Cycles with Annotations

Non-Wellfounded Proof Theory
of
Modal Fixpoint Logics

Johannes Kloibhofer

Cycles with Annotations

Non-Wellfounded Proof Theory of Modal Fixpoint Logics

ILLC Dissertation Series DS-2026-05

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 107
1098 XG Amsterdam
phone: +31-20-525 6051
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

The research for this thesis has been made possible by a grant from the Dutch Research Council (NWO), project number 617.001.857.

Cover design by Iris Sophia van der Hut.
Printed and bound by Ipskamp Printing.

ISBN: 978-94-6536-037-9

Cycles with Annotations

Non-Wellfounded Proof Theory of Modal Fixpoint Logics

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Aula der Universiteit
op vrijdag 13 maart 2026, te 11.00 uur

door Johannes Kloibhofer
geboren te Freistadt

Promotiecommissie

Promotor:	prof. dr. Y. Venema	Universiteit van Amsterdam
Copromotor:	prof. B. Afshari	University of Gothenburg
Overige leden:	dr. B.D. ten Cate	Universiteit van Amsterdam
	dr. N.J.S. Enqvist	Stockholm University
	dr. I. van der Giessen	Universiteit van Amsterdam
	dr. M. Girlando	Universiteit van Amsterdam
	prof. dr. R. Iemhoff	Universiteit Utrecht
	prof. dr. S.J.L. Smets	Universiteit van Amsterdam
	prof. dr. T. Studer	University of Bern

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

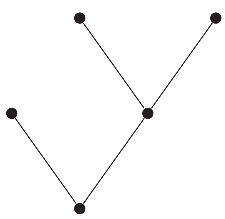
Contents

Chapter 1

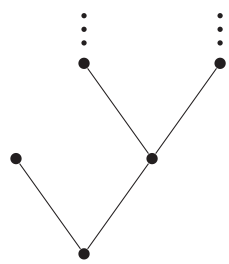
Introduction

Taking notes helps. When working through a complicated mathematical proof, it is the markings and small remarks in the margin of the paper, that help to understand it. By taking notes of previous definitions and facts, one can write more structured arguments that simplify checking the proof afterwards. This is not only something I have learned informally in the process of my PhD, but it is also what this thesis is about in a formal context: we show how to add annotations to formal mathematical proofs and study the advantages of these annotated proofs.

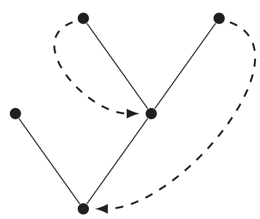
The particular proofs we study may contain infinitely long branches or cycles, and are called *non-wellfounded proofs*. In order to disallow absurd circular reasoning, one needs to be careful which infinite branches and cycles are allowed. This is done by formulating a so-called *soundness condition* that determines the "good" infinite branches and cycles. The main challenge in non-wellfounded proof theory is to handle the soundness condition, particularly to design cyclic proof systems with a simple soundness condition. In this thesis, we show how to add annotations to infinitary proof systems to obtain such cyclic proof systems in a uniform way. We apply this method to design annotated cyclic proof systems for the modal μ-calulus, as well as for extensions and fragments thereof, and use these systems to obtain results on interpolation and cut elimination.



a *finite* proof



an *infinite* proof



a *cyclic* proof

Before we go into more details and explain the concepts mentioned above, let us introduce the theoretical background of this work. We start by motivating *proof theory* in general, and *non-wellfounded proof theory* in particular.

In proof theory, proofs are formal mathematical objects, allowing their study by means of mathematical tools. *Finite* proofs start with *axioms* – formulas that are assumed to be valid. For instance, that for every proposition p, either p or its negation $\overline{p}$ holds. Valid formulas are then combined by *rules*. One such rule states that if two formulas φ and ψ are both valid, then we may deduce that their conjunction $\varphi \wedge \psi$ is valid as well. A proof of a formula φ is then a *finite tree*: its leaves are labeled with axioms, its internal nodes are formed by rule applications, and its root is labeled with φ. Since axioms are valid and the rules preserve validity, we can conclude that φ is valid as well. As such, proofs provide a certificate that a formula is valid in some logical system.

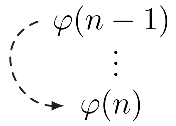
The proofs we consider in this thesis differ in one fundamental way: we allow *infinite branches* or *cycles* in the proof tree and call them *infinite proofs* and *cyclic proofs*, respectively. How can we still ensure that only valid formulas are deduced? A soundness condition needs to be defined that exactly carves out those infinite branches and cycles that embody valid reasoning.

To illustrate how such a soundness condition may look, consider an example in the natural numbers. Suppose we want to show that a formula $\varphi(n)$ holds for all natural numbers n. One approach, called *proof search*, proceeds by successively applying rules to $\varphi(n)$ in an attempt to find a proof. When carrying out proof search for $\varphi(n)$, we might apply rules until finding the formula $\varphi(n-1)$ in the proof tree. We can then continue in the same manner until we find $\varphi(n-2)$, and so forth. This creates an infinite branch, where the same formula φ is occurring infinitely often, yet with different inputs.

$$\begin{array}{c} \vdots \\ \varphi(n-2) \\ \vdots \\ \varphi(n-1) \\ \vdots \\ \varphi(n) \end{array}$$

Note that the natural number n in this proof is arbitrary, and therefore this branch is infinitely long. However, for every fixed n, the formula $\varphi(0)$ is reached after finitely many steps. At that point we may stop, assuming we can prove that $\varphi(0)$ holds. Thus, although the proof is infinite in principle, for every instantiated n the relevant part of the proof is finite; we can thus convince ourselves that this is a valid form of reasoning. We can therefore define the *soundness condition* on such proofs as follows: on every infinite branch there is a variable n that decreases infinitely often.

Cyclic proofs consist of finitary encodings of infinite proofs. In the above example, the proof of $\varphi(n-1)$ is analogous to the proof of $\varphi(n)$, with $n-1$ substituted for n. Instead of repeating an analogous proof over and over again, we can encode the infinite branch by repetitions of the steps taken from $\varphi(n)$ to reach $\varphi(n-1)$. This can be depicted by the following cycle, which introduces a *back-edge* from $\varphi(n-1)$ to $\varphi(n)$:



These cyclic proofs then satisfy the following soundness condition: on every infinite path through the proof there is a variable n that decreases infinitely often.

In order to make precise the proofs we are working with, we need to introduce the logical systems they are designed for. The logics we consider are extensions of modal logic, used in particular to model phenomena in theoretical computer science.

Modal Logic The language of modal logic is obtained by adding the modalities $\Diamond$ and $\Box$ to classical propositional logic, which can be interpreted in multiple ways. Originally, modal logic was studied to talk about possibility and necessity [LL32]. In this reading, $\Diamond\varphi$ is interpreted as "possibly φ" and $\Box\varphi$ as "necessarily φ". In some contexts, multiple modalities are considered; we denote them by $\langle a \rangle$, $[a]$ for each a in a given set. In epistemic logics, the modalities model knowledge and belief. Given a set of agents, modalities $\langle a \rangle$ and $[a]$ for each agent a are considered, where $[a]\varphi$ is interpreted as "agent a knows φ" or "agent a believes φ".

A different reading of modalities is given in so-called *program logics*, where the modalities are interpreted as programs; their meaning is given as

$$\begin{aligned} \langle a \rangle\varphi &\quad\equiv\quad \text{“after some run of the program } a\text{, the formula } \varphi \text{ holds”} \\ [a]\varphi &\quad\equiv\quad \text{“after every run of the program } a\text{, the formula } \varphi \text{ holds”} \end{aligned}$$

In this setting, programs are non-deterministic, meaning that they may have multiple distinct runs.

PDL Propositional Dynamic Logic, in short PDL, takes this one step further by interpreting modalities as *regular expressions* of programs [FL79]. That is, programs can be combined by the constructors introduced below.

If a and b are programs, then $a;b$ is a program interpreted as first running a, then b; and $a \cup b$ is a program that is interpreted as either running a or b. Via the test program $\psi?$ formulas can be turned into programs; the program $\psi?$ is interpreted as testing ψ, continuing if ψ holds and aborting otherwise. Maybe the most interesting program constructor is the Kleene star a^* which is interpreted as running the program a an arbitrary finite number of times. Modalities are then interpreted as programs as indicated above; for instance, $\langle a^* \rangle \varphi$ holds if after running the program a some finite number of times, the formula φ holds. As such, PDL can describe if-clauses and while loops,[1] and consequently make statements about imperative programs. For an overview of PDL we refer to [TB23].

Running a program finitely many times is equivalent to either running it zero times or at least once. In other words,

$$\langle a^* \rangle \varphi \text{ is equivalent to } \varphi \vee \langle a \rangle \langle a^* \rangle \varphi.$$

Thus, the formula $\langle a^* \rangle \varphi$ is a *fixpoint* of the function

$$x \mapsto \varphi \vee \langle a \rangle x.$$

In fact, $\langle a^* \rangle \varphi$ is its *least fixpoint*, meaning that additionally, if ψ is another fixpoint (meaning that ψ is equivalent to $\varphi \vee \langle a \rangle \psi$), then $\langle a^* \rangle \varphi$ implies ψ.

Modal μ-calculus In PDL, fixpoints of certain functions are expressible, as for instance witnessed by the formula $\langle a^* \rangle \varphi$ above. In the modal μ-calculus on the other hand, least and greatest fixpoints of all positive functions are expressible: the formula $\mu x.\varphi$ describes the least fixpoint, and the formula $\nu x.\varphi$ describes the greatest fixpoint of the function $x \mapsto \varphi(x)$ for any modal logic formula $\varphi(x)$ where x only occurs positively. By definition, fixpoint formulas satisfy the following fixpoint property: $\mu x.\varphi$ is equivalent to its *unfolding* $\varphi[\mu x.\varphi/x]$, where $\mu x.\varphi$ is substituted for x in φ. Additionally, in the modal μ-calculus these fixpoint operators might be nested, allowing the logic to model complex phenomena. The modal μ-calculus was introduced by Kozen [Koz83] and since then it has been intensively studied, with widespread applications including in program verification. Surveys on this logic can be found in [BS07] and [DGL16]. This thesis studies the proof theory of the modal μ-calculus, with particular attention to extensions and fragments thereof.

Proof theory of the modal μ-calculus Before we can talk about the proof theory of the modal μ-calculus, we need to clarify some notions. We work within the proof-theoretic framework of *sequent calculus*. In this setting, one considers finite sets Γ of formulas, called *sequents*, with the goal of proving that the disjunction $\bigvee \Gamma$ of all formulas in Γ is valid.[2] As exemplified by the rule for conjunction, rules are written as follows:

$$\frac{\varphi, \Gamma \qquad \psi, \Gamma}{\varphi \wedge \psi, \Gamma} \wedge$$

[1] Given a program a, the while-loop **while** p **do** a is described by the program $(p?; a)^*; \neg p?$.

[2] If $\Gamma = \varphi_1, ..., \varphi_n$, then $\bigvee \Gamma = \varphi_1 \vee \cdots \vee \varphi_n$.

This formulation indicates that the conclusion $\varphi \wedge \psi, \Gamma$ can be deduced from the premises φ, Γ and ψ, Γ for any formulas φ, ψ and any sequent Γ.

We consider the infinitary proof system NW for the modal μ-calculus as introduced by Niwiński and Walukiewicz [NW96]. When proving a sequent containing a fixpoint formula $\mu x.\varphi$, they rely on the fixpoint property: $\mu x.\varphi$ is equivalent to $\varphi[\mu x.\varphi/x]$. That is, they obtain the rule

$$\frac{\varphi[\mu x.\varphi/x], \Gamma}{\mu x.\varphi, \Gamma}\ \mu$$

However, there is something odd about this rule: the premise $\varphi[\mu x.\varphi/x], \Gamma$ is syntactically more complex than the conclusion $\mu x.\varphi, \Gamma$. In other rules, the formulas in the premise of the rule are subformulas of the conclusion, as can be seen in the conjunction rule above. Because the rule for μ lacks this subformula property, we can not guarantee that proof search terminates: we may obtain infinite branches.

In order to decide which infinite branches are allowed, a soundness condition is needed. This condition is formulated in terms of *traces*: a trace on an infinite branch is a sequence of formulas along the branch. Figure 1.2 illustrates such traces.

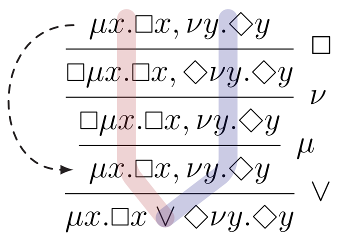


Figure 1.2: NW-proof of $\mu x.\Box x \vee \nu y.\Diamond y$. It contains one infinite path (the infinite unfolding of the cycle) with the ν-trace $\mu x.\Box x \vee \Diamond\nu y.\Diamond y \rightsquigarrow \nu y.\Diamond y \rightsquigarrow \nu y.\Diamond y \rightsquigarrow \Diamond\nu y.\Diamond y \rightsquigarrow \nu y.\Diamond y \rightsquigarrow \cdots$.

An NW-proof satisfies the soundness condition if on all infinite branches there is a *ν-trace*: a trace on which the "most important" fixpoint formula occurring infinitely often is a greatest fixpoint. In the above example, the blue trace is a ν-trace and therefore this infinite branch satisfies the soundness condition. Informally, ν-traces are "good" because greatest fixpoints (ν-formulas) may be unfolded infinitely often, while least fixpoints (μ-formulas) may be unfolded only finitely often.

Two main themes The soundness condition on NW-proofs has two major drawbacks. First, it is based on *traces*, whose dynamics along infinite branches can be complicated, making the condition complex and hard to work with. A simpler alternative is provided by *path-based* soundness conditions, which are formulated in terms of the proof paths themselves and do not refer to formulas along the branch. Second, it is a *global* soundness condition, meaning that it is formulated in terms of all infinite branches through the proof. For cyclic proof systems, one can also introduce *local* soundness conditions. Whereas a global soundness condition is formulated in terms of all infinite paths through a cyclic proof, a local one is defined solely in terms of the finitely many cycles in the proof.

Considerable effort has gone into adding extra structure to proof systems with a trace-based global soundness condition to obtain cyclic proof systems with a path-based local soundness condition. For the modal μ-calculus, this has been pioneered by Walukiewicz [Wal93], Jungteerapanich [Jun10] and Stirling [Sti14].

Taking proof systems with a trace-based global soundness condition as a starting point, this thesis is organized around two main themes. First, we show how to add extra structure in the form of *annotations* to such proof systems to obtain proof systems with a path-based soundness condition. We transform the latter into cyclic proof systems with a local soundness condition. In the second theme, we demonstrate how to use such annotated cyclic proof systems to derive results about their underlying logic. Moreover, we establish fundamental results about annotated cyclic proof systems that enhance our understanding of them and provide a foundation for future research. We will now motivate both themes in a bit more detail.

➢ ***Obtaining cycles with annotations*** In this theme we use techniques from automata theory to get a better grasp of global soundness conditions. Infinite branches in a proof tree can be seen as a stream (infinite word) over some finite alphabet, and we define an automaton $\mathbb{A}$ that operates on such streams to check whether an infinite branch β is good – that is, whether it carries a ν-trace. Then the idea is to decorate nodes of β with states of $\mathbb{A}$, such that the stream of decorations corresponds to the run of $\mathbb{A}$ on the branch β; it follows that β is good iff this stream of decorations is a successful run of $\mathbb{A}$. Hence, the soundness condition based on traces can be replaced by the acceptance condition of $\mathbb{A}$, which is a much simpler path-based condition.

The natural definition of the automaton $\mathbb{A}$ turns out to be non-deterministic. However, for the above strategy to work one requires $\mathbb{A}$ to be *deterministic*. To see this, observe that two accepted infinite branches could generally require two distinct runs of $\mathbb{A}$. If $\mathbb{A}$ is non-deterministic, these two runs might already diverge before the two branches split. Thus, we first need to transform $\mathbb{A}$ into an equivalent deterministic automaton $\mathbb{A}^D$, which brings us to the relevance of determinization methods for stream automata. We show how different determinization methods give rise to different annotated proof systems. Having such an annotated infinitary proof system at hand, we can then transform it to an annotated cyclic proof system with a *local* soundness condition.

Based on an abstract notion of proof systems, trace-based systems have been translated into path-based systems in [LW24] using a particular determinization method. Our construction, on the other hand, is uniform with respect to both the proof systems and the determinization method.

➢ ***Using cycles with annotations*** Cyclic proof systems with annotations allow us to transform proofs while making sure that the soundness condition is preserved. We consider two classic proof-theoretic results using annotated cyclic proofs: *interpolation* and *cut elimination*.

A logic has *interpolation*, if for every valid implication $\varphi \to \psi$ there is a formula θ (the interpolant) in the common vocabulary of φ and ψ such that $\varphi \to \theta$ and $\theta \to \psi$ are valid. That is, informally, θ contains only the relevant information making $\varphi \to \psi$ valid. This property has various applications including in model checking and knowledge representation. Proof-theoretically, this can be shown by defining the interpolant θ on the basis of a proof π of $\varphi \to \psi$, and transforming π to respective proofs of $\varphi \to \theta$ and $\theta \to \psi$ [Mae61]. This method has also been adapted to cyclic proof systems in [Sha14; AL19; MV21b]. We show how to apply this technique to certain annotated cyclic proof systems.

Cut elimination is the central result of proof theory. The *cut rule* states that one can prove a set of formulas Γ by proving, for some formula φ, both Γ or φ, and Γ or $\overline{\varphi}$ (the negation of φ):

$$\frac{\Gamma, \varphi \qquad \overline{\varphi}, \Gamma}{\Gamma}\ \mathsf{cut}$$

The cut rule is a generalization of modus pones and may be interpreted as inserting a "lemma" φ in the proof of Γ. This rule is regularly added to a proof system to show its completeness. However, the cut rule can be problematic, as the cut formula φ does not depend on Γ. For example, with the inclusion of the cut rule, proof search may become undecidable and many desirable properties of the system are lost. Cut elimination is the process of transforming proofs with cuts to proofs not making use of the cut rule. As such, cut elimination may have consequences such as the subformula property or consistency.

For finitary proof systems, there is a standard technique for cut elimination tracing back to Gentzen [Gen35]. Many adaptations of this method have been proposed for infinitary proof systems, for instance in [FS13; BDS16; DP18; SS20; MSZ24]. However, for cyclic proof systems, cut elimination is yet unexplored. We show how to apply cut elimination to an annotated cyclic proof system.

Chapter overview

We give an overview of the chapters of this thesis, state our main contributions, and clarify the material this work builds on. All chapters have been significantly revised compared to the papers they are based on.

Chapter 2: Modal fixpoint logics and their proof theory In this preliminary chapter we set the stage. First, we introduce the logics that we study: the modal μ-calculus, extensions and fragments thereof, and (Converse) PDL. We state basic facts about these logics and define their game semantics. For PDL, we also provide a proof that the game semantics and the standard relational semantics are equivalent. Then, we introduce non-wellfounded proof systems in an abstract way. We define the notions of derivations, infinite proofs and cyclic proofs that we will use throughout this thesis. We end the chapter with a definition of the infinitary proof system NW for the modal μ-calculus. This chapter was written specifically for this thesis and does not contain original material.

Chapter 3: Determinization of ω-automata ω-Automata are automata acting on infinite words. We define ω-automata and introduce two methods transforming non-deterministic ω-automata to deterministic ones. The first method directly determinizes non-deterministic parity automata using a construction based on binary trees. The second method determinizes non-deterministic parity automata with ε-transitions extending the well-known Safra construction. These constructions are bespoke so that they can be used in the Chapters 4 and 5, respectively, to obtain annotated proof systems. The content of this chapter is based on parts of the papers [DKMV23] and [KV25].

Chapter 4: Cyclic proof systems for the modal μ-calculus In this chapter we tackle the first theme of this thesis and study how to obtain annotated proof systems. A special focus is put on investigating different proof systems for the modal μ-calculus. In Section 4.1 we define a uniform construction that, given a proof system where the soundness condition can be checked by a deterministic ω-automata, yields an annotated infinitary proof systems. This construction is uniform with respect to the proof system and the ω-automata. In Section 4.2 we will use this construction and the determinization method from Section 3.2 to obtain the infinitary proof system BT^∞ and the cyclic proof system BT for the modal μ-calculus. These proof systems are related to other proof systems in the literature. In Section 4.3 we study the annotated cyclic proof system Clo introduced by Afshari and Leigh [AL17] and show that it is incomplete. Section 4.1 was written explicitly for this thesis, Section 4.2 is based on [DKMV23] and Section 4.3 is based on an unpublished manuscript [Klo23].

Chapter 5: Interpolation for the two-way modal μ-calculus In this chapter we see the interplay of both themes mentioned above. We introduce multiple non-wellfounded proof systems for the two-way modal μ-calculus and use them to prove that the logic has interpolation. In Section 5.1 we introduce a trace-based infinitary proof system that is inspired by work on the alternation-free two-way modal μ-calculus by Rooduijn and Venema [RV23]. We show that this system is sound and complete. In the following section, Section 5.2, we use the determinization method from Section 3.3 and the uniform construction from Section 4.1 to define an annotated cyclic proof system JS_2. We then simplify the occurring annotations to obtain a refined annotated cyclic proof system Circ_2 in Section 5.3. Consequently, we use the system Circ_2 to show that the two-way modal μ-calculus enjoys Craig interpolation. This chapter is based on the paper [KV25]. Section 5.3 is written specifically for this thesis; this section is added to fix a mistake in [KV25].

Chapter 6: Interpolation for Converse PDL In this chapter we define an annotated cyclic proof system for Converse PDL and employ it to show that the logic has interpolation. In Section 6.1 we introduce the proof system, define a split version of it in the next section, and then show its soundness and completeness in Section 6.3. In the definition of the proof system we take inspiration from the proof systems introduced in Chapter 5. In the subsequent sections we use the cyclic proof system to show that Converse PDL has the Craig interpolation property. This proof is more intricate than the one for the two-way modal μ-calculus. Inspired by recent work on PDL [BGHRDV25], we carry out the interpolation proof on an auxiliary structure defined from the cyclic proof. This chapter is based on the paper [KTV25].

Chapter 7: Cut elimination for the alternation-free modal μ-calculus In the last chapter we show how annotated cyclic proof systems can be used to show cut elimination. Cut elimination is carried out within the Focus system defined by Marti and Venema [MV21a] for the alternation-free modal μ-calculus. Due to the simple shape of annotations in this system, we are able to perform proof transformations eliminating cuts inductively. One key aspect of our strategy is that we treat cuts outside of cycles differently than cuts residing inside cycles. This chapter is based on an, as for now, unpublished manuscript [AK25], and builds on the paper [AK24].

Chapter 2

Modal fixpoint logics

In this preliminary chapter, we introduce the logics we are working with and the proof-theoretic setting we utilize to study them.

Before we define the logics, we first fix terminology on graphs and trees. Because the semantics of our logics will be defined game-theoretically, we also define infinite two-player games. Then we introduce the modal μ-calculus; all other logics of study will evolve around it. We investigate an extension of the μ-calculus, the two-way modal μ-calculus, and fragments thereof: the alternation-free modal μ-calculus, PDL and Converse PDL. In the last two sections, we introduce non-wellfounded proofs – infinitary and cyclic proofs – in an abstract way and we introduce the infinitary proof system NW for the modal μ-calculus.

Our presentation will be largely self-contained. However, some familiarity with basic modal logic and finitary sequent calculi, as presented for instance in [BRV01] and [Tak87], will be helpful for the reader. This chapter does not contain original material, apart from the adequacy proof of the game semantics for PDL, which originates from [KTV25], and maybe its presentation.

2.1 Graphs and trees

In order to define proofs and games, we first need to specify our notion of graphs and trees. Because we are only interested in *countable* graphs and trees, we may restrict nodes to be naturals. This allows us to take the set of all trees without worrying about set-theoretic considerations.

2.1.1. Definition. A *graph* (V, E) is a set $V \subseteq \mathbb{N}$ with a binary relation $E \subseteq V \times V$. We say that (V, E) is *strongly connected*, if for all nodes $u, v \in V$ there is an E-path from u to v. A graph (V, E) is *connected* if for all nodes $u, v \in V$ there is an $E \cup \breve{E}$-path from u to v, where $\breve{E}$ is the converse relation of E. A *cycle* in a graph (V, E) is a non-empty E-path in which the first and last vertices are equal. We call a graph without cycles *acyclic*.

We call two graphs (V, E) and (V', E') *isomorphic*, if there is a bijection $f : V \to V'$ preserving the relation E, that is, uEv iff $f(u)E'f(v)$ for all $u, v \in T$.

2.1.2. Definition. A *tree* $(T, \lessdot)$ is a connected and acyclic graph with a unique node $r \in T$, called the *root*, such that for every node $v \in T$ there is a unique $\lessdot$-path from the root r to v.

We call the binary relation $\lessdot$ of a tree $(T, \lessdot)$ the *parent relation* and, if $v \lessdot u$, say that v is the *parent* of u and that u is the *child* of v. We call a node u a *descendant* of a node v if there are nodes $v = v_0, \ldots, v_n = u$ with $n > 0$ and v_{i+1} being a child of v_i for $i = 0, ..., n-1$. In this case we call v an *ancestor* of u. We will picture proof trees "growing upwards", that is, if two nodes are connected, then the node above is the child of the node below.

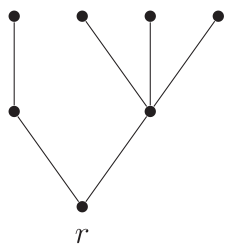


Figure 2.1: A tree

2.1.3. Definition. We say that a graph (V, E) is a *subgraph* of a graph (V, E) if $V' \subseteq V$ and the relations E' and E coincide on V'.

2.1.4. Definition. We say that a tree $(T', \lessdot')$ is a *subtree* of a tree $(T, \lessdot)$ if $T' \subseteq T$ is connected and the relations $\lessdot'$ and $\lessdot$ coincide on T'. $(T', \lessdot')$ is a *maximal subtree* of $(T, \lessdot)$ if it is a subtree which is upward closed, that is, if $t \in T'$ and $t \lessdot s$ then $s \in T'$.

2.2 Infinite games

We briefly define infinite two-player games to the extent needed in this thesis; for more details we refer to [GTW02]. We fix two players that we shall refer to as $\exists$ (Eloise, female) and $\forall$ (Abelard, male) and use P as a variable ranging over the set $\{\exists, \forall\}$.

Games

2.2.1. Definition. An *infinite two-player game* is a quadruple $\mathbb{G} = (V, E, O, W)$ where (V, E) is a graph; O is a map $V \to \{\exists, \forall\}$; and W is a set of infinite paths in (V, E).

Henceforth, we will call infinite two-player games simply *games*. We will refer to (V, E) as the *board* of the game. Elements of V will be called *positions*, and $O(v)$ is the *owner* of v. We write $V_P := O^{-1}(P)$ for the set of positions owned by P. Given a position v for player P, the set $E[v]$ denotes the set of *moves* that are *admissible for P* at v. The set W is called the *winning condition* of the game.

For a finite sequence $s = v_0...v_n$ we define $\mathsf{first}(s) := v_0$ and $\mathsf{last}(s) := v_n$.

2.2.2. Definition. A *match* $\mathcal{M}$ of the game $\mathbb{G} = (V, E, O, W)$ is a path through the graph (V, E). Such a match $\mathcal{M}$ is *full* if it is maximal as a path, that is, either finite with $E[\mathsf{last}(\mathcal{M})] = \varnothing$, or infinite. If the last position of a match has no E-successors, the owner of that positions *gets stuck* and loses the match. Infinite matches are won by $\exists$ if the match, as an E-path, belongs to the set W and won by $\forall$ otherwise.

Given these definitions, it should be clear that it does not matter which player owns a state that has a unique successor; for this reason we often take O to be a *partial* map, provided $O(v)$ is defined whenever $|E[v]| \neq 1$.

An *initialized game* is a pair consisting of a game $\mathbb{G}$ and an element v of V, usually denoted as $\mathbb{G}@v$. A match of $\mathbb{G}@v$ is a match of $\mathbb{G}$ starting at v.

Strategies Let PM_P denote the collection of partial matches $\mathcal{M}$ ending in a position $\mathsf{last}(\mathcal{M}) \in V_P$, and define $PM_P@v$ as the set of partial matches in PM_P starting at position v.

2.2.3. Definition. A *strategy* for a player P is a function $f : PM_P \to V$. A match $\mathcal{M} = (v_i)_{i<\kappa}$ of length $\kappa \leq \omega$ is *guided* by a P-strategy f, in short f*-guided*, if $f(v_0 v_1 \cdots v_{n-1}) = v_n$ for all $n < \kappa$ such that $v_0 \cdots v_{n-1} \in PM_P$.

2.2.4. Definition. A P-strategy f is *winning for P from v* if P wins all f-guided full matches starting at v. A position v is a *winning position* for player $P \in \{\exists, \forall\}$ if P has a winning strategy in the game $\mathbb{G}@v$; the set of these positions is denoted as $Win_P(\mathbb{G})$. The game $\mathbb{G}$ is *determined* if every position is winning for either $\exists$ or $\forall$.

2.2.5. Definition. A strategy is *positional* if it only depends on the last position of a partial match, namely, if $f(\mathcal{M}) = f(\mathcal{M}')$ whenever $\mathsf{last}(\mathcal{M}) = \mathsf{last}(\mathcal{M}')$.

Positional strategies can and will be presented as a map $f : V_P \to V$.

2.2.6. Definition. A strategy f for P is a *finite-memory strategy*, if there is a finite memory set M, an element $m_I \in M$ and a map $(h_1, h_2) : V \times M \to V \times M$ such that for all partial matches $v_0 \dots v_n \in PM_P$ and $m_0, ..., m_n \in M$, if $m_0 = m_I$ and $m_{i+1} = h_2(v_i, m_i)$ for all $i < k$, then $f(v_0 \dots v_n) = h_1(v_n, m_n)$.

Given an initialized game $\mathbb{G}@v$, the set of partial matches can be presented as the *game tree* $(T_{\mathbb{G}}, \lessdot_{\mathbb{G}})$ consisting of all partial matches $v_0 \dots v_n$, where $v_0 \dots v_n \lessdot_{\mathbb{G}} v_0 \dots v_n v_{n+1}$ if $v_{n+1} \in E[v_n]$. According to this definition, the set of branches in $(T_{\mathbb{G}}, \lessdot_{\mathbb{G}})$ corresponds to the set of matches of $\mathbb{G}@v$. Given a strategy f for P, the *strategy tree* for P in $\mathbb{G}@v$ is the maximal subgraph of its game tree, where every partial match $\mathcal{M} \in PM_P$ has one unique child $f(\mathcal{M})$ if $E[\mathsf{last}(\mathcal{M})] \neq \varnothing$. If V is finite and the strategy f is positional or a finite-memory strategy, then the strategy tree is regular. That is, it only has finitely many distinct maximal subtrees up to isomorphism. Note that the game tree is always regular whenever V is finite.

Determinacy

2.2.7. Definition. A *parity game* is a game $\mathbb{G} = (V, E, O, W_\Omega)$ in which the winning condition W_Ω is given by a bounded priority map $\Omega : V \to \mathbb{N}$ as follows: $\mathcal{M} \in W_\Omega$ iff $\max\{\Omega(v) \mid v \text{ occurs infinitely often in } \mathcal{M}\}$ is even.

Such a parity game is usually denoted as $\mathbb{G} = (V, E, O, \Omega)$. The following theorem is independently due to Emerson and Jutla [EJ99] and Mostowski [Mos91].

2.2.8. Theorem (Positional Determinacy). *Let $\mathbb{G} = (V, E, O, \Omega)$ be a parity game. Then $\mathbb{G}$ is determined, and both players have positional winning strategies.*

Let Σ be a finite set, called an alphabet. We call a set $L \subseteq \Sigma^\omega$ a *language* over Σ. A language L is *regular*, if there is a parity automata $\mathbb{A}$ such that $L = \mathcal{L}(\mathbb{A})$. We will formally define parity automata in Chapter 3.

2.2.9. Definition. We say that $\mathbb{G} = (V, E, O, W)$ is a *regular game*, if there is finite set C and a coloring function $\chi : V \to C$ such that $\{\chi(\mathcal{M}) \mid \mathcal{M} \in W\}$ is a regular language over C, where χ extends to matches in the natural way by $\chi(v_0 v_1 \dots) := \chi(v_0)\chi(v_1)\dots$.

The following proposition follows from [BL69] and the positional determinacy of parity games.

2.2.10. Proposition. *Let $\mathbb{G} = (V, E, O, W)$ be a regular game. Then $\mathbb{G}$ is determined, and both players have finite-memory winning strategies.*

2.3 The modal μ-calculus

The modal μ-calculus is the archetypal modal fixpoint logic. It extends basic modal logic with explicit least and greatest fixpoint operators, allowing the formulation of various recursive phenomena. The version used today was introduced by Kozen [Koz83] in 1983 and has since become a key tool in the formal study of the behavior of programs and the dynamics of processes in general. Despite its expressive power, the μ-calculus still has the finite model property [Koz88] and reasonable computational properties; its model checking problem is in quasi-polynomial time [CJKLS17] and its satisfiability problem is EXPTIME complete [EJ99].

Many of the metalogical results on the μ-calculus have been obtained by characterizing the logic with so-called alternating automata. Through this automata-theoretic approach, Janin and Walukiewicz [JW96] showed that, for bisimulation-invariant properties, the modal μ-calculus has the same expressive power as monadic second-order logic. Building on their work, D'Agostino and Hollenberg proved that the logic also enjoys uniform interpolation [DH00]. In Chapter 4 we will see that automata theory plays an important role in the proof theory of the modal μ-calculus as well.

2.3.1 Syntax

Throughout this thesis we fix a countably infinite set Prop of *proposition letters* and a countably infinite set Var of *variables* with $\mathsf{Prop} \cap \mathsf{Var} = \varnothing$.

2.3.1. Definition. Let Act be a set of *actions*. The set $\mathcal{L}_\mu(\mathsf{Act})$ of *formulas* of the modal μ-calculus is generated by the grammar

$$\varphi := \bot \mid \top \mid p \mid \overline{p} \mid x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle a \rangle \varphi \mid [a]\varphi \mid \mu x.\varphi \mid \nu x.\varphi,$$

where $p \in \mathsf{Prop}$, $x \in \mathsf{Var}$ and $a \in \mathsf{Act}$.

2.3.2. Remark. Usually, the specific choice of the set Act of actions is not important, and we will simply write $\mathcal{L}_\mu$ instead of $\mathcal{L}_\mu(\mathsf{Act})$. If $\mathsf{Act} = \{a\}$ is a singleton, we will write $\Box, \Diamond$ instead of $[a], \langle a \rangle$, respectively.

