{\Phi} \subseteq \Phi$ the sub-structure formed only by those paths in Φ that end in a repetition pair. No structural change occurred between T and ρ, so $g_{\rho,T}$ is the same function as before. Having changed the input $\bar{\mathsf{a}}$ with $\bar{\mathsf{b}}$ we can show that $\alpha = g_{\rho,T}(\bar{\mathsf{a}}, \varphi) < g_{\rho,T}(\bar{\mathsf{b}}, \varphi) = \alpha'$. The conclusion comes from the following inductive argument. We already proved that $g_{s,T}$ is monotone with respect to $\preceq$, and in this case it is also strictly increasing. We will prove, in fact, that for all $\Phi_s \in \hat{\Phi}$: $g_{s,T}(\bar{\mathsf{b}}, \psi) \geq g_{s,T}(\bar{\mathsf{a}}, \psi) + \omega$. That can be done by showing that the order of all $\Phi_s \in \hat{\Phi}$ is increased by the substitution, including Φ_ρ. By induction looking at the definition of $g_{s,T}$:

1. we know by assumption that $\mathsf{b}_{t_j}(\psi) \geq \mathsf{a}_{t_j}(\psi) + \omega$ for all the leaves Φ_{t_j} in $\hat{\Phi}$ and $\psi \in \Phi_{t_j}$.
2. by hypothesis we have
   (a) $\psi \equiv p$ is not possible[6]
   (b) for $\psi \equiv \psi_0 \vee \psi_1$ we have
   $$\begin{aligned} g_{\Phi_s,T}(\bar{\mathsf{b}}, \psi_0 \vee \psi_1) &=_{df} \min(g_{\Phi_s,T}(\bar{\mathsf{b}}, \psi_i)) \\ &\geq_{IH} \min(g_{\Phi_s,T}(\bar{\mathsf{a}}, \psi_i)) + \omega \\ &=_{df} g_{\Phi_s,T}(\bar{\mathsf{a}}, \psi_0 \vee \psi_1) + \omega \end{aligned}$$
   the second step following from the fact that either both $\psi_i \in \Phi_s$, or that ψ_i is and $g_{\Phi_s,T}(\bar{\mathsf{a}}, \psi_{1-i}) \geq g_{\Phi_s,T}(\bar{\mathsf{a}}, \psi_i) + \omega$ already (by Definition 4.22)
   (c) for $\psi \equiv \psi_0 \wedge \psi_1$ we have
   $$\begin{aligned} g_{\Phi_s,T}(\bar{\mathsf{b}}, \psi_0 \wedge \psi_1) &=_{df} \sup(g_{\Phi_s,T}(\bar{\mathsf{b}}, \psi_i)) \\ &\geq_{IH} \sup(g_{\Phi_s,T}(\bar{\mathsf{a}}, \psi_i)) + \omega \\ &=_{df} g_{\Phi_s,T}(\bar{\mathsf{a}}, \psi_0 \wedge \psi_1) + \omega \end{aligned}$$
   the second step following from the fact that either both $\psi_i \in \Phi_s$, or that ψ_i is and $g_{\Phi_s,T}(\bar{\mathsf{a}}, \psi_{1-i}) < g_{\Phi_s,T}(\bar{\mathsf{a}}, \psi_i)$ already (by Definition 4.22)
   (d) for $\psi \equiv \Diamond\psi_0$ we have
   $$\begin{aligned} g_{\Phi_s,T}(\bar{\mathsf{b}}, \Diamond\psi_0) &=_{df} \min(g_{\Phi_{s_j},T}(\bar{\mathsf{b}}, \psi_0)) \\ &\geq_{IH} \min(g_{\Phi_{s_j},T}(\bar{\mathsf{a}}, \psi_0)) + \omega \\ &=_{df} g_{\Phi_s,T}(\bar{\mathsf{a}}, \Diamond\psi_0) + \omega \end{aligned}$$

[6] By definition, in fact, $p \in \Phi_s$ only if Θ_s is a leaf.

the second step following from the fact that for all the successor states Θ_{s_j}, either $\Theta_{s_j} \in \hat{\Phi}$, or $g_{\Theta_{s_j},T}(\bar{\mathsf{a}}, \psi_0) \geq g_{\Theta_s,T}(\bar{\mathsf{a}}, \Diamond\psi_0) + \omega$ already (by Definition 4.22 (4.a)).

(e) for $\psi \equiv \Box\psi_0$ we have

$$\begin{aligned} g_{\Phi_s,T}(\bar{\mathsf{b}}, \Box\psi_0) &=_{df} \sup(g_{\Phi_{s_j},T}(\bar{\mathsf{b}}, \psi_0)) \\ &\geq_{IH} \sup(g_{\Phi_{s_j},T}(\bar{\mathsf{a}}, \psi_0)) \\ &=_{df} g_{\Phi_s,T}(\bar{\mathsf{a}}, \Box\psi_0) \end{aligned}$$

the second step following from the fact that among all the successor states, there are some such that $\Theta_{s_j} \in \hat{\Phi}$. In this case we need to ensure that $\sup(g_{\Phi_{s_j},T}(\bar{\mathsf{b}}, \psi_0)) \geq \sup(g_{\Phi_{s_j},T}(\bar{\mathsf{a}}, \psi_0)) + \omega$, that is, that the $g_{\Phi_{s_j},T}(\bar{\mathsf{b}}, \psi_0))$ are also increasing and hence maintaining the limit jump. This is the case indeed, as it appears if we consider that all the $g_{\Phi_{s_j},T}(\bar{\mathsf{a}}, \psi_0)$ were increasing, that the function g is not changed and $\bar{\mathsf{b}}$ cannot be infinitely decreasing at the bar. As a consequence

$$g_{\Phi_s,T}(\bar{\mathsf{b}}, \Box\psi_0) \geq g_{\Phi_s,T}(\bar{\mathsf{a}}, \Box\psi_0) + \omega$$

(f) for $\psi \equiv \sigma x.\psi_0$ we have

$$\begin{aligned} g_{\Phi_s,T}(\bar{\mathsf{b}}, (\sigma+1)x.\psi_0) &=_{df} 1 + (g_{\Phi_s,T}(\bar{\mathsf{b}}, \psi_0[\sigma x.\psi_0])) \\ &\geq_{IH} 1 + (g_{\Phi_s,T}(\bar{\mathsf{a}}, \psi_0[\sigma' x.\psi_0])) + \omega \\ &=_{df} g_{\Phi_s,T}(\bar{\mathsf{a}}, (\sigma'+1)x.\psi_0)) \end{aligned}$$

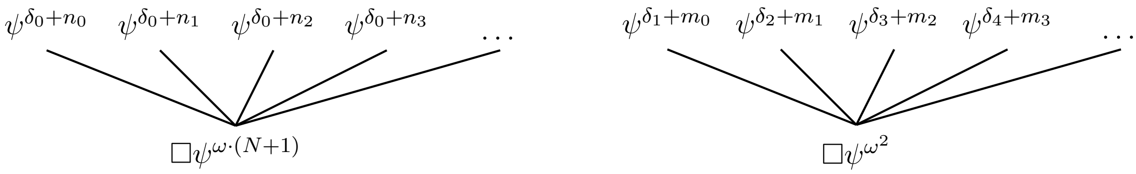
(remember that for a formula $\alpha x.\psi$ to occur not in the scope of a modality it means that $\alpha = \sigma + 1$ for some sigma, see Corollary 3.10).

We conclude that at any state $\Phi_s \in \hat{\Phi}$, the result of changing the initial assignment $\bar{\mathsf{a}}$ to $\bar{\mathsf{b}}$ produces an increment at least to the next limit ordinal. Since $\Phi_\rho \in \hat{\Phi}$ and $\varphi \in \Phi_\rho$, and the fact that the assignment given by $g_{\Phi_s,T}$ reflects the conservative feature of the ordinal assignment (Lemma 4.20 and Lemma 4.24), we have that the new model is conservative for $\varphi^{\alpha'}$ and $\alpha' \geq \alpha + \omega$. □

Notice that after the substitutions, we have obtained another model entirely. This means that if we want to re-apply the Lemma we can, but clearly the bar T has to be re-defined, and it will probably be made of a different set of states, and the same is true for Φ and $\hat{\Phi}$. Apart from that, we can clearly iterate the result and conclude that from $\alpha > \omega \cdot N$, we can define a conservative model for any $\alpha' \geq \alpha + \omega$. As a consequence, we have that the process has an upper bound in the first ordinal of the form ω^n bigger than α. To be able to make that step, we need to push the argument as we did for the primary case.

Lemma 4.29. *Given a conservative model Θ for a formula φ^α and $\omega^{n+1} > \alpha > \omega \cdot N$ there is a conservative model of $\varphi^{\alpha'}$ with $\alpha' > \omega^{n+1}$.*

Proof. Let's prove the lemma for $n = 1$, the general statement resulting from the same argument. In case we have a model for φ^δ and $\omega \cdot (N+1) > \delta > \omega \cdot N$ we can always apply Lemma 4.28 once and obtain a conservative model for a new φ^α, hence let's assume that we have a conservative model for φ^α and $\alpha > \omega \cdot (N+1)$. Define the structure Φ as in Definition 4.22, a bar T as in Definition 4.27 and find $\hat{\Phi}$ as in the proof of the previous Lemma. By construction[7] we have that on each path in $\hat{\Phi}$ there is at least a state where $\Theta_r \vDash \Box\psi^{\omega\cdot(N+1)}$, and $\Box\psi \in \Phi_r$. Moreover, by conservativity and Definition 4.22(4.c) each Θ_r has an infinite number of successor states in $\hat{\Phi}$ with ordinals bigger than $\omega \cdot N$. It follows that we can apply Lemma 4.28 to each one of them: once to the first successor, twice to the second, … We obtain an infinite series of states with increasing ordinals by at least one limit ordinal with respect to the previous one. In the example let's call δ_n the ordinal $\omega \cdot (N+n)$:



By Lemma 4.20, Lemma 4.24 and conservativity we have that $\Theta_r \vDash \Box\psi^{\omega^2}$. As we pointed out, every path in $\hat{\Phi}$ has a state like Θ_r, hence we can consider the set of those states as a bar where all the relevant ordinals have been raised over ω^2. We conclude that $\Theta_\rho \vDash \varphi^{\omega^2+\beta}$ for some β. The same procedure applies to any $n \geq 1$, hence we obtain ω^{n+1} and prove the Lemma. ☐

A combination of the last two Lemmas allows to conclude the proof for formulas $\varphi \in \Sigma_1^{ML}$.