We call formulas of the form $\bot, \top, p, \overline{p}, x, \varphi \vee \psi$ and $\varphi \wedge \psi$ *propositional*, formulas of the form $\langle a \rangle \varphi$ *diamond-formulas* and formulas of the form $[a]\varphi$ *box-formulas*; diamond and box-formulas are also called *modal*. We refer to formulas of the form $\mu x.\varphi$ as *μ-formulas*, and to formulas of the form $\nu x.\varphi$ as *ν-formulas*; formulas of either kind are called *fixpoint formulas*, and the symbols μ and ν themselves are *fixpoint operators*. This terminology is in line with the intended semantics, where $\mu x.\varphi$ and $\nu x.\varphi$ describe, respectively, the least and greatest fixpoint of φ

with respect to x. We use $\eta, \lambda \in \{\mu, \nu\}$ to denote arbitrary fixpoint operators. Formulas that do not contain fixpoint operators are called *fixpoint-free formulas*.

Note that we assume formulas to be in *negation normal form*, that is, only proposition letters can be negated and we do not have an explicit negation in the language. However, we can define such a negation operation as follows.

2.3.3. Definition. We inductively extend the map $p \mapsto \overline{p}$ to a full-blown *negation operation* on all formulas $\varphi \in \mathcal{L}_\mu$:

$$\begin{array}{lllllllll} \overline{\bot} & := & \top & \overline{\varphi \wedge \psi} & := & \overline{\varphi} \vee \overline{\psi} & \overline{\mu x.\varphi} & := & \nu x.\overline{\varphi} \\ \overline{\top} & := & \bot & \overline{\varphi \vee \psi} & := & \overline{\varphi} \wedge \overline{\psi} & \overline{\nu x.\varphi} & := & \mu x.\overline{\varphi} \\ \overline{x} & := & x & \overline{[a]\varphi} & := & \langle a \rangle \overline{\varphi} & & & \\ \overline{\overline{p}} & := & p & \overline{\langle a \rangle \varphi} & := & [a]\overline{\varphi} & & & \end{array}$$

Note that $\overline{\overline{\varphi}} = \varphi$ for every formula φ.

We use standard terminology and notation for the binding of variables by the fixpoint operators and for the substitution operation.

2.3.4. Definition. In a fixpoint formula $\eta x.\psi$, the *scope* of ηx is defined as its subformula ψ. The scope of modalities is defined analogously. We call an occurrence of x in a formula φ *bound* if it is in the scope of a fixpoint operator ηx and *free* otherwise.

We define $\mathsf{BV}(\varphi)$ as the set of bound variables occurring in a formula φ and $\mathsf{FV}(\varphi)$ as the set of free variables occurring in φ.

Note that $\mathsf{BV}(\varphi)$ and $\mathsf{FV}(\varphi)$ are not necessarily disjoint. We call a formula φ a *sentence* if $\mathsf{FV}(\varphi) = \varnothing$. Unless otherwise noted we will assume that every formula is a sentence.

2.3.5. Definition. Given a formula φ with free variable x, and a sentence χ, we define the *substitution* of χ for x in φ, written $\varphi[\chi/x]$, to be the formula obtained from φ by replacing every free occurrence of x in φ with χ.

Because χ is a sentence, we make sure that no variable capture may occur in a substitution. An important use of the substitution operation concerns the *unfolding* $\varphi[\eta x.\varphi/x]$ of a fixpoint formula $\eta x.\varphi$.

2.3.6. Definition. A variable x is *guarded* in a formula ψ, if every free occurrence of x in ψ is in the scope of a modality. A formula φ is called *guarded*, if in every subformula $\eta x.\psi$ of φ, every free occurrence of x is guarded in ψ.

Assuming formulas to be guarded is not a real restriction, as shown by the following lemma. A proof can for instance be found in [DGL16].

2.3.7. Lemma. *Every formula is equivalent to a guarded one.*

2.3.8. Definition. We define the set of subformulas of a formula φ as usual. We say that ψ is a *direct subformula* of φ, written as $\psi \lhd \varphi$, if either

1. $\varphi = \chi_0 \circ \chi_1$ and $\psi \in \{\chi_0, \chi_1\}$ with $\circ \in \{\wedge, \vee\}$;
2. $\varphi = \triangle\psi$ with $\triangle \in \{\langle a\rangle, [a] \mid a \in \mathsf{Act}\}$; or
3. $\varphi = \eta x.\psi$ with $\eta \in \{\mu, \nu\}$.

We let $\unlhd$ be the reflexive and transitive closure of $\lhd$ and say that ψ is a *subformula* of φ if $\psi \unlhd \varphi$.

In basic modal logic, the semantic truth of a formula φ can be captured by the truth of its subformulas. For the modal μ-calculus, this is more complicated. Consider the formula $\eta x.\varphi$ and its subformula φ. As we will see later on, the truth of $\eta x.\varphi$ depends on the truth of φ in *different models*, where an interpretation to the free variable x is given.

The more natural notion is therefore that of the *closure* of φ, obtained by replacing clause 3 in Definition 2.3.8 with one that unfolds fixpoint formulas. Note that because $\eta x.\varphi$ will be interpreted as a fixpoint of φ with respect to x, it is semantically equivalent to its *unfolding* $\varphi[\eta x.\varphi/x]$.

2.3.9. Definition. Given two formulas $\varphi, \psi \in \mathcal{L}_\mu$ we write $\varphi \to_C \psi$ if either

1. $\varphi = \chi_0 \circ \chi_1$ and $\psi \in \{\chi_0, \chi_1\}$ with $\circ \in \{\wedge, \vee\}$;
2. $\varphi = \triangle\psi$ with $\triangle \in \{\langle a\rangle, [a] \mid a \in \mathsf{Act}\}$; or
3. $\psi = \varphi[\eta x.\varphi/x]$ with $\eta \in \{\mu, \nu\}$.

We let $\twoheadrightarrow_C$ be the reflexive and transitive closure of $\to_C$ and write $\varphi \equiv_C \psi$ if $\varphi \twoheadrightarrow_C \psi$ and $\psi \twoheadrightarrow_C \varphi$.

The *closure* $\mathsf{Clos}(\varphi)$ of φ is the least set of formulas containing φ that is closed under the relation $\twoheadrightarrow_C$.

The following facts about the closure are standard, a proof can for example be found in [Ven20].

2.3.10. Lemma. *Let φ be a formula. Then*

1. $\mathsf{Clos}(\varphi)$ *is finite;*
2. *If φ is a sentence, then every formula in* $\mathsf{Clos}(\varphi)$ *is a sentence;*
3. *If φ is guarded, then every formula in* $\mathsf{Clos}(\varphi)$ *is guarded.*

2.3.11. Definition. For a set of formulas Φ we define $\overline{\Phi} := \{\overline{\varphi} \mid \varphi \in \Phi\}$. We define $\mathsf{Clos}(\Phi) := \bigcup_{\varphi\in\Phi} \mathsf{Clos}(\varphi)$ and $\mathsf{Clos}^{\neg}(\Phi) := \mathsf{Clos}(\Phi) \cup \mathsf{Clos}(\overline{\Phi})$.

Note that, if Φ is finite, then so are $\mathsf{Clos}(\Phi)$ and $\mathsf{Clos}^{\neg}(\Phi)$.

2.3.12. DEFINITION. A *trace* is a sequence $(\varphi_n)_{n<\kappa}$, with $\kappa \leq \omega$, such that $\varphi_n \to_C \varphi_{n+1}$, for all $n+1 < \kappa$.

The following lemma plays a crucial role in the definition of the game semantics and proof systems for the modal μ-calculus. A proof can for instance be found in [Ven20].

2.3.13. LEMMA. *Any infinite trace $\tau = (\varphi_n)_{n<\omega}$ features a unique fixpoint formula φ that occurs infinitely often on τ and is a subformula of φ_n for cofinitely many n.*

If the fixpoint formula φ in the previous lemma is a μ-formula, we call τ a *μ-trace*, and if it is a ν-formula, we call τ a *ν-trace*.

2.3.14. DEFINITION. Let Fix be the set of fixpoint formulas in $\mathcal{L}_\mu$. We define a *dependence order* on Fix by setting $\eta x.\varphi \leq_d \lambda y.\psi$ if $\eta x.\varphi \equiv_C \lambda y.\psi$ and $\lambda y.\psi \trianglelefteq \eta x.\varphi$.

Note that bigger in $\leq_d$ means being of higher priority.

2.3.15. DEFINITION. Let $\mathbb{N}^+ := \mathbb{N} \setminus \{0\}$. We define the *priority function* $\Omega_\mu : \mathsf{Fix} \to \mathbb{N}^+$ to be the minimal-valued function such that

1. $\Omega_\mu(\eta x.\varphi) \leq \Omega_\mu(\lambda y.\psi)$ if $\eta x.\varphi \leq_d \lambda y \psi$ and
2. $\Omega_\mu(\eta x.\varphi)$ is even iff $\eta = \nu$.

We extend Ω_μ to a function $\Omega_\mu : \mathcal{L}_\mu \to \mathbb{N}^+$ by setting $\Omega_\mu(\varphi) = 1$ if φ is not a fixpoint formula.

2.3.16. REMARK. Note that the priority function Ω_μ is well-defined. That is, there is a minimum-valued function $\mathsf{Fix} \to \mathbb{N}^+$ satisfying conditions 1 and 2. In order to see this, assume that Ω_0 and Ω_1 are functions from $\mathsf{Fix} \to \mathbb{N}^+$ satisfying conditions 1 and 2. Then it easy to see that the function

$$\begin{aligned} \Omega_2 : \quad & \mathsf{Fix} \to \mathbb{N}^+ \\ & \eta x.\varphi \mapsto \min\{\Omega_0(\eta x.\varphi), \Omega_1(\eta x.\varphi)\} \end{aligned}$$

satisfies conditions 1 and 2 aswell. We can therefore define Ω_μ as the pointwise minimum of all such functions.

The following lemma follows easily from the definitions and Lemma 2.3.13.

2.3.17. LEMMA. *Let $\tau = (\varphi_n)_{n<\omega}$ be an infinite trace. Then τ is a ν-trace iff $\max\{\Omega_\mu(\varphi) \mid \varphi$ occurs infinitely often on $\tau\}$ is even.*

2.3.2 Semantics

Formulas of the modal μ-calculus are interpreted in Kripke models. Traditionally, the meaning of formulas is given by *relational semantics*. However, it is more convenient for us to work with an alternative *game semantics*. We define both semantics and refer to [DGL16] for a proof of their equivalence.

2.3.18. Definition. A *Kripke model* $\mathbb{S} = (S, R, V)$ consists of a set S of states; a family of binary relations $R = \{R_a \subseteq S^2 \mid a \in \mathsf{Act}\}$ on S; and a valuation $V : \mathsf{Prop} \to \mathcal{P}(S)$. A *pointed model* is a pair $(\mathbb{S}, s)$ where $\mathbb{S}$ is a Kripke model and $s \in S$.

Game semantics

2.3.19. Definition. Let $\mathbb{S} = (S, R, V)$ be a Kripke model. The *evaluation game* $\mathcal{E}_\mu(\mathbb{S})$ is the following infinite two-player game. Its positions are pairs of the form $(\varphi, s) \in \mathcal{L}_\mu \times S$, and its ownership function and admissible moves are given in Table 2.1. As usual, finite matches are lost by the player who got stuck. Infinite matches of the form $(\varphi_n, s_n)_{n<\omega}$ are won by $\exists$ if the induced trace $(\varphi_n)_{n<\omega}$ is a ν-trace, and won by $\forall$ if it is a μ-trace.

Position	Owner	Admissible moves
$(\bot, s)$	$\exists$	$\varnothing$
$(\top, s)$	$\forall$	$\varnothing$
(p, s) with $s \in V(p)$	$\forall$	$\varnothing$
(p, s) with $s \notin V(p)$	$\exists$	$\varnothing$
$(\overline{p}, s)$ with $s \in V(p)$	$\exists$	$\varnothing$
$(\overline{p}, s)$ with $s \notin V(p)$	$\forall$	$\varnothing$
$(\varphi \vee \psi, s)$	$\exists$	$\{(\varphi, s), (\psi, s)\}$
$(\varphi \wedge \psi, s)$	$\forall$	$\{(\varphi, s), (\psi, s)\}$
$(\langle a \rangle \varphi, s)$	$\exists$	$\{(\varphi, t) \mid (s, t) \in R_a\}$
$([a]\varphi, s)$	$\forall$	$\{(\varphi, t) \mid (s, t) \in R_a\}$
$(\eta x.\varphi, s)$	-	$\{(\varphi[\eta x.\varphi/x], s)\}$

Table 2.1: The evaluation game $\mathcal{E}_\mu(\mathbb{S})$

The evaluation game $\mathcal{E}_\mu(\mathbb{S})$ can be presented as a parity game by defining the priority function

$$\begin{aligned} \Omega : \mathcal{L}_\mu \times S &\to \mathbb{N}, \\ (\varphi, s) &\mapsto \Omega_\mu(\varphi). \end{aligned}$$

Because of Lemma 2.3.17 this is an equivalent definition. Theorem 2.2.8 therefore implies that the evaluation game is positionally determined and we may assume that both players play positional strategies.

2.3.20. Remark. Note that in a match starting at a position (φ, s), all positions are of the form (ψ, t) with $\psi \in \mathsf{Clos}(\varphi)$. Therefore, if the model $\mathbb{S}$ is finite, matches in $\mathcal{E}_\mu(\mathbb{S})$ only reach finitely many different positions.

2.3.21. Definition. Let $\mathbb{S}, s$ be a pointed model, let f be a strategy for $\exists$ in $\mathcal{E}_\mu(\mathbb{S})$ and let φ be a formula. We write $\mathbb{S}, s \Vdash_f \varphi$ if f is winning for $\exists$ at (φ, s). We define $\mathbb{S}, s \Vdash \varphi$ if $\mathbb{S}, s \Vdash_f \varphi$ for some strategy f for $\exists$. We say that a formula φ is *satisfiable*, if there exists $\mathbb{S}, s$ such that $\mathbb{S}, s \Vdash \varphi$ and *unsatisfiable* otherwise. A formula φ is called *valid*, if $\mathbb{S}, s \Vdash \varphi$ for all pointed models $\mathbb{S}, s$.

We write $\varphi \vDash \psi$ for the *local consequence relation* meaning that $\mathbb{S}, s \Vdash \varphi$ implies $\mathbb{S}, s \Vdash \psi$ for every pointed model $\mathbb{S}, s$. We say that two formulas φ and ψ are *semantically equivalent*, written as $\varphi \equiv \psi$, if $\varphi \vDash \psi$ and $\psi \vDash \varphi$.

2.3.22. Example. Define the following $\mathcal{L}_\mu$-formulas:

$$\begin{aligned}\varphi &:= \nu x.\Diamond(x \wedge \mu y.\Diamond y \vee p),\\ \chi &:= \nu x.\mu y.\Diamond[(x \wedge p) \vee (y \wedge \overline{p})].\end{aligned}$$

Then $\mathbb{S}, s \Vdash \varphi$ if there is an infinite path in $\mathbb{S}$ starting from s, such that at each state on the path, a state satisfying p is reachable.
We have $\mathbb{S}, s \Vdash \chi$ if there is an infinite path in $\mathbb{S}$ starting from s on which infinitely often p holds. We therefore have $\chi \vDash \varphi$.

Relational semantics We present the standard, relational semantics of $\mathcal{L}_\mu$, and we show its equivalence to the game semantics. In the relational semantics, given a Kripke model $\mathbb{S}$, formulas are inductively interpreted as subsets over the carrier of $\mathbb{S}$. For the modal clause of this definition we need the fact that any binary relation R on a set S induces two operations on $\mathcal{P}(S)$:

$$\begin{array}{lcl}\langle R\rangle(U) & := & \{s \in S \mid R[s] \cap U \neq \varnothing\}\\ {[R]}(U) & := & \{s \in S \mid R[s] \subseteq U\}.\end{array}$$

We are mainly interested in the semantics of sentences in Kripke models. However, in order to define the relational semantics, we define the meaning of a formula inductively on its subformulas. We therefore need a definition that also covers formulas with free variables. As usual, this is done by an interpretation of the free variables in a Kripke model.

2.3.23. Definition. Let $\mathbb{S} = (S, R, V)$ be a Kripke model and let $X \in \mathcal{P}(S)$. An *interpretation* $\mathcal{I}$ of the variables in $\mathbb{S}$ is a function $\mathsf{Var} \to \mathcal{P}(S)$. Given such an interpretation $\mathcal{I}$, we define the interpretation $\mathcal{I}[x \mapsto X]$ by

$$\mathcal{I}[x \mapsto X](y) := \begin{cases} X & \text{if } y = x\\ \mathcal{I}(y) & \text{otherwise}\end{cases}$$

2.3.24. Definition. Given a Kripke model $\mathbb{S} = (S, R, V)$, we inductively define the meaning $[\![\varphi]\!]^{\mathbb{S}}_{\mathcal{I}} \subseteq S$ of a formula φ in $(\mathbb{S}, \mathcal{I})$ as follows:

$$
\begin{array}{lll}
[\![\top]\!]^{\mathbb{S}}_{\mathcal{I}} & := & S \\
[\![\bot]\!]^{\mathbb{S}}_{\mathcal{I}} & := & \varnothing \\
[\![p]\!]^{\mathbb{S}}_{\mathcal{I}} & := & V(p) \\
[\![\overline{p}]\!]^{\mathbb{S}}_{\mathcal{I}} & := & S \setminus V(p) \\
[\![x]\!]^{\mathbb{S}}_{\mathcal{I}} & := & \mathcal{I}(x) \\
[\![\varphi \vee \psi]\!]^{\mathbb{S}}_{\mathcal{I}} & := & [\![\varphi]\!]^{\mathbb{S}}_{\mathcal{I}} \cup [\![\psi]\!]^{\mathbb{S}}_{\mathcal{I}} \\
[\![\varphi \wedge \psi]\!]^{\mathbb{S}}_{\mathcal{I}} & := & [\![\varphi]\!]^{\mathbb{S}}_{\mathcal{I}} \cap [\![\psi]\!]^{\mathbb{S}}_{\mathcal{I}} \\
[\![\langle a \rangle \varphi]\!]^{\mathbb{S}}_{\mathcal{I}} & := & \langle R_a \rangle ([\![\varphi]\!]^{\mathbb{S}}_{\mathcal{I}}) \\
[\![[a] \varphi]\!]^{\mathbb{S}}_{\mathcal{I}} & := & [R_a]([\![\varphi]\!]^{\mathbb{S}}_{\mathcal{I}}) \\
[\![\mu x.\varphi]\!]^{\mathbb{S}}_{\mathcal{I}} & := & \bigcap \{X \subseteq S \mid [\![\varphi]\!]^{\mathbb{S}}_{\mathcal{I}[x \mapsto X]} \subseteq X\} \\
[\![\nu x.\varphi]\!]^{\mathbb{S}}_{\mathcal{I}} & := & \bigcup \{X \subseteq S \mid X \subseteq [\![\varphi]\!]^{\mathbb{S}}_{\mathcal{I}[x \mapsto X]}\}
\end{array}
$$

If φ is a sentence, the interpretation $\mathcal{I}$ may be omitted, and we write $[\![\varphi]\!]^{\mathbb{S}}$ for the meaning of φ in $\mathbb{S}$.

2.3.25. Remark. Let $(L, \leq)$ be a complete lattice. A *prefixpoint* of a function $f : L \to L$ is an element $a \in L$ such that $f(a) \leq a$. If f is monotone, then the Knaster-Tarski Theorem [Tar55] states that the least fixpoint of f coincides with the infimum of all prefixpoints, that is,

$$\mu x.f = \bigwedge \{a \mid f(a) \leq a\}.$$

Note that $(\mathcal{P}(S), \subseteq)$ is a complete lattice and that $\mu x.\varphi$ is interpreted as the intersection of all prefixpoints of the function

$$
\begin{aligned}
\varphi_x : \mathcal{P}(S) &\to \mathcal{P}(S) \\
X &\mapsto [\![\varphi]\!]^{\mathbb{S}}_{\mathcal{I}[x \mapsto X]}
\end{aligned}
$$

Because x only occurs positively in φ, the function φ_x is monotone. Therefore, $\mu x.\varphi$ is interpreted as the least fixed point of φ_x. Dually, the interpretation of $\nu x.\varphi$ is the greatest fixpoint of φ_x.

Recall that $\mathbb{S}, s \Vdash \varphi$ if $\exists$ has a winning strategy in $\mathcal{E}_\mu(\mathbb{S})@(\varphi, s)$. The following theorem states that the game semantics coincides with the relational semantics. For a proof we refer to [DGL16].

2.3.26. Theorem (Adequacy). *For every pointed model $\mathbb{S}, s$ and formula φ it holds that*

$$\mathbb{S}, s \Vdash \varphi \quad \textit{iff} \quad s \in [\![\varphi]\!]^{\mathbb{S}}.$$

2.4 Extensions and fragments of the modal μ-calculus

2.4.1 The two-way modal μ-calculus

The language of the two-way modal μ-calculus is obtained from the modal μ-calculus by adding for each modality a, a modality $\breve{a}$ which in the semantics will be interpreted as the converse of the accessibility relation for a. This addition enables the logic to reason about the past; logics of this kind are also referred to as tense logics [Nis80]. The ability to argue about past behaviors is attractive from the perspective of formal program verification [LPZ85], but also in the area of description logics, where converse modalities correspond to inverse roles [GL94].

Compared to its one-way version, surprisingly little seems to be known about this logic. Remarkably, the two-way μ-calculus lacks the finite model property, as witnessed by the formula

$$\nu x.(\langle a\rangle x \wedge \mu y.[\breve{a}]y)$$

stating that there is an infinite a-path along which all backward $\breve{a}$-paths are finite. A key result by Vardi [Var98] states that the satisfiablity problem for the two-way μ-calculus still can be solved in exponential time. Many other questions remain open, such as the existence of a complete axiomatization or whether the logic has uniform interpolation.

2.4.1. Definition. Let Act be a set of actions. The language of the *two-way* modal μ-calculus $\mathcal{L}^2_\mu(\mathsf{Act})$ is precisely the same as that of $\mathcal{L}_\mu(\mathsf{Act})$ with the additional assumption that there is an involution operation $\breve{\cdot} : \mathsf{Act} \to \mathsf{Act}$ such that for every $a \in \mathsf{Act}$ it holds that $\breve{a} \neq a$ and $\breve{\breve{a}} = a$.

As for $\mathcal{L}_\mu$, we simply write $\mathcal{L}^2_\mu$ whenever the specific set of actions Act is not important.

2.4.2. Definition. Let φ be a $\mathcal{L}^2_\mu$-formula. We define the *vocabulary of* φ, written $\mathsf{Voc}(\varphi)$, to be the set of proposition letters and actions occurring in φ with the proviso that we include both a and $\breve{a}$ in the vocabulary of any expression in which a or $\breve{a}$ occurs.

All notions for $\mathcal{L}_\mu$ are defined analogously for $\mathcal{L}^2_\mu$. The only exception is, that we interpret $\mathcal{L}^2_\mu$-formulas only on models where the converse relation $R_{\breve{a}}$ is interpreted as the converse of R_a.

2.4.3. Definition. A *two-way Kripke model* is a Kripke model that satisfies the following property:

$$R_{\breve{a}} = \{(s,t) \mid (t,s) \in R_a\} \text{ for every } a \in \mathsf{Act}.$$

An $\mathcal{L}_\mu^2$-formula φ is *satisfiable*, if there is a two-way Kripke model that satisfies φ and *unsatisfiable* otherwise.

2.4.4. Example. Consider the $\mathcal{L}_\mu^2$-formula $\nu x.\langle a\rangle\langle\breve{a}\rangle x$. We have that

$$\mathbb{S}, s \Vdash \nu x.\langle a\rangle\langle\breve{a}\rangle x$$

iff there is an infinite $(a, \breve{a})$-path in $\mathbb{S}$ starting from s. It therefore holds that $\mathbb{S}, s \Vdash \langle a\rangle p \to \nu x.\langle a\rangle\langle\breve{a}\rangle x$ for every two-way Kripke model $\mathbb{S}$ and $s \in \mathbb{S}$. Thus, $\langle a\rangle p \vDash \nu x.\langle a\rangle\langle\breve{a}\rangle x$.

2.4.5. Example. Let s, t be nodes in Kripke model $\mathbb{S}$. We say that t is a-reachable from s if there is an a-path from s to t. We have

$$\mathbb{S}, s \Vdash \nu x.[a]x \wedge \mu y.\langle\breve{a}\rangle y \vee q$$

if for every a-reachable node t there is an $\breve{a}$-reachable node r from t where q holds. For two-way Kripke models a node t is a-reachable from a node s iff s is $\breve{a}$-reachable from t. This implies that $\mathbb{S}, s \Vdash q \to \nu x.[a]x \wedge \mu y.\langle\breve{a}\rangle y \vee q$ for all two-way Kripke models and thus $q \to \nu x.[a]x \wedge \mu y.\langle\breve{a}\rangle y \vee q$ is valid.

2.4.2 The alternation-free modal μ-calculus

Although the modal μ-calculus is defined in a general way, many important concepts can already be captured within specific fragments of the logic. In fact, a range of dynamic and temporal logics – such as PDL, LTL, and CTL – can be expressed as fragments of the modal μ-calculus by appropriately restricting the use of fixpoint operators. In particular, many of these logics fall within the *alternation-free modal μ-calculus*: the fragment of the μ-calculus where least and greatest fixpoints are not interleaved. While the alternation-free μ-calculus is strictly less expressive than $\mathcal{L}_\mu$ over all models [Bra98], it attains the same expressive power over certain classes of frames [AF09; GKL14]. Moreover, the logic enjoys Craig interpolation [DAg18], further reinforcing its importance as a logic worthy of independent study.

2.4.6. Definition. We call a formula $\varphi \in \mathcal{L}_\mu$ *alternation-free* if for any subformula $\eta x.\psi$ of φ no free occurrence of x in ψ is in the scope of an $\overline{\eta}$-operator.

The language of the *alternation-free modal μ-calculus* $\mathcal{L}_\mu^{af}$ is the set of all alternation-free $\mathcal{L}_\mu$-formulas.

An alternative inductive definition of $\mathcal{L}_\mu^{af}$ can be found in [MV21a]. Recall that $\varphi \equiv_C \psi$ if there is a trace from φ to ψ and vice versa.

2.4.7. Example. The following $\mathcal{L}_\mu$-formula is alternation-free:

$$\nu x.\Box(x \wedge \mu y.\Box y \vee p),$$

whereas the following $\mathcal{L}_\mu$-formula is not alternation-free:

$$\nu x.\mu y.\Box[(x \wedge p) \vee (y \wedge \overline{p})].$$

2.4.8. Definition. We call a formula φ *magenta*, if there is $\mu x.\psi$ such that $\varphi \equiv_C \mu x.\psi$, and *navy*, if there is $\nu x.\psi$ such that $\varphi \equiv_C \nu x.\psi$.

The following proposition summarises key properties of $\mathcal{L}_\mu^{af}$.

2.4.9. Proposition. *Let φ be an alternation-free formula. Then*

1. *its negation $\overline{\varphi}$ is alternation-free,*
2. *every subformula of φ is alternation-free,*
3. *every formula in $\mathsf{Clos}(\varphi)$ is alternation-free,*
4. *there is a guarded and alternation-free formula φ' such that $\varphi' \equiv \varphi$, and*
5. *φ is not both magenta and navy.*

Proof:
Items 1–3 are immediate and item 4 follows from the standard translation of a μ-calculus formula to a guarded one (see for instance [DGL16]). We prove item 5. Towards a contradiction assume that $\xi \in \mathcal{L}_\mu^{af}$ is both magenta and navy. Then there is a pair of formulas $\eta x.\varphi, \overline{\eta} y.\psi \in \mathsf{Clos}(\xi)$ with $\eta x.\varphi \twoheadrightarrow_C \overline{\eta} y.\psi$ and $\overline{\eta} y.\psi \twoheadrightarrow_C \eta x.\varphi$.

For any trace $\chi_1...\chi_n$ there is $i \in \{1, ..., n\}$ such that χ_i is a subformula of every formula on the trace and such that χ_i is a fixpoint formula if χ_n is one. This can be shown by induction on the length of the trace.

Therefore, we can find such formulas $\eta x.\varphi, \overline{\eta} y.\psi$ that also satisfy $\eta x.\varphi \trianglelefteq \overline{\eta} y.\psi$. Moreover, without loss of generality we may assume that all fixpoint formulas occurring on the trace $\tau : \eta x.\varphi \cdots \overline{\eta} y.\psi$ are η-formulas (apart from $\overline{\eta} y.\psi$). Otherwise take the subtrace of τ from $\eta x.\varphi$ to the first $\overline{\eta}$-formula $\overline{\eta} y.\psi'$.

Due to Proposition 2.4.9.3 all formulas on τ are alternation-free. Thus there is no free occurrence of z in ψ for any η-formula $\eta z.\delta$ on τ. As all fixpoint formulas occurring on τ are η-formulas it follows inductively that $\overline{\eta} y.\psi$ is a subformula of every formula on τ, in particular $\overline{\eta} y.\psi \trianglelefteq \eta x.\varphi$. Yet this contradicts $\eta x.\varphi \trianglelefteq \overline{\eta} y.\psi$.
□

2.4.10. Remark. Combining item 3 and 5 of Proposition 2.4.9 yields that for an alternation-free formula φ, no formula in $\mathsf{Clos}(\varphi)$ is both magenta and navy. Conversely, it is not so hard to see that if no formula in $\mathsf{Clos}(\varphi)$ is both magenta and navy, then φ is alternation-free. Therefore, we can alternatively characterize $\mathcal{L}_\mu^{af}$ as the set of $\mathcal{L}_\mu$-formulas φ, where for any formulas of the form $\mu x.\psi$ and $\nu y.\chi$ in $\mathsf{Clos}(\varphi)$ it holds that $\mu x.\psi \not\equiv_C \nu y.\chi$.

2.5 PDL and Converse PDL

Propositional Dynamic Logic (abbreviated: PDL) was introduced by Fischer and Ladner [FL79] in 1979 as a propositional formalism to reason about the behavior of programs. The language of PDL features an infinite collection of modalities, the intended interpretation of $\langle\alpha\rangle\varphi$ being that "after some execution of the program α, the formula φ holds". The inductive structure of programs is reflected by the syntax of PDL, where complex programs are constructed from atomic ones and formula tests, by means of program constructors for sequential composition, nondeterministic choice and iteration.

Converse PDL or CPDL, also defined in [FL79], extends PDL with a converse operator on programs, which facilitates backwards reasoning about programs. PDL and CPDL also have applications in for instance knowledge representation [BL07], where the program expressions represent *roles* between objects, and the program constructions correspond to natural operations on such roles; in particular, the converse operator corresponds to *inverse roles.*

PDL and CPDL both have the small-model property and an Exptime-complete satisfiability problem, as established by Fischer and Ladner [FL79] and Pratt [Pra80]. A natural axiomatisation was given by Segerberg [Seg77] and proved to be complete by Parikh [Par78] and others. Generally, PDL and related formalisms have been recognized as important modal fixpoint logics for quite some time now, see for instance Troquard and Balbiani [TB23] for a recent survey.

Interestingly, PDL and CPDL correspond to fragments of $\mathcal{L}_\mu$ and $\mathcal{L}_\mu^2$, respectively. In fact, most notions for PDL resemble those for $\mathcal{L}_\mu$, adjusted to account for the different syntax of PDL. In Subsection 2.5.4 we will clarify the exact relationship between PDL and $\mathcal{L}_\mu$.

2.5.1 Syntax

Recall that we fixed a countably infinite set Prop of proposition letters. For defining the language of PDL we will also fix a countably infinite set of actions Act. In this setting, actions will also be called *atomic programs.*

2.5.1. Definition. The sets of *formulas* and *programs* of PDL are given by the following mutual induction:

$$\varphi ::= \top \mid \bot \mid p \mid \overline{p} \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle\alpha\rangle\varphi \mid [\alpha]\varphi$$
$$\alpha ::= a \mid \alpha;\alpha \mid \alpha \cup \alpha \mid \alpha^* \mid \varphi?$$

where $p \in \mathsf{Prop}$ and $a \in \mathsf{Act}$.

In this section we will simply write formulas for PDL-formulas. We refer to formulas of the form $\langle\alpha\rangle\varphi$ and $[\alpha]\varphi$ as, respectively, *diamond* and *box formulas.* Formulas of the form $\langle\alpha^*\rangle\varphi$ or $[\alpha^*]\varphi$ are called *fixpoint formulas*.

Substitution is defined as for $\mathcal{L}_\mu$-formulas. We can define the negation of PDL-formulas similarly to that of $\mathcal{L}_\mu$-formulas.

2.5.2. Definition. We inductively extend the map $p \mapsto \overline{p}$ to a full-blown *negation operation* on formulas as follows:

$$\begin{array}{lllllllllll} \overline{\top} & := & \bot & & & & \overline{\varphi \wedge \psi} & := & \overline{\varphi} \vee \overline{\psi} & \overline{[\alpha]\varphi} & := & \langle\alpha\rangle\overline{\varphi} \\ \overline{\bot} & := & \top & \overline{\overline{p}} & := & p & \overline{\varphi \vee \psi} & := & \overline{\varphi} \wedge \overline{\psi} & \overline{\langle\alpha\rangle\varphi} & := & [\alpha]\overline{\varphi} \end{array}$$

Note that $\overline{\overline{\varphi}} = \varphi$ for every formula φ.

In the context of PDL, the closure of a formula is commonly referred to as the *Fischer-Ladner closure*, named after its introduction in [FL79]. We will refer to it simply as the *closure* to highlight similarities with the modal μ-calculus.

2.5.3. Definition. Let φ and ψ be formulas. To simplify this definition, we write $(\alpha)\varphi$ to denote $\langle\alpha\rangle\varphi$ or $[\alpha]\varphi$. We write $\varphi \to_C \psi$ if

1. $\varphi = \chi_0 \circ \chi_1$ and $\psi \in \{\chi_0, \chi_1\}$ with $\circ \in \{\wedge, \vee\}$;
2. $\varphi = (a)\psi$;
3. $\varphi = (\alpha;\beta)\chi$ and $\psi = (\alpha)(\beta)\chi$;
4. $\varphi = (\alpha \cup \beta)\chi$ and $\psi \in \{(\alpha)\chi, (\beta)\chi\}$;
5. $\varphi = \langle\tau?\rangle\chi$ and $\psi \in \{\tau, \chi\}$, or $\varphi = [\tau?]\chi$ and $\psi \in \{\overline{\tau}, \chi\}$; or
6. $\varphi = (\alpha^*)\chi$ and $\psi \in \{(a)(\alpha^*)\chi, \chi\}$.

We let $\twoheadrightarrow_C$ be the reflexive and transitive closure of $\to_C$ and write $\varphi \equiv_C \psi$ if $\varphi \twoheadrightarrow_C \psi$ and $\psi \twoheadrightarrow_C \varphi$.

The *closure* $\mathsf{Clos}(\varphi)$ of φ is the least set of formulas containing φ that is closed under the relation $\twoheadrightarrow_C$.

As for $\mathcal{L}_\mu$, we have the following lemma. A proof can be found in [FL79].

2.5.4. Lemma. $\mathsf{Clos}(\varphi)$ *is finite for every formula* φ.

2.5.5. Definition. For a set of formulas Φ we define $\overline{\Phi} := \{\overline{\varphi} \mid \varphi \in \Phi\}$. We define $\mathsf{Clos}(\Phi) := \bigcup_{\varphi\in\Phi} \mathsf{Clos}(\varphi)$ and $\mathsf{Clos}^{\neg}(\Phi) := \mathsf{Clos}(\Phi) \cup \mathsf{Clos}(\overline{\Phi})$.

By Lemma 2.5.4, if Φ is finite, then $\mathsf{Clos}(\Phi)$ and $\mathsf{Clos}^{\neg}(\Phi)$ are also finite.

2.5.6. Definition. A *trace* is a sequence $(\varphi_n)_{0\leq n<\kappa}$ (with $\kappa \leq \omega$) such that $\varphi_n \rightarrow_C \varphi_{n+1}$ for all $i < \kappa$.

The following lemma resembles Lemma 2.3.13 for traces in the modal μ-calculus.

2.5.7. Lemma. *Let $t = (\varphi_n)_{n<\omega}$ be an infinite trace. Then infinitely many φ_n are fixpoint formulas, and either cofinitely many φ_n are diamond formulas, or cofinitely many φ_n are box formulas.*

Proof:
To prove this proposition we first have to introduce some notations. The *length* $|e|$ of an expression e is defined by a mutual induction on formulas and programs. Atomic formulas and programs have length one, and we set $|\varphi\odot\psi| := 1+|\varphi|+|\psi|$ if $\odot \in \{\wedge, \vee\}$; $|\alpha \odot \beta| := 1 + |\alpha| + |\beta|$ if $\odot \in \{\cup, ;\}$; $|\varphi?| := 1 + |\varphi|$; $|\beta^*| := 1 + |\beta|$. The key clause in the definition is that we put $|\langle\alpha\rangle\varphi| := |\alpha| + |\varphi|$.

For a list $\vec{\delta}$ of programs, the formula $\Diamond(\vec{\delta}, \psi)$ is inductively defined as follows: $\Diamond(\varepsilon, \psi) := \psi$, and $\Diamond(\gamma\vec{\delta}, \psi) := \langle\gamma\rangle\Diamond(\vec{\delta}, \psi)$.

Let $\mathsf{Inf}(t)$ denote the set of formulas that occur infinitely often on t, and let $\varphi \in \mathsf{Inf}(t)$ be of minimal length. It is obvious that φ must be a fixpoint formula, since in all other cases the direct derivatives of φ are shorter than φ itself.

We only consider the case where φ is a diamond fixpoint formula, say, $\varphi = \langle\alpha^*\rangle\psi$. (The proof in the case where $\varphi = [\alpha^*]\psi$ is completely analogous.) Let k be such that $\mathsf{Inf}(t) = \{\varphi_n \mid n \geq k\}$, and such that $\varphi_k = \varphi$. The proposition then follows from the claim below.

<u>`Claim 1`</u>: For all $n \geq k$ the formula φ_n is of the form $\varphi_n = \Diamond(\vec{\delta}, \varphi)$, for some list $\vec{\delta} = \delta_1 \cdots \delta_m$ of programs, where each δ_i is shorter than α.

`Proof of Claim 1`: The claim can be proved by a straightforward induction on n. In the base case, where $n = k$, we have $\varphi_n = \varphi = \Diamond(\varepsilon, \varphi)$.

For the induction step we assume as our induction hypothesis that $\varphi_n = \Diamond(\vec{\delta}, \varphi)$ for some program list $\vec{\delta}$, and we make a case distinction. In case $\vec{\delta} = \varepsilon$ we have $\varphi_n = \varphi = \langle\alpha^*\rangle\psi$, so that $\varphi_{n+1} \in \{\psi, \langle\alpha\rangle\langle\alpha^*\rangle\psi\}$. However, by the assumption on k the formula φ_{n+1} cannot be shorter than φ, so that we find $\varphi_{n+1} = \langle\alpha\rangle\langle\alpha^*\rangle\psi = \Diamond(\alpha, \varphi)$.

In case $\vec{\delta} \neq \varepsilon$ we may write $\vec{\delta} = \beta\vec{\gamma}$ and we make a further case distinction as to the nature of β.

Case $\beta = a$ for some $a \in Act$. Here we find $\varphi_{n+1} = \Diamond(\vec{\gamma}, \varphi)$.

Case $\beta = \tau?$. We find that $\varphi_{n+1} \in \{\tau, \Diamond(\vec{\gamma}, \varphi)\}$, but since (by assumption on k) φ_{n+1} cannot be shorter than φ only the second option is possible.

Case $\beta = \beta_0; \beta_1$. We obtain $\varphi_{n+1} = \Diamond(\beta_0\beta_1\vec{\gamma}, \varphi)$.

Case $\beta = \beta_0 \cup \beta_1$. We obtain $\varphi_{n+1} = \Diamond(\beta_0\vec{\gamma}, \varphi)$ or $\varphi_{n+1} = \Diamond(\beta_1\vec{\gamma}, \varphi)$.

Case $\beta = \beta_0^*$. We find that either $\varphi_{n+1} = \Diamond(\vec{\gamma}, \varphi)$ or $\varphi_{n+1} = \Diamond(\beta_0\beta_0^*\vec{\gamma}, \varphi)$.

In all cases it is straightforward to verify that φ_{n+1} has the required shape. ⊣

This finishes the proof of the Claim and, hence, that of the Proposition. □

2.5.2 Semantics

PDL-formulas are interpreted in Kripke models. The meaning of formulas can be given by *relational semantics* as well as by *game semantics*. In this thesis we will work with the game semantics, yet this is not the usual approach in the literature. We therefore also define relational semantics and include a proof of their equivalence.

Game semantics

2.5.8. Definition. Let $\mathbb{S} = (S, R, V)$ be a Kripke model. The *evaluation game* $\mathcal{E}_{\mathsf{PDL}}(\mathbb{S})$ is the following infinite two-player game. Its positions are pairs of the form (φ, s), where φ is a formula and $s \in S$, and its ownership function and admissible moves are given in the Table 2.2. Note that the left projection $(\varphi_n)_{n<\kappa}$ of any (partial) match $(\varphi_n, s_n)_{n<\kappa}$ is a trace. An infinite match is won by $\forall$ if its left projection features infinitely many diamond fixpoint formulas (or, equivalently, cofinitely many diamond formulas) and by $\exists$ else.

2.5.9. Remark. The evaluation game $\mathcal{E}_{\mathsf{PDL}}(\mathbb{S})$ can be formulated as a parity game, simply consider the priority map Ω mapping positions of the form $([\alpha]\varphi, s)$ to 2 and all other positions to 1. As such, the game is positionally determined.

2.5.10. Definition. Let $\mathbb{S}, s$ be a pointed model, let g be a strategy for $\exists$ in $\mathcal{E}_{\mathsf{PDL}}(\mathbb{S})$ and let φ be a formula. We write $\mathbb{S}, s \Vdash_g \varphi$ if g is winning for $\exists$ at (φ, s), and $\mathbb{S}, s \Vdash \varphi$ if $\mathbb{S}, s \Vdash_g \varphi$ for some strategy g for $\exists$. We say that a formula φ is *satisfiable* if there exists $\mathbb{S}, s$ such that $\mathbb{S}, s \Vdash \varphi$ and *unsatisfiable* otherwise.

We define the relation $\vDash$ for PDL-formulas as for $\mathcal{L}_\mu$-formulas. That is, we write $\varphi \vDash \psi$ if $\mathbb{S}, s \Vdash \varphi$ implies $\mathbb{S}, s \Vdash \psi$ for every pointed model $\mathbb{S}, s$. We say that two formulas φ and ψ are *semantically equivalent*, written as $\varphi \equiv \psi$, if $\varphi \vDash \psi$ and $\psi \vDash \varphi$.

Position		Owner	Admissible moves
$(\bot, s)$		$\exists$	$\varnothing$
$(\top, s)$		$\forall$	$\varnothing$
(p, s)	with $s \in V(p)$	$\forall$	$\varnothing$
(p, s)	with $s \notin V(p)$	$\exists$	$\varnothing$
$(\overline{p}, s)$	with $s \in V(p)$	$\exists$	$\varnothing$
$(\overline{p}, s)$	with $s \notin V(p)$	$\forall$	$\varnothing$
$(\varphi \vee \psi, s)$		$\exists$	$\{(\varphi, s), (\psi, s)\}$
$(\varphi \wedge \psi, s)$		$\forall$	$\{(\varphi, s), (\psi, s)\}$
$(\langle a \rangle \varphi, s)$		$\exists$	$\{(\varphi, t) \mid (s, t) \in R_a\}$
$([a]\varphi, s)$		$\forall$	$\{(\varphi, t) \mid (s, t) \in R_a\}$
$(\langle \alpha; \beta \rangle \varphi, s)$		-	$\{(\langle \alpha \rangle \langle \beta \rangle \varphi, s)\}$
$([\alpha; \beta]\varphi, s)$		-	$\{([\alpha][\beta]\varphi, s)\}$
$(\langle \alpha \cup \beta \rangle \varphi, s)$		$\exists$	$\{(\langle \alpha \rangle \varphi, s), (\langle \beta \rangle \varphi, s)\}$
$([\alpha \cup \beta]\varphi, s)$		$\forall$	$\{([\alpha]\varphi, s), ([\beta]\varphi, s)\}$
$(\langle \alpha^* \rangle \varphi, s)$		$\exists$	$\{(\langle \alpha \rangle \langle \alpha^* \rangle \varphi, s), (\varphi, s)\}$
$([\alpha^*]\varphi, s)$		$\forall$	$\{([\alpha][\alpha^*]\varphi, s), (\varphi, s)\}$
$(\langle \psi? \rangle \varphi, s)$		$\forall$	$\{(\psi, s), (\varphi, s)\}$
$([\psi?]\varphi, s)$		$\exists$	$\{(\overline{\psi}, s), (\varphi, s)\}$

Table 2.2: The evaluation game $\mathcal{E}_{\mathsf{PDL}}(\mathbb{S})$

Relational semantics In this section we present the standard, relational semantics of PDL. Given a Kripke model $\mathbb{S}$, formulas and programs are inductively interpreted as, respectively, subsets of and binary relations over the carrier of $\mathbb{S}$. Recall that any binary relation R on a set S induces two operations on $\mathcal{P}(S)$:

$$\begin{array}{lll} \langle R \rangle (U) & := & \{s \in S \mid R[s] \cap U \neq \varnothing\} \\ [R](U) & := & \{s \in S \mid R[s] \subseteq U\}. \end{array}$$

2.5.11. Definition. Given a Kripke model $\mathbb{S} = (S, R, V)$, by a mutual recursion on formulas and programs we define the meaning $[\![\varphi]\!]^{\mathbb{S}} \subseteq S$ of a formula φ in $\mathbb{S}$:

$$\begin{array}{lllllll} [\![\top]\!]^{\mathbb{S}} & := & S & \quad & [\![\bot]\!]^{\mathbb{S}} & := & \varnothing \\ [\![p]\!]^{\mathbb{S}} & := & V(p) & & [\![\overline{p}]\!]^{\mathbb{S}} & := & S \setminus V(p) \\ [\![\varphi \vee \psi]\!]^{\mathbb{S}} & := & [\![\varphi]\!]^{\mathbb{S}} \cup [\![\psi]\!]^{\mathbb{S}} & & [\![\varphi \wedge \psi]\!]^{\mathbb{S}} & := & [\![\varphi]\!]^{\mathbb{S}} \cap [\![\psi]\!]^{\mathbb{S}} \\ [\![\langle \alpha \rangle \varphi]\!]^{\mathbb{S}} & := & \langle R_\alpha \rangle ([\![\varphi]\!]^{\mathbb{S}}) & & [\![[\alpha]\varphi]\!]^{\mathbb{S}} & := & [R_\alpha]([\![\varphi]\!]^{\mathbb{S}}) \end{array}$$

and we extend the maps $R = \{R_a \mid a \in \mathsf{Act}\}$ to provide accessibility relations $R_\alpha \subseteq S \times S$ to arbitrary programs α:

$$\begin{array}{lll} R_{\alpha;\beta} & := & R_\alpha ; R_\beta \\ R_{\alpha \cup \beta} & := & R_\alpha \cup R_\beta \\ R_{\alpha^*} & := & R_\alpha^* \\ R_{\varphi?} & := & \{(s, s) \mid s \in [\![\varphi]\!]^{\mathbb{S}}\} \end{array}$$

Here we define $R_\alpha; R_\beta$ as $\{(s,u) \mid (s,t) \in R_\alpha \text{ and } (t,u) \in R_\beta \text{ for some } t\}$, and define R^*_α as $\bigcup_{n\in\mathbb{N}} R^n_\alpha$, where R^n_α is defined inductively by $R^0_\alpha := \text{id}$ and $R^{n+1}_\alpha := R^n_\alpha; R_\alpha$.

Adequacy We show the equivalence of the game semantics and the relational semantics. In the evaluation game $\mathcal{E}_{\mathsf{PDL}}(\mathbb{S})$ we let $Win_\exists(\mathbb{S})$ denote the winning positions for $\exists$, and we write $(\!|\varphi|\!)^\mathbb{S} := \{s \in S \mid (\varphi, s) \in Win_\exists(\mathbb{S})\}$.

2.5.12. Theorem (Adequacy). *For every model* $\mathbb{S}$ *and formula* φ *we have*

$$[\![\varphi]\!]^\mathbb{S} = (\!|\varphi|\!)^\mathbb{S}. \tag{2.1}$$

Proof:
By a mutual induction on formulas and programs we will show that every formula φ satisfies (2.1), while for every program α we have

$$(\!|\langle\alpha\rangle\psi|\!)^\mathbb{S} = \langle R^\mathbb{S}_\alpha\rangle((\!|\psi|\!)^\mathbb{S}) \text{ and } (\!|[\alpha]\psi|\!)^\mathbb{S} = [R^\mathbb{S}_\alpha]((\!|\psi|\!)^\mathbb{S}), \text{ for every } \mathbb{S} \text{ and } \psi. \tag{2.2}$$

The proof of (2.1) is routine, so we confine ourselves to a few examples. The case where φ is atomic is immediate by the definitions. In the induction step, where $\varphi = \varphi_0 \vee \varphi_1$ is a disjunction, we reason as follows: $s \in [\![\varphi]\!]^\mathbb{S} = [\![\varphi_0]\!]^\mathbb{S} \cup [\![\varphi_1]\!]^\mathbb{S}$ iff $s \in [\![\varphi_0]\!]^\mathbb{S}$ or $s \in [\![\varphi_1]\!]^\mathbb{S}$ iff (IH) $s \in (\!|\varphi_0|\!)^\mathbb{S}$ or $s \in (\!|\varphi_1|\!)^\mathbb{S}$ iff $s \in (\!|\varphi_0 \vee \varphi_1|\!)^\mathbb{S}$, where the last equivalence is based on an obvious game-theoretical observation. For the case where $\varphi = \langle\alpha\rangle\varphi'$ we reason as follows. By definition we have $[\![\langle\alpha\rangle\varphi']\!]^\mathbb{S} = \langle R^\mathbb{S}_\alpha\rangle([\![\varphi']\!]^\mathbb{S})$, and by applications of the induction hypothesis (for φ' and α, respectively), we find that $\langle R^\mathbb{S}_\alpha\rangle([\![\varphi']\!]^\mathbb{S}) = \langle R^\mathbb{S}_\alpha\rangle((\!|\varphi'|\!)^\mathbb{S}) = (\!|\langle\alpha\rangle\varphi'|\!)^\mathbb{S})$. Clearly, then we have $[\![\langle\alpha\rangle\varphi']\!]^\mathbb{S} = (\!|\langle\alpha\rangle\varphi'|\!)^\mathbb{S})$.

For the proof of (2.2) we only cover the statement on diamond formulas, and we leave the cases where α is atomic or of the form $\beta \cup \gamma$ as exercises for the reader.

In the case where $\alpha = \tau?$ is a test, it is easy to see that $(\!|\langle\tau?\rangle\psi|\!)^\mathbb{S} = (\!|\tau|\!)^\mathbb{S} \cap (\!|\psi|\!)^\mathbb{S}$. For the right hand side we have $s \in \langle R^\mathbb{S}_{\tau?}\rangle((\!|\psi|\!)^\mathbb{S})$ iff there is a $t \in R_{\tau?}[s] \cap (\!|\psi|\!)^\mathbb{S}$ iff $s \in [\![\tau]\!]^\mathbb{S} \cap (\!|\psi|\!)^\mathbb{S}$ iff (by induction hypothesis on τ) $s \in (\!|\tau|\!)^\mathbb{S} \cap (\!|\psi|\!)^\mathbb{S}$, as required.

In the case where α is of the form $\alpha = \beta;\gamma$ we apply the induction hypothesis to β and γ, respectively, and find that $(\!|\langle\beta\rangle\langle\gamma\rangle\psi|\!)^\mathbb{S} = \langle R^\mathbb{S}_\beta\rangle((\!|\langle\gamma\rangle\psi|\!)^\mathbb{S}) = \langle R^\mathbb{S}_\beta\rangle\big(\langle R^\mathbb{S}_\gamma\rangle((\!|\psi|\!)^\mathbb{S})\big)$. But then we are done, since we obviously have that $\langle R^\mathbb{S}_{\beta;\gamma}\rangle$ is the composition of $\langle R^\mathbb{S}_\beta\rangle$ and $\langle R^\mathbb{S}_\gamma\rangle$.

The key case in the proof is where $\alpha = \beta^*$ is an iteration. For the inclusion $\supseteq$ we observe that $\langle R^\mathbb{S}_{\beta^*}\rangle(A) = \bigcup_{n\in\mathbb{N}}\langle R^\mathbb{S}_\beta\rangle^n(A)$, for all $A \subseteq S$, so that it suffices to show that $\langle R^\mathbb{S}_\beta\rangle^n((\!|\psi|\!)^\mathbb{S}) \subseteq (\!|\langle\beta^*\rangle\psi|\!)^\mathbb{S}$ for all $n \in \mathbb{N}$. This inclusion we can establish by a straightforward inner induction on n. In the base step we have $\langle R^\mathbb{S}_\beta\rangle^0((\!|\psi|\!)^\mathbb{S}) = (\!|\psi|\!)^\mathbb{S}$, and it is obvious that $(\!|\psi|\!)^\mathbb{S} \subseteq (\!|\langle\beta^*\rangle\psi|\!)^\mathbb{S})$. In the inductive step we find $\langle R^\mathbb{S}_\beta\rangle^{n+1}((\!|\psi|\!)^\mathbb{S}) = \langle R^\mathbb{S}_\beta\rangle\langle R^\mathbb{S}_\beta\rangle^n((\!|\psi|\!)^\mathbb{S})$. Respective applications of the inner induction hypothesis (on n) and the outer induction hypothesis (on β) show that $\langle R^{\mathbb{S}}_{\beta}\rangle\langle R^{\mathbb{S}}_{\beta}\rangle^{n}((\!|\psi|\!)^{\mathbb{S}}) \subseteq \langle R^{\mathbb{S}}_{\beta}\rangle((\!|\langle\beta^{*}\rangle\psi|\!)^{\mathbb{S}})) \subseteq (\!|\langle\beta\rangle\langle\beta^{*}\rangle\psi|\!)^{\mathbb{S}}$. Finally, it is obvious that $(\!|\langle\beta\rangle\langle\beta^{*}\rangle\psi|\!)^{\mathbb{S}} \subseteq (\!|\langle\beta^{*}\rangle\psi|\!)^{\mathbb{S}})$, so that we are done.

For the opposite inclusion $\subseteq$ of (2.2) in the case where $\alpha = \beta^{*}$ we have to do more work. To reduce notational clutter we let A denote the right hand side of the equation, that is, $A := \langle R^{\mathbb{S}}_{\beta^{*}}\rangle((\!|\psi|\!)^{\mathbb{S}})$; it is an easy consequence of the definition of $R^{\mathbb{S}}_{\beta^{*}}$ that

$$A = (\!|\psi|\!)^{\mathbb{S}} \cup \langle R^{\mathbb{S}}_{\beta}\rangle A. \tag{2.3}$$

We will show that $(\!|\langle\beta^{*}\rangle\psi|\!)^{\mathbb{S}} \subseteq A$ by providing $\forall$ with a winning strategy in the game $\mathcal{E}_{\mathsf{PDL}}(\mathbb{S})@(\langle\beta^{*}\rangle\psi, s)$ for an arbitrary state $s \notin A$. In this proof we abbreviate $\mathcal{E} := \mathcal{E}_{\mathsf{PDL}}$.

In order to define this strategy we use an auxiliary structure. Take a fresh proposition letter p and consider the model $\mathbb{S}_A := \mathbb{S}[p \mapsto A]$; that is, we modify the valuation of $\mathbb{S}$ so that in $\mathbb{S}_A$ the proposition letter p is interpreted as the set A. Fix some winning positional strategy g for $\forall$; that is, g is winning for every position in $Win_{\forall}(\mathcal{E}(\mathbb{S}_A))$. Observe that we have

$$\langle R^{\mathbb{S}}_{\beta}\rangle(A) = \langle R^{\mathbb{S}}_{\beta}\rangle([\![p]\!]^{\mathbb{S}_A}) = \langle R^{\mathbb{S}_A}_{\beta}\rangle((\!|p|\!)^{\mathbb{S}_A}) = (\!|\langle\beta\rangle p|\!)^{\mathbb{S}_A}, \tag{2.4}$$

where we use the fact that p does not occur in β in the second equality, and the induction hypothesis on β in the last one. Then for any state t in S it follows from $t \notin A$, (2.3) and (2.4) that $t \notin (\!|\langle\beta\rangle p|\!)^{\mathbb{S}_A}$, which by determinacy of $\mathcal{E}(\mathbb{S}_A)$ means that g is winning for $\forall$ in $\mathcal{E}(\mathbb{S}_A)@(\langle\beta\rangle p, t)$:

$$t \notin A \text{ implies } g \text{ is winning for } \forall \text{ in } \mathcal{E}(\mathbb{S}_A)@(\langle\beta\rangle p, t). \tag{2.5}$$

Furthermore, observe that since $\mathbb{S}_A$ is an expansion of $\mathbb{S}$, we may see g as a strategy for $\mathcal{E}(\mathbb{S})$ as well.

We can now define $\forall$'s strategy h in $\mathcal{E}(\mathbb{S})$ as follows:

- at a position of the form $\langle\tau?\rangle\chi$ play as follows:
  - pick (τ, u) if $(\tau, u) \in Win_{\forall}(\mathcal{E}(\mathbb{S}))$ and continue with the strategy g;
  - otherwise, pick χ.
- at any other position (χ, u) play g if $(\chi, u) \in Win_{\forall}(\mathcal{E}(\mathbb{S}))$;
- otherwise play randomly.

In order to show that h is winning for $\forall$ in $\mathcal{E}(\mathbb{S})@(\langle\beta^{*}\rangle\psi, s)$, we need the following Claim.

Claim Let $\pi = (\varphi_i, s_i)_{i \leq n}$ be some partial h-guided match of $\mathcal{E}(\mathbb{S})@(\langle\beta^{*}\rangle\psi, s)$ where $s_0 = s \notin A$. If at any position $(\langle\beta^{*}\rangle\psi, t)$ in π Eloise picks $(\langle\beta\rangle\langle\beta^{*}\rangle\psi, t)$, and $\varphi_n = \langle\beta^{*}\rangle\psi$, then $s_n \notin A$.

Proof of Claim First of all, note that at the start $(\varphi_0, s_0) = (\langle\beta^*\rangle\psi, s)$ of the match $\exists$ picks $(\varphi_1, s_1) = (\langle\beta\rangle\langle\beta^*\rangle\psi, s)$ by assumption, and that the position $(\langle\beta\rangle p, s) \in Win_\exists(\mathcal{E}(\mathbb{S}_A))$. Now let m with $1 \leq m \leq n$ be minimal such that $\varphi_m = \langle\beta^*\rangle\psi$. We first prove that $s_m \notin A$, inductively this implies that $s_n \notin A$ as intended. In order to do so, we claim that the match $\pi' = (\varphi_i, s_i)_{1\leq i\leq m}$ is of the form $\rho'[\langle\beta^*\rangle\psi/p]$ for some g-guided $\mathcal{E}(\mathbb{S}_A)$ match ρ'.

To see this, we show that for every i with $1 \leq i \leq m$ there are program lists λ_i such that

(†) $\varphi_i = \Diamond(\lambda_i, \langle\beta^*\rangle\psi)$ for all i, while

(‡) the sequence $(\Diamond(\lambda_i, p), s_i)_{1\leq i\leq m}$ is a g-guided partial $\mathcal{E}(\mathbb{S}_A)$-match.

This statement is obvious for $i = 1$, as $\varphi_1 = \langle\beta\rangle\langle\beta^*\rangle\psi$ by assumption, meaning that $\lambda_1 := \beta$. In the induction step we assume that we have defined the program lists $\lambda_1, \ldots, \lambda_i$ satisfying (†) and (‡) for some $1 \leq i < m$. Since m is minimal with $\varphi_m = \langle\beta^*\rangle\psi$, the program list λ_i is nonempty and so it must be of the form $\gamma\mu$ for some program γ and program list μ.

The only case of interest is where γ is a test, say, $\gamma = \tau?$. The position (φ_i, s_i) in π' is $(\varphi_i, s_i) = ((\Diamond(\tau?\mu, \langle\beta^*\rangle\psi), s_i) = (\langle\tau?\rangle\Diamond(\mu, \langle\beta^*\rangle\psi), s_i)$. We claim that $\varphi_{i+1} = \Diamond(\mu, \langle\beta^*\rangle\psi)$. To see this, note that we cannot have $\varphi_{i+1} = \tau$, since all subsequent formulas in π would have to belong to the closure of τ, and this clearly does not hold for the formula $\varphi_m = \langle\beta^*\rangle\psi$. But since $\forall$ played according to his strategy h, this means that (τ, s_i) is a winning position for $\exists$, in both $\mathcal{E}(\mathbb{S})$ and $\mathcal{E}(\mathbb{S}_A)$. Hence, by our assumption that g is a winning strategy for $\forall$, in $\mathcal{E}(\mathbb{S}_A)$, at position $(\Diamond(\tau?\mu, p), s_i)$, it will tell $\forall$ to move to position $(\Diamond(\mu, p), s_i)$. In other words, the new position in $\mathcal{E}(\mathbb{S})$ is $(\varphi_{i+1}, s_{i+1}) = (\Diamond(\mu, \langle\beta^*\rangle\psi), s_{i+1})$ while the new position in $\mathcal{E}(\mathbb{S}_A)$ is $(\Diamond(\mu, p), s_{i+1})$. Obviously then, if we define $\lambda_{i+1} := \mu$ the conditions (†) and (‡) hold for $i + 1$, as required.

For the case $i = m$ the statements (†) and (‡) imply that $s_m \notin A$. This implies $s_n \notin A$ and we may consider the claim to be proved.

Now consider an arbitrary h-guided full match $\pi = (\varphi_i, s_i)_{i\leq\kappa}$ of $\mathcal{E}(\mathbb{S})@(\langle\beta^*\rangle\psi, s)$, where $s \notin A$. To see why π must be won by $\forall$, we distinguish cases.

Let n be maximal such that the assumptions of the Claim are satisfied, meaning that $\varphi_n = \langle\beta^*\rangle\psi$ and Eloise picks $(\langle\beta\rangle\langle\beta^*\rangle\psi, t)$ at any position $(\langle\beta^*\rangle\psi, t)$ in $(\varphi_i, s_i)_{i\leq n}$.

If n is undefined, then there are infinitely many positions of the form $(\langle\beta^*\rangle\psi, t)$ in π. By definition of the winning conditions, it then constitutes a win for $\forall$.

Otherwise write $t := s_n$. It follows by the Claim that $t \notin A$, and, hence, by (2.3) that $t \notin (\!|\psi|\!)^{\mathbb{S}}$. By the determinacy of the evaluation game this means that $(\psi, t) \in Win_\forall(\mathcal{E}(\mathbb{S}))$. Hence if $\exists$ picks position $(\varphi_{n+1}, s_{n+1}) = (\psi, t)$ then the remainder of π will simply be guided by $\forall$'s winning strategy g, resulting in a win for $\forall$. Now assume that $\exists$ picks position $(\varphi_{n+1}, s_{n+1}) = (\langle\beta\rangle\langle\beta^*\rangle\psi, t)$. In the case that $\forall$ picks a test formula at some position after stage $n+1$, the remaining tail of π is guided by his winning strategy g and so he wins π. But if $\forall$ never picks a test formula, it means that for all $i > n$ the formula φ_i is the form $(\Diamond(\lambda, \langle\beta^*\rangle\psi), t)$ for some nonempty list of programs λ. Since π is a full match this can only be the case if π is infinite¿ But then π has a tail of diamond formulas and thus is won by $\forall$.

In other words, we have proved that h is winning for $\forall$ in $\mathcal{E}(\mathbb{S})@(\langle\beta^*\rangle\psi, s)$ indeed, and this suffices to prove the inclusion $\subseteq$ of (2.2) in the case where $\alpha = \beta^*$. ☐

2.5.3 Converse PDL

Recall that we fixed a countably infinite set of actions Act.

2.5.13. Definition. The language of CPDL is precisely the same as that of PDL, with the additional assumption that there is an involution operation $\breve{\cdot} : \mathsf{Act} \to \mathsf{Act}$ such that for every $a \in \mathsf{Act}$ it holds that $\breve{a} \neq a$ and $\breve{\breve{a}} = a$.

In our set-up, the atomic actions come in pairs $a, \breve{a}$. This has some technical advantages over the approach where the converse operation $\breve{}$ is an explicit syntactic symbol. This resembles our choice of working with the negation normal form of formulas compared to allowing an explicit syntactic negation symbol.

However, as for negation, we may extend the converse operator to arbitrary programs by putting

$$(\alpha;\beta)^{\smile} := \breve{\beta};\breve{\alpha} \qquad (\alpha\cup\beta)^{\smile} := \breve{\alpha}\cup\breve{\beta} \qquad (\alpha^*)^{\smile} := \breve{\alpha}^* \qquad (\tau?)^{\smile} := \tau?$$

Thus we may indeed think of CPDL as extending PDL with a converse operation on programs.

2.5.14. Definition. Let φ be a CPDL-formula. We define the *vocabulary of* φ, written $\mathsf{Voc}(\varphi)$, to be the set of proposition letters and actions occurring in φ with the proviso that we include both a and $\breve{a}$ in the vocabulary of any expression in which a or $\breve{a}$ occurs.

All notions defined for PDL are defined analogously for CPDL. As for $\mathcal{L}^2_\mu$-formulas, we interpret CPDL-formulas only on two-way Kripke models. Recall that a two-way Kripke model is a Kripke model that satisfies $R_{\breve{a}} = \{(s,t) \mid (t,s) \in R_a\}$ for every $a \in \mathsf{Act}$.

2.5.15. Definition. A CPDL-formula φ is *satisfiable*, if there is a two-way Kripke model that satisfies φ and *unsatisfiable* otherwise.

2.5.16. Example. Consider the CPDL-formulas $\langle a^*\rangle p$ and $\langle a^*; p?; \breve{a}^*\rangle q$. We have

$$\mathbb{S}, s \Vdash \langle a^*\rangle p$$

iff there is an a-path from s to some node t where p holds; and we have

$$\mathbb{S}, s \Vdash \langle a^*; p?; \breve{a}^*\rangle q$$

iff there is an a-path from s to some node t where p holds, and such that there is an $\breve{a}$-path from t to some node r where q holds.

If $\mathbb{S}$ is a two-way Kripke model, than there is an a-path from s to some node t iff there is an $\breve{a}$-path from t to s. Thus, for every two-way Kripke model $\mathbb{S}$ and $s \in \mathbb{S}$ it holds that

$$\mathbb{S}, s \Vdash \langle a^*\rangle p \to (q \to \langle a^*; p?; \breve{a}^*\rangle q).$$

We therefore have $\langle a^*\rangle p \vDash q \to \langle a^*; p?; \breve{a}^*\rangle q$.

2.5.4 PDL as a fragment of the modal μ-calculus

Comparing the two sections on the modal μ-calculus and PDL, one notices a lot of similarities. This is not a coincidence; in fact, PDL corresponds to a fragment of the modal μ-calculus. We will define this fragment – the completely additive μ-calculus $\mathcal{L}_\mu^{ca}$ – and motivate the translations between PDL and $\mathcal{L}_\mu^{ca}$ presented in [CV14]. A semantic characterization of the completely additive μ-calculus can be found in [FV18].

2.5.17. Definition. We say that a variable x in ψ is in the scope of a fixpoint operator η, if a free occurrence of x in ψ is in the scope of a fixpoint operator η. Analogously for modalities. We say that a variable x in ψ is in the scope of an *essential* conjunction, if there is a conjunction $\varphi_0 \wedge \varphi_1 \trianglelefteq \psi$ such that a free occurrence of x in ψ is in φ_0 and another free occurrence of x in ψ is in φ_1. Analogously for an essential disjunction.

The *completely additive μ-calculus* $\mathcal{L}_\mu^{ca}$ consists of all formulas $\varphi \in \mathcal{L}_\mu$ where

(i) for any subformula $\mu x.\psi$ of φ, the variable x in ψ is not in the scope of a $\Box$-modality, an essential conjunction or a ν-operator; and

(ii) for any subformula $\nu x.\psi$ of φ, the variable x in ψ is not in the scope of a $\Diamond$-modality, an essential disjunction or a μ-operator.

An inductive definition of $\mathcal{L}_\mu^{ca}$ can be found in [CV14]. It should be clear from the definitions that every formula in $\mathcal{L}_\mu^{ca}$ is alternation-free. That is, $\mathcal{L}_\mu^{ca}$ is also a fragment of $\mathcal{L}_\mu^{af}$.

2.5.18. Remark. An alternative characterization of the completely additive μ-calculus could be given by referring to traces instead of subformulas: A formula $\psi \in \mathcal{L}_\mu$ is in $\mathcal{L}_\mu^{ca}$, if for every μ-formula $\mu x.\varphi$ in $\mathsf{Clos}(\psi)$ there is no box-formula $\Box\chi$ such that $\mu x.\varphi \equiv_C \Box\chi$, no conjunction $\chi_0 \wedge \chi_1$ such that $\mu x.\varphi \equiv_C \chi_0 \equiv_C \chi_1$, and no ν-formula $\nu y.\chi$ such that $\mu x.\varphi \equiv_C \nu y.\chi$; and where the dual condition holds for every ν-formula $\nu x.\varphi \in \mathsf{Clos}(\psi)$.

The following lemma follows easily from the definition.