Theorem 4.30 (Closure Ordinal for Σ_1^{ML})**.** *For any formula $\varphi \in \Sigma_1$ with at most one μ-quantifier, either the closure ordinal is an $\alpha \leq \omega \cdot 2^{2\cdot|FL(\varphi)|}$, or there is none.*

Proof. Assume that $CO(\varphi) = \kappa > \omega \cdot N$ for $N = 2^{2\cdot|FL(\varphi)|}$. By Lemma 4.28 and Lemma 4.29 we know that there exists a conservative model for $\kappa' > \kappa$. By Lemma 3.18 then $CO(\varphi) \geq \kappa'$, contradicting the hypothesis. ☐

Corollary 4.31 (Upper bound)**.** *ω^2 is the upper bound on closure ordinals for formulas in Σ_1^{ML}.*

[7] We can convince ourselves of this fact considering that the order of Φ_ρ is necessarily bigger than the order of the repetition leaves in $\hat{\Phi}$, and the only step where the order decreases in Definition 4.22 of Φ is (4.c) for some $\Box\psi$.

4.4 Σ_1^W and future steps

Despite the machinery involved, the result so far is interesting, although quite restricted. Our final goal is to use such a machinery to prove a bound on the whole Σ_1 fragment, and possibly more. So far we have been able to test the argument with primary formulas, and then adjust it to the complications coming from a more general structure of the formula, i.e. allowing for any ψ with one μ-quantifier. The last step is to tackle the question about multiple least fixed point occurring in the same formula, and yet we find already in an undefined situation. The concept of the closure ordinal of a formula with just one μ quantifier can be described informally by counting the number of times that the formula has been folded starting from the top.

When multiple instances of μ appear in the same formula, the concept of closure ordinal ceases to be immediate. In a direct translation of the informal *number of steps* description, we should list the ordinals appearing in the formula $\alpha_0, \alpha_1, \ldots, \alpha_n$ and take the sum of them. The ordinal so obtained corresponds to the number of iterations of all least fixed point above. Another way of defining the closure ordinal could be to count the least ordinal that is sufficient for each fixed point to be fully satisfied. As in the case of the nabla operator, we could argue that if $\alpha_j = \max(\alpha_0, \alpha_1, \ldots, \alpha_n)$, then it can be seen as the closure ordinal, because each quantifier certainly reaches its fixed point in that amount of steps. This interpretation can be said to come from the semantic definition of closure ordinal as the least ordinal such that $\|\varphi^\alpha\| = \|\varphi^{\alpha+1}\|$. Whatever the final definition will be in the end, for the time being we want to keep both interpretations open: the informal *number–of–steps* and the formal *least–general–ordinal.*

Instead of jumping to the Σ_1 fragment already, let's make a step into another fragment, that nonetheless extends the results obtained so far. Let's call it Σ_1^W, or sigma-weak fragment that introduces gradually the presence of multiple least fixed points. The weak fragment of Σ_1 is defined as follows:

Definition 4.32 (Σ_1^W)**.** *The set of formulas of Σ_1^W is defined by*

$$\varphi ::= p \mid \neg p \mid x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \mu x.\varphi$$

where every $\mu x.\varphi$ has no free variables.

We work with formulas that are guarded, and assume that each variable occurs bounded at most once. The following are examples of formulas in this fragment, assuming $x \notin FV(\psi)$:

(4.1) $\quad (p \wedge \Box\mu x.(\Box x)) \vee (\Diamond\mu y.\psi(y)) \qquad \mu x.(\varphi(x)) \wedge \mu y.(\psi(y))$

(4.2) $\quad \mu x.(\Box(p \wedge x) \vee \Diamond\mu y.\Box y) \qquad \Box\mu x.(\varphi(x) \vee \Box\Diamond\mu y.\psi(y))$

In the examples above we have considered only two fixed point quantifiers, from now on identified by μ_x and μ_y. We will continue to consider the number of different quantifiers to be two, the generalisation to three or more being a direct consequence. Clearly there are two possibilities for formulas in this fragment: (a.) none of the two quantifiers is in the scope of the other 4.1, or (b.) one is in the scope of the other: $\mu_x \sqsubset \mu_y$ 4.2.