2.5.19. Lemma. *Let $\varphi \in \mathcal{L}_\mu^{ca}$. Then*

1. *every subformula of φ is in $\mathcal{L}_\mu^{ca}$ and*
2. *every formula in $\mathsf{Clos}(\varphi)$ is in $\mathcal{L}_\mu^{ca}$.*

2.5.20. Theorem. PDL *and $\mathcal{L}_\mu^{ca}$ are equivalent. That is, for every* PDL*-formula φ there is a $\mathcal{L}_\mu^{ca}$-formula ψ such that $\varphi \equiv \psi$, and vice versa.*

A proof of this theorem can be found in [CV14]. We only give a short motivation of the translations.

In one direction, program constructors are translated into the language of the μ-calculus by an inductively defined function f from PDL to $\mathcal{L}_\mu^{ca}$. The crucial step is the translation of a fixpoint formula of the form $\langle\alpha^*\rangle\varphi$, where f is defined as follows:

$$f(\langle\alpha^*\rangle\varphi) := \mu x.f(\varphi) \vee f(\langle\alpha\rangle x).$$

Importantly, the resulting fixpoint formula is in $\mathcal{L}_\mu^{ca}$.

For the converse direction, a function g from $\mathcal{L}_\mu^{ca}$ to PDL is defined inductively. In the crucial inductive step, it is first shown that every fixpoint formula in $\mathcal{L}_\mu^{ca}$ of the form $\mu x.\varphi$ is equivalent to one of the form

$$\mu x.\psi \vee \langle\alpha_x\rangle x,$$

where x does not occur in ψ and the program α_x is obtained by induction hypothesis. It then follows that

$$\mu x.\varphi \equiv \langle\alpha_x^*\rangle\psi.$$

The syntaxes of CPDL and $\mathcal{L}_\mu^2$ correspond to the syntaxes of PDL and $\mathcal{L}_\mu$, respectively, while the assumption on the set of actions in CPDL and $\mathcal{L}_\mu^2$ are the same. Therefore, the above argumentation transfers to the logics including converse modalities. That is, CPDL corresponds to a fragment of $\mathcal{L}_\mu^2$, called the *completely additive two-way μ-calculus $\mathcal{L}_\mu^{2ca}$*.

2.6 Non-wellfounded proofs

Traditionally, proofs consist of *finite* labeled trees. However, for logics modelling inductive or fixpoint behavior, finding cut-free finitary proof systems is difficult, and so-called *non-wellfounded* proof systems turned out to be preferable. In such systems, proofs are finitely branching but may contain infinite branches or cycles within the proof tree. To ensure soundness, non-wellfounded proofs must satisfy a *global soundness condition*, which enforces a notion of "progress" along infinite paths. The precise formulation of this condition determines the key subtleties and characteristics of the proof system.

In the context of the modal μ-calculus, non-wellfounded proof systems were first studied by Niwinski and Walukiewicz [Wal93; NW96]. Since then, such systems have been explored in a variety of settings [San02; Bro06; BDS16; Sim17; KPP21]. In this section, we will define non-wellfounded proof systems in an abstract way.

Rules We fix a set $\mathcal{S}$ and call elements of $\mathcal{S}$ *sequents*. In this thesis, sequents will consist of sets or multisets of (annotated) formulas depending on the specific system.

2.6.1. Definition. Let $\Gamma, \Gamma_1, ..., \Gamma_n$ be sequents. A *finitary rule* is an expression of the following form, where R is the name of the rule:

$$\frac{\Gamma_1 \quad \cdots \quad \Gamma_n}{\Gamma_0}\ \mathsf{R}$$

We call a rule R, where $n = 0$, an *axiom*. We call the sequent Γ_0 the *conclusion* and $\Gamma_1, ..., \Gamma_n$ the *premises* of R.

A *discharge rule* is of the following form, where D is the name of the rule:

$$\begin{array}{c} [\Gamma_2]^{\dagger} \\ \vdots \\ \dfrac{\Gamma_1}{\Gamma_0}\ \mathsf{D}_{\dagger} \end{array}$$

We call Γ_0 the *conclusion*, Γ_1 the *premise*, and Γ_2 a *discharged assumption* of the rule $\mathsf{D}_{\dagger}$. Each discharge rule is marked with a unique *discharge token* taken from a fixed infinite set $\mathsf{Tokens} = \{\dagger, \ddagger, ‡, ...\}$.

Rules are either finitary rules or discharge rules.

Sets of rules will usually be specified by *rule schemata*. For instance, the rule schema

$$\vee : \frac{\varphi, \psi, \Gamma}{\varphi \vee \psi, \Gamma}$$

defines all disjunction rules, where φ and ψ are formulas and Γ is a sequent. Whenever a rule schema is defined, we write its name on the left; when referring to a specific rule, we place the name on the right. We will often abuse of language and refer to rule schemata simply as rules when it is clear from the context.

Derivations

2.6.2. Definition. A *derivation system* $\mathcal{D}$ is a set of rules.

A $\mathcal{D}$*-derivation* $\pi = (T, \lessdot, \mathsf{S}, \mathsf{R})$ is a quadruple such that $(T, \lessdot)$ is a, possibly infinite, tree with nodes T and parent relation $\lessdot \subseteq T \times T$; S is a function that maps every node $u \in T$ to a sequent S_u; R is a function that maps every node $u \in T$ to its *label* R_u, which is either (i) the name of a rule in $\mathcal{D}$, (ii) a discharge token or (iii) a special value o (signalling an open assumption). To qualify as a derivation, such a quadruple is required to satisfy the following conditions:

1. If a node is labeled with the name of a rule, then it has as many children as the rule has premises, and the sequents at the node and its children match the specification of the rule.

2. If a node is labeled with a discharge token or with o then it is a leaf.

3. For every leaf l that is labeled with a discharge token $\dagger \in \mathsf{Tokens}$ there is an ancestor $c(l)$ of l that is labeled with $\mathsf{D}_\dagger$ and such that the sequents at l, $c(l)$ and its child match the specification of D. In this case we call l a *repeat leaf* and $c(l)$ the *companion* of l.

A $\mathcal{D}$-derivation of a sequent Γ is a $\mathcal{D}$-derivation where the root is labeled with Γ.

Note that a companion node v labeled with $\mathsf{D}_\dagger$ may have multiple repeat leaves. That is, there might be multiple leaves l labeled with $\dagger$ such that $c(l) = v$.

The name *repeat leaf* is a bit premature at this point. However, in specific proof systems leaves and companions are labeled with (almost) the same sequents, motivating this notion.

2.6.3. Definition. Let $\pi = (T, \lessdot, \mathsf{S}, \mathsf{R})$ be a derivation. We will be working with the following two graphs associated to π.

(i) The usual *proof tree* $\mathcal{T}_\pi := (T, \lessdot)$.

(ii) The *proof tree with back edges* $\mathcal{T}_\pi^C := (T, \lhd)$, where $\lhd$ is the parent relation plus back-edges for each repeat leaf, that is, $\lhd = \lessdot \cup \{(l, c(l)) \mid l \text{ is a repeat leaf}\}$.

For a repeat leaf l, we call the path β_l in $\mathcal{T}_\pi$ from $c(l)$ to l the *repeat path*. A *path* in a $\mathcal{D}$-derivation $\pi = (T, \lessdot, \mathsf{S}, \mathsf{R})$ is a path in $\mathcal{T}_\pi^C$. A *branch* of $\mathcal{D}$ is a path in $\mathcal{D}$ starting at the root. A $\mathcal{D}$*-path* is a path in some $\mathcal{D}$-derivation π.

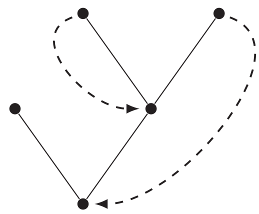


Figure 2.2: A depiction of a proof π by drawing its proof tree with back edges $\mathcal{T}_\pi^C$. We draw back-edges $(l, c(l))$ for repeat leaves l with dashed arrows.

2.6.4. Definition. A *strongly connected subgraph* of a derivation π is a non-empty set A of nodes in π, such that for every $u, v \in A$ there is a $\lhd$-path in A from u to v. A strongly connected subgraph is called *trivial* if it consists of exactly one node and is called *proper* otherwise. A *cluster* in π is a maximal strongly connected subgraph in π.

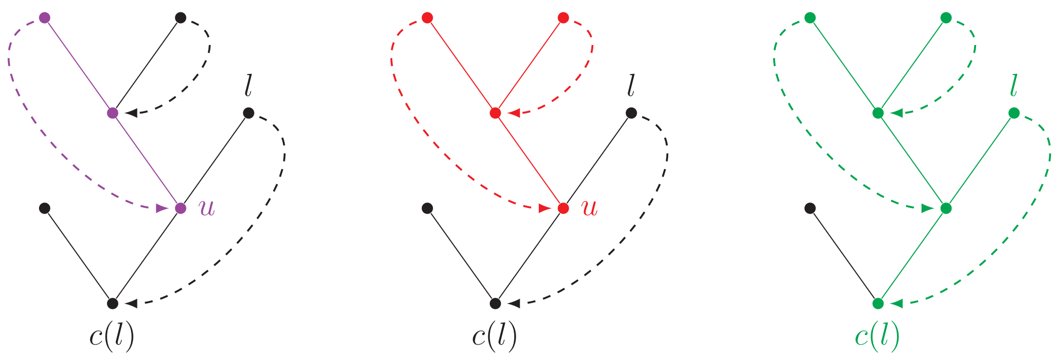
Every proper strongly connected subgraph A in a derivation π has a lowest node $u \in A$ such that all other nodes in A are descendants of u in $\mathcal{T}_\pi$. This node u is always a companion node and we call u the *root* of A.

2.6.5. Definition. Let u be a companion node in a derivation π. The *strongly connected subtree* $\mathrm{scst}(u)$ of u in π is the maximal strongly connected subgraph A of π such that u is the root of A.

We call a leaf l in π *outermost* if $c(l)$ is the root of a cluster C in π.

If l is an outermost leaf in π, then $\mathrm{scst}(c(l))$ is a proper cluster C with root $c(l)$.

2.6.6. Example. Consider the following depiction of a derivation π. This derivation has 6 proper strongly connected subgraphs and 1 proper cluster. We indicated a companion node u and an outermost leaf l with its companion node $c(l)$. In the left figure, we marked a (non-maximal) strongly connected subgraph purple. In the middle figure, the strongly connected subtree of u is marked red. In the right figure, the strongly connected subtree of $c(l)$ is marked green. As l is an outermost leaf, this is a proper cluster.



2.6.7. Definition. Let $\pi = (T, \lessdot, \mathsf{S}, \mathsf{R})$ and $\pi' = (T', \lessdot', \mathsf{S}', \mathsf{R}')$ be derivations. Then π' is a *(maximal) subderivation* of π if $(T', \lessdot')$ is a (maximal) subtree of $(T, \lessdot)$ and S', R' and S, R coincide on T'. We say that π and π' are isomorphic, written as $\pi \sim \pi'$, if there is a bijection $f : T \to T'$ preserving the relations $\lessdot$, S and R. We call a derivation π *regular* if it has finitely many distinct maximal subderivations, up to isomorphism.

Let v be a node in a derivation $\pi = (T, \lessdot, \mathsf{S}, \mathsf{R})$. Let $(T_v, \lessdot')$ be the subtree of $(T, \lessdot)$ rooted at v. If v is the root of some cluster in π, then $\pi_v := (T_v, \lessdot', \mathsf{S}|_{T_v}, \mathsf{R}|_{T_v})$ is a maximal subderivation of π; in this case we call π_v the *maximal subderivation of* π *rooted at* v. If on the other hand v is not the root of some cluster, then π_v contains leaves labeled with a discharge token $\dagger$ without a companion node labeled with $\mathsf{D}_\dagger$, and thus π_v does not qualify as a derivation.

Infinite proofs

2.6.8. Definition. An *infinitary proof system* is a pair $\mathcal{P} = (\mathcal{D}, \mathcal{G})$, where $\mathcal{D}$ is a derivation system and $\mathcal{G}$ is a set of infinite $\mathcal{D}$-paths, called the *global soundness condition*.

Given an infinite path $\beta = v_0 v_1 \ldots$, the global soundness condition $\mathcal{G}$ is usually specified by a condition on its stream of labels $(\mathsf{S}_{v_0}, \mathsf{R}_{v_0})(\mathsf{S}_{v_1}, \mathsf{R}_{v_1})\ldots$. In some proof systems this necessitates to add extra information to the names of rules. As an example, consider the following (branch of a) conjunction rule

$$\frac{\varphi, \varphi \wedge \psi, \varphi \wedge \chi}{\varphi \wedge \psi, \varphi \wedge \chi} \wedge$$

Just by looking at the sequents and the name of the rule, it is impossible to know if φ "descends" from $\varphi \wedge \psi$ or from $\varphi \wedge \chi$. Yet, for certain proof systems this kind of information may be essential to define the global soundness condition. In those systems, the name of the rule will incorporate the *principal formula* – the formula in the conclusion to which the rule is applied. For each proof system we consider, we will formally define the rules and the notion of a principal formula.

2.6.9. Definition. Let $\mathcal{P} = (\mathcal{D}, \mathcal{G})$ be an infinitary proof system. An *infinitary* $\mathcal{P}$*-proof* π is a $\mathcal{D}$-derivation without discharge rules, where all leaves are labeled with axioms and all infinite branches are in $\mathcal{G}$. We say that $\mathcal{P}$ *proves* a sequent Γ, written $\mathcal{P} \vdash \Gamma$, if there is an infinitary $\mathcal{P}$-proof π, where the root is labeled with Γ. If $\mathcal{P}$ is clear from the context, we will omit it and just write $\vdash \Gamma$.

Cyclic proofs In infinite proofs all infinite branches have to satisfy a certain global soundness condition. Cyclic proofs have to adhere to a soundness condition as well; this condition can be given in different ways.

2.6.10. Definition. A *cyclic proof system* is a pair $\mathcal{P} = (\mathcal{D}, \mathcal{G})$, where $\mathcal{D}$ is a derivation system and $\mathcal{G}$ is a set of finite $\mathcal{D}$-derivations, called the *soundness condition*.

2.6.11. Definition. Let $\mathcal{P} = (\mathcal{D}, \mathcal{G})$ be a cyclic proof system. Let $\mathcal{A}$ be a set of sequents, called the set of *assumptions*. A *cyclic $\mathcal{P}$-proof* π with assumptions $\mathcal{A}$ is a finite $\mathcal{D}$-derivation in $\mathcal{G}$, where every leaf labeled with o is labeled with a sequent in $\mathcal{A}$. We say that $\mathcal{P}$ *proves* a sequent Γ with assumptions $\mathcal{A}$, written $\mathcal{A} \vdash_{\mathcal{P}} \Gamma$, if there is a cyclic $\mathcal{P}$-proof π with assumptions $\mathcal{A}$, where the root is labeled with Γ. If $\mathcal{A}$ is empty, we write $\mathcal{P} \vdash \Gamma$. If $\mathcal{P}$ is clear from the context, we will omit it and just write $\mathcal{A} \vdash \Gamma$ or $\vdash \Gamma$.

2.6.12. Definition. The soundness condition $\mathcal{G}$ on cyclic proofs can be given in different ways. In particular, there are two ways to define a *local* soundness condition:

- A *path-based* condition is given as a set of finite $\mathcal{D}$-paths $\mathcal{G}_p$. The corresponding set of finite derivations $\mathcal{G}$ is defined as follows.

  Let l be a a repeat leaf in a $\mathcal{D}$-derivation π with companion $c(l)$, and let β_l be the *repeat path* of l in $\mathcal{T}_\pi$ from $c(l)$ to l. We call l a *discharged leaf* if the path β_l is in $\mathcal{G}_p$. We call a leaf *closed* if it is either a discharged leaf or labeled with an axiom, and call it *open* otherwise.

  The set of finite derivations $\mathcal{G}$ is defined as all finite derivations, where all repeat leaves are discharged.

- A *subgraph-based* condition is given as a collection of sets of $\mathcal{D}$-nodes $\mathcal{G}_s$. The set of finite derivations $\mathcal{G}$ is defined as all finite derivations π, where all proper strongly connected subgraphs of π are in $\mathcal{G}_s$.

In concrete cases, we will not distinguish between a derivation system and a proof system, and we will denote both – whether or not they include the soundness condition – by the same name.

2.6.13. Definition. Let π be a $\mathcal{D}$-derivation. Recall that we call a leaf l outermost if $c(l)$ is the root of a proper cluster in π. The *unfolding* of an outermost leaf l in π is the derivation obtained from π by replacing l with the maximal subderivation $\pi_{c(l)}$ of π rooted at $c(l)$.[1]

The *infinite unfolding* π^* of π is the $\mathcal{D}$-derivation obtained from π by recursively unfolding outermost leaves, and removing all discharge rules.

The infinite unfolding is commonly used to translate cyclic proofs to infinitary ones. For specific systems, the definition of the infinite unfolding may differ from our general definition – we will define it precisely whenever differences arise.

[1]In order to guarantee that D rules are labeled with unique discharge tokens, in $\pi_{c(l)}$ discharge tokens may be replaced by fresh discharge tokens not occurring in π.

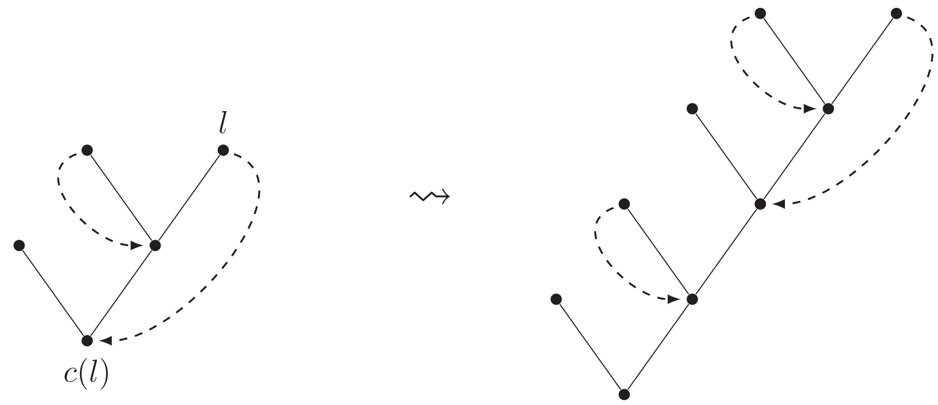


Figure 2.4: The unfolding of an outermost leaf l.

2.7 NW-proofs

In the last section of this chapter, we will introduce the infinitary proof system NW for the modal μ-calculus. This system is directly based on the infinite two-player tableau-style game introduced by Niwiński and Walukiewicz [NW96] with two modifications. First, we present their system in the shape of a proof system. This change of perspective can be justified by identifying winning strategies for one of the players in the game with NW-proofs. Second, our presentation is dual to the one in [NW96], meaning that we are concerned with proving validity and consequently interpret sequents disjunctively.

Throughout this section *sequents* are sets of $\mathcal{L}_\mu$-formulas. For simplicity we assume that the set of actions Act is a singleton and we denote modalities by $\Box$ and $\Diamond$. The system could straightforwardly be generalized to the case of multiple modalities. Given a sequent Γ, we define $\Diamond\Gamma := \{\Diamond\varphi \mid \varphi \in \Gamma\}$. The rules of the derivation system NW are given in Figure 2.5.

$$\mathsf{Ax1}\colon \frac{}{p, \bar{p}} \qquad \vee\colon \frac{\varphi, \psi, \Gamma}{\varphi \vee \psi, \Gamma} \qquad \mu\colon \frac{\varphi[\mu x.\varphi/x], \Gamma}{\mu x.\varphi, \Gamma} \qquad \Box\colon \frac{\varphi, \Gamma}{\Box\varphi, \Diamond\Gamma}$$

$$\mathsf{Ax2}\colon \frac{}{\top} \qquad \wedge\colon \frac{\varphi, \Gamma \qquad \psi, \Gamma}{\varphi \wedge \psi, \Gamma} \qquad \nu\colon \frac{\varphi[\nu x.\varphi/x], \Gamma}{\nu x.\varphi, \Gamma} \qquad \mathsf{weak}\colon \frac{\Gamma}{\varphi, \Gamma}$$

Figure 2.5: Rules of NW

In the rules $\vee$, $\wedge$, $\Box$, μ and ν the single explicitly written formula in the conclusion is called the *principal formula* and the explicitly written formulas in the premises are called *auxiliary formulas*.

All formulas in the conclusion and premise of $\Box$ are called *active*. For the other rules we call the formulas in the context Γ *inactive* and the other formulas *active*. Note that, as sequents are sets, formulas might be both active and inactive at the same time. The notions of principal, auxiliary and active formulas transfer to other proof systems presented in this thesis as expected.

Following our general definition of proofs in Section 2.6, *rules* in the NW system formally are pairs (R, ξ), where R is the name of the rule and ξ is either its principal formula or "nil", if R does not have a principal formula. This guarantees that the notions of principal, auxiliary, active and inactive formulas are always uniquely defined by rules. Whenever it is clear from the context, we will sometimes omit the principal formula and just write R for the pair (R, ξ).

In order to define the global soundness condition of NW, which determines whether an NW-derivation qualifies as a proper *proof*, one must keep track of the development of individual formulas along infinite branches.

2.7.1. Definition. Let Γ be the conclusion and Γ' be a premise of a rule R in Figure 2.5. Let $\varphi \in \Gamma$ and $\psi \in \Gamma'$. The *trace relation* $\mathsf{T}_{\Gamma,\mathsf{R},\Gamma'} \subseteq \Gamma \times \Gamma'$ consists of pairs (φ, ψ) such that ψ "originates from" φ, that is, if either

(i) φ and ψ are inactive and $\varphi = \psi$, or

(ii) φ and ψ are active and either

  (a) φ is the principal and ψ is an auxiliary formula of R, or

  (b) $\mathsf{R} = \Box$ and $\varphi = \Diamond\psi$.

If $(\varphi, \psi) \in \mathsf{T}_{\Gamma,\mathsf{R},\Gamma'}$, we say that ψ is a *direct descendant* of φ.

2.7.2. Definition. Let $\beta = (v_i)_{i<\kappa}$ with $\kappa \leq \omega$ be a path in an NW-derivation π. An NW-*trace* on β is a sequence $\tau = (\varphi_i)_{i<\kappa}$ of formulas such that $\varphi_i \in \mathsf{S}_{v_i}$ for $i < \kappa$ and such that φ_{i+1} is a direct descendant of φ_i whenever $i + 1 < \kappa$.

We obtain the *tightening* $\widehat{\tau}$ of such an NW-trace τ by erasing all φ_{i+1} from τ for which φ_i and φ_{i+1} are inactive in the rule application at v_i, in other words, where φ_{i+1} is a direct descendant of φ_i by virtue of item (i) in Definition 2.7.1.

Note that the tightening $\widehat{\tau}$ of an NW-trace τ is a trace of $\mathcal{L}_\mu$-formulas as defined in Section 2.3. We call τ a *ν-trace* if its tightening $\widehat{\tau}$ is a ν-trace. In particular, this implies that both τ and $\hat{\tau}$ are infinite. An infinite path β in π is called *successful*, if there is a ν-trace on β.

Whenever it is clear from the context, we will identify NW-traces with traces as defined in Section 2.3 and call NW-traces simply traces.

Let $\varphi \in \mathsf{S}_u$ and $\psi \in \mathsf{S}_v$ be such that there is a path β from u to v. Then we call ψ a *descendant* of φ, if there is a trace on β starting from φ and ending at ψ. In this case we call φ an *ancestor* of ψ. These notions will also be used in other proof systems in this thesis.

2.7.3. Remark. The traces on a path β form a rather intricate structure. A formula φ may have multiple direct descendants – for instance, if φ is the principal formula of a $\vee$-rule. Moreover, since sequents are sets, a formula φ may also have multiple direct ancestors. Consequently, the traces on β do not form a tree, but rather a directed acyclic graph.

2.7.4. Definition. The infinitary proof system NW is defined by the rules in Figure 2.5 together with all successful paths.

We refer to soundness conditions defined in terms of traces as *trace-based* soundness conditions, and call proof systems with such a soundness condition *trace-based.* In Chapter 5 we will see another example of such a trace-based proof system for the two-way modal μ-calculus.

Soundness and Completeness of NW for guarded formulas follows from the results by Niwiński and Walukiewicz [NW96]. It follows from [Stu08] and [FL13] that the result in fact also holds for arbitrary formulas. By Theorem 6.3 in [NW96], NW-proofs can be assumed to be regular, and this observation applies to unguarded formulas as well.

2.7.5. Theorem (Soundness and Completeness). *Let Γ be a sequent. Then $\bigvee \Gamma$ is valid iff $\mathsf{NW} \vdash \Gamma$ iff Γ has a regular NW proof.*

2.7.6. Example. Define the following formulas:

$$\varphi := \nu x.\Diamond(x \wedge \mu y.\Diamond y \vee p),$$
$$\psi := \mu x.\nu y.\Box[(x \vee \overline{p}) \wedge (y \vee p)].$$

In Example 2.3.22 we saw that the formula $\overline{\psi} \to \varphi$ is valid. We can therefore give an NW-proof π of ψ, φ. For convenience, we define the formula

$$\chi := \nu y.\Box[(\psi \vee \overline{p}) \wedge (y \vee p)].$$

Note that $\psi \to_C \chi$, that is, χ is in the closure of ψ. The proof π is given as follows. Here the proof π_0 is isomorphic to the proof π and π_1 is isomorphic to the maximal subderivation of π rooted at χ, φ. The proof ρ is given below.

$$
\begin{prooftree}
\AxiomC{π_0}
\noLine
\UnaryInfC{ψ, φ}
\RightLabel{weak}
\UnaryInfC{$\psi, \overline{p}, \varphi$}
\AxiomC{}
\RightLabel{Ax1}
\UnaryInfC{$\overline{p}, p$}
\RightLabel{weak}
\UnaryInfC{$\psi, \overline{p}, \Diamond(\mu y.\Diamond y \vee p), p$}
\RightLabel{$\vee$}
\UnaryInfC{$\psi, \overline{p}, \Diamond(\mu y.\Diamond y \vee p) \vee p$}
\RightLabel{μ}
\UnaryInfC{$\psi, \overline{p}, \mu y.\Diamond y \vee p$}
\RightLabel{$\wedge$}
\BinaryInfC{$\psi, \overline{p}, \varphi \wedge \mu y.\Diamond y \vee p$}
\RightLabel{$\vee$}
\UnaryInfC{$\psi \vee \overline{p}, \varphi \wedge \mu y.\Diamond y \vee p$}
\AxiomC{π_1}
\noLine
\UnaryInfC{χ, φ}
\AxiomC{ρ}
\noLine
\UnaryInfC{$\chi, \mu y.\Diamond y \vee p$}
\RightLabel{$\wedge$}
\BinaryInfC{$\chi, \varphi \wedge \mu y.\Diamond y \vee p$}
\RightLabel{weak}
\UnaryInfC{$\chi, p, \varphi \wedge \mu y.\Diamond y \vee p$}
\RightLabel{$\vee$}
\UnaryInfC{$\chi \vee p, \varphi \wedge \mu y.\Diamond y \vee p$}
\RightLabel{$\wedge$}
\BinaryInfC{$(\psi \vee \overline{p}) \wedge (\chi \vee p), \varphi \wedge \mu y.\Diamond y \vee p$}
\RightLabel{$\Box$}
\UnaryInfC{$\Box[(\psi \vee \overline{p}) \wedge (\chi \vee p)], \Diamond(\varphi \wedge \mu y.\Diamond y \vee p)$}
\RightLabel{ν}
\UnaryInfC{$\chi, \Diamond(\varphi \wedge \mu y.\Diamond y \vee p)$}
\RightLabel{ν}
\UnaryInfC{χ, φ}
\RightLabel{μ}
\UnaryInfC{ψ, φ}
\end{prooftree}
$$

The proof ρ is given as follows, where the proof ρ' is isomorphic to ρ.

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{Ax1}
\UnaryInfC{$\overline{p}, p$}
\RightLabel{weak}
\UnaryInfC{$\psi, \overline{p}, \Diamond(\mu y.\Diamond y \vee p), p$}
\RightLabel{$\vee$}
\UnaryInfC{$\psi, \overline{p}, \Diamond(\mu y.\Diamond y \vee p) \vee p$}
\RightLabel{μ}
\UnaryInfC{$\psi, \overline{p}, \mu y.\Diamond y \vee p$}
\RightLabel{$\vee$}
\UnaryInfC{$\psi \vee \overline{p}, \mu y.\Diamond y \vee p$}
\AxiomC{ρ'}
\noLine
\UnaryInfC{$\chi, \mu y.\Diamond y \vee p$}
\RightLabel{weak}
\UnaryInfC{$\chi, p, \mu y.\Diamond y \vee p$}
\RightLabel{$\vee$}
\UnaryInfC{$\chi \vee p, \mu y.\Diamond y \vee p$}
\RightLabel{$\wedge$}
\BinaryInfC{$(\psi \vee \overline{p}) \wedge (\chi \vee p), \mu y.\Diamond y \vee p$}
\RightLabel{$\Box$}
\UnaryInfC{$\Box[(\psi \vee \overline{p}) \wedge (\chi \vee p)], \Diamond(\mu y.\Diamond y \vee p)$}
\RightLabel{ν}
\UnaryInfC{$\chi, \Diamond(\mu y.\Diamond y \vee p)$}
\RightLabel{weak}
\UnaryInfC{$\chi, \Diamond(\mu y.\Diamond y \vee p), p$}
\RightLabel{$\vee$}
\UnaryInfC{$\chi, \Diamond(\mu y.\Diamond y \vee p) \vee p$}
\RightLabel{μ}
\UnaryInfC{$\chi, \mu y.\Diamond y \vee p$}
\end{prooftree}
$$

We need to argue that every infinite branch of π carries a ν-trace. Let γ be the only infinite branch of ρ. Then the trace σ defined as[2]

$$\chi \ \to_C \ \Box[(\psi \vee \overline{p}) \wedge (\chi \vee p)] \ \to_C \ (\psi \vee \overline{p}) \wedge (\chi \vee p) \ \to_C \ \chi \vee p \ \to_C \ \chi \ \to_C \ \cdots$$

is a ν-trace on γ.

In π there are infinitely many infinite branches. We start by studying traces on finite parts of the infinite branches. Let β_0 be the path in π from the root of π to the root of π_0. Then τ_0 defined as

$$\varphi \ \to_C \ \Diamond(\varphi \wedge \mu y.\Diamond y \vee p) \ \to_C \ \varphi \wedge \mu y.\Diamond y \vee p \ \to_C \ \varphi$$

[2]Formally, this is the tightening of an NW-trace. As mentioned before, we will identify NW-traces with their tightenings an their corresponding traces of $\mathcal{L}_\mu$-formulas.

is a trace on β_0. Analogously, let β_1 be the path in π from the root of π to the root of π_1. Then τ_1 defined as

$$\varphi \ \rightarrow_C \ \Diamond(\varphi \wedge \mu y.\Diamond y \vee p) \ \rightarrow_C \ \varphi \wedge \mu y.\Diamond y \vee p \ \rightarrow_C \ \varphi$$

is a trace on β_1.

Any infinite branch α of π is a path in $(\beta_0, \beta_1)^\omega$ or in $(\beta_0, \beta_1)^*\gamma$. In the first case, a concatenation of the paths τ_0 and τ_1 is a ν-trace on α. In the second case, σ is the tail of a ν-trace on α. We have shown that every infinite branch of π carries a ν-trace and therefore π is an NW-proof.

Chapter 3

Determinization of ω-automata

ω-Automata were first introduced by Büchi [Büc62] as a variation of finite state taking *infinite words* as inputs. Such infinite words are accepted by the automaton if a run of the automaton passes infinitely many accepting states, automata with such an acceptance condition are nowadays called Büchi automata. More intricate acceptance conditions on ω-automata were presented by Muller [Mul63] and Rabin [Rab69] and it turned out that all of those ω-automata recognize the same languages, called the ω-regular languages [McN66]. The only exception is the class of Büchi automata with a deterministic transition function, which is strictly less expressive. We will elaborate on the notable distinctions between deterministic and non-deterministic automata in more detail later.

There are manifold applications of the theory of ω-automata to logic – in fact, Büchi introduced the concept for the purpose of showing decidability of the monadic second-order theory of the natural numbers with successor. A more thorough investigation of the connections between ω-automata and logic can be found in [GTW02].

Modal μ-calculus The modal μ-calculus is closely related to automata theory. To begin with, the decidability of $\mathcal{L}_\mu$ has been reduced to the non-emptiness problem of automata on infinite trees [SE89]. Most of its theoretical results were obtained by characterizing the modal μ-calculus with so-called alternating automata, see also Chapter 2.

The proof theory of the modal μ-calculus has multiple connections to ω-automata as well. Niwinski and Walukiewicz [NW96] introduced a tableau system and used parity automata to obtain an exponential bound on the size of tableaus. Parity automata are more general versions of Büchi automata, where priorities are assigned to all states and a run of the automaton is successful if the maximal priority that is occurring infinitely often, is even. By determinizing this parity automaton Walukiewicz [Wal93] could prove the completeness of a Hilbert style system, yet with a different induction axiom from the expected one. Jungteerapanich [Jun10] and Stirling [Sti14] took it further and explicitly studied the proof system obtained by building the deterministic automaton into the syntax.

Determinization of finite word automata As already highlighted above in its application to the modal μ-calculus, the *determinization* of ω-automata is one of the central problems in automata theory. We start by outlining the strategy used to determinize an automaton over *finite words*:

A *finite word automata* over a set Σ, called an *alphabet*, is a tuple $\mathbb{A} = (A, \Delta, a_I, F)$ consisting of a finite set of *states* A; a *transition function* $\Delta : A \times \Sigma \to \mathcal{P}(A)$; an *initial state* $a_I \in A$ and a set of *accepting states* $F \subseteq A$. Such a finite word automata is called *deterministic* if the range of Δ only consists of singletons.

A *run* r of such an automaton $\mathbb{A}$ on a finite word $w = z_0 z_1 ... z_n \in \Sigma^*$ is a sequence $a_0 a_1 ... a_n \in A^*$ such that $a_0 = a_I$ and $a_{i+1} \in \Delta(a_i, z_i)$ for $i = 0, ..., n-1$. The automaton $\mathbb{A}$ *accepts* r if $a_n \in F$. A word w is *accepted* by $\mathbb{A}$ if there is an accepting run r of $\mathbb{A}$ on w.

Let $\mathbb{A}$ be a non-deterministic finite word automaton; we want to find a deterministic automaton $\mathbb{A}^P$ that accepts the same language. After an input of a word w multiple states a of A might be reached. In order to simulate $\mathbb{A}$ with a deterministic automaton one considers all those states a together. We therefore define the states of $\mathbb{A}^P$ to be subsets of A and call them *macrostates* in contrast to the states of $\mathbb{A}$. Formally, we define the initial macrostate of $\mathbb{A}^P$ to be the singleton $\{a_I\}$ and, given a macrostate A_0 and input letter z, we define the transition function δ of $\mathbb{A}^P$ as $\delta(A_0, z) := \bigcup_{a \in A_0} \Delta(a, z)$. We call this definition of the transition function a *macro-move*. The accepting states of $\mathbb{A}^P$ will be those subsets of A that intersect with F. It can be seen quite easily that the resulting automaton $\mathbb{A}^P$ is equivalent to $\mathbb{A}$, yet it might be exponentially bigger. This construction is called the *powerset construction* and was first introduced by Rabin and Scott [RS59].

Determinization of ω-automata Let us now move on to ω-automata and more specifically to their simplest versions – Büchi automata. These are defined as finite word automata, where inputs are infinite words w that are accepted iff an (infinite) run of the automaton on w contains infinitely many occurrences of states in F. McNaughton [McN66] first proved that non-deterministic Büchi automata have the same expressivity as deterministic Muller automata. In this setting, the problem is much harder. To begin with, note that the powerset construction of the Büchi automaton $\mathbb{A}$, which gives $\mathbb{A}^P$, might accept more words than $\mathbb{A}$ does: It is possible that the run of $\mathbb{A}^P$ on w is accepted, even though $\mathbb{A}$ has no run on w that visits states in F infinitely often – only infinitely many runs that visit states in F only finitely often.

Therefore, more complicated constructions are needed. One possible way is to add more structure to the macrostates in the powerset construction, meaning that states of the deterministic automaton will consist of subsets of A *with extra information*. In fact, all mentioned and introduced constructions in this thesis are of that shape. The first optimal construction that simulates a non-deterministic Büchi automaton A by a deterministic Rabin automaton was given by Safra [Saf88]. States of the deterministic automaton consist of trees – so-called Safra-trees – where nodes are labeled by subsets of A. This result has been one of the hallmarks of the theory of ω-automata, and the Safra-construction is still the most widely used determinization method.

Since the Safra-construction is essentially ad-hoc, other determinization methods with more underlying theory have been developed. Muller and Schupp [MS95] simulate alternating tree automata by non-deterministic automata, and as a corollary also obtain a determinization method for Büchi automata. This method has later been simplified by Kähler and Wilke [KW08]. At the core of this approach are *profile trees*[1], these are trees that encode all the essential information of the infinite runs of the Büchi automaton. Building on these works, a particularly neat determinization method was developed by Fogarty et al. [FKVW15], where macrostates encode levels of the profile tree.

Contributions We introduce a new determinization method that is also based on profile trees. Yet, differently from [FKVW15], states of our deterministic automaton will consist of labeled binary trees, that encode the profile tree up to some level. This construction is bespoke so that it can be used in Chapter 4 to obtain a new proof system for the modal μ-calculus.

All determinization constructions mentioned above have only been developed for Büchi automata, the reason being that other kinds of ω-automata may first be translated to non-deterministic Büchi automata, and then be determinized. In our application this is not the desired approach, as we also want the deterministic automaton to be of a certain shape; in particular we want the states of the deterministic automaton to be based on macrostates. Therefore we also generalize our binary tree construction to directly apply to parity automata.

In Section 3.3 we generalize the Safra construction for parity automata with ε-transitions – transitions that do not consume an input letter. We will use this construction in the proof theory of the two-way modal μ-calculus in Chapter 5.

3.1 ω-automata with ε-transitions

We define automata operating on streams (infinite words). In addition to basic transitions we allow ε-transitions: transitions without an input letter. These automata are called *ω-automata with ε-transitions*.

[1] Some authors also call them *reduced split trees*.

3.1.1. Definition. Let Σ be a finite set, called an *alphabet*. An *ω-automaton* over Σ is a quadruple $\mathbb{A} = (A, \Delta, a_I, \mathrm{Acc})$, where A is a finite set; $\Delta : A \times \Sigma \to \mathcal{P}(A)$ is the *transition function* of $\mathbb{A}$; $a_I \in A$ its *initial state*; and $\mathrm{Acc} \subseteq A^\omega$ its *acceptance condition*. An ω-automaton is called *deterministic* if $|\Delta(a, z)| = 1$ for all pairs $(a, y) \in A \times \Sigma$.

3.1.2. Definition. A *run* of such an ω-automaton $\mathbb{A}$ on a stream $w = z_0 z_1 z_2 ... \in \Sigma^\omega$ is a stream $a_0 a_1 a_2 ... \in A^\omega$ such that $a_0 = a_I$ and $a_{i+1} \in \Delta(a_i, z_i)$ for all $i \in \omega$. A stream w is *accepted* by $\mathbb{A}$ if there is a run r of $\mathbb{A}$ on w with $r \in \mathrm{Acc}$.

The acceptance condition can be given in different ways:

- A *Büchi* condition is given as a subset $F \subseteq A$. The corresponding acceptance condition is the set of runs, which contain infinitely many states in F.
- A *parity* condition is given as a priority map $\Omega : A \to \mathbb{N}$. The corresponding acceptance condition is the set of runs α such that $\max\{\Omega(a) \mid a \text{ occurs infinitely often in } \alpha\}$ is even.
- A *Rabin* condition is given as a set $R = ((G_i, B_i))_{i \in I}$ of pairs of subsets of A. The corresponding acceptance condition is the set of runs α for which there exists $i \in I$ such that α contains infinitely many states in G_i and finitely many in B_i.

Automata with these acceptance conditions are called *Büchi*, *parity* and *Rabin automata*, respectively.

3.1.3. Definition. We define $\mathcal{L}(\mathbb{A})$ to be the set of all accepting streams of $\mathbb{A}$. Two automata $\mathbb{A}$ and $\mathbb{B}$ are called *equivalent*, if $\mathcal{L}(\mathbb{A}) = \mathcal{L}(\mathbb{B})$.

A set Σ^ω over an alphabet Σ is called a *language*. Every automata $\mathbb{A}$ over Σ defines a language $\mathcal{L}(\mathbb{A})$ over Σ. We call a language L *regular*, if there is a parity automata $\mathbb{A}$ such that $L = \mathcal{L}(\mathbb{A})$.

3.1.4. Remark. As shown for example in [GTW02], (deterministic) parity automata, (deterministic) Rabin automata, and Büchi automata all accept the same class of languages. Therefore, a language L is called regular if it is accepted by any (or all) of these automata.

3.1.5. Definition. An *ω-automaton with ε-transitions* $\mathbb{A} = (A, \Delta, a_I, \mathrm{Acc})$ is defined analogously to an ω-automaton, where the transition function $\Delta := (\Delta_b, \Delta_\varepsilon)$ is a pair consisting of the *basic transition function* $\Delta_b : A \times \Sigma \to \mathcal{P}(A)$ and the *ε-transition function* $\Delta_\varepsilon : A \to \mathcal{P}(A)$. For simplicity we always assume that $\Delta_\varepsilon(a_I) = \varnothing$.

An *extended run* of such an ω-automaton with ε-transitions $\mathbb{A}$ on a stream $w = z_0 z_1 z_2 ... \in \Sigma^\omega$ is a stream $(a_0, n_0)(a_1, n_1)(a_2, n_2)... \in (A \times \mathbb{N})^\omega$ such that $(a_0, n_0) = (a_I, 0)$, and for all $i \in \omega$ either

1. $a_{i+1} \in \Delta_b(a_i, z_{n_i})$ and $n_{i+1} = n_i + 1$ or

2. $a_{i+1} \in \Delta_\varepsilon(a_i)$ and $n_{i+1} = n_i$.

Additionally we assume that $\sup\{n_i \mid i \in \omega\} = \omega$ guaranteeing that every run contains infinitely many basic transitions; in other words, we do not allow runs that from some point onwards only consist of ε-transitions.

A *run* of $\mathbb{A}$ on a stream w is a stream $a_0a_1a_2... \in A^\omega$ such that there are natural numbers $n_0, n_1, ...$ in a way that $(a_0, n_0)(a_1, n_1)...$ is an extended run of $\mathbb{A}$ on w. As before, a stream w is *accepted* by $\mathbb{A}$ if there is a run r of $\mathbb{A}$ on w with $r \in \mathrm{Acc}$.

3.2 Determinization with binary trees

In this section we introduce a determinization method for Büchi and parity automata without ε-transitions. Our approach is inspired by a determinization method based on profile trees in [FKVW15], yet, differently to their construction, states of the deterministic automaton will be based on *binary trees*. We start by fixing notations on binary trees. In Subsection 3.2.2 we motivate and define the construction for Büchi automata and then generalize the approach to parity automata in Subsection 3.2.3.

3.2.1 Binary trees

We let 2^* denote the set of *binary strings* and write $<$ for the lexicographical order on 2^*. We write $\preccurlyeq$ for the initial substring relation given by $s \preccurlyeq t$ if $sr = t$ for some r and define $s \prec t$ if $s \preccurlyeq t$ and $s \neq t$. *Substitution* for binary strings is defined in the following way: Let $s, t, \tilde{s}, u \in 2^*$ be such that $s = t\tilde{s}$, then $s[u/t]$ denotes the binary string $u\tilde{s}$.

A *binary tree* is a finite set of binary strings $T \subseteq 2^*$ such that $s0 \in T \Rightarrow s \in T$ and $s0 \in T \Leftrightarrow s1 \in T$. We let $\mathrm{leaves}(T) := \{s \in T \mid s0 \notin T\}$ denote the set of *leaves* of T, and $\mathrm{minL}(T)$ the *minimal* leaf of T, that is, the unique leaf of the form $0 \cdots 0$. A set of binary strings L is a *set of leaves of a binary tree* if $\mathrm{tree}(L) := \{s \in 2^* \mid \exists t \in L : s \preccurlyeq t\}$ is a binary tree and L is an antichain, meaning that for all $s \neq t \in L$ we have $s \not\preccurlyeq t$.

3.2.2 Büchi automata

Let Σ be an alphabet. In this subsection we fix a non-deterministic Büchi automaton $\mathbb{B} = (B, \Delta, b_I, F)$ over Σ. We will define a deterministic Rabin automaton $\mathbb{B}^D$ that is equivalent to $\mathbb{B}$. We start by motivating our construction, in particular we introduce the crucial notion of a *profile tree*.

Motivation Let $w = (z_i)_{i \in \omega}$ be an infinite word in Σ^ω. In order to decide if $\mathbb{B}$ accepts w one might study all possible runs of $\mathbb{B}$ on w. Yet, we are only interested in acceptance, hence the exact shape of paths through $\mathbb{B}$ might be ignored and we only have to remember if states in F are passed. This gives rise to the following notion:

The *split tree* of $\mathbb{B}$ on w is the pair $T_S = (T, l)$, where T is the full infinite binary tree and l labels every node s with $B_s \subseteq B$, such that $l(\varepsilon) := \{b_I\}$ and for $|s| = i$:

$$l(s1) := \Delta(B_s, z_i) \cap F \quad \text{and} \quad l(s0) := \Delta(B_s, z_i) \cap \overline{F},$$

where we define $\Delta(B_s, z_i) := \bigcup_{b \in B_s} \Delta(b, z_i)$. It describes all possible runs of $\mathbb{B}$ on w by binary strings, where 1s keep track of visited states in F.

The *profile tree* T_P, introduced in[2] [FKWV13], is a pruned version of the split tree, where

1. at each level all but the (lexicographically) greatest occurrence of a state b are removed and
2. nodes labeled by the empty set are deleted.

This results in a tree of bounded width, where every node has 0,1 or 2 children. Importantly, the profile tree still contains all information that is needed to decide if $\mathbb{B}$ accepts w.

3.2.1. Lemma ([FKVW15], Theorem 3.7). *The profile tree T_P contains an infinite branch with infinitely many 1s iff $\mathbb{B}$ accepts w.*

Our determinization method takes the profile tree T_P as a starting point as follows: We want the states of the deterministic automaton to be initial subtrees of the profile tree up to a certain level. As this does not give a bound on the size of states we opt to only encode the essential information of such a subtree of the profile tree.

The crucial observation is the following: Let T be a subtree of the profile tree up to some level n. Whenever a node s in T has exactly one child $t = s1$ or $t = s0$, then we may identify s and t and remove the edge between the nodes. If $t = s1$ we thereby remove the information, that an accepting state in F has been passed and we thus color s green. All descendants of s are colored red, which marks that those nodes are located in a part of the tree, that is not stable yet. Doing this for all such nodes results in a colored binary tree.

The binary tree construction starts with the binary tree of one node labeled with b_I and, in every step of the transition function adds one level of the profile tree and then compresses and colors the tree. We accept a run of the resulting deterministic automaton, if there is a node that is always present, labeled green infinitely often and red only finitely often. Figure 3.1 contains an example of this determinization construction.

[2] Note that this concept already occurs in [KW08] with the different name *reduced split tree*.

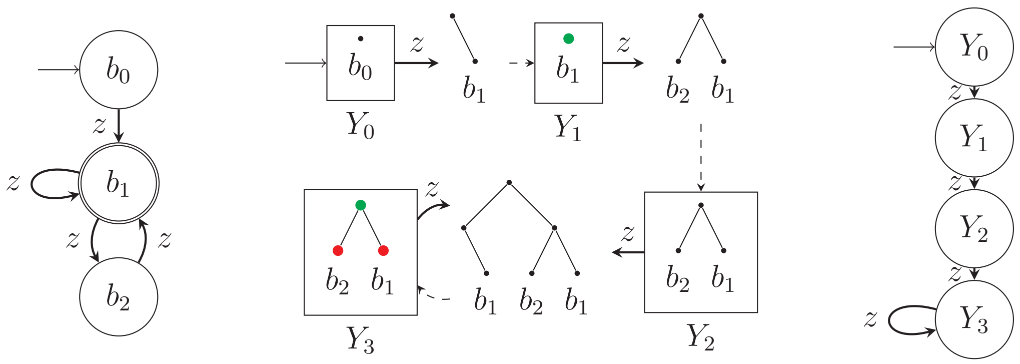


Figure 3.1: A nondeterministic Büchi automaton $\mathbb{B}$ on the left and its determinization $\mathbb{B}^D$ on the right. The diagram in the middle shows the internal structure of the states Y_0, Y_1, Y_2 and Y_3 of $\mathbb{B}^D$. Binary trees are represented in the obvious way (i.e., the root is the string ε, and for every node the left child appends 0 and the right child appends 1). The transitions of $\mathbb{B}^D$ are split in two parts: In the first part one level of the split tree is added, corresponding to the steps 1 and 2 in the construction of the transition function defined on the next page. In the second part (the dashed arrows) the tree is pruned and compressed, corresponding to the steps 3 and 4. The acceptance condition of $\mathbb{B}^D$ is such that the word z^ω is accepted by $\mathbb{B}^D$ because the string ε is always in play, colored green infinitely often and never red.

Construction In our formal definition we change our perspective on this construction: Instead of defining states Y of the deterministic automaton as binary trees, where the leaves are labeled by subsets of B, we define Y to consist of subsets of B where each state b is mapped to a binary string – its location in the binary tree. The reason for this different view on the construction is that it better aligns with its use in Chapter 4: We will define the states of the automaton to consist of formulas and the binary strings will annotate those formulas.

We define the deterministic Rabin automaton $\mathbb{B}^D := (B^D, \delta, b'_I, R)$ as follows: Its carrier set B^D of $\mathbb{B}^D$ consists of all triples $Y = (B_Y, f, c)$, where

- B_Y is a subset of B,
- $f : B_Y \to 2^*$ is a map, such that[3] $\mathrm{ran}(f)$ is a set of leaves of a binary tree and
- c is a map from $\mathrm{tree}(\mathrm{ran}(f)) \to \{\mathrm{green}, \mathrm{red}, \mathrm{white}\}$.

We call a subset $B_Y \subseteq B$ a *macrostate* and call $Y \in B^D$ a *BT-state*, in other words a BT-state is a macrostate with additional information. We define T^Y

[3] For a function $f : X \to Y$ we let $\mathrm{ran}(f) \subseteq Y$ be the *range* of f, that is, $\mathrm{ran}(f) := \{f(x) \mid x \in X\}$.

to be the binary tree tree(ran(f)), that has ran(f) as its leaves and say that a binary string s is *in play* if $s \in T^Y$. If it is clear from the context we occasionally abbreviate T^Y by T. We will usually denote a BT-state by a set of pairs (b, s), written as b^s, where $b \in B_Y$ and $s = f(b)$ and deal with the map c implicitly. We call c a *coloring map* and say that a string s is colored col if $c(s) = \text{col}$, where $\text{col} = \text{white}, \text{green}, \text{red}$.

The initial BT-state b'_I consists of the singleton $\{b_I^\varepsilon\}$, where $c(\varepsilon) = \text{white}$. To define the transition function δ let Y be in B^D and $z \in \Sigma$. We define $\delta(Y, z) := Y'$, where we build up Y' in the following steps starting from the empty set. Note that $T^{Y'}$ is a binary tree only after completing all four steps and not necessarily at intermediate steps.

1. Move: For every $a^s \in Y$ and $b \in \Delta(a, z)$, add b^s to Y'.

2. Append: For every $a^s \in Y'$, where $a \notin F$, change a^s to a^{s0}. For every $a^s \in Y'$, where $a \in F$, change a^s to a^{s1}.

3. Resolve: For any a^s and a^t in Y', where $s < t$, remove a^s.

4. Compress/Colour: Let $c(t) = \text{white}$ for every $t \in T^{Y'}$. We compress and color T in the following way, until there exists no *witness* $t \in T$ such that (a) or (b) is applicable:

   (a) For any $t \in T$, such that $t0 \in T$ and $t1 \notin T$, change every $a^s \in Y'$, where $t0 \preccurlyeq s$, to $a^{s[t/t0]}$. For any $s \in T$, where $t \prec s$, let $c(s) = \text{red}$.

   (b) For any $t \in T$, such that $t0 \notin T$ and $t1 \in T$, change every $a^s \in Y'$, where $t1 \preccurlyeq s$, to $a^{s[t/t1]}$. For any $s \in T$ such that $t = s0 \cdots 0$, let $c(s) = \text{green}$, if $c(s) \neq \text{red}$. In particular, let $c(t) = \text{green}$ if $c(t) \neq \text{red}$. For any $s \in T$, where $t \prec s$, let $c(s) = \text{red}$.

The Rabin automaton $\mathbb{B}^D$ accepts a run if there is a binary string s, which is in play cofinitely often such that s is colored green infinitely often and red only finitely often.

Step 4 of the transition function δ is defined in a seemingly non-deterministic way. The next proposition shows that the procedure gives a unique BT-state, regardless of how the witnesses are chosen.

3.2.2. Proposition. *1. The transition function δ is well-defined. In particular, the result of step 4 does not depend on the order in which the witnesses $t \in T$ are chosen.*

2. The string ε is never colored red.

3. The length of binary strings occurring in ran(f) *is bounded by the size of B.*

Proof:
1. After step 3 of the transition function Y' consists of a finite set of binary strings such that for all $s \neq t$ in Y' it holds that $s \not\preccurlyeq t$. Thus $T^{Y'}$ describes a tree, where every node has at most two children and its leaves are labeled by disjoint sets of states. Step 4 of δ identifies a node t with its child t', if t' is its unique child. This results in a unique binary tree T'. It remains to show that the coloring of T' does not depend on which nodes are identified first. Therefore, we will give an equivalent presentation of step 4 of δ.

Let T^3 be the tree $T^{Y'}$ after step 3 of the construction. Let $\sim$ be the equivalence relation on T^3 generated by all pairs of nodes (s,t) such that t is the unique child of s. Let T^E be the quotient of T^3 over $\sim$, written as $T^E := \{[s]_\sim \mid s \in T^3\}$. We can define a parent relation on T^E as follows: $[s]_\sim$ is the parent of $[t]_\sim$ iff $s \not\sim t$ and there is $s' \sim s$ and $t' \sim t$ such that s' is the parent of t'. If $[s]_\sim$ and $[t]_\sim$ are siblings in T^E, then there are $s' \sim s$ and $t' \sim t$ such that s' and t' are siblings in T^3, hence they inherit an order on the siblings. Thus T^E is a binary tree, which can be given as a set of binary strings. We can define a coloring map c on T^E as follows: $[s]_\sim$ is colored red if it has an ancestor $[t]_\sim$ with $|[t]_\sim| > 1$. An equivalence class $[s]_\sim$ is colored green if it is not red and it has a minimal descendant (meaning that every child in the ancestor path is the minimal child with respect to the sibling order) $[t]_\sim$ such that there are $t' \sim t'' \sim t$ where more 1s are occurring in t' than in t''. All other nodes are colored white. It can be readily seen that the colored binary tree T^E is isomorphic to $T^{Y'}$ after step 4 independent of which nodes are identified first.

2. The string ε is not a real superstring of any other string, thus it can never be colored red. Item 3 follows because the length of any path from the root to a leaf is bounded by the size of the tree. □

3.2.3. THEOREM. *The automata $\mathbb{B}^D$ and $\mathbb{B}$ are equivalent.*

Proof:
We need to show that $\mathcal{L}(\mathbb{B}^D) = \mathcal{L}(\mathbb{B})$. Let $w = z_0z_1... \in \Sigma^\omega$ and let $\rho = Y_0Y_1Y_2...$ be the run of $\mathbb{B}^D$ on w.

"$\supseteq$": Suppose that $\mathbb{B}$ accepts a run $r = b_0b_1b_2..$ on w. Let s_n be the binary string such that $b_n^{s_n} \in Y_n$ for $n \in \omega$. Let t be the maximal string which is a substring of cofinitely many s_n and colored red only finitely often. Note that such a string always exists, as ε satisfies these conditions due to Proposition 3.2.2.2. By definition, t is in play cofinitely often and colored red only finitely often. We will show that t is colored green infinitely often. Let t' be the maximal string of the form $t0\cdots0$, such that $t' \preccurlyeq s_n$ for cofinitely many n. Note that t' might be colored red infinitely often. Let N be such that $t' \preccurlyeq s_n$ and $c(t) \neq \text{red}$ in Y_n for all $n \geq N$.

Now we distinguish the following cases:

1. $t' = s_n$ for infinitely many n,
2. $t'1 \preccurlyeq s_n$ for cofinitely many n,
3. $t'1 \preccurlyeq s_n$ for infinitely many n and $t'0 \preccurlyeq s_n$ for infinitely many n, while $t' = s_n$ for only finitely many n.

These three cases cover all possibilities, as t' is the maximal string of the form $t0 \cdots 0$ such that $t' \preccurlyeq s_n$ for cofinitely many n.

First assume that $s_n = t'$ for infinitely many n. Let $m \geq N$ be such that $s_m = t'$. As r is accepted, 1 will be appended to s_n at step 2 of δ for some $n \geq m$. As $s_n = t'$ infinitely often, this 1 will need to be removed at some possibly later stage. But a 1 can only get removed in step 4(b), which means that t is colored green, as $c(t)$ is never red. Thus $c(t) =$ green infinitely often.

Second, assume that $t'1 \preccurlyeq s_n$ for cofinitely many n. Let $M \geq N$ be such that $t'1 \preccurlyeq s_n$ for all $n \geq M$. The definition of t implies that $t'1$ is colored red infinitely often. If $t'1$ is colored red at stage n, then there is a witness $u \preccurlyeq t'$ in step 4 of δ. As t is never colored red it follows that $t \preccurlyeq u$. If $t \preccurlyeq u \prec t'$ then we are in step 4(a) of the construction of δ and s_n is replaced by $s_n[u/u0]$. Yet $t'1 \not\preccurlyeq s_n[u/u0]$, which contradicts our assumption. Thus the witness in step 4 of δ has to be t'. In this case $t'1 \in T^{Y_n}$ and $t'0 \notin T^{Y_n}$, thus t is colored green.

Third, consider the case where $t'0 \preccurlyeq s_n$ for infinitely many n and $t'1 \preccurlyeq s_n$ for infinitely many n, while $s_n = t'$ for only finitely many n. Then it holds for infinitely many $n \geq N$ that $t'1 \preccurlyeq s_n$ and $t'0 \preccurlyeq s_{n+1}$. As $s_{n+1} < s_n$, this is only possible if t' is the witness in step 4(b) of the construction, meaning that $t'1 \in T$ and $t'0 \notin T$. As t is never colored red this implies that t is colored green.

Thus in every case t is colored green infinitely often and we have proved the first direction.

"$\subseteq$": Conversely, suppose that there is a binary string t which is in play in ρ cofinitely often and which is colored green infinitely often and red only finitely often. Let N be such that t is in play and never colored red for any $i \geq N$. For $i \geq N$ we define

$$A_i := \{b^s \in Y_i \mid t \preccurlyeq s\}.$$

We first show

For all $b^s \in A_N$ there is a path from b_I to b in $\mathbb{B}$ on input $z_0 ... z_{N-1}$. (3.1)

For all $i > N$ and $b^{s_b} \in A_{i+1}$ there exists $a^{s_a} \in A_i$ such that $b \in \Delta(a, z_i)$ (3.2)

Statement (3.1) follows, as the transition function is just a refined version of a macro-move. For (3.2) let $b^{s_b} \in A_{i+1}$. Due to step 1 of the transition function there is $a^{s_a} \in Y_i$ with $b \in \Delta(a, z_i)$. We choose such an a^{s_a} where s_a is maximal and claim that $t \preccurlyeq s_a$. To see that we take a look at $\delta(Y_i, z_i)$. After step 2 of δ there is $b^{s_a 0}$ or $b^{s_a 1}$ in Y_i', which will not be removed in step 3 as we chose the maximal s_a. Let u be maximal with $u \preccurlyeq s_a$ and $u \preccurlyeq t$. If $u = t$ we are done. Else $u \prec t \preccurlyeq s_b$ and $u \preccurlyeq s_a$. In step 4 of δ the string s_a gets compressed to s_b. If $t \not\preccurlyeq s_a$, then u is a witness in step 4 of the transition function. As $s_a \in T^{Y_i'}$ this implies that $t \notin T^{Y_i'}$. Yet $t \in T^{Y_{i+1}}$ and this is only possible if $c(t) = \text{red}$ at the end of step 4, which contradicts our definition of t.

We next define the *trace tree* $\mathcal{T}^{\mathbb{B}}$. It will consist of the root b_I and nodes (a^{s_a}, i), where $i \geq N$ and $a^{s_a} \in A_i$. We define a partial order $<_{\mathcal{T}}$ on the nodes of $\mathcal{T}^{\mathbb{B}}$ in the following way: $(a^{s_a}, i) <_{\mathcal{T}} (b^{s_b}, i)$ iff $s_a < s_b$. The parent of (a^{s_a}, N) is b_I. For $i \geq N$ the unique parent of $(b^{s_b}, i+1)$ is an element (a^{s_a}, i) such that $b \in \Delta(a, y_i)$ which is maximal with respect to $<_{\mathcal{T}}$. Such an element always exists due to (3.2); if there exist more than one, choose one of them.

The trace tree $\mathcal{T}^{\mathbb{B}}$ is an infinite, finitely branching tree. By König's Lemma there exists an infinite branch $b_I(a_N^{s_N}, N)(a_{N+1}^{s_{N+1}}, N+1)$.... We let r be the infinite branch such that the infinite string $s_N s_{N+1}$... is minimal with respect to the lexicographical order. Due to (3.1) there exists a path r' from b_I to a_N in $\mathbb{B}$ on input $z_0...z_{N-1}$. Combined with (3.2) this implies that $r = r'a_N a_{N+1}...$ is a run of $\mathbb{B}$ on $w = z_0 z_1....$

It remains to show that r is successful, in other words that $a_j \in F$ for infinitely many $j \geq N$. Towards a contradiction assume that there is $M \geq N$ such that $a_i \notin F$ for all $i \geq M$. We have $a_i^{s_i} \in A_i$ for $i \geq M$, where $s_i = tu_i$ for some u_i for all $i \geq M$. Let v_i be minimal such that $s_i = tv_i0 \cdots 0$ for $i \geq M$. As $a_i \notin F$ for all $i \geq M$, only zeros are added to s_i, hence the length of v_i is not increasing. Now r is the minimal infinite path in $\mathcal{T}^{\mathbb{B}}$, thus all lexicographically bigger paths of r are finite. This implies that at some point $k \geq M$ it holds that $s_k = t0 \cdots 0$. The next time when t gets colored green, $t0 \cdots 0$ has to be a witness in step 4(b), which is only possible if $a_j \in F$ for some $j \geq k$. □

3.2.4. LEMMA. *Let $\mathbb{B}$ be a Büchi automaton with n states. The automaton $\mathbb{B}^D$ has $2^{\mathcal{O}(n \log n)}$ BT-states and the Rabin condition consists of $\mathcal{O}(2^n)$ pairs.*

Proof:
A BT-state of $\mathbb{B}^D$ consists of a colored binary tree with k leaves, where k ranges from $0, ..., n$ and a function labeling the leaves of the binary tree with disjoint non-empty subsets of B. The number of binary trees with $k+1$ leaves C_k is called the *k-th Catalan number* [Sta15]. It holds that $C_k = \frac{1}{k+1}\binom{2k}{k} \leq 2^{2k}$. A binary tree with k leaves has $2k-1$ nodes, thus there are 3^{2k-1} possible coloring maps c. The k leaves can be labeled by disjoint subsets of B in $(k+1)^n$ different ways, in particular there are at most $(k+1)^n$ possibilities to label k leaves with

disjoint non-empty subsets of B. In total we have

$$\begin{aligned}|B^D| &\leq \sum_{k=0}^{n} C_{k-1} \cdot 3^{2n-1} \cdot (k+1)^n \\ &\leq 6^{2n-1} \cdot (n+1)^{n+1} = 2^{\mathcal{O}(n \log n)}.\end{aligned}$$

The number of Rabin pairs is the number of binary strings, which may occur in a BT-state. Due to Proposition 3.2.2.3 this is bounded by $\sum_{k=0}^{n} 2^k \leq 2^{n+1}$. □

3.2.5. Remark. Given a Büchi automaton of n states, the asymptotically minimal equivalent deterministic Rabin automaton has $2^{\mathcal{O}(n \log n)}$ states and a Rabin condition of $\mathcal{O}(n)$ pairs [Saf88]. Hence the number of BT-states of $\mathbb{B}^D$ is asymptotically optimal. With some adaptations we could also match the optimal Rabin condition by adding a labeling function as outlined below.

Let $L = \{1, \ldots, 2n-1\}$ be a set of potential labels. BT-states are defined as before, where additionally we add an injective labeling function $l : T^Y \to L$. For the initial state we let $l(\varepsilon) = 1$. The steps $1 - 4$ in the transition function remain the same and we add a final step 5 in which we define the new labeling function l': We let $l'(s) = l(s)$ for all s that already occurred in T^Y and for all $s \in T^{Y'} \setminus T^Y$ we let $c(s) = \text{red}$ and choose new, distinct labels in L, meaning that they do not occur in $\text{ran}(l)$. The binary tree $T^{Y'}$ has at most n leaves, hence it has at most $2n-1$ many nodes and this is always possible.

The new acceptance condition has the following form: The automaton accepts a run if there is a label $k \in L$, such that $c(l^{-1}(k))$ is green infinitely often and red only finitely often. Here $c(l^{-1}(k))$ is defined to be red if $k \notin \text{ran}(l)$. This is a Rabin condition with $\mathcal{O}(n)$ pairs. Notably, we still have $n^{\mathcal{O}(n)}$ BT-states, thus the determination method is asymptotically optimal.

3.2.6. Question. In [LP19] the determinization methods of Safra [Saf88] and Muller-Schupp [MS95; KW08; FKVW15] are unified; in particular, bijections between states of both kind of deterministic automata are given. It would be interesting to see how our construction fits into their framework.

3.2.3 Parity automata

Next we extend the binary tree determinization method to parity automata. Clearly, one could first translate a parity automata to an equivalent Büchi automata and then apply the construction for Büchi automata. Yet, the shape of the resulting deterministic automaton would not fit our needs, in particular the states would not be based on subsets of states of the parity automaton. We thus opt to give a direct determinization construction for parity automata.

For this subsection let Σ be an alphabet and $\mathbb{A} = (A, \Delta_A, a_I, \Omega)$ be a parity automaton without ε-transitions. Let m be the maximal even priority of Ω.

Motivation In order to present the intuitive idea behind the construction for parity automata we first transform $\mathbb{A}$ into an equivalent nondeterministic Büchi automaton $\mathbb{B}$. We can then use the determinization for Büchi automata for certain *subsets* of $\mathbb{B}$.

Let m be the maximal even priority of Ω. For any successful run r in $\mathbb{A}$ there is an even $k = 0, 2, ..., m$ such that from some point onwards all states occurring in r have priority lower or equal than k and such that r contains infinitely many occurrences of states of priority k. Looking at it from a different perspective, some tail of r is a successful run of the Büchi automaton $\mathbb{A}_k$, where $\mathbb{A}_k$ is a copy of $\mathbb{A}$ only containing states of priority lower or equal than k with accepting states being the ones of priority k. Formally, for even $k = 0, 2, ...m$, we define $\mathbb{A}_0, \mathbb{A}_2, ..., \mathbb{A}_m$ as follows: $\mathbb{A}_k = (A_k, \Delta_k, F_k)$ with $A_k = \{a_k \mid a \in A \text{ and } \Omega(a) \leq k\}$, $\Delta_k = \Delta_A|_{A_k}$ and $F_k = \{a_k \in A_k \mid \Omega(a) = k\}$. Note that $\mathbb{A}_0, \mathbb{A}_2, ..., \mathbb{A}_m$ do not qualify as automata under our definition, as they lack an initial state.

Now we define the nondeterministic Büchi automaton $\mathbb{B} = (B, \Delta_B, b_I, F)$:[4]

$$B = A \cup \bigcup_{\substack{k=0 \\ k \text{ even}}}^{m} A_k, \qquad b_I = a_I, \qquad F = \bigcup_{\substack{k=0 \\ k \text{ even}}}^{m} F_k,$$

$$\Delta_B = \Delta_A \cup \bigcup_{\substack{k=0 \\ k \text{ even}}}^{m} \Delta_k \cup \{(a, y, b_k) \in A \times \Sigma \times A_k \mid b \in \Delta_A(a, y)\}.$$

It can easily be verified that the automata $\mathbb{B}$ and $\mathbb{A}$ are equivalent. If we were just interested in any deterministic automaton that is equivalent to $\mathbb{A}$ we could now apply the determinization method from above to $\mathbb{B}$ and be done. Yet, we want our deterministic automaton to be of a certain shape and will thus proceed as sketched below.

Although $\mathbb{A}_k$ is not an automaton for $k = 0, ..., m$ we can define the Büchi automata $\mathbb{A} \cup \mathbb{A}_k = (A \cup A_k, \Delta_B|_{A \cup A_k}, a_I, F_k)$. The intuition behind the determinization of the parity automaton $\mathbb{A}$ is the following: We apply the binary tree construction to every automaton $\mathbb{A} \cup \mathbb{A}_k$ for $k = 0, 2, ..., m$. The annotation of a state $a \in \mathbb{A}$ will then be the tuple $(s_0, s_2, ..., s_m)$, where s_k is the annotation of the state a_k in the determinization of $\mathbb{A} \cup \mathbb{A}_k$ for $k = 0, 2, ..., m$. Because there are no paths from A_k to A_j if $k \neq j$ and none of the accepting states of $\mathbb{B}$ are in the set A, this results in an equivalent automaton.