Definition 4.33 ($\sqsubset$). *Given a formula φ and two subformulas $\mu x.\psi, \mu y.\xi \in FL(\varphi)$, we say that $\mu x.\psi \sqsubset \mu y.\xi$ iff $\mu x.\psi \in FL(\mu y.\xi)$.*

Notice that for $y \in FV(\psi)$ we don't have $\mu x.\psi \in FL(\mu y.\xi)$ because $\mu x.\psi[\mu y.\xi/y] \in FL(\mu y.\xi)$ according to Definition 3.3. Before presenting the arguments, we need to specify what is the meaning of an annotation φ^α in this context. We can refer to Definition 3.1 at the beginning, noting that in this context it is not necessarily true that the ordinal replacing μ_x corresponds to α, as it was informally until now. It could be the case, for example, that $(\beta x.(\Box(p \wedge x) \vee \Diamond\gamma y.\Box y))^\alpha$, with $\alpha \geq \beta, \gamma$. To facilitate our work in this section, let's call $g(\varphi)$ the ordinal of the formula φ as resulting from Definition 3.1, with the addition of the condition that if μ is the main connective of φ, then $g(\varphi)$ is either the sum or the maximum of the ordinals annotated in φ.[8] The definition below is a clarification with respect to Definition 3.1.

Definition 4.34. *The meaning of φ^α with $\overline{x}$ fixed point variables is $\varphi[\bot/\overline{x}]$ when $\alpha = 0$; otherwise:*

$$
\begin{array}{lll}
p^\alpha = p & (\neg p)^\alpha = \neg p & \\
(\varphi \wedge \psi)^\alpha = \varphi^\alpha \wedge \psi^\alpha & (\varphi \vee \psi)^\alpha = \varphi^\alpha \vee \psi^\alpha & \\
(\Diamond\varphi)^\alpha = \Diamond\varphi^\alpha & (\Box\varphi)^\alpha = \Box\varphi^\alpha & (\mu x.\varphi)^\alpha = (\alpha x.\varphi^\alpha)
\end{array}
$$

Let's go back to the two categories of formulas in Σ_1^W. From now on we assume the existence of a model Θ and a generic formula φ with two fixed point formulas $\mu x.\psi$ and $\mu y.\xi$ such that $\mu_y \sqsubset \mu_x$. In the case (a.) it is straightforward to argue that the closure ordinal of the main formula, if it exists, it is bounded by ω^2 as well. Whichever the definition of closure ordinal chosen, the sum or the maximum, each one of the subformulas falls under the scope of Theorem 4.30, hence they either have no closure ordinal, or it is an $\alpha_i < \omega^2$. Depending on the structure of the formula, the closure ordinal -if exists- is given by one or both subformulas, but it certainly cannot exceed ω^2 via sum or max operations.

In the second case the motivations are similar, but some extra effort must be taken to be able to assert that. In fact, given $\mu_y \sqsubset \mu_x$ we have that the ordinal assigned to the innermost quantified formula, say $\mu y.\psi$ could contribute to the final ordinal assigned to μ_x, and hence to the whole formula φ. Luckily, the contribution of μ_y can be proved to be not enough to extend the existing upper bound over ω^2. The condition that $x \notin FV(\psi)$ allows us to establish a bound on the closure ordinal for the innermost formula. We know that the subformula $\mu y.\psi$ is independent from the main formula in determining its ordinal. Since $\mu y.\psi$ is in the scope of Theorem 4.30, we already know that its ordinal is at most some $\alpha_y < \omega^2$ (under the assumption that a closure ordinal exists for such a formula).

[8]The choice of the letter g is to suggest a connection with the function $g_{s,T}(\bar{\mathsf{a}}, \varphi)$ defined above.

Let's assume that $\mu y.\psi$ has a closure ordinal $< \omega^2$. To determine the effect of $g(\mu y.\psi)$ on $g(\varphi)$, we can look at the subformula $\mu y.\psi$ as if it was substituted by a fresh propositional constant p_y, whose assigned ordinal is not 0 as usual, but it is the same as μ_y at each given state. Let's call φ_{p_y} the formula $\varphi[p_y/\mu y.\psi]$ and φ_p the formula $\varphi[p/\mu y.\psi]$ with $g(p) = 0$ at all states. We prove that $CO(\varphi_{p_y}) = CO(\varphi)$, and that if $CO(\varphi_p) < \omega^2$ then $CO(\varphi_{p_y}) < \omega^2$.

Lemma 4.35. *If $CO_\Theta(\varphi_p) < \omega^2$ and the ordinal of $p_y < \omega^2$ at all states, then $CO_\Theta(\varphi_{p_y}) < \omega^2$.*