Construction To make those intuitions formal we need some definitions.

A *treetop* L is a set of leaves of a binary tree, where potentially the minimal leaf is missing, meaning that L is an antichain such that $\text{tree}(L) := \{s \in 2^* \mid \exists t \in$

[4]For easier notation we represent the transition function $B \times \Sigma \to \mathcal{P}(B)$ by its corresponding relation on $B \times \Sigma \times B$.

$L : s \preccurlyeq t\} \cup \{s0 \mid s = 0 \cdots 0 \text{ and } s1 \in L\}$ is a binary tree. Recall that minL(T) is the minimal leaf of a tree T.

For even m let $\text{TSeq}(m) := \{(s_0, s_2, ..., s_m) \mid s_0, s_2, ..., s_m \in 2^*\}$ be the set of sequences of length $\frac{m}{2} + 1$, where $s_0, s_2, ..., s_m$ are binary strings. Let π_k be the projection function, which maps $\sigma = (s_0, ..., s_m)$ to s_k for $k = 0, ..., m$.

The partial order $<_c$ on TSeq(m) is the converse lexicographic order defined as follows: Let $(s_0, ..., s_m) < (t_0, ..., t_m)$ if there exists $k = 0, 2, ..., m$ such that $s_k < t_k$ and $s_j = t_j$ for $j = k + 2, ..., m$.

We now define the deterministic Rabin automaton $\mathbb{A}^D = (A^D, \delta_A, a'_I, R_A)$. Recall that m is the maximal even priority of $\mathbb{A}$. Its carrier set A^D of $\mathbb{A}^D$ consists of all tuples $(A_Y, f, c_0, ..., c_m)$, where

- A_Y is a subset of A,
- $f : A_Y \to \text{TSeq}(m)$ is a map, such that[5] $\text{ran}(\pi_k \circ f)$ is a treetop for $k = 0, 2, ..., m$ and
- c_k is a coloring map from $\text{tree}(\text{ran}(\pi_k \circ f)) \to \{\text{green}, \text{red}, \text{white}\}$ for $k = 0, 2, ..., m$.

We will call $A_Y \in A^D$ a *BT-state*. We define T_k^Y to be the binary tree $\text{tree}(\text{ran}(\pi_k \circ f))$ for $k = 0, 2, ..., m$ and say that a binary string s is *in play at priority* k if $s \in T_k^Y$. If the context is clear we will abbreviate T_k^Y with T_k. Again we usually denote a BT-state by a set of pairs (a, σ), written as a^σ, where $a \in A_Y$ and $\sigma = f(a)$ and deal with the coloring maps $c_0, ..., c_m$ implicitly.

The initial BT-state a'_I consists of the singleton $\{a_I^{(\varepsilon,...,\varepsilon)}\}$. To define the transition function δ_A let Y be in A^D and $z \in \Sigma$. We define $\delta_A(Y, z) := Y'$, where Y' is constructed in the following steps:

1. (a) <u>Move:</u> For every $a^\sigma \in Y$ and $b \in \Delta_A(a, z)$, add b^σ to Y'.
   (b) <u>Reduce:</u> For every $a^\sigma \in Y'$, change a^σ to $a^{\sigma'}$, where σ' is obtained from $\sigma = (s_0, ..., s_m)$ by replacing every s_j with $j < \Omega(a)$ by $\text{minL}(T_j^Y)$.
2. <u>Append:</u> For every $a^\sigma \in Y'$ and $\sigma = (s_0, ..., s_m)$, change a^σ to $a^{\sigma'}$, where $\sigma' = (s_0 0, ..., s_{k-2} 0, s_k 1, s_{k+2} 0, ..., s_m 0)$ if $\Omega(a) = k$ is even, and $\sigma' = (s_0 0, ..., s_m 0)$ if $\Omega(a) = k$ is odd.
3. <u>Resolve:</u> For any a^σ and a^τ in Y', where $\sigma <_c \tau$, remove a^σ.
4. <u>Compress/Colour:</u> Do for every $k = 0, 2, ..., m$: Let $c_k(t) = \text{white}$ for every $t \in T_k$. Now we compress and color T_k inductively in the following way, until there exists no *witness* $t \in T_k$ such that (a) or (b) is applicable:

[5] Recall that we let $\text{ran}(f) := \{f(x) \mid x \in X\}$ be the *range* of a function $f : X \to Y$.

(a) For any $t \in T_k$, such that $t0 \in T_k$ and $t1 \notin T_k$, change every $a^\sigma \in Y'$ with $\sigma = (s_0, ..., s_m)$ and $t0 \preccurlyeq s_k$ to $a^{\sigma'}$, where $\sigma' = (s_0, ..., s_k[t/t0], ..., s_m)$. For any $s \in T_k$, where $t \prec s$, let $c_k(s) = \text{red}$.

(b) For any $t \in T_k$, such that $t0 \notin T_k$, $t1 \in T_k$ and $t \neq 0 \cdots 0$, change every $a^\sigma \in Y'$, where $\sigma = (s_0, ...s_m)$, and $t1 \preccurlyeq s_k$, to $a^{\sigma'}$, where $\sigma' = (s_0, ..., s_k[t/t1], ..., s_m)$. For any $s \in T_k$ such that $t = s0 \cdots 0$, let $c_k(s) = \text{green}$, if $c_k(s) \neq \text{red}$. In particular, let $c_k(t) = \text{green}$ if $c_k(t) \neq \text{red}$. For any $s \in T_k$, where $t \prec s$, let $c_k(s) = \text{red}$.

The automaton $\mathbb{A}^D$ accepts a run if there is $k = 0, 2, ..., m$ and a binary string s, which is in play at priority k cofinitely often and such that $c_k(s)$ is green infinitely often and red only finitely often.

3.2.7. Theorem. *The automata $\mathbb{A}^D$ and $\mathbb{A}$ are equivalent.*

Proof:
This proof follows the same lines as the proof of Theorem 3.2.3. We will focus on the differences and omit some steps which are analogous to the case of Büchi automata. We show $\mathcal{L}(\mathbb{A}^D) = \mathcal{L}(\mathbb{A})$. Let $w = z_0 z_1 ... \in \Sigma^\omega$ and $\rho = Y_0 Y_1 Y_2 ...$ be the run of $\mathbb{A}^D$ on w.

"$\supseteq$": Suppose that $\mathbb{A}$ accepts a run $r = a_0 a_1 a_2 ..$ on w. Let $k = 0, ..., m$ be maximal such that $\Omega(a_j) = k$ for infinitely many $j \in \omega$. Because r is accepted, k is even. Let s_n be the binary string such that $a_n^{\sigma_n} \in Y_n$ and $s_n = \pi_k(\sigma_n)$ for $n \in \omega$. Let t be the maximal string which is a substring of cofinitely many s_n and such that $c_k(t) = \text{red}$ only finitely often. Note that $t \neq 0 \cdots 0$: Towards a contradiction assume that $t = 0 \cdots 0$. As r is accepted, an 1 is appended to s_n at some point after the last time that $c_k(t)$ is red. This 1 can never get deleted again, and $c_k(t1)$ is never red because this would imply that $c_k(t)$ is red as well. Thus $t1$ is also a substring of cofinitely many s_n and $c_k(t1) = \text{red}$ only finitely often.

By definition, t is in play at priority k cofinitely often and $c_k(t) = \text{red}$ only finitely often, we thus need to show that $c_k(t)$ is green infinitely often. Let t' be the maximal string of the form $t0 \cdots 0$, such that $t' \preccurlyeq s_n$ for cofinitely many n. Let $N \in \omega$ be such that for all $n \geq N$ it holds that $t' \preccurlyeq s_n$, $c_k(t) \neq \text{red}$ in Y_n and $\Omega(a_n) \leq k$.

The rest of the first direction can be proved analogously to the proof of Theorem 3.2.3.

"$\subseteq$": Conversely, suppose that there is $k = 0, 2, ..., m$ and a binary string t, which is in play at priority k in ρ cofinitely often and such that $c_k(t)$ is green infinitely often and red only finitely often. In particular, $t \neq 0 \cdots 0$. Let N be such that t is in play at priority k and $c_k(t) \neq \text{red}$ for all $i \geq N$. For $i \geq N$ we define

$$A_i = \{a^\sigma \in Y_i \mid t \preccurlyeq \pi_k(\sigma)\}.$$

We first show

For all $i \geq N$ and $a^\sigma \in A_i$ it holds $\Omega(a) \leq k$. (3.3)

For all $a^\sigma \in A_N$ there is a path from a_I to a in $\mathbb{A}$ on input $z_0 ... z_{N-1}$. (3.4)

For all $i \geq N$ and $b^{\sigma_b} \in A_{i+1}$ there exists $a^{\sigma_a} \in A_i$ such that $b \in \Delta(a, z_i)$ (3.5)

For (3.3) assume that $\Omega(a) > k$. Then σ_k is replaced by $\min\mathrm{L}(T_k)$ at step 1(b) of the transition function, which results in $\sigma_k = 0 \cdots 0$ at the end of the transition function. Yet $t \neq 0 \cdots 0$, hence $t \not\preccurlyeq \sigma_k$. Statement (3.4) follows, as the transition function is a refined version of a macro-move.

For (3.5) let $b^{\sigma_b} \in A_{i+1}$. Due to step 1 of the transition function there is $a^{\sigma_a} \in Y_i$ with $b \in \Delta(a, z_i)$. We choose a^{σ_a} with that property such that σ_a is maximal, let $s_a := \pi_k(\sigma_a)$ and claim that $t \preccurlyeq s_a$. We show that by analyzing $\delta(Y_i, z_i)$. Due to (3.3) the binary string s_a is not reduced in step 1(b) and after step 2 there is b^σ in Y_i', where $\pi_k(\sigma) = s_a0$ or $\pi_k(\sigma) = s_a1$. The rest of the transition function resembles the Büchi case and we can therefore, analogous to the proof of Theorem 3.2.3, show that $t \preccurlyeq s_a$.

Using König's Lemma we obtain a run $r = r' a_N a_{N+1}$ of $\mathbb{A}$ on input w analogously as in the proof of Theorem 3.2.3. For showing that r is successful it suffices that $\Omega(a_j) = k$ for infinitely many $j \geq N$, as $\Omega(a_j) \leq k$ for all $j \geq N$ due to (3.3). This can be done as in the proof of Theorem 3.2.3. □

3.2.8. LEMMA. *Let $\mathbb{A}$ be a parity automaton of size n and highest even priority m. The automaton $\mathbb{A}^D$ has $2^{\mathcal{O}(mn \log n)}$ BT-states and $\mathcal{O}(m \cdot 2^n)$ Rabin pairs.*

Proof:
We can alternatively represent a BT-state in $\mathbb{A}^D$ as a tuple of BT-states of the automata $(\mathbb{A} \cup \mathbb{A}_k)^D$ for $k = 0, 2, ..., m$, where $\mathbb{A} \cup \mathbb{A}_k$ is the Büchi automaton defined above. Hence Lemma 3.2.4 yields that $\mathbb{A}^D$ has $2^{\mathcal{O}(mn \log n)}$ possible BT-states. The number of Rabin pairs is the number of possible binary strings for $k = 0, 2, ..., m$, which is $\mathcal{O}(m \cdot 2^n)$. □

3.3 Safra construction for parity automata with ε-transitions

In this subsection we will introduce a different determinization method – the well-known Safra construction [Saf88]. This construction is usually studied for Büchi automata only, but here we generalize it to parity automata with ε-transitions. The motivation behind this generalization is its intended use in obtaining a proof system for the two-way modal μ-calculus in Chapter 5. We fix an alphabet Σ and an ε-parity automaton $\mathbb{A} = (A, \Delta, a_I, \Omega)$. Let m be the maximal even priority of Ω, let m' be the maximal priority of Ω and let $n = |A|$ be the size of $\mathbb{A}$.

Motivation There are two dual perspectives on the Safra construction. Usually, states of the deterministic automaton are taken to be so-called Safra-trees – certain trees, where nodes are labeled by subsets of A.

In the other perspective [Koz06] states of the deterministic automaton consist of subsets A_0 of A, where each state $a \in A_0$ is labeled by a stack. This stack corresponds to the address of a in the Safra-tree. We choose the latter perspective, as it better aligns with its intended use in the non-wellfounded proof theory of the two-way modal μ-calculus, see Chapter 5

One main difficulty to generalize the Safra-construction to parity automata with ε-transitions is to incorporate ε-moves into the transition function without finding ourselves in a loop that only consists of ε-transitions. Recall that the transition function consists of the basic transition function $\Delta_b : A \times \Sigma \to \mathcal{P}(A)$ and the ε-transition function $\Delta_\varepsilon : A \to \mathcal{P}(A)$. The idea is that, after every basic transition, we consider all states that might be reached by finitely many ε-transitions. The following definition captures these states, where we also pay attention to the priority of the passed states.

3.3.1. Definition. Let $a \in A$ and $k = 0, 1, ..., m'$. The *k-priority ε-closure* of a, written $\varepsilon\text{Clos}_k(a)$, consists of all states $b \in A$ for which there is a Δ_ε-path $a = a_0 a_1 \cdots a_N = b$ in $\mathbb{A}$ with $\max\{\Omega(a_i) \mid i = 1, \ldots, N\} = k$.

As in the Safra construction for Büchi automaton, we annotate states with sequences of names. Yet, in the context of parity automata, names will have *priorities* and states of priority k will be labeled by sequences of names of priority at least k. Whenever a state of even priority k is passed, a name of priority k is added and all names of priority less than k are removed. A name x will be marked successful, whenever all occurrences of x are covered by names of the same priority. As in the Büchi case a run of the deterministic automaton is successful if there is a name that is always present from some point on and that is successful infinitely often.

Construction For each even number $k = 0, 2, \ldots, m$ we fix a set of *k-names* X_k, such that $|X_k| = 4n$ and $X_k \cap X_l = \varnothing$ if $k \neq l$. We define the set of *names* $X := X_0 \uplus X_2 \uplus \cdots \uplus X_m$ and use the symbols $\mathsf{x}, \mathsf{y}, \mathsf{z}, ...$ for names in X. We call a non-repeating sequence of k-names τ_k a *k-stack* and let T_k be the set of all k-stacks. The empty sequence will be denoted by ε. We define the set of all stacks T to be $T_m \cdots T_2 \cdot T_0$, for clarity $T := \{\tau_m \cdots \tau_2 \cdot \tau_0 \mid \tau_m \in T_n, \ldots, \tau_0 \in T_0\}$. In case $\tau_i = \varepsilon$ for all $i < k$ we may write $\tau_m \cdots \tau_k$ rather than $\tau_m \cdots \tau_k \cdots \tau_0$. For a stack τ we define $\tau \downharpoonright l$ to be the stack obtained from τ by removing all k-names, where $k < l$.

Each non-repeating sequence of names θ defines a *linear order* $<_\theta$ on names by setting $\mathsf{x} <_\theta \mathsf{y}$ if x occurs before y in θ. This order extends to an order on stacks as follows: $\sigma <_\theta \tau$ if either

- $\sigma \downharpoonright k$ is a proper extension of $\tau \downharpoonright k$ for some $k \leq m$, or
- σ is lexicographically $<_\theta$-smaller than τ, meaning that σ and τ can be written as $\sigma = \rho \cdot \mathsf{x} \cdot \sigma'$ and $\tau = \rho \cdot \mathsf{y} \cdot \tau'$ with $\mathsf{x} <_\theta \mathsf{y}$.

3.3.2. Proposition. *Let $T_0 \subseteq T$ be a set of stacks and let θ be a non-repeating sequence of names containing all names in T_0. Then $<_\theta$ is a linear order on T_0.*

We now define the deterministic Rabin automaton $\mathbb{A}^S := (A^S, \delta_A, a'_I, R_A)$. Its carrier set A^S consists of all tuples $Y = (A_Y, f, \theta, c)$, where

- A_Y is a subset of A,
- $f : A_Y \to T$ maps each state $a \in A_Y$ to a stack $\tau \in T$, such that $\tau = \tau \downharpoonright k$, where $k = \Omega(a)$,
- θ is a non-repeating sequence of all names occurring in ran(f).
- c is a map from ran$(f) \to \{\text{green}, \text{white}\}$.

We call a subset $A_Y \subseteq A$ a *macrostate* and call $Y \in A^S$ a *Safra-state*, meaning that a Safra-state is a macrostate with extra information. We will present a Safra-state $S_0 \in A^S$ by a set of pairs (a, τ), usually written as a^τ, where $a \in A_0$, $\tau \in T$ and $f(a) = \tau$, and deal with θ and c implicitly. The sequence θ will be called the *control*. We say a name is *active* if it appears in θ. An active k-name is *visible* if it is the last k-name in some stack and *invisible* otherwise. The function c is called the *coloring map* and we say that a name x is colored green/white, if $c(\mathsf{x}) = \text{green}/c(\mathsf{x}) = \text{white}$.

The initial Safra-state is $a'_I := \{a_I^\varepsilon\}$. To define the transition function δ_A let Y be in A^S and $z \in \Sigma$. We define $\delta_A(Y, z) := Y'$, where Y' is constructed in the following steps. Note that intermediate positions in this construction are not necessarily Safra-states; in particular there may be multiple stacks associated with some states.

1. Basic move: For every $a^\tau \in Y$ and $b \in \Delta_b(a, z)$, add $b^{\tau \downharpoonright k}$ to Y', where $\Omega(b) = k$.
2. Cover: For every $a^\tau \in Y'$, where $\Omega(a) = k$ is even, change a^τ to $a^{\tau \cdot \mathsf{x}}$, where x is a fresh k-name that is not active in $Y \cup Y'$. If two different states are labeled by the same stack, we add the same name x. Add x as the last element in θ.
3. ε-Move: For every $a^\tau \in Y'$, odd k and $b \in \varepsilon\text{Clos}_k(a)$, add $b^{\tau \downharpoonright k}$ to Y'. For every $a^\tau \in Y'$, even k and $b \in \varepsilon\text{Clos}_k(a)$, add $b^{\tau \downharpoonright k \cdot \mathsf{x}}$ to Y', where x is a fresh k-name that is not active in $Y \cup Y'$. If two different states are labeled by the same stack, we add the same name x. Add x as the last element in θ.

4. Thin: For any a^σ and a^τ in Y', where $\sigma <_\theta \tau$, remove a^τ.

5. Reset: Colour any invisible name x green and change $a^{\sigma\cdot\mathsf{x}\cdot\tau}$ to $a^{\sigma\cdot\mathsf{x}}$ for every $a^{\sigma\cdot\mathsf{x}\cdot\tau} \in Y'$.

Any name removed in this process is also removed from θ.

The automaton $\mathbb{A}^S$ accepts a run if some name x is active cofinitely often and colored green infinitely often.

3.3.3. Remark. Let Y be a Safra-state in A^S, then A_Y has size at most n. Thus there are at most n active k-names in Y for every $k = 0, 2, ..., m$ – otherwise there would be an invisible k-name inducing the Safra-state to change in step 5 of δ_A. In step 2 of δ_A at most n fresh k-names are introduced, resulting in at most $2n$ many distinct stacks after step 2. In step 3 up to $2n$ names are added. Thus in total at most $4n$ many k-names are needed for each $k = 0, 2, ...m$.

3.3.4. Remark. The map δ_A is formulated in a seemingly non-deterministic way, but this is only superficially so: all choices can be made *canonical*, based on arbitrary but fixed orders $<_A$ on states and $<_X$ on names in X as follows: Note that the order $<_X$ induces an order $<_T$ on the set of stacks T. Whenever all $a^\tau \in Y'$ are transformed, start with the smallest element according to the lexicographic order of $<_A$ and $<_T$. Whenever a fresh k-name is added, add the $<_X$-smallest such name and in step 5 also treat names according to $<_X$.

Importantly, the particular choices of $<_A$ and $<_X$ do not matter. More precisely, two automata $\mathbb{A}^S_1$ and $\mathbb{A}^S_2$ based on the orders $<^1_A, <^1_X$ and $<^2_A, <^2_X$, respectively, are "isomorphic" in the following sense: There is a bijection $g : X \to X$ such that for every given word w and runs $Y_0Y_1...$ of $\mathbb{A}^S_1$ and $Z_0Z_1...$ of $\mathbb{A}^S_2$ on w, the Safra-state Z_j is obtained from Y_j by replacing every name x with $g(x)$ for every $j \in \omega$. In particular, $\mathbb{A}^S_1$ accepts w iff $\mathbb{A}^S_2$ accepts w.

3.3.5. Theorem. *The automata $\mathbb{A}^S$ and $\mathbb{A}$ are equivalent.*

Proof:
We need to show that $\mathcal{L}(\mathbb{A}^S) = \mathcal{L}(\mathbb{A})$. We fix a word $w = z_0z_1... \in \Sigma^\omega$ and let $\rho = Y_0Y_1...$ be the unique run of $\mathbb{A}^S$ on w.

"$\supseteq$": Let $r = (a_0, n_0)(a_1, n_1)...$ be an extended run of $\mathbb{A}$ on w such that $\mathbb{A}$ accepts the run $a_0a_1....$ We want to show that $\mathbb{A}^S$ accepts the run ρ on w.

We define a sequence of natural numbers $m(0) < m(1) < \cdots$ such that $m(j) = \max\{i \mid n_i = j\}$ for $j \geq 0$. Intuitively, $m(j)$ is the last index in the run r such that j-many basic transitions were applied. In other words, at index $m(j)$ in the run r the $j+1$-th basic transition is applied.

`Claim 1`: For every $j \in \omega$ there is a unique stack τ_j such that $a^{\tau_j}_{m(j)}$ is in the Safra-state Y_j.

`Proof of Claim 1`: By induction on j. It holds that $a_0 = a_I$ and $m(j) = 0$, as we assume that $\Delta_\varepsilon(a_I) = \varnothing$. By definition $Y_0 = \{a_I^\varepsilon\}$.

Now assume that $a_{m(j)}^{\tau_j} \in Y_j$. After step 1 of the transition function we find $a_{m(j)+1}^{\tau_j} \in Y_j'$. In the extended run r between $(a_{m(j)+1}, j)$ and $(a_{m(j+1)}, j)$ all transitions are ε-transitions. Therefore, after step 3 of the transition function $a_{m(j+1)}^{\tau'} \in Y_j'$ for some τ'. After that, elements are removed such that we end up with a unique τ_{j+1} with $a_{m(j+1)}^{\tau_{j+1}} \in Y_{j+1}$. ⊣

We will now analyze the sequence $(\tau_j)_{j\in\omega}$. Let $h := \liminf |\tau_j|$, that is, h is the maximal number such that cofinitely many τ_j have size at least h. Let J_0 be such that $|\tau_j| \geq h$ for all $j \geq J_0$. For $0 \leq l \leq h$ we let $\tau[l]$ denote the stack consisting of the first l names in τ. We say that $\tau_j[l]$ is constant for $j \geq J$ if for all $i, j \geq J$ it holds that $\tau_i[l] = \tau_j[l]$.

<u>`Claim 2`</u>: There exists $J \in \omega$ such that $\tau_j[h]$ is constant for $j \geq J$.

`Proof of Claim 2`: By induction on l we prove that there exist $J_l \geq J_0$ such that $\tau_j[l]$ is constant for $j \geq J_l$ for all $0 \leq l \leq h$. For $l = 0$ this is trivial. Now assume that it holds for $l < h$. For simpler notation we write $g := J_l$ and let x and σ_g be such that $\tau_g = \tau_g[l] \cdot \mathsf{x} \cdot \sigma_g$. Let θ_j denote the control in the Safra-state Y_j. The only way that $\tau_j[l+1]$ might change for $j \geq J_l$ is in step 4 of the transition function, if $\tau_j = \tau_j[l] \cdot \mathsf{y} \cdot \sigma_j$ with $\mathsf{y} <_{\theta_j} \mathsf{x}$. As every newly introduced name is added as the last element in θ this implies that already $\mathsf{y} <_{\theta_g} \mathsf{x}$. If $\tau_i[l+1]$ changes again, then there is $\mathsf{z} <_{\theta_i} \mathsf{y}$, which already implies $\mathsf{z} <_{\theta_g} \mathsf{y}$ and so on. As there are only finitely many names below x in $<_{\theta_g}$ the stack $\tau_j[l+1]$ can only change finitely many times for $j \geq J_l$ and thus for some $J_{l+1} \geq J_l$ it must hold that $\tau_j[l+1]$ is constant for $j \geq J_{l+1}$. ⊣

Let $J \in \omega$ be as given in Claim 2 and let x be the h-th name in τ_J. For $j \geq J$ the name x is always active. We want to show that x is colored green infinitely often. $\mathbb{A}$ accepts the extended run r, thus there is an even k such that $\Omega(a_j) = k$ for infinitely many j and $\Omega(a_j) \leq k$ for all $j \geq T$ for some $T \in \omega$. We may assume that J is picked big enough such that $J \geq m(T)$. Therefore, for some $j \geq J$ a k-name y is added to the stack τ_j. But we have $|\tau_i| = h$ for some $i \geq j$, and this can only happen in step 4 of the transition function if x was invisible. This implies that x is colored green. Note that then x is a k-name as well. Repeating this argument yields that x is colored green infinitely often in ρ.

"$\subseteq$": Assume that $\mathbb{A}^S$ accepts the run ρ on w. Let x be a k-name that is active cofinitely often and colored green infinitely often. Let $t(0) < t(1) < \cdots$ be the minimal indices such that x is in play in Y_j for every $j \geq t(0)$ and such that x is colored green in $Y_{t(i)}$ for every $i \in \omega$.

For $j \in \omega$ let $Y_{t(j)} = (A_j, f_j, \theta_j, c_j)$. For $p, q \in \omega$ let $w[p, q)$ denote the segment $z_p ... z_{q-1}$ of the infinite word $w = z_0 z_1 ....$ In particular $w = w[0, t_0) \cdot w[t_0, t_1) \cdots$.

An $z_0...z_k$*-labeled path* in $\mathbb{A}$ is a partial run $a_0...a_{k+1}$ of $\mathbb{A}$ on input $z_0...z_n$ starting at state a_0. Our goal is to find certain $[t(j), t(j+1))$-labeled paths in $\mathbb{A}$ which can be composed to an infinite successful run of $\mathbb{A}$ on the word w.

For $j \in \omega$ let B_j be the set of states in the macrostate A_j which contain x in their stack. Formally, $B_j := \{b \in A_j \mid \mathsf{x} \text{ occurs in } f_j(b)\}$.

`Claim 3`: For every $a \in B_0$ there is an $w[0, t_0)$-labeled path from a_I to a.

`Proof of Claim 3`: For all $i = 0, \ldots, t(0)$ let $C_i \subseteq A$ be the macrostate in Y_i. For all $b \in C_{i+1}$ there is $a \in C_i$ such that there exists $c \in A$ with $\Delta_b(a, z_i) = c$ and $b \in \varepsilon\text{Clos}(c)$. This follows from the definition of step 1 and 3 of the transition function. The other steps only manipulate stacks but do not change macrostates. The claim then follows by induction. ⊣

`Claim 4`: For all $j > 0$ and all $b \in B_{j+1}$ there is a state $a \in B_j$ and a $w[t_j, t_{j+1})$-labeled path $c_0 \cdots c_h$ with $a = c_0$, $b = c_h$ and $\max\{\Omega(c_j) \mid i = 1, \ldots, h\} = k$.

`Proof of Claim 4`: As in the proof of Claim 3 we can show that there is $a \in A_j$ and a $w[t_j, t_{j+1})$-labeled path $c_0 \cdots c_h$ with $a = c_0$ and $b = c_h$. Because x is in play in Y_j for all $j \geq t_0$ the name x can never be introduced in the transition function. Thus we may conclude that x was already present in the stack τ_j of a in $Y_{t(j)}$, meaning that $a \in B_j$. It remains to show that there is such a path where $\max\{\Omega(c_j) \mid i = 1, ..., h\} = k$. In $Y_{t(j)}$ the name x is visible in all stacks, where x occurs. In $Y_{t(j+1)}$ the name x is colored green, indicating that after step 4 of the transition function in $Y'_{t(j+1)-1}$ the name x is invisible. This can only happen if a k-name y was added to the stack τ_j in $Y_{t(j)+1}...Y_{t(j+1)}$ in step 2 or 3 of the transition function. But then $\Omega(c_j) = k$ for some $j = 1, ..., h$. As x is always in play we also have $\Omega(c_j) \leq k$ for all $j = 1, ..., h$ and thus $\max\{\Omega(c_j) \mid i = 1, ..., h\} = k$. ⊣

We will now glue those paths together to obtain an infinite path through $\mathbb{A}$. This can be achieved using König's Lemma. Let $G := (V, E)$ where

$$\begin{aligned} V :=& \{a_I\} \cup \{(a, j) \mid j \in \omega \text{ and } a \in B_j\}, \\ E :=& \{(a_I, (a, 0)) \mid a \in B_0\} \cup \\ & \{((a, j), (b, j+1)) \mid b \in B_{j+1} \text{ and } a \in B_j \text{ as in Claim 4}\} \end{aligned}$$

Clearly, G is a connected, finitely branching and infinite graph. Hence we can apply König's Lemma to obtain an w-labeled path $r = a_0 a_1 \cdots$ in $\mathbb{A}$, where $\Omega(a_j) \leq k$ for cofinitely many $j \in \omega$ and $\Omega(a_j) = k$ for infinitely many $j \in \omega$. In particular we find $r \in \text{Acc}$. □

3.3.6. Lemma. *Let $\mathbb{A}$ be a parity automaton with ε-transitions of size n and highest even priority m. The automaton $\mathbb{A}^S$ has $2^{\mathcal{O}(mn \log m \log n)}$ Safra-states and $\mathcal{O}(mn)$ Rabin pairs.*

Proof:
There are 2^n many subsets A_Y of A. Whenever a name x in the construction is introduced, it is fresh and added as the last element of a unique stack. Therefore, a name x either appears as the first element of any stack in which it occurs or there is a unique name y to the left of x in any such stack. We can thus represent the function $f : A_Y \to T$ as two maps $f_I : A_Y \to X$ and $f_S : X \to X \cup \{\text{nil}\}$. Hence, there are at most $(2mn)^n \cdot (2mn)^{2mn}$ such functions. The control θ is a non-repeating sequence of up to mn names, hence there are at most $(mn)!$ such controls. Lastly, there are at most 2^{mn} coloring maps c. In total that amounts to

$$2^n \cdot (2mn)^n \cdot (2mn)^{2mn} \cdot (mn)! \cdot 2^{mn} = 2^{\mathcal{O}(mn \log m \log n)}$$

many Safra-states. The number of Rabin pairs is the number of names, namely $2mn$. □

3.3.7. Remark. A closer inspection on the construction reveals that the control θ is only needed to order names *of the same priority*. Hence, we could replace θ by non-repeating sequences $\theta_0, ..., \theta_m$, where θ_k is a non-repeating sequence of all k-names occurring in $\text{ran}(f)$ for $k = 0, ..., m$. This change would slightly reduce the number of Safra-states to $2^{\mathcal{O}(mn \log n)}$. The above complexity also coincides with the complexity of the binary tree construction for parity automata without ε-transitions, see Lemma 3.2.8. However, we did not opt for this construction in order to simplify the presentation.

Chapter 4

Cyclic proof systems for the modal μ-calculus

Given the importance of the modal μ-calculus, there is a natural interest in the development and study of derivation systems for its validities. And indeed, already in [Koz83] Kozen proposed a Hilbert-style axiomatization. A sequent-style reformulation Koz of this axiomatization is defined as the *finitary* proof system obtained by adding the rules ind and cut to the rules of NW defined in Figure 2.5.

$$\mathsf{ind}: \frac{\varphi[\bigwedge \overline{\Gamma}/x], \Gamma}{\nu x.\varphi, \Gamma} \qquad \mathsf{cut}: \frac{\Gamma, \varphi \qquad \overline{\varphi}, \Gamma}{\Gamma}$$

Figure 4.1: Additional rules of Koz

Despite the naturality of this axiom system, Kozen only established a partial completeness result, and it took a substantial amount of time before Walukiewicz [Wal00] managed to prove soundness and completeness for the full language.

Even though Kozen's axiomatization is finitary it has a major drawback. Both rules ind and cut are *not analytic*: formulas in a premise of the rule might be outside of the closure of its conclusion. This makes proof search unfeasible, resulting in a less attractive system from a proof-theoretic perspective. For this reason, different proof systems for $\mathcal{L}_\mu$ have been developed. Special emphasis has been placed on finding cut-free and analytic proof systems for $\mathcal{L}_\mu$.

Infinitely branching systems One attempt to obtain analytic proof systems is infinitely branching ones, which allow rules with infinitely many premises. The first system of this kind was introduced by Kozen [Koz88], who used the finite model theorem of $\mathcal{L}_\mu$ to show completeness. Notably, Kozen's system contains cut. Later, Jäger, Kretz and Studer [JKS08] refined this system to a cut-free system JKS. Let $(\nu x)^0\varphi := \varphi[\top/x]$ and $(\nu x)^{n+1}\varphi := \varphi[(\nu x)^n\varphi/x]$. The system JKS is defined as the system obtained by adding the following rule to the rules of NW and requiring all branches to be finite.

$$\nu^\omega\colon \frac{\cdots \quad (\nu x)^n\varphi, \Gamma \quad \cdots \quad \text{for all } n \in \omega}{\nu x.\varphi, \Gamma}$$

Figure 4.2: Infinitely branching rule in JKS

Completeness of this system is proved by a canonical counter-model construction. It is also noted that the rule ν^ω might be "finitized" to a rule ν^n, which only consists of the first n premises: Relying on the small model theorem an exponential function f can be defined such that in a proof of a sequent Γ all occurrences of ν^ω might be replaced by ν^n, where $n := f(|\mathsf{Clos}(\Gamma)|)$. This gives a finitary, cut-free proof system for $\mathcal{L}_\mu$. Yet, as the number of premises in ν^n rules depend on the size of the end-sequent, the system is not well-suited for proof search and one might argue how natural it is.

Infinitary systems In infinitary systems, all rules are finitely branching, but branches may be infinitely long. Niwiński and Walukiewicz [NW96] introduced the first such system as a two-player tableau-style game. We presented their system in the form of the infinitary proof system NW in Chapter 2. NW is cut-free and analytic, yet, as a trade-off, one has to distinguish between successful and unsuccessful branches, where an infinite branch is *successful* if it carries a ν-trace: a trace which is dominated by a *greatest* fixpoint operator.

This condition is easy to formulate but not so nice to work with. One could describe the subsequent developments in non-wellfounded proof theory for the modal μ-calculus as a series of modifications of the system NW which aim at getting a grip on the complexities and intricacies of the above-mentioned traces, and in particular, to use the resulting "trace management" for the introduction of cyclic proof systems.

Cyclic systems As NW-proofs may assumed to be regular, clearly one can define a cyclic proof system from the rules of NW. The resulting global soundness condition of such cyclic NW-proofs states that every infinite path through a proof must be successful – meaning that a cyclic proof is nothing more than a finitary representation of an infinitary NW-proof. The complicated global soundness condition makes the proof system impractical for theoretical and practical considerations. In fact, checking if a given NW-derivation is a cyclic proof is PSPACE-complete [Nol21].

In the search for cyclic proof systems for $\mathcal{L}_\mu$ with an easier-to-check soundness condition, the study of ω-automata turned out to be extremely valuable. Already

Niwiński and Walukiewicz [NW96] observed that infinite matches of their game, corresponding to infinite branches in an NW derivation, can be seen as infinite words over some finite alphabet. It follows that ω-automata can be used to determine whether such a match/branch carries a ν-trace. Niwiński and Walukiewicz used this perspective to link their results to the exponential-time complexity of the satisfiability problem for the μ-calculus.

The natural choice for an ω-automaton checking whether an infinite NW-branch carries a ν-trace is a non-deterministic parity automaton $\mathbb{A}_\mu$ – called the *tracking automaton* for NW-proofs. A key contribution of Walukiewicz [Wal93] was to bring the tracking automaton into the syntax of the proof system. The basic idea would be to decorate each sequent in a derivation with a state of the tracking automaton; starting from the root, the successive states decorating the sequents on a given branch simply correspond to a run of the automaton on this branch. For this idea to work, one needs the stream automaton to be *deterministic*. To see this, observe that two successful but distinct branches in a derivation would generally require two distinct runs, and in the case of a nondeterministic automaton, these two runs might already diverge before the two branches split. As the natural choice for the tracking automaton is non-deterministic, one thus first needs to determinize it. This explains the relevance of determinization constructions in the proof theory of the modal μ-calculus.

Walukiewicz used the Safra construction to determinize the tracking automaton; by doing so, he managed to translate NW-proofs into a finitary system with a different induction rule from Kozen's, and proved that this system is complete. Jungteerapanich [Jun10] and Stirling [Sti14] went one step further and studied the proof system obtained by building the determinization of $\mathbb{A}_\mu$ into the syntax. This results in path-based cyclic proof systems – we will identify their calculi under the name JS. Afshari and Leigh [AL16] further modified the proof system JS to obtain a cyclic proof system with a simplified discharge condition.

In [LW24], Leigh and Wehr take a rather general approach, working with abstract proof systems and examining how determinization constructions can be used in the design of derivation systems. Their analysis is confined to the Safra construction.

Our contributions The approach taken by Walukiewicz, Jungteerapanich and Stirling is restricted to the proof system NW and the specific deterministic automaton checking the trace condition. We generalize both.

Their proof systems were inspired by the Safra construction – the most well-known determinization construction. We define a uniform construction that, given any deterministic automaton $\mathbb{A}$ checking the global soundness condition of NW-derivations, yields a sound and complete proof system $\mathsf{NW}^{\mathbb{A}}$ for $\mathcal{L}_\mu$. The sequents of $\mathsf{NW}^{\mathbb{A}}$ are the states of $\mathbb{A}$ and the soundness condition on infinite paths coincides with the acceptance condition of $\mathbb{A}$. This condition is usually much simpler than the soundness condition of NW. Our point is that distinct determinization constructions lead to distinct sequent systems, and that we may look for alternatives to the Safra construction.

We then proceed to apply our construction to the deterministic automaton $\mathbb{A}_\mu^D$ – the automaton obtained by applying the binary tree construction from Section 3.2.3 to the tracking automaton $\mathbb{A}_\mu$. Up to minor syntactic changes this will result in the infinitary proof system BT^∞. From the infinitary system we can then obtain the cyclic proof system BT by a standard procedure. The system BT is subgraph-based, which means that to determine whether a BT derivation constitutes a valid proof, one needs to examine all strongly connected subgraphs of the proof tree.

Our construction is not only uniform with respect to different automata but also with respect to different proof systems. This makes the construction also applicable to other proof systems. We will use the construction in Chapter 5 to obtain a proof system for the two-way modal μ-calculus.

Moreover, we make the use of automata theory *explicit*. While Jungteerapanich and Stirling took the Safra construction merely as an inspiration, we directly work with the deterministic automaton and utilize the correctness of the determinization method in the soundness and completeness proof.

In the last section of this chapter we consider a different cyclic proof system for the modal μ-calculus: The proof system Clo introduced by Afshari and Leigh [AL17]. Our main result is the incompleteness of Clo. This is shown by giving a valid sequent that is not provable in Clo.

4.1 Using deterministic ω-automata to obtain proof systems

4.1.1 A Uniform construction

For this subsection we fix an arbitrary infinitary proof system $\mathcal{P} = (\mathcal{D}, \mathcal{G})$ over a set of sequents $\mathcal{S}$. We show how one can use a deterministic ω-automaton recognizing the global soundness condition $\mathcal{G}$ to obtain a different infinitary proof system with an easier condition on infinite branches proving the same sequents. Recall that a derivation system $\mathcal{D}$ is a set of rules and $\mathcal{G}$ is a class of infinite $\mathcal{D}$-paths.

First, we introduce the alphabet $\Sigma := \mathcal{S} \times \mathcal{D} \times \mathcal{S}$. Let $\gamma = (v_n)_{n\in\omega}$ be an infinite branch in a $\mathcal{P}$-proof π. We define $w(\gamma) \in \Sigma^\omega$ to be the infinite word $(\mathsf{S}(v_0), \mathsf{R}(v_0), \mathsf{S}(v_0))(\mathsf{S}(v_0), \mathsf{R}(v_0), \mathsf{S}(v_1))(\mathsf{S}(v_1), \mathsf{R}(v_1), \mathsf{S}(v_2))\ldots$. The reason why we repeat $\mathsf{S}(v_0)$ and $\mathsf{R}(v_0)$ is to better fit runs of an automaton starting at the same initial state with proof branches starting from any sequent.

4.1.1. Definition. Let $\mathcal{P} = (\mathcal{D}, \mathcal{G})$ be an infinitary proof system over a set of sequents $\mathcal{S}$. Let $\mathbb{A} = (A, \delta, a_I, \mathrm{Acc})$ be a deterministic ω-automaton over the alphabet Σ such that an infinite $\mathcal{P}$-path γ is in $\mathcal{G}$ iff $w(\gamma) \in \mathcal{L}(\mathbb{A})$.

Let $f : A \to \mathcal{S}$ be surjective and $g : \mathcal{S} \to A$ be injective such that $f(g(\Gamma)) = \Gamma$ and such that $\delta(a_I, (\Gamma, \mathsf{R}, \Gamma')) = g(\Gamma')$ for all sequents Γ and Γ'.

We define the infinitary proof system $\mathcal{P}^{\mathbb{A}} := (\mathcal{D}^{\mathbb{A}}, \mathcal{G}^{\mathbb{A}})$ over the set of sequents A as follows. For any rule

$$\mathsf{R}\colon \frac{\Gamma_1 \quad \cdots \quad \Gamma_n}{\Gamma_0}$$

in $\mathcal{D}$, let the following rule be in $\mathcal{D}^{\mathbb{A}}$:

$$\mathsf{R}^{\mathbb{A}}\colon \frac{a_1 \quad \cdots \quad a_n}{a_0}$$

where $\delta(a_0, (\Gamma_0, \mathsf{R}, \Gamma_i)) = a_i$ for $i = 1, ..., n$ and $f(a_i) = \Gamma_i$ for $i = 0, 1, ..., n$.

The global soundness condition $\mathcal{G}^{\mathbb{A}}$ is defined as the class of all infinite $\mathcal{D}^{\mathbb{A}}$-paths γ, where $a_I\gamma$ is in $\mathrm{Acc} \subseteq A^\omega$.

The automaton $\mathbb{A}$ checks the global soundness condition of $\mathcal{P}$, where the functions f and g provide a correspondence between the sequents of $\mathcal{P}$ and of $\mathcal{P}^{\mathbb{A}}$. The aim of this definition is to show that the proof systems $\mathcal{P}$ and $\mathcal{P}^{\mathbb{A}}$ prove the same sequents – up to the correspondence given by g.

4.1.2. Lemma. *For any sequent $\Gamma \in \mathcal{S}$ there is a $\mathcal{P}$-proof π of Γ iff there is a $\mathcal{P}^{\mathbb{A}}$-proof ρ of $g(\Gamma)$. The proof ρ is regular iff π is so.*

Proof:
“ $\Rightarrow$ ”: Let π be a $\mathcal{P}$-proof of Γ. We define a $\mathcal{P}^{\mathbb{A}}$-proof ρ and a bijection $h : \pi \to \rho$ inductively. Let r^π be the root of π. We let the root r^ρ of ρ be labeled with $g(\Gamma)$ and define $h(r^\pi) = r^\rho$. By definition $f(g(\Gamma)) = \Gamma$

Let u in π be labeled with R and Γ_0 with children $u_1, ..., u_n$ labeled with the respective sequents $\Gamma_1, ..., \Gamma_n$. If $h(u) = v$ is labeled with a_0, then v is labeled with the rule $\mathsf{R}^{\mathbb{A}}$ and we let v have n children $a_1, ..., a_n$ labeled with the respective sequents $a_1, ..., a_n$, where $\delta(a_0, (\Gamma_0, R, \Gamma_i)) = a_i$ for $i = 1, ..., n$. By induction hypothesis $f(a_0) = \Gamma_0$ and therefore we obtain that $f(a_i) = \Gamma_i$ for $i = 1, \ldots, n$ as well. We define $h(u_i) = v_i$ for $i = 1, ..., n$.

This results in a $\mathcal{D}^{\mathbb{A}}$-derivation ρ of $g(\Gamma)$. It remains to show that every infinite branch in ρ is successful. Let $\gamma = a_0 a_1 ...$ be such an infinite branch in ρ. The bijection $h : \pi \to \rho$ extends to a bijection from paths in π to paths in ρ. Let $\beta = \Gamma_0\Gamma_1 ...$ be the corresponding branch in π obtained by applying h^{-1} to γ.

By assumption $w(\beta) \in \mathcal{L}(\mathbb{A})$, where it follows from the definition that $w(\beta) = (\Gamma_0, R_0, \Gamma_0)(\Gamma_0, R_0, \Gamma_1)(\Gamma_1, R_1, \Gamma_2)....$ We take a look at the run r of $\mathbb{A}$ on $w(\beta)$ and claim that $r = a_I\gamma = a_I a_0 a_1 ....$ Clearly, the first state of r is a_I and, by definition of g, it holds that $\delta(a_I, (\Gamma_0, R_0, \Gamma_0)) = g(\Gamma_0) = a_0$. The claim then follows by induction, as $\delta(a_m, (\Gamma_m, R_m, \Gamma_{m+1})) = a_{m+1}$ for $m \in \omega$. Therefore $a_I\gamma \in \text{Acc}$ and thence $\gamma \in \mathcal{G}^{\mathbb{A}}$.

" $\Rightarrow$ ": Conversely, let ρ be a $\mathcal{P}^{\mathbb{A}}$-proof of $g(\Gamma)$. We obtain a $\mathcal{P}$-proof π of Γ by translating every node v in ρ labeled with a and $\mathsf{R}^{\mathbb{A}}$ to a node u in π labeled with $f(a)$ and R. In particular, the root of π is labeled with $f(g(\Gamma)) = \Gamma$. This results in a $\mathcal{P}$ derivation of Γ; it remains to show that $\mathcal{P}$ satisfies the global soundness condition.

Let $\beta = \Gamma_0\Gamma_1...$ be an infinite branch in π, we need to show that $\beta \in \mathcal{G}$. Let $\gamma = a_0a_1...$ be the corresponding branch in ρ. Note that $\Gamma_i = f(a_i)$ for all $i \in \omega$ and that by assumption $a_I\gamma \in \text{Acc}$. We claim that the run r of $\mathbb{A}$ on $w(\beta)$ is $a_I\gamma$.

It holds that $w(\beta) = (\Gamma_0, R_0, \Gamma_0)(\Gamma_0, R_0, \Gamma_1)(\Gamma_1, R_1, \Gamma_2)...$, where $\Gamma_0 = \Gamma$. The first state of r is a_I and $\delta(a_I, (\Gamma, R_0, \Gamma)) = g(\Gamma) = a_0$. Again our claim follows by induction using the equality $\delta(a_m, (\Gamma_m, R_m, \Gamma_{m+1})) = a_{m+1}$ for $m \in \omega$. Thus $\mathbb{A}$ accepts the run r and therefore $\beta \in \mathcal{G}$.

In both directions regular proofs are translated to regular proofs. We therefore have shown the lemma. □

4.1.2 Tracking automaton

In the previous subsection, we constructed a method transforming an *abstract* infinitary proof system to a different proof system with an easier condition on infinite paths. We now aim to apply this method to the *concrete* proof system NW. An infinite path in NW is successful if it carries a ν-trace – an infinite ancestry path of formulas, where the most important formula is a ν-formula. This condition is hard to work with, as traces on an infinite path behave non-deterministically: they might split and merge.

Yet, we can define an ω-automaton $\mathbb{A}_\mu$ that checks whether such an infinite path carries a ν-trace. The natural choice for this automaton is a non-deterministic parity automaton that follows all possible traces – we will define the tracking automaton in this subsection. We then use the binary tree construction from Chapter 3, which yields a deterministic automaton $\mathbb{A}_\mu^D$. This automaton satisfies the conditions of Lemma 4.1.2 and we obtain a new proof system with an easier condition on infinite paths.

For the rest of this section, sequents are *sets* of $\mathcal{L}_\mu$-formulas. As in Section 2.7, we assume that the set of actions Act is a singleton and we denote modalities by $\Box$ and $\Diamond$. We fix a sequent Φ such that $\mathsf{Clos}(\Phi) = \Phi$. For a proof of Γ one may define Φ as $\mathsf{Clos}(\Gamma)$. Let Ω_μ be the priority function on $\mathcal{L}_\mu$ defined in Section 2.3.

We will define a nondeterministic parity automaton that checks if there is a ν-trace on a given infinite path γ in an NW-derivation. Recall that rules in NW are pairs consisting of the name of the rule and its principal formula.

4.1.3. Definition. The alphabet Σ consists of all triples $(\Gamma, \mathsf{R}, \Gamma')$, where $\Gamma \subseteq \Phi$ is the conclusion and $\Gamma' \subseteq \Phi$ is the premise of a rule R of NW in Figure 2.5.

We define the following nondeterministic parity automaton $\mathbb{A}_\mu := (A, \Delta, a_I, \Omega_A)$ over Σ:

- $A := a_I \cup \mathsf{Clos}(\Phi) \cup \{\eta x.\psi^* \mid \eta x.\psi \in \mathsf{Clos}(\Phi)\}$,
- For each $\chi \in A$ and $(\Gamma, \mathsf{R}, \Gamma') \in \Sigma$:
  1. if $\chi = a_I$, then $\Delta(\chi, (\Gamma, \mathsf{R}, \Gamma')) := \Gamma'$,
  2. if $\chi = \eta x.\psi$ is the principal formula of R, then $\Delta(\chi, (\Gamma, \mathsf{R}, \Gamma')) := \{\eta x.\psi^*\}$,
  3. if $\chi = \eta x.\psi^*$, then $\Delta(\chi, (\Gamma, \mathsf{R}, \Gamma')) := \{\chi' \mid (\psi[\eta x.\psi/x], \chi') \in \mathsf{T}_{\Gamma,\mathsf{R},\Gamma'}\}$,
  4. else $\Delta(\chi, (\Gamma, \mathsf{R}, \Gamma')) := \{\chi' \mid (\chi, \chi') \in \mathsf{T}_{\Gamma,\mathsf{R},\Gamma'}\}$.
- For all states of the form $\eta x.\psi^*$ let $\Omega_A(\eta x.\psi^*) := \Omega_\mu(\eta x.\psi)$. For all other states χ let $\Omega_A(\chi) := 1$.

We call $\mathbb{A}_\mu$ the *tracking automaton* for NW.

Let $\gamma = (v_n)_{n\in\omega}$ be an infinite branch in an NW-proof π. We define $w(\gamma) \in \Sigma^\omega$ to be the infinite word $(\mathsf{S}(v_0), \mathsf{R}(v_0), \mathsf{S}(v_0))(\mathsf{S}(v_0), \mathsf{R}(v_0), \mathsf{S}(v_1))(\mathsf{S}(v_1), \mathsf{R}(v_1), \mathsf{S}(v_2))\ldots$.

4.1.4. Lemma. *Let γ be an infinite branch in an NW-derivation. Then γ carries a ν-trace iff $w(\gamma) \in \mathcal{L}(\mathbb{A}_\mu)$.*

Proof:
The automaton $\mathbb{A}_\mu$ tracks all traces on γ. A run r is accepted iff the state of maximal priority occurring infinitely often in r is of the form $\nu x.\psi^*$. Therefore, $\mathbb{A}_\mu$ accepts $w(\gamma)$ iff there is a ν-trace on γ. □

We aim to apply Lemma 4.1.2 to the infinitary proof system NW using the deterministic Rabin automaton $\mathbb{A}_\mu^D$. To do so we define the infinitary proof system $\mathsf{NW}^D := \mathsf{NW}^{\mathbb{A}_\mu^D}$. Given a sequent Γ we define $\Gamma^\varepsilon := \{\varphi^{(\varepsilon,\ldots,\varepsilon)} \mid \varphi \in \Gamma\}$. Note that Γ^ε is a BT state in $\mathbb{A}_\mu^D$. In the following subsection we will also see that Γ^ε is a so-called annotated sequent.

4.1.5. Lemma. *For any sequent Γ there is an NW-proof π of Γ iff there is a NW^D-proof ρ of Γ^ε. The proof ρ is regular iff π is so.*

Proof:
Lemma 4.1.4 states that the tracking automaton $\mathbb{A}_\mu$ exactly captures the successful infinite paths in NW-derivations. Therefore, the correctness of the binary tree determinization, Theorem 3.2.7, implies that an infinite NW-branch γ carries a

ν-trace iff $w(\gamma) \in \mathcal{L}(\mathbb{A}_\mu^D)$. It remains to define the functions f and g in a suitable way. We define

$$
\begin{aligned}
f : A^D \to& \Phi \\
Y \mapsto& \{\varphi \in \mathcal{L}_\mu \mid \varphi^\sigma \in Y \text{ for some } \sigma\} \cup \\
&\{\psi[\eta x.\psi/x] \mid (\eta x.\psi^*)^\sigma \in Y \text{ for some } \sigma\}
\end{aligned}
\qquad
\begin{aligned}
g : \Phi &\to A^D \\
\Gamma &\mapsto \Gamma^\varepsilon
\end{aligned}
$$

Clearly, the desired conditions are fulfilled. Thus the lemma directly follows from Lemma 4.1.2 applied to NW and the automaton $\mathbb{A}_\mu^D$. □

Even though the condition on infinite paths in NW^D is much simpler, there is a bit of syntactic clutter that we would like to avoid. Therefore we will first define the infinitary proof system BT^∞ from scratch without the syntactic clutter and then show that, up to some minor modifications, BT^∞ is equivalent to NW^D.

4.2 BT-proofs

We present two non-wellfounded proof systems for the modal μ-calculus, the infinitary system BT^∞ and the cyclic system BT. The system BT^∞ almost coincides with NW^D and thus annotated sequents in BT correspond to macrostates of $\mathbb{A}_\mu^D$, where $\mathbb{A}_\mu$ is the tracking automaton checking the global soundness condition of NW. The main difference between BT^∞ and NW^D is that in BT^∞ the transition function of $\mathbb{A}_\mu^D$ is split into multiple rules. Therefore, the rules of BT^∞ resemble the steps of the definition of the transition function of $\mathbb{A}_\mu^D$.

The cyclic proof system BT corresponds to the regular fragment of BT^∞ with a subgraph-based global soundness condition.

4.2.1 Definition of proof systems

Recall that we fixed a set of $\mathcal{L}_\mu$-formulas Φ such that $\mathsf{Clos}(\Phi) = \Phi$ and a priority function $\Omega_\mu : \mathcal{L}_\mu \to \mathbb{N}^+$. Let m be the maximal even priority of Ω_μ on Φ, that is, the maximal even number in $\{\Omega_\mu(\varphi) \mid \varphi \in \Phi\}$. We will use notation introduced in Subsection 3.2.3. Note that the range of the priority function Ω_μ is $\mathbb{N}^+$ and we can therefore omit the priority 0. We let $\mathrm{TSeq}(m) := \{(s_2, s_4, ..., s_m) \mid s_2, s_4, ..., s_m \in 2^*\}$ be the set of sequences of binary strings of length $\frac{m}{2}$.

Annotated sequents are sets of pairs (φ, σ), usually written as φ^σ, where $\varphi \in \mathsf{Clos}(\Phi)$ and $\sigma \in \mathrm{TSeq}(m)$. For an annotated sequent Γ we let Γ^N be the set of annotations occurring in Γ, formally $\Gamma^N := \{\sigma \in \mathrm{TSeq}(m) \mid \exists \varphi \text{ s.t. } \varphi^\sigma \in \Gamma\}$. We let Γ_k^N be the set of binary strings occurring at the k-th position of the annotations in Γ, formally $\Gamma_k^N := \pi_k[\Gamma^N]$. We say that a string s *occurs in* Γ_k^N if there exists $t \in \Gamma_k^N$ such that $s \preccurlyeq t$, meaning that s is an initial substring of t.

For $\sigma = (s_2, ..., s_m) \in \mathrm{TSeq}(m)$ we define $\sigma \cdot 1_k := (s_2, ..., s_k 1, ..., s_m)$ and $\sigma \cdot 0_k := (s_2, ..., s_k 0, ..., s_m)$ for even $k = 2, ..., m$. For an annotated sequent Γ we let $\Gamma^{\cdot 0_k}$ denote the annotated sequent $\{\varphi^{\sigma \cdot 0_k} \mid \varphi^\sigma \in \Gamma\}$.

Let Γ be an annotated sequent and $\varphi^\sigma \in \Gamma$. We define $\sigma \upharpoonright k^\Gamma$ to be the tuple of binary strings obtained from $\sigma = (s_2, ..., s_m)$ by replacing every s_j with $j > k$ by $\mathrm{minL}(\mathrm{tree}(\Gamma_j^N))$. If the context Γ is clear we write $\sigma \upharpoonright k$ instead of $\sigma \upharpoonright k^\Gamma$.

$$\mathsf{Ax1}: \frac{}{p^\sigma, \overline{p}^\tau} \qquad \vee: \frac{\varphi^\sigma, \psi^\sigma, \Gamma}{(\varphi \vee \psi)^\sigma, \Gamma} \qquad \Box: \frac{\varphi^\sigma, \Gamma}{\Box\varphi^\sigma, \Diamond\Gamma} \qquad \mathsf{D}_\dagger: \begin{array}{c} [\Gamma]^\dagger \\ \vdots \\ \dfrac{\Gamma}{\Gamma} \end{array}$$

$$\mathsf{Ax2}: \frac{}{\top^\sigma} \qquad \wedge: \frac{\varphi^\sigma, \Gamma \quad \psi^\sigma, \Gamma}{(\varphi \wedge \psi)^\sigma, \Gamma} \qquad \mathsf{weak}: \frac{\Gamma}{\varphi^\sigma, \Gamma}$$

$$\mu: \frac{\varphi[\mu x.\varphi/x]^{\sigma \upharpoonright \Omega_\mu(\mu x.\varphi)}, \Gamma}{\mu x.\varphi^\sigma, \Gamma} \qquad \nu: \frac{\varphi[\nu x.\varphi/x]^{\sigma \upharpoonright k \cdot 1_k}, \Gamma^{\cdot 0_k}}{\nu x.\varphi^\sigma, \Gamma} \quad \text{where } k = \Omega_\mu(\nu x.\varphi)$$

$$\mathsf{Compress}_k^{s0}: \frac{\varphi_1^{(..., st_1, ...)}, ..., \varphi_n^{(..., st_n, ...)}, \Gamma}{\varphi_1^{(..., s0t_1, ...)}, ..., \varphi_n^{(..., s0t_n, ...)}, \Gamma} \quad \text{where } s \text{ does not occur in } \Gamma_k^N$$

$$\mathsf{Compress}_k^{s1}: \frac{\varphi_1^{(..., st_1, ...)}, ..., \varphi_n^{(..., st_n, ...)}, \Gamma}{\varphi_1^{(..., s1t_1, ...)}, ..., \varphi_n^{(..., s1t_n, ...)}, \Gamma} \quad \begin{array}{l} \text{where } s \text{ does not occur in } \Gamma_k^N \\ \text{and } s \neq 0 \cdots 0 \end{array}$$

Figure 4.3: Rules of BT

The rules $\mathsf{Compress}_k^{s0}$ and $\mathsf{Compress}_k^{s1}$ are schemata for even $k = 2, ..., m$ and $s \in 2^*$. In these rules the notation $\varphi_i^{(..., st_i, ...)}$ is to be understood in a way that st_i is the binary string in the k-th position of the annotation. We will write Compress for any of those rules and write $\mathsf{Compress}_k^s$ for either $\mathsf{Compress}_k^{s0}$ or $\mathsf{Compress}_k^{s1}$.

Note that, if one ignores the annotations, the rules Ax1, Ax2, $\vee$, $\wedge$, μ, ν and $\Box$ in Figure 4.3 are the same as the rules of NW. Annotated sequents in the BT system correspond to macrostates of $\mathbb{A}_\mu^D$, where $\mathbb{A}_\mu$ is the tracking automaton checking the trace condition in an NW-proof. The rules of BT correspond to the transition function δ_A of $\mathbb{A}_\mu^D$, where the transformations of δ_A are distributed over multiple rules: Step 1(a) of δ_A is carried out in every rule and step 1(b) and step 2 correspond to the modification of the annotations in the rules R_μ and R_ν. Notably, we do not add zeros to the annotations if the zeros would get deleted anyway in step 4 of the transition function. Step 3 corresponds to the following rule Resolve, which is a special case of weak. In the application of this rule we require that $\varphi^\tau \notin \Gamma$.

$$\text{Resolve: } \frac{\varphi^\sigma, \Gamma}{\varphi^\sigma, \varphi^\tau, \Gamma} \quad \text{where } \sigma > \tau$$

The rules Compress are additional and correspond to step 4 of δ_A.

4.2.1. Definition. A BT-*derivation* π is a derivation defined from the rules in Figure 4.3, with the extra condition that the rules are applied with the following priority: first Resolve, then Compress, and then all other rules.

Just as annotated sequents correspond to macrostates of the deterministic automaton $\mathbb{A}^D_\mu$, the condition on infinite BT^∞-paths corresponds to the acceptance condition of $\mathbb{A}^D_\mu$: We say that a pair (k, s) is *preserved* at a node, if s is in play at position k at the corresponding macrostate and not marked red; and *progresses* if it is marked green.

4.2.2. Definition. Let π be a BT-derivation and A be a set of nodes in π. Let $k \in \{2, 4, ..., m\}$ and $s \in 2^*$. We say that the pair (k, s)

- is *preserved* on A if
  - s occurs in $\mathsf{S}(v)^N_k$ for every v in A and
  - if $\mathsf{R}(v) = \mathsf{Compress}^t_k$ for a node v in A, then $t \not\preceq s$,
- *progresses* (infinitely often) on A if there is $s' = s0 \cdots 0$ such that $\mathsf{R}(v) = \mathsf{Compress}^{s'1}_k$ for some v in A (for infinitely many $v \in A$).

An infinite path $\beta = (u_i)_{i \in \omega}$ in π is *successful* if there are N and (k, s) such that (k, s) is preserved and progresses infinitely often on $\{u_i \mid i \geq N\}$.

A set of nodes A in π is *successful* if there is (k, s) that is preserved and progresses on A.

4.2.3. Definition. The *infinitary proof system* BT^∞ is defined by the BT rules together with all infinite successful paths.

The *cyclic proof system* BT is subgraph-based and defined by the BT rules together with all successful finite sets of nodes. Recall that this means that a BT-*proof* is a finite BT-derivation such that for each *strongly connected subgraph* A in π there exists (k, s) that is preserved and progresses on A.

4.2.4. Example. Define the following formulas:

$$\varphi := \nu x.\Diamond(x \wedge \mu y.\Diamond y \vee p),$$
$$\psi := \mu x.\nu y.\Box[(x \vee \overline{p}) \wedge (y \vee p)].$$

In Example 2.7.6 we gave an NW-proof of ψ, φ. We now present a BT-proof of the same sequent. For convenience we define the following auxiliary formulas:

$$\begin{aligned}
\chi &:= \nu y.\Box[(\psi \vee \overline{p}) \wedge (y \vee p)],\\
\gamma &:= (\psi \vee \overline{p}) \wedge (\chi \vee p),\\
\delta &:= \mu y.\Diamond y \vee p.
\end{aligned}$$

The fixed set of $\mathcal{L}_\mu$-formulas Φ can be defined as $\Phi := \mathsf{Clos}(\overline{\psi}, \varphi)$. Note that only formulas in Φ will occur in a proof of $\overline{\psi}, \varphi$. We need to consider the priorities of the fixpoint formulas in Φ as defined in Definition 2.3.15:

$$\begin{aligned}
\Omega_\mu(\varphi) &= 2 & \qquad \Omega_\mu(\psi) &= 3\\
\Omega_\mu(\delta) &= 1 & \qquad \Omega_\mu(\chi) &= 2
\end{aligned}$$

Therefore, 2 is the only even priority of formulas in Φ and annotations will consist of sequences of binary strings of length 1, that is, of binary strings. The following is a BT-proof π of ψ, φ, where ρ is the BT-proof of χ^{01}, δ^{1} given below. For space reasons we omit the (finite) proof π_0 of $\psi^{01}, \overline{p}^{01}, \delta^{1}$. To improve readability, the annotations in this example are colored blue.

$$\pi :\quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\lceil \chi^{0}, \Diamond(\varphi\wedge\delta)^{1}\rceil^{\dagger}}{\chi^{0}, \Diamond(\varphi\wedge\delta)^{11}}\,\mathsf{Compress}_2^{11}}{\chi^{00}, \Diamond(\varphi\wedge\delta)^{11}}\,\mathsf{Compress}_2^{00}}{\chi^{0}, \varphi^{1}}\,\nu}{\chi^{00}, \varphi^{1}}\,\mathsf{Compress}_2^{00}}{\psi^{01}, \varphi^{1}}\,\mu}{\psi^{01}, \overline{p}^{01}, \varphi^{1}}\,\mathsf{weak} \qquad \pi_0}{\psi^{01}, \overline{p}^{01}, \varphi\wedge\delta^{1}}\,\wedge}{\psi\vee\overline{p}^{01}, \varphi\wedge\delta^{1}}\,\vee \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\lceil \Box\gamma^{01}, \Diamond(\varphi\wedge\delta)^{10}\rceil^{\ddagger}}{\Box\gamma^{011}, \Diamond(\varphi\wedge\delta)^{10}}\,\mathsf{Compress}_2^{011}}{\chi^{01}, \Diamond(\varphi\wedge\delta)^{1}}\,\nu}{\chi^{01}, \Diamond(\varphi\wedge\delta)^{11}}\,\mathsf{Compress}_2^{11}}{\chi^{010}, \Diamond(\varphi\wedge\delta)^{11}}\,\mathsf{Compress}_2^{010}}{\chi^{01}, \varphi^{1}}\,\nu \qquad \rho}{\chi^{01}, \varphi\wedge\delta^{1}}\,\wedge}{\chi^{01}, p^{01}, \varphi\wedge\delta^{1}}\,\mathsf{weak}}{\chi\vee p^{01}, \varphi\wedge\delta^{1}}\,\vee}{\gamma^{01}, \varphi\wedge\delta^{1}}\,\wedge}{\Box\gamma^{01}, \Diamond(\varphi\wedge\delta)^{1}}\,\Box}{\Box\gamma^{01}, \Diamond(\varphi\wedge\delta)^{10}}\,\mathsf{Compress}_2^{10}}{\Box\gamma^{01}, \Diamond(\varphi\wedge\delta)^{10}}\,\mathsf{D}_{\ddagger}}{\chi^{0}, \Diamond(\varphi\wedge\delta)^{1}}\,\nu}{\chi^{0}, \Diamond(\varphi\wedge\delta)^{1}}\,\mathsf{D}_{\dagger}}{\chi^{\varepsilon}, \varphi^{\varepsilon}}\,\nu}{\psi^{\varepsilon}, \varphi^{\varepsilon}}\,\mu$$

$$
\rho : \quad
\begin{prooftree}
\AxiomC{}
\RightLabel{Ax1}
\UnaryInfC{$\overline{p}^{01}, p^{1}$}
\RightLabel{weak}
\UnaryInfC{$\psi^{01}, \overline{p}^{01}, \Diamond(\mu y.\Diamond y \lor p)^{1}, p^{1}$}
\RightLabel{$\lor$}
\UnaryInfC{$\psi^{01}, \overline{p}^{01}, \Diamond(\mu y.\Diamond y \lor p) \lor p^{1}$}
\RightLabel{μ}
\UnaryInfC{$\psi^{01}, \overline{p}^{01}, \mu y.\Diamond y \lor p^{1}$}
\RightLabel{$\lor$}
\UnaryInfC{$\psi \lor \overline{p}^{01}, \mu y.\Diamond y \lor p^{1}$}
\AxiomC{$[\chi^{01}, \mu y.\Diamond y \lor p^{1}]^{\ddagger\!\!\ddagger}$}
\RightLabel{weak}
\UnaryInfC{$\chi^{01}, p^{01}, \mu y.\Diamond y \lor p^{1}$}
\RightLabel{$\lor$}
\UnaryInfC{$\chi \lor p^{01}, \mu y.\Diamond y \lor p^{1}$}
\RightLabel{$\land$}
\BinaryInfC{$\gamma^{01}, \mu y.\Diamond y \lor p^{1}$}
\RightLabel{$\Box$}
\UnaryInfC{$\Box\gamma^{01}, \Diamond(\mu y.\Diamond y \lor p)^{1}$}
\RightLabel{$\mathsf{Compress}_2^{10}$}
\UnaryInfC{$\Box\gamma^{01}, \Diamond(\mu y.\Diamond y \lor p)^{10}$}
\RightLabel{$\mathsf{Compress}_2^{011}$}
\UnaryInfC{$\Box\gamma^{011}, \Diamond(\mu y.\Diamond y \lor p)^{10}$}
\RightLabel{ν}
\UnaryInfC{$\chi^{01}, \Diamond(\mu y.\Diamond y \lor p)^{1}$}
\RightLabel{weak}
\UnaryInfC{$\chi^{01}, \Diamond(\mu y.\Diamond y \lor p)^{1}, p^{1}$}
\RightLabel{$\lor$}
\UnaryInfC{$\chi^{01}, \Diamond(\mu y.\Diamond y \lor p) \lor p^{1}$}
\RightLabel{μ}
\UnaryInfC{$\chi^{01}, \mu y.\Diamond y \lor p^{1}$}
\RightLabel{$\mathsf{D}_{\ddagger\!\!\ddagger}$}
\UnaryInfC{$\chi^{01}, \mu y.\Diamond y \lor p^{1}$}
\end{prooftree}
$$

In order to show that π is a BT-proof, every strongly connected subgraph of π must be successful. In π there are three discharged leaves l_0, l_1, l_2 discharged by $\dagger, \ddagger, \ddagger\!\!\ddagger$, respectively. Strongly connected subgraphs of π consist of repeat paths of those leaves.