Sketch of the proof. Fix a new Definition of $g_{s,T}(\bar{\mathsf{a}}, \varphi)$ so that to propositional constants an ordinal other than 0 can be assigned. Take a bar T in the given model Θ. At each state the value of $g_{s,T}(\bar{\mathsf{a}}, \varphi_{p_y})$ either depends on the value of $g_{s,T}(\bar{\mathsf{a}}, p_y)$ or not. If it does not, then $g_{s,T}(\bar{\mathsf{a}}, \varphi_{p_y}) = g_{s,T}(\bar{\mathsf{a}}, \varphi_p)$ because it doesn't depend on p either. If it does depend on $g_{s,T}(\bar{\mathsf{a}}, p_y)$, then in the worst case scenario $g_s(\bar{\mathsf{a}}, \varphi_{p_y})$ is sent to $g_{s,T}(\bar{\mathsf{a}}, p_y)$ by some supremum-condition. In either cases, the value of $g_{s,T}(\bar{\mathsf{a}}, \varphi_{p_y})$ never exceeds the threshold of ω^2, because both $g_s(\bar{\mathsf{a}}, \varphi_p)$ and $g_{s,T}(\bar{\mathsf{a}}, p_y)$ are smaller (Theorem 4.30 and assumption). In both interpretations of CO, we either take the sum of $g_{s,T}(\bar{\mathsf{a}}, \varphi_{p_y})$ and $g_{s,T}(\bar{\mathsf{a}}, p_y)$ or their maximum. As a result, if $CO(\varphi_p) < \omega^2$ and $g_{s,T}(\bar{\mathsf{a}}, p_y) < \omega^2$ at all states, there is not enough increment to reach ω^2, hence also $CO(\varphi_{p_y}) < \omega^2$. □

Lemma 4.36. $CO(\varphi_{p_y}) = CO(\varphi)$

Sketch of the proof. Since $g(\mu y.\psi) = g(p_y)$ at each state by definition, and that is the only change between φ_{p_y} and φ, then $g_s(\bar{\mathsf{a}}, \varphi_{p_y}) = g_s(\bar{\mathsf{a}}, \varphi)$ for every s. □

Those results relies on the assumption that both φ_p and $\mu y.\psi$ have a closure ordinal. We need also to consider the cases where at least one of them has no CO.

Lemma 4.37. *If $\mu y.\psi$ has no closure ordinal, then $CO(\varphi) < \omega^2$ or it doesn't exists.*

Sketch of the proof. If $\mu y.\psi$ has no closure ordinal then in principle we could assign any ordinal to the proposition p_y in φ_{p_y}. Now: if $g_s(\bar{\mathsf{a}}, \varphi_{p_y})$ does not depend on $g(p_y)$, then clearly $CO(\varphi_{p_y}) = CO(\varphi_p)$, and we know from Theorem 4.30 that $CO(\varphi_p) < \omega^2$ if it exists. If instead $g_s(\bar{\mathsf{a}}, \varphi_{p_y})$ does depend on $g(p_y)$, then there is no limit to the value of $g(p_y)$ that can be increased arbitrarily, hence also $g_s(\bar{\mathsf{a}}, \varphi_{p_y})$. As a result there is no closure ordinal for $CO(\varphi_{p_y})$, and by Lemma 4.36, not one for φ. □

Lemma 4.38. *If φ_p has no closure ordinal, then there is no $CO(\varphi)$.*

Sketch of the proof. φ_p has no closure ordinal. By changing p with p_y we have that $g_s(\bar{\mathsf{a}}, \varphi_p) \leq g_s(\bar{\mathsf{a}}, \varphi_{p_y})$. By the definition of $g_s(\bar{\mathsf{a}})$, in fact, there is no way in which increasing the ordinal of a propositional constant would determine a lower outcome, hence the impossibility of a bound in the closure ordinal of φ_{p_y}. By Lemma 4.36 the same holds for φ. □

Combining all these lemmas we can prove the Theorem

Theorem 4.39. *For every formula $\varphi \in \Sigma_1^W$, if a closure ordinal exists it is less than ω^2.*

Proof. For any φ in the fragment, we can order the quantifiers with respect to the inclusion relation $\sqsubset$. Whenever it is the case that $\mu_x \not\sqsubset \mu_y$ and $\mu_y \not\sqsubset \mu_x$, the closure ordinal of their combination does not exceed the sum of them, that is known to be $< \omega^2$. Consider the case $\mu_y \sqsubset \mu_x$ where the innermost subformula is $\mu y.\xi$ and it immediate predecessor is $\mu x.\psi$. There are four possible cases:

1. both $(\mu x.\psi)[p/\mu y.\xi]$ and $\mu y.\xi$ have a closure ordinal. By Lemma 4.30 we know that $CO((\mu x.\psi)_p) < \omega^2$ and $CO(\mu y.\xi) < \omega^2$. By Lemma 4.35 and Lemma 4.36 we know that $CO(\mu x.\psi) < \omega^2$.

2. $\mu y.\xi$ does not have a closure ordinal. By Lemma 4.37 then $CO(\mu x.\psi) < \omega^2$ or it doesn't exists.

3. $(\mu x.\psi)[p/\mu y.\xi]$ does not have a closure ordinal. By Lemma 4.38 then also $\mu x.\psi$ has not a closure ordinal.

4. neither $(\mu x.\psi)[p/\mu y.\xi]$ nor $\mu y.\xi$ have a closure ordinal. By the same argument of the previous case also $\mu x.\psi$ has not a closure ordinal.

We have that $CO(\mu x.\psi) < \omega^2$ or it doesn't exists. We can move to the next $\mu z.\chi$ and repeat the same argument for $\mu_x \sqsubset \mu_z$. As a result we have that $CO(\varphi) < \omega^2$, if it exists. $\square$

4.5 Conclusion

In the last chapter we laid the foundations for our future inquiry on closure ordinals for the modal μ-calculus in general. The definition of conservative well-annotations taken from Kozen has been the major tool to replicate the argument in [AL13] on primary formulas, a small subset of the Σ_1 fragment. With the application of a pumping-like argument, the impossibility of a bound equal or bigger than ω^2 for primary formulas was established, and a condition for the application of that process has been given in the form of the existence of a repetition trace. Once the general machinery have been tested, an attempt to extend directly the procedure to a bigger fragment of Σ_1, that is disjunctive formulas, proved already to be unfeasible. As a consequence we restricted our interest towards formulas with no particular structure but only one least fixed point quantifier. The notion of trace has been replaced by that of a structure Φ, that focuses on all the paths that is sufficient to involve in the pumping process, in order to guarantee the ordinal increment. A different representation of the ordinals was defined in terms of a function $g_{s,T}$. With these modifications, a new repetition condition in terms of paths in Φ was defined, and the same process of the primary case works for the fragment of Σ_1 with only one least fixed point, showing that ω^2 is still an upper bound.

To extend the result to the whole Σ_1 we need to address the presence of multiple quantifiers. That raises the question not only of the interaction of several cycles, but already about the meaning of a closure ordinal for more than one fixed point. A step in the direction of a solution to the first issue was taken in the previous section, where we described how the method can be implemented for an intermediate fragment like Σ_1^W. To the question about the meaning of closure ordinals of multiple variables we didn't give an answer, but we tried to keep both major possibilities into consideration, concluding that there is no significant difference between them with respect to closure ordinals in Σ_1^W. We don't know at the moment if the same could be achieved for Σ_1 or more. In any case, we believe that this neutral approach could potentially give some insight about the effects of both choices.

The first and main task for subsequent work is to complete the proof of the bound for Σ_1. The aspect that remains unanswered in the present work concerns the way in which the effects of the interaction between multiple nested μ quantifiers will reflect on a function like g in Definition 4.19. With the limitations assumed in Σ_1^W it was possible to treat the innermost fixed point formula as a propositional constant with an arbitrary ordinal assignment, thanks to the fact that the interaction between the two was limited. While the same method doesn't seem to be expandable right away, the framework developed in the present work suggests that with some minor modification the same result can be obtained. A study on the nature of the interaction between two nested μ-formulas will be also the starting point to understand whether a change in the function g is sufficient to approach the study of greatest fixed point, or a deeper modification is necessary. It is possible that a different definition of Φ will be necessary.

The main challenge to extend the results to Π_2/Σ_2 will be the definition of a framework that is able to keep together both kinds of fixed point in a functional way. Very little appears to be known about the closure ordinals beyond the alternation-free μ-calculus, [AL13] being one of the most advanced results on the topic. We have to consider, also, that there are not even examples of formulas with a closure ordinal greater than ω^2, at least in the language and semantics given here. Some motivation for looking at ω^ω as a potential next bound comes from two recent works. One is Milanese's master thesis [Mil18] where a bound of ω^ω is given to formulas in the context of bidirectional models. ω^ω also appears to be a necessary lower bound to prove the soundness of the infinitary calculus $K^{\kappa}_{\mu+}$ for the full μ-calculus,[9] as showed in [AJL19] by Afshari, Jäger and Leigh. Another work that could inspire some future development is the one from Gouveia and Santocanale. In [GS18] they study κ-continuous formulas, that is a generalisation of the notion of continuous fragment seen in [Fon08]. They show that for $\aleph_1$-continuous formulas the closure ordinal is the first uncountable ordinal ω_1. These examples suggest possible directions for inquiry, but they cannot be used directly in the present framework. In any case, we believe that an attempt can be made starting from the work presented here, if only with the goal of having a better understanding of closure ordinals at higher levels.

Finally a possible tangent work could be started from the attempted work on disjunctive formulas. The realisation that using disjunctive formulas was not going to be a shortcut towards the analysis of Σ_1 made us abandon the question on their behaviour with respect to closure ordinals. However, there is a chance that an approach like the one adopted here could be a starting point for a research about the changes in ordinals that occur with semantically equivalent but syntactically different formulas.

[9] That is μ-calculus extended with converse modalities.

Bibliography

[AHL15] Bahareh Afshari, Stefan Hetzl, and Graham E. Leigh. "Herbrand Disjunctions, Cut Elimination and Context-Free Tree Grammars". In: *TLCA*. 2015.

[AHL20] Bahareh Afshari, Stefan Hetzl, and Graham E. Leigh. "Herbrand's theorem as higher order recursion". In: *Annals of Pure and Applied Logic* 171.6 (2020), p. 102792.

[AJL19] Bahareh Afshari, Gerhard Jäger, and Graham E. Leigh. "An Infinitary Treatment of Full Mu-Calculus". In: *Logic, Language, Information, and Computation*. Ed. by Rosalie Iemhoff, Michael Moortgat, and Ruy de Queiroz. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 17–34.