In the subproof ρ of π, there is only one strongly connected subgraph A_2 consisting of the repeat path of the discharged leaf l_2. The pair $(2, 01)$ is successful, as 01 occurs in the annotations of every sequent in A_0 and a $\mathsf{Compress}_2^{011}$ rule is applied in A_0.

There are three more proper strongly connected subgraphs in π: A_0 consisting of the repeat path of l_0, A_1 consisting of the repeat path of l_1, and A_{01} consisting of the repeat path of both l_0 and l_1. In all three strongly connected subgraphs, the pair $(2, 1)$ is successful, as 1 occurs in the annotations of every sequent and a $\mathsf{Compress}_2^{11}$ rule is applied on both repeat paths. We have therefore shown that π is a BT-proof of ψ, φ.

4.2.2 Infinitary proof system BT^∞

The system BT^∞ informally resembles the infinitary proof system NW^D. This connection will be formalized in the next lemma.

4.2.5. Lemma. *There is a BT^∞-proof π of an annotated sequent Γ iff there is an NW^D-proof of Γ. The proof π is regular iff ρ is so.*

Proof:
Let ρ be a NW^D proof of Γ. We make the following changes to obtain a BT^∞-proof π of Γ.

1. Every element $(\eta x.\psi^*)^\sigma$ occurring in a sequent Y in ρ is replaced by $\eta x.\psi^\sigma$.

2. Every rule $\mathsf{R}^{\mathbb{A}^D}$ is split up in the rule R and multiple occurrences of Resolve and Compress.

This results in a BT-derivation π of Γ. The global soundness conditions of NW^D and BT^∞ coincide, thus π is a BT^∞-proof.

Conversely, let π be a BT^∞-proof of Γ. Recall that in π the rules Resolve and Compress are applied whenever applicable. We can therefore translate π to an NW^D-proof ρ by changing the following:

1. Any Resolve and Compress rules are merged with the rule applied at their parent-node.

2. Any formula $\varphi[\eta x.\varphi/x]^\sigma$ occurring as the auxiliary formula of an η rule is replaced with $(\eta x.\varphi^*)^\sigma$.

This gives a NW^D-proof, as the global soundness condition in NW^D and BT^∞ coincide.

In both directions regular proofs are translated to regular proofs. □

4.2.6. Theorem (Soundness and Completeness). *Let Γ be a sequent. Then there is a regular BT^∞-proof of Γ^ε iff $\bigvee \Gamma$ is valid.*

Proof:
This follows from the soundness and completeness of regular NW-proofs Theorem 2.7.5 together with Lemma 4.1.5 and Lemma 4.2.5. □

4.2.7. Remark. In BT^∞-proofs we demand that the rules Resolve and Compress are applied whenever applicable. We do so to be able to prove the soundness of the system using the uniform construction from Subsection 4.1.1. We do note though that this extra condition on the order of the applied rules is not necessary. We could still prove soundness of the system, yet the proof would become more complicated. We sketch two possible ways to do so below.

One possibility would be to adapt the binary tree determinization in a way such that in the definition of the transition function the steps 3 and 4 might not be executed for all elements – corresponding to sequents in a proof, where Resolve and Compress are not applied even though applicable. With this adaption the proof of the converse direction of Theorem 3.2.7 still goes through and yields

$\mathcal{L}(\mathbb{A}_\mu^D) \subseteq \mathcal{L}(\mathbb{A}_\mu)$. This inclusion is enough to show that $\mathsf{NW}^D \vdash \Gamma^\varepsilon$ implies $\mathsf{NW} \vdash \Gamma$ following the same lines as our proof. Making these adaptions precise becomes very technical and was therefore omitted in exchange for more readability.

Another way would be to directly prove soundness without using the determinization method explicitly. This could be done by adapting the proof of the converse direction of the correctness of the determinization method, Theorem 3.2.7, directly on the proof tree. We use a similar approach in the soundness proof of JS_2^∞ in Chapter 5.

4.2.3 Cyclic proof system BT

As NW-proofs can be assumed to be regular and annotations are added deterministically, we can also assume BT^∞-proofs to be regular. A standard argument then transforms regular BT^∞-proofs into BT-proofs and vice versa.

4.2.8. LEMMA. *There is a* BT*-proof of an annotated sequent* Γ *iff there is a regular* BT^∞*-proof of* Γ.

Proof:
Let π be a regular BT^∞-proof of an annotated sequent Γ. For a node $v \in \pi$ we let π_v be the maximal subtree of π rooted at v. We define the equivalence relation $\sim$ by setting $v \sim u$ if π_v is isomorphic to π_u. As π is regular, there are only finitely many distinct equivalence classes.

Let P_π be the set of infinite paths in π. We will define functions f_c, f_l from $P_\pi \to \omega$ that select appropriate positions for discharge rules and corresponding leaves. Let $\beta = (\beta_i)_{i\in\omega}$ be in P_π. As β is successful there exist $t < t'$ such that

1. $v_t \sim v_{t'}$ and
2. there exists (k, s) which is preserved and progresses on $\{v_t, ..., v_{t'}\}$.

Choose (t, t') minimally with respect to the lexicographic order and define $f_c(\beta) = t$ and $f_l(\beta) = t'$. Let $C = \{\beta_{f_c(\beta)} \mid \beta \in P_\pi\}$ and $L = \{\beta_{f_l(\beta)} \mid \beta \in P_\pi\}$ be the set of companions and leaves of our new proof, respectively. We let the companion of $l = \beta_{f^l(\beta)}$ be $c(l) = \beta_{f^c(\beta)}$. We define π_L to be the subtree of π up to the leaves L. Then we insert a D rule at every companion node $v \in C$ with discharged assumptions $l \in L$ such that $c(l) = v$. Because of König's Lemma π_L is indeed a finite tree. Therefore it holds that π_L is a BT-derivation. The following claim shows that π_L is a BT-proof and therefore shows the first direction of the lemma.

<u>`Claim 1`</u>: On every strongly connected subgraph A of π_L there exists (k, s) such that (k, s) is preserved and progresses on A.

`Proof of Claim 1`: It is easy to see that every infinite path β in π_L corresponds to an infinite path γ in π. Let A be a strongly connected subgraph of π_L and $\beta = (\beta_i)_{i\in\omega}$ be an infinite path in π_L that visits every node in A infinitely often and no other node infinitely often. Let γ be the corresponding infinite path in π. As γ is successful there is N and (k, s) such that (k, s) is preserved and progresses infinitely often on $\{\gamma_i \mid i \geq N\}$. Yet this yields that (k, s) is preserved and progresses on A. $\dashv$

For the other direction we can show that the infinite unfolding π^* of a BT-proof π is a BT^∞-proof. Recall that the infinite unfolding π^* of π is the BT-derivation obtained from π by recursively unfolding outermost leaves, and removing all discharge rules. It follows that π^* is a BT-proof as every infinite branch in π^* can be seen as a path in a strongly connected subgraph of π. Thus the condition for the infinite branches in π^* follows from the analogous condition for strongly connected subgraphs of π. □

4.2.9. Theorem (Soundness and Completeness). *Let* Γ *be a sequent. Then there is a* BT*-proof of* Γ^ε *iff* $\bigvee \Gamma$ *is valid..*

Proof:
This is a combination of Theorem 4.2.6 and Lemma 4.2.8. □

4.2.10. Remark. In the proof system JS introduced by Jungteerapanich [Jun10] and Stirling [Sti14] annotated sequents are of the form $\theta : \varphi_1^{a_1}, ..., \varphi_n^{a_n}$, where $\varphi_1, ..., \varphi_n$ are $\mathcal{L}_\mu$-formulas, $a_1, ..., a_n$ are sequences of names, and the so-called *control* θ is a linear order on all names occurring in the sequent. In contrast to JS, our sequents consist of formulas with annotations and nothing else; that is, no control. On the other hand, the soundness condition of BT is less local: It is subgraph-based whereas JS is a path-based cyclic proof system. In this sense, the control in JS gives information on the structure of the cyclic proof tree.

Interestingly, we could also add a control to our sequents and obtain a path-based soundness condition, if desired. We sketch how such a system BT^c could be defined. We let sequents of BT^c be of the form $\theta_2, ..., \theta_m : \Gamma$, where Γ is a BT-sequent and θ_k is a linear order on the strings occurring in Γ_k^N for even $k = 2, ..., m$. The BT-rules can be adjusted to BT^c-rules in a way that newly introduced strings are added as the last elements in the linear orders. We can then define a path-based soundness condition on BT^c by enforcing repeat paths to be successful, where a finite path β is successful, if the set of nodes occurring on β is successful. Note that for a repeat in BT^c, also the linear orders $\theta_2, ..., \theta_m$ coincide at the repeat leaves and their companion nodes. We can use a similar argument to the one for the cyclic system JS_2 in Chapter 5 to show that BT^c is sound and complete.

Similarly, in [AEL22] a control was added to a cyclic system for the first-order μ-calculus introduced by [SD03] to obtain a path-based system.

In [AL16], Afshari and Leigh defined a path-based cyclic proof system Circ, that is obtained by simplifying the annotations in JS. Even though annotations are simplified, this system still contains a control. Inspired by Circ, we will introduce the cyclic proof system Circ_2 for the two-way modal μ-calculus in Chapter 5. In this system we are able to omit the control, while maintaining a path-based soundness condition. Adjusting our system Circ_2 to the simpler setting of $\mathcal{L}_\mu$ would therefore give a cyclic proof system that combines the advantages of the systems BT and JS; that is, sequents consisting of annotated formulas only as in BT, and a path-based soundness condition as in JS.

We end this subsection with a remark on the complexity of proof-checking, that is, checking if a BT-derivation is a BT-proof.

4.2.11. Remark. Given a BT-derivation π, checking if π is a BT-proof is in coNP. We can give the following algorithm in NP, that checks if π is not a BT-proof: Choose non-deterministically a strongly connected subgraph A and a pair (k, s) and check if (k, s) is preserved and progresses on A. The latter can be done in polynomial time. The complexity of proof checking can be compared to PSPACE in NW and linear time in JS. Note that, if we add a control to the BT-proof system as mentioned in Remark 4.2.10, the soundness condition boils down to checking paths between leafs and its companions; in that case proof checking could also be done in linear time.

4.3 Incompleteness of Clo

In the rest of this chapter we study a proof system for the modal μ-calculus with a very simple form of annotations: the cyclic system Clo. We prove that Clo is incomplete by giving a valid sequent Θ that is not provable in Clo. We thereby argue that, for a cyclic, cut-free proof system with a local validity condition to be complete, the use of complex annotations – such as those in BT or in JS– is in some sense necessary.

The proof system Clo was introduced by Afshari and Leigh in [AL17] as a cyclic, cut-free proof system for the modal μ-calculus. In that paper they intend to prove the completeness of Kozen's axiomatization Koz for the modal μ-calculus in a proof-theoretic way. This is done by a series of translations starting from Jungteerapanich and Stirling's annotated proof system JS to Clo and further to Koz. Apart from its prominent role in this completeness proof Clo has also attracted interest as a stand-alone proof system: It is cut-free, cyclic and has very simple annotations and discharge condition. As such the completeness of Clo also played a crucial role in showing completeness of the natural axiomatization for game logic [EHKMV19].

Our contribution is to show that Clo is in fact incomplete. This shows that JS-proofs cannot be translated to Clo-proofs and thus breaks the completeness proof for Koz in [AL17].

4.3.1 Clo-proofs

We define the cyclic proof system Clo. Formulas in Clo will be annotated by sequences of *names*. Our presentation will differ slightly from [AL17], as we do not assume that formulas are of a certain form. We thus define names for every ν-formula instead of every variable symbol.

Let Φ be a sequent. For each ν-formula $\varphi = \nu x.\varphi'$ in $\mathsf{Clo}(\Phi)$ fix an infinite set N_φ of *names* for φ. We assume $\mathsf{N}_\varphi \cap \mathsf{N}_\psi = \varnothing$ if $\varphi \neq \psi$ and let $\mathsf{N} = \bigcup\{\mathsf{N}_\varphi \mid \varphi \text{ is a } \nu\text{-formula in } \mathsf{Clo}(\Phi)\}$. The dependence order on ν-formulas extends to a partial order on the names as follows: $\mathsf{x} \leq \mathsf{y}$ if $\mathsf{x} \in \mathsf{N}_\varphi, \mathsf{y} \in \mathsf{N}_\psi$ and $\varphi \leq_\Phi \psi$. We also write $\psi \leq \mathsf{x}$ for a fixpoint formula ψ and a name x, if $\mathsf{x} \in N_\varphi$ and $\psi \leq_\Phi \varphi$. For a sequence of names σ and a name x we write $\mathsf{x} \leq \sigma$ if $\mathsf{x} \leq \mathsf{y}$ for all $\mathsf{y} \in \sigma$. Let $\sqsubseteq$ denote the reflexive subsequence relation on N^*, where we write $\sigma \sqsubseteq \tau$ if τ contains all names occurring in σ and whenever a name x occurs to the left of a name y in τ, then x also occurs to the left of y in σ.

An *annotated formula* is a pair (φ, σ), denoted by φ^σ, where φ is a closed formula and σ is a finite sequence of decreasing names. An *annotated sequent* is a set of annotated formulas.

Names play a double role in Clo: They mark successful repeats and they act as discharge tokens for the discharge rule $\nu{-}\mathsf{clo}$. In particular, every $\nu{-}\mathsf{clo}$ rule is marked with an unique name. For convenience we assume that every $\nu{-}\mathsf{clo}$ rule has at least one discharged assumption; it might have multiple ones though.

4.3.1. Definition. The cyclic proof system Clo is the proof system defined from the rules in Figure 4.4 and all finite repeat paths. In particular, a Clo-*proof* is a finite Clo-derivation.

4.3.2. Remark. Note that, differently to the presentation in [AL17], we employ a set of names for all ν-formulas rather than for all variables. The reason is that we do not assume any syntactic condition on formulas: In particular, a variable x may be bound by fixpoint operators at different places in a formula. If one assumes that every variable is the fixpoint variable of a unique fixpoint formula $\eta x.\varphi$, then naming variables is equivalent to naming fixpoint formulas.

In the discharge rule $\nu{-}\mathsf{clo}$ in Clo, discharged leaves are labeled with different sequents than their companions. Accordingly, the notion of an infinite unfolding of a proof must be adapted. Recall that we call a repeat leaf l *outermost*, if its companion $c(l)$ is the root of some proper cluster.

$$\mathsf{Ax1}\colon \frac{}{p^\sigma, \overline{p}^\tau} \qquad \vee\colon \frac{\varphi^\sigma, \psi^\sigma, \Gamma}{(\varphi \vee \psi)^\sigma, \Gamma} \qquad \eta\colon \frac{\varphi[\eta x.\varphi/x]^\sigma, \Gamma}{\eta x.\varphi^\sigma, \Gamma} \text{ where } \eta x.\varphi \leq \sigma$$

$$\mathsf{Ax2}\colon \frac{}{\top^\sigma} \qquad \wedge\colon \frac{\varphi^\sigma, \Gamma \qquad \psi^\sigma, \Gamma}{(\varphi \wedge \psi)^\sigma, \Gamma} \qquad \Box\colon \frac{\varphi^\sigma, \Gamma}{\Box\varphi^\sigma, \Diamond\Gamma} \qquad \mathsf{weak}\colon \frac{\Gamma}{\varphi^\sigma, \Gamma}$$

$$\mathsf{exp}\colon \frac{\varphi_1^{\sigma_1}, ..., \varphi_n^{\sigma_n}}{\varphi_1^{\tau_1}, ..., \varphi_n^{\tau_n}} \quad \text{where } \sigma_i \sqsubseteq \tau_i \text{ for } i = 1, ..., n$$

$$\begin{array}{l} \qquad\qquad [\nu x.\varphi^{\sigma \mathsf{x}}, \Gamma]^{\mathsf{x}} \\ \qquad\qquad\qquad \vdots \\ \nu{-}\mathsf{clo}_{\mathsf{x}}\colon \dfrac{\varphi[\nu x.\varphi/x]^{\sigma\mathsf{x}}, \Gamma}{\nu x.\varphi^\sigma, \Gamma} \quad \text{where } \mathsf{x} \in \mathsf{N}_{\nu x.\varphi} \text{ fresh and } \mathsf{x} \leq \sigma \end{array}$$

Figure 4.4: Rules of Clo

4.3.3. Definition. Let ρ be a Clo-derivation. For an outermost repeat leaf l in ρ labeled with $\Gamma, \nu x.\varphi^{\sigma\mathsf{x}}$, we define the Clo-derivation ρ_l as

$$\dfrac{\begin{array}{c}\rho_v \\ \varphi[\nu x.\varphi/x]^{\sigma\mathsf{x}}, \Gamma\end{array}}{\nu x.\varphi^{\sigma\mathsf{x}}, \Gamma} \ \nu$$

where ρ_v is the maximal subderivation of ρ rooted at the child node of $c(l)$.

The *infinite unfolding* ρ^* of ρ is defined as the Circ_2-derivation obtained from ρ by recursively replacing outermost leaves l with ρ_l.

4.3.4. Proposition. Clo *is sound.*

Proof:
Let ρ be a Clo-proof of a sequent Γ. Let ρ^* be the infinite unfolding of ρ. Replacing $\nu{-}\mathsf{clo}$ rules with ν, removing nodes labeled with exp and removing all annotations in ρ^* yields an NW-derivation π of Γ. The names occurring in ρ^* give a ν-trace for every infinite path in π and thus π is an NW-proof. Hence, the soundness of Clo follows from the soundness of NW. □

4.3.2 Proof of incompleteness

In the main part of this section we prove the incompleteness of Clo. This is done by defining a sequent Θ, which is probable in NW, but shown to be unprovable in Clo.

Definition of Θ

Define the following formulas

$$
\begin{aligned}
\varphi &:= \Diamond(\bar{p} \wedge (\Box x \vee \Diamond \nu y.\Box(p \wedge (\Box x \vee \Diamond y))))\\
\psi &:= \Box(p \wedge (\Box \nu x.\varphi \vee \Diamond y)))\\
\chi &:= \Box \nu x.\varphi \vee \Diamond \nu y.\psi
\end{aligned}
$$

.

In order to understand the relation between these formulas, observe that

$$\varphi[\nu x.\varphi/x] = \Diamond(\bar{p} \wedge \chi) \qquad \text{and} \qquad \psi[\nu y.\psi/y] = \Box(p \wedge \chi).$$

For the rest of this section let

$$\Theta := \nu x.\varphi, \nu y.\psi$$

We want to show that $\bigvee \Theta$ is a valid μ-calculus formula that is not provable in Clo.

The reason for the excessive use of modalities in the definition of Θ is to restrict possibilities in the proof search of Θ: For most sequents occurring in the proof search only one rule apart from weakenings will be applicable.

Validity of $\bigvee \Theta$

The following is an NW-proof π of Θ, where the subproofs π' are isomorphic to the whole proof π:

$$
\dfrac{
  \dfrac{
    \dfrac{}{\bar{p}, p}\,\mathsf{Ax1}
    \qquad
    B: \dfrac{\dfrac{\dfrac{\overset{\pi'}{\nu x.\varphi, \nu y.\psi}}{\Box \nu x.\varphi, \Diamond \nu y.\psi}\,\Box}{\Box \nu x.\varphi, \Diamond \nu y.\psi, p}\,\mathsf{weak}}{\chi, p}\,\vee
  }{\bar{p} \wedge \chi, p}\,\wedge
  \qquad
  \dfrac{
    C: \dfrac{\dfrac{\dfrac{\overset{\pi'}{\nu x.\varphi, \nu y.\psi}}{\Box \nu x.\varphi, \Diamond \nu y.\psi}\,\Box}{\bar{p}, \Box \nu x.\varphi, \Diamond \nu y.\psi}\,\mathsf{weak}}{\bar{p}, \chi}\,\vee
    \qquad
    \dfrac{\dfrac{\overset{\pi'}{\nu x.\varphi, \nu y.\psi}}{\Box \nu x.\varphi, \Diamond \nu y.\psi}\,\Box}{\chi}\,\vee
  }{\bar{p} \wedge \chi, \chi}\,\wedge
}{
  \dfrac{
    \dfrac{
      \dfrac{\bar{p} \wedge \chi, p \wedge \chi}{\varphi[\nu x.\varphi/x], \psi[\nu y.\psi/y]}\,\Box
    }{\varphi[\nu x.\varphi/x], \nu y.\psi}\,\nu
  }{\nu x.\varphi, \nu y.\psi}\,\nu
}\,\wedge
$$

As there are no μ-formulas, all traces in π are ν-traces. Hence, on every infinite branch of π there is a ν-trace implying that π is an NW-proof. Using the fact that NW is a sound proof system it follows that $\bigvee \Theta$ is a valid μ-calculus formula.

Proof idea

The more difficult task is to show that Θ is not provable in Clo. To gather some intuition we first consider the above NW-proof π and see why it cannot be translated to a Clo-proof ρ. At the node B both formulas are descendants of $\nu x.\varphi$, but not of $\nu y.\psi$ at the root. Contrary, at the node C both formulas are descendants of $\nu y.\psi$, but not of $\nu x.\varphi$ at the root. Replacing the ν rules by $\nu-\mathsf{clo}$ rules yields the following Clo-derivation ρ_0:[1]

$$
\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\bar{p}^{\mathsf{x}}, p^{\mathsf{y}}}\ \mathsf{Ax1} \qquad \dfrac{\dfrac{\dfrac{B:\ [\nu x.\varphi^{\mathsf{x}}, \nu y.\psi]^{\mathsf{x}}}{\nu x.\varphi^{\mathsf{x}}, \nu y.\psi^{\mathsf{x}}}\ \mathsf{exp}}{\Box\nu x.\varphi^{\mathsf{x}}, \Diamond\nu y.\psi^{\mathsf{x}}, p^{\mathsf{y}}}\ \mathsf{weak},\Box}{\chi^{\mathsf{x}}, p^{\mathsf{y}}}\ \vee}{\bar{p}\wedge\chi^{\mathsf{x}}, p^{\mathsf{y}}}\ \wedge \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{C:\ \varphi[\nu x.\varphi/x], \nu y.\psi^{\mathsf{y}}}{\varphi[\nu x.\varphi/x]^{\mathsf{y}}, \nu y.\psi^{\mathsf{y}}}\ \mathsf{exp}}{\nu x.\varphi^{\mathsf{y}}, \nu y.\psi^{\mathsf{y}}}\ \nu}{\bar{p}^{\mathsf{x}}, \Box\nu x.\varphi^{\mathsf{y}}, \Diamond\nu y.\psi^{\mathsf{y}}}\ \mathsf{weak},\Box}{\bar{p}^{\mathsf{x}}, \chi^{\mathsf{y}}}\ \vee \qquad \cdots}{\bar{p}\wedge\chi^{\mathsf{x}}, \chi^{\mathsf{y}}}\ \wedge}{\bar{p}\wedge\chi^{\mathsf{x}}, p\wedge\chi^{\mathsf{y}}}\ \wedge}{\varphi[\nu x.\varphi/x]^{\mathsf{x}}, \psi[\nu y.\psi/y]^{\mathsf{y}}}\ \Box}{\varphi[\nu x.\varphi/x]^{\mathsf{x}}, \nu y.\psi}\ \nu-\mathsf{clo}_{\mathsf{y}}}{\nu x.\varphi, \nu y.\psi}\ \nu-\mathsf{clo}_{\mathsf{x}}
$$

The leaf B can be discharged by $\nu-\mathsf{clo}_{\mathsf{x}}$, yet it is impossible to discharge C by $\nu-\mathsf{clo}_{\mathsf{y}}$, as the formula $\varphi[\nu x.\varphi/x]$ is annotated with x at the companion node and there is no way to obtain the same annotation at C. Thus we can discharge C neither by $\nu-\mathsf{clo}_{\mathsf{x}}$ nor by $\nu-\mathsf{clo}_{\mathsf{y}}$. We will see that essentially the same problem occurs in all Clo-derivations of Θ.

As Clo is cut-free we can do proof search adapted for cyclic, annotated proofs in the following way: First we consider all NW-proofs of Θ. Then we show that in all of those NW-proofs it is impossible to replace some ν rules by $\nu-\mathsf{clo}$ rules with discharged assumptions in order to obtain a Clo-proof.

NW-proofs of Θ

Recall that $\Theta = \nu x.\varphi, \nu y.\psi$. We want to consider all possible NW-proofs of Θ. Thus let π be any NW-proof of Θ and r be any node in π labeled with the sequent Θ. We want to have a look at the subtree rooted at r, where leaves are axioms or nodes labeled with Θ.

We begin by claiming that the first two applied rules at r have to be ν rules with respective principal formulas $\nu x.\varphi$ and $\nu y.\psi$, in no particular order. The only other rules which may be applied are instances of weak. Yet this is impossible, as both $\nu x.\varphi$ and $\nu y.\psi$, respectively, are not valid and therefore not provable.

[1]We omit the rightmost branch, as it is not important for this example.

Hence the first two applied rules are ν rules; for now the order in which the formulas are unfolded is not important. The lowest part of the proof-tree rooted at r looks as follows:

$$\dfrac{\dfrac{\dfrac{\bar{p}\wedge\chi, p\wedge\chi}{\varphi[\nu x.\varphi/x], \psi[\nu y.\psi/y]}\;\Box}{\varphi[\nu x.\varphi/x], \nu y.\psi}\;\nu}{\nu x.\varphi, \nu y.\psi}\;\nu$$

First we assume that weak is only applied for literals, then the proof tree rooted at r looks as follows, up to the order of the applied rules:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A: \dfrac{}{\bar{p}, p}\;\mathsf{Ax1} \qquad \dfrac{\dfrac{B: \dfrac{\nu x.\varphi, \nu y.\psi}{\Box\nu x.\varphi, \Diamond\nu y.\psi}\;\Box}{\Box\nu x.\varphi, \Diamond\nu y.\psi, p}\;\mathsf{weak}}{\chi, p}\;\vee}{\bar{p}\wedge\chi, p}\;\wedge \qquad \dfrac{\dfrac{\dfrac{C: \dfrac{\nu x.\varphi, \nu y.\psi}{\Box\nu x.\varphi, \Diamond\nu y.\psi}\;\Box}{\bar{p}, \Box\nu x.\varphi, \Diamond\nu y.\psi}\;\mathsf{weak}}{\bar{p}, \chi}\;\vee \qquad \dfrac{D: \dfrac{\nu x.\varphi, \nu y.\psi}{\Box\nu x.\varphi, \Diamond\nu y.\psi}\;\Box}{\chi}\;\vee}{\bar{p}\wedge\chi, \chi}\;\wedge}{\bar{p}\wedge\chi, p\wedge\chi}\;\wedge}{\varphi[\nu x.\varphi/x], \psi[\nu y.\psi/y]}\;\Box}{\varphi[\nu x.\varphi/x], \nu y.\psi}\;\nu}{\nu x.\varphi, \nu y.\psi}\;\nu$$

Note that, in whatever order the rules are applied, we end up with the same four nodes A,B,C and D.

At last we have to check the case in which an instance of weak is applied to a non-literal. Leaving those weakenings aside which lead to invalid formulas, we end up with only two possible NW-proofs (up to the order of the unfoldings):

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{A: \dfrac{}{\bar{p}, p}\;\mathsf{Ax1} \qquad \dfrac{\dfrac{B: \dfrac{\nu x.\varphi, \nu y.\psi}{\Box\nu x.\varphi, \Diamond\nu y.\psi}\;\Box}{\Box\nu x.\varphi, \Diamond\nu y.\psi, p}\;\mathsf{weak}}{\chi, p}\;\vee}{\bar{p}\wedge\chi, p}\;\wedge \qquad \dfrac{\dfrac{C: \dfrac{\nu x.\varphi, \nu y.\psi}{\Box\nu x.\varphi, \Diamond\nu y.\psi}\;\Box}{\chi}\;\vee}{\bar{p}\wedge\chi, \chi}\;\mathsf{weak}}{\bar{p}\wedge\chi, p\wedge\chi}\;\wedge}{\varphi[\nu x.\varphi/x], \psi[\nu y.\psi/y]}\;\Box}{\varphi[\nu x.\varphi/x], \nu y.\psi}\;\nu}{\nu x.\varphi, \nu y.\psi}\;\nu$$

$$
\begin{prooftree}
\AxiomC{}
\RightLabel{Ax1}
\UnaryInfC{$A:\ \bar{p}, p$}
\AxiomC{$C:\ \nu x.\varphi, \nu y.\psi$}
\RightLabel{$\Box$}
\UnaryInfC{$\Box \nu x.\varphi, \Diamond \nu y.\psi$}
\RightLabel{weak}
\UnaryInfC{$\bar{p}, \Box \nu x.\varphi, \Diamond \nu y.\psi$}
\RightLabel{$\vee$}
\UnaryInfC{$\bar{p}, \chi$}
\RightLabel{$\wedge$}
\BinaryInfC{$\bar{p}, p \wedge \chi$}
\AxiomC{$B:\ \nu x.\varphi, \nu y.\psi$}
\RightLabel{$\Box$}
\UnaryInfC{$\Box \nu x.\varphi, \Diamond \nu y.\psi$}
\RightLabel{$\vee$}
\UnaryInfC{χ}
\RightLabel{weak}
\UnaryInfC{$\chi, p \wedge \chi$}
\RightLabel{$\wedge$}
\BinaryInfC{$\bar{p} \wedge \chi, p \wedge \chi$}
\RightLabel{$\Box$}
\UnaryInfC{$\varphi[\nu x.\varphi/x], \psi[\nu y.\psi/y]$}
\RightLabel{ν}
\UnaryInfC{$\varphi[\nu x.\varphi/x], \nu y.\psi$}
\RightLabel{ν}
\UnaryInfC{$\nu x.\varphi, \nu y.\psi$}
\end{prooftree}
$$

We have considered all possible NW-proofs rooted at a node labeled with Θ, where leaves are axioms or nodes labeled with Θ. We can state some immediate observations.

Let π be an NW-proof of Θ.

1. We call a node in π which is labeled with the sequent Θ an *unfolding node*. An unfolding node and its child are always labeled with ν rules with principal formulas $\nu x.\varphi$ and $\nu y.\psi$, in no particular order.

2. The *unfolding tree* $\mathcal{U}_\pi = (U, P)$ of π is the tree consisting of all unfolding nodes in π, such that $P(u, v)$ if v is a descendant of u with no other unfolding nodes in π between u and v.

3. Let v be an unfolding node and $u \in \mathcal{U}_\pi$ its parent, meaning that $P(u, v)$ holds. We call v an *x-node* if there are no traces from $\nu y.\psi$ at u to $\nu x.\varphi$ or $\nu y.\psi$ at v. We call v a *y-node* if there are no traces from $\nu x.\varphi$ at u to $\nu x.\varphi$ or $\nu y.\psi$ at v.

4.3.5. Lemma. *Let π be an* NW*-proof of Θ. Every unfolding node of π has either two or three children in $\mathcal{U}_\pi$, where exactly one is an x-node and exactly one is a y-node.*

Proof:
In the above proofs the nodes B,C and D are unfolding nodes, where the nodes B are x-nodes, the nodes C are y-nodes and the nodes D are neither. $\Box$

Clo-derivations of Θ

4.3.6. Definition. Let π be an NW-proof of a sequent Γ and let ρ be a Clo-derivation of Γ. We say that ρ is *obtained from* π if π can be transformed to ρ by

1. changing some ν rules to $\nu{-}\mathsf{clo}$ rules with discharged assumptions, such that the proof tree is pruned at discharged assumptions and

2. adding annotations and exp rules accordingly.

Clo-derivations ρ obtained from an NW-proof π have a very similar structure to the NW-proof π. Apart from nodes labeled with exp, ρ consists of the same nodes as π, where the tree is pruned at discharged assumptions. We transfer the concepts of unfolding nodes, x-nodes and y-nodes to Clo-derivations and get similar results for Clo-derivations obtained from NW-proofs.

4.3.7. DEFINITION. Let ρ be a Clo-derivation of Θ. We call a node v in ρ an *unfolding node* if it is labeled with $\nu x.\varphi^\sigma, \nu y.\psi^\tau$ for some annotations σ, τ and with a rule different from exp. The *unfolding tree* $\mathcal{U}_\rho$ of ρ, *x-nodes* and *y-nodes* are defined analogously as for NW-proofs. We call an unfolding node u *root-like* if no node in the maximal subtree of ρ rooted at u is discharged by a $\nu{-}\mathsf{clo}$ rule at an ancestor-node of u.

4.3.8. LEMMA. *Let π be an* NW*-proof of Θ and ρ be a* Clo*-derivation obtained from π. Every root-like unfolding node of ρ has either two or three children in $\mathcal{U}_\rho$, where exactly one is an x-node and exactly one is a y-node.*

Proof:
The same statement holds for every unfolding node in π. We want to show that it transfers to root-like unfolding nodes in ρ.

Towards that aim let u be a root-like unfolding node in ρ and let u' be the corresponding node in π from which u is obtained. Clearly, u' is an unfolding node in π. Let v' be a child of u' in $\mathcal{U}_\pi$ and τ' be the path from u' to v'. As no node in the maximal subtree of ρ rooted at u is discharged by an ancestor-node of u, the path τ' is transformed to a path τ from u to v in ρ, where only ν rules are changed to $\nu{-}\mathsf{clo}$ rules and nodes labeled with exp are added. Thus v is an unfolding node in ρ, which is an x-node (a y-node) iff v' is an x-node (a y-node).

Therefore u in $\mathcal{U}_\rho$ and u' in $\mathcal{U}_\pi$ have the same number of children of the same type and the statement follows from Lemma 4.3.5. □

4.3.9. LEMMA. *Let ρ be a* Clo*-derivation of Θ obtained from an* NW*-proof π. Let u and v be unfolding nodes in ρ, such that v is a child of u in $\mathcal{U}_\rho$.*

Let u' be either u or a descendant of u below v. If v is an x-node and u' is labeled with $\nu{-}\mathsf{clo}_\mathsf{y}$ with principal formula $\nu y.\psi$, then none of its discharged assumptions are in the maximal subtree of ρ rooted at v.

The same holds, if v is a y-node and the principal formula at u' is $\nu x.\varphi$.

Proof:
All nodes between u and u' are labeled with ν, $\nu-\mathsf{clo}$ or exp. This implies that any trace starting from $\nu y.\psi$ at u' also gives a trace starting from $\nu y.\psi$ at u. As v is an x-node, there are no traces from $\nu y.\psi$ at u to $\nu x.\varphi$ or $\nu y.\psi$ at v and therefore also none starting from $\nu y.\psi$ at u'. Hence the name y introduced