[AL13] Bahareh Afshari and Graham E. Leigh. "On closure ordinals for the modal mu-calculus". In: *Computer Science Logic 2013 (CSL 2013)*. Ed. by Simona Ronchi Della Rocca. Vol. 23. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2013, pp. 30–44.

[AF98] Jeremy Avigad and Solomon Feferman. "Chapter V - Gödel's Functional ("Dialectica") Interpretation". In: *Handbook of Proof Theory*. Ed. by Samuel R. Buss. Vol. 137. Studies in Logic and the Foundations of Mathematics. Elsevier, 1998, pp. 337–405.

[Bar84] H. P. Barendregt. *The lambda calculus : its syntax and semantics*. Elsevier Science Pub., 1984.

[Bar88] Jon Barwise. "Three views of common knowledge". In: *Proceedings of the 2nd conference on Theoretical aspects of reasoning about knowledge*. TARK '88. Morgan Kaufmann Publishers Inc., 1988, pp. 365–379.

[BT17a] Stefano Berardi and Makoto Tatsuta. "Classical System of Martin-Löf's Inductive Definitions is not Equivalent to Cyclic Proofs". In: (2017). DOI: `10.48550/ARXIV.1712.09603`.

[BT17b] Stefano Berardi and Makoto Tatsuta. "Equivalence of Inductive Definitions and Cyclic Proofs under Arithmetic". In: *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science*. LICS '17. IEEE Press, 2017.

[BT17c] Stefano Berardi and Makoto Tatsuta. "Equivalence of Intuitionistic Inductive Definitions and Intuitionistic Cyclic Proofs under Arithmetic". In: (2017). DOI: `10.48550/ARXIV.1712.03502`.

[Bra98] J.C. Bradfield. "The modal mu-calculus alternation hierarchy is strict". In: *Theoretical Computer Science* 195.2 (1998), pp. 133–153.

[BS07] Julian Bradfield and Colin Stirling. "Modal mu-calculi". In: *Handbook of Modal Logic*. Netherlands: Elsevier, 2007, pp. 721–756.

[BS11] J. Brotherston and A. Simpson. "Sequent calculi for induction and infinite descent". In: *Journal of Logic and Computation* 21.6 (2011), pp. 1177–1216.

[Bro05] James Brotherston. "Cyclic Proofs for First-Order Logic with Inductive Definitions". In: *Automated Reasoning with Analytic Tableaux and Related Methods*. Ed. by Bernhard Beckert. Springer Berlin Heidelberg, 2005, pp. 78–92.

[Bro06] James Brotherston. "Sequent calculus proof systems for inductive definitions". PhD thesis. 2006.

[Cza10] Marek Czarnecki. "How fast can the fixpoints in modal mu-calculus be reached". In: *Fixed Points in Computer Science* (2010), pp. 35–39.

[Dam94] Mads Dam. "CTL* and ECTL* as Fragments of the Modal mu-Calculus". In: *Theor. Comput. Sci.* 126.1 (1994), pp. 77–96.

[DG02] Mads Dam and Dilian Gurov. "mu-Calculus with Explicit Points and Approximations". In: *Journal of Logic and Computation* 12.2 (2002), pp. 255–269.

[Das20] Anupam Das. "A circular version of Gödel's T and its abstraction complexity". In: (2020). DOI: 10.48550/ARXIV.2012.14421.

[EJS93] E. A. Emerson, C. S. Jutla, and A. P. Sistla. "On model-checking for fragments of mu-calculus". In: *Computer Aided Verification*. Ed. by Costas Courcoubetis. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1993, pp. 385–396.

[FL79] Michael J. Fischer and Richard E. Ladner. "Propositional dynamic logic of regular programs". In: *Journal of Computer and System Sciences* 18.2 (1979), pp. 194–211.

[Fon10] G. M. M. Fontaine. "Modal fixpoint logic: some model theoretic questions". PhD thesis. Amsterdam Institute for Logic, Language and Computation, 2010.

[Fon08] Gaëlle Fontaine. "Continuous Fragment of the mu-Calculus". In: *Computer Science Logic*. Ed. by Michael Kaminski and Simone Martini. Springer Berlin Heidelberg, 2008, pp. 139–153.

[FV18] Gaëlle Fontaine and Yde Venema. "Some model theory for the modal mu-calculus: syntactic characterisations of semantic properties". In: *Logical Methods in Computer Science ; Volume 14* (2018), Issue 1, 18605974.

[GK03] Philipp Gerhardy and Ulrich Kohlenbach. "Extracting Herbrand Disjunctions by Functional Interpretation". In: *BRICS Report Series* 03-32 (2003).

[GS18] Maria Joa O Gouveia and Luigi Santocanale. "$\aleph_1$ and the modal mu-calculus". In: (2018). DOI: 10.23638/LMCS-15\%284\%3A1\%292019.

[Het12] Stefan Hetzl. "Applying Tree Languages in Proof Theory". In: *Language and Automata Theory and Applications*. Ed. by Adrian-Horia Dediu and Carlos Martin-Vide. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012, pp. 301–312.

[JW95] David Janin and Igor Walukiewicz. “Automata for the modal mu-calculus and related results”. In: *Mathematical Foundations of Computer Science 1995*. Ed. by Jiří Wiedermann and Petr Hájek. Vol. 969. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1995, pp. 552–562.

[JW96] David Janin and Igor Walukiewicz. “On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic”. In: *CONCUR '96: Concurrency Theory*. Ed. by Ugo Montanari and Vladimiro Sassone. Vol. 1119. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer Berlin Heidelberg, 1996, pp. 263–277.

[Jun10] Natthapong Jungteerapanich. “Tableau systems for the modal mu-calculus”. PhD thesis. The University of Edinburgh, 2010.

[Klo92] J. W. Klop. “Term Rewriting Systems”. In: *Handbook of Logic in Computer Science*. Ed. by S. Abramsky, D. Gabbay, and T. Maibaurn. Vol. 2. Oxford University Press, 1992, pp. 1–116.

[Koz83] Dexter Kozen. “Results on the propositional mu-calculus”. In: *Theoretical Computer Science* 27.3 (1983), pp. 333–354.

[Koz88] Dexter Kozen. “A Finite Model Theorem for the Propositional mu-Calculus”. In: *Studia Logica: An International Journal for Symbolic Logic* 47.3 (1988), pp. 233–241.

[KP84] Dexter Kozen and Rohit Parikh. “A decision procedure for the propositional mu-calculus”. In: *Logics of Programs*. Ed. by Edmund Clarke and Dexter Kozen. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1984, pp. 313–325.

[KV03] Orna Kupferman and Moshe Y. Vardi. “$\Pi_2 \cap \Sigma_2 \equiv AFMC$”. In: *Automata, Languages and Programming*. Ed. by Jos C. M. Baeten et al. Springer Berlin Heidelberg, 2003, pp. 697–713.

[Mar71] Per Martin-Löf. “Hauptsatz for the Intuitionistic Theory of Iterated Inductive Definitions”. In: *Proceedings of the Second Scandinavian Logic Symposium*. Ed. by J.E. Fenstad. Vol. 63. Studies in Logic and the Foundations of Mathematics. Elsevier, 1971, pp. 179–216.

[Mil18] Gian Carlo Milanese. “An exploration of closure ordinals in the modal mu-calculus”. Master Thesis. 2018.

[Niw86] Damian Niwiński. “On fixed-point clones”. In: *Automata, Languages and Programming*. Ed. by Laurent Kott. Vol. 226. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 1986, pp. 464–473.

[NW96] Damian Niwiński and Igor Walukiewicz. “Games for the mu-calculus”. In: *Theoretical Computer Science* 163.1 (1996), pp. 99–116.

[RS59] M. O. Rabin and D. Scott. “Finite Automata and Their Decision Problems”. In: *IBM Journal of Research and Development* 3.2 (1959), pp. 114–125.

[SW12] Sylvain Salvati and Igor Walukiewicz. *Recursive Schemes, Krivine Machines, and Collapsible Pushdown Automata*. Tech. rep. 2012.

[San02] Luigi Santocanale. “A Calculus of Circular Proofs and Its Categorical Semantics”. In: *Foundations of Software Science and Computation Structures*. Ed. by

Mogens Nielsen and Uffe Engberg. Springer Berlin Heidelberg, 2002, pp. 357–371.

[Sim17] Alex Simpson. "Cyclic Arithmetic Is Equivalent to Peano Arithmetic". In: *Foundations of Software Science and Computation Structures*. Ed. by Javier Esparza and Andrzej S. Murawski. Springer Berlin Heidelberg, 2017, pp. 283–300.

[SD03] Christoph Sprenger and Mads Dam. "On global induction mechanisms in a mu-calculus with explicit approximations". In: *RAIRO - Theoretical Informatics and Applications* 37.4 (2003), pp. 365–391.

[SE84] Robert S. Streett and E. Allen Emerson. "The propositional mu-calculus is elementary". In: *Automata, Languages and Programming*. Ed. by Jan Paredaens. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 1984, pp. 465–472.

[SE89] Robert S. Streett and E. Allen Emerson. "An automata theoretic decision procedure for the propositional mu-calculus". In: *Information and Computation* 81.3 (June 1, 1989), pp. 249–264.

[TS00] A. S. Troelstra and H. Schwichtenberg. *Basic Proof Theory*. 2nd ed. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2000.

[Tro98] A.S. Troelstra. "Chapter VI - Realizability". In: ed. by Samuel R. Buss. Vol. 137. Studies in Logic and the Foundations of Mathematics. Elsevier, 1998, pp. 407–473.

[Wal00] Igor Walukiewicz. "Completeness of Kozen's Axiomatisation of the Propositional mu-Calculus". In: *Information and Computation* 157.1 (2000), pp. 142–182.

[Wir04] C.P. Wirth. "Descente Infinie + Deduction". In: *Logic Journal of IGPL* 12.1 (2004), pp. 1–96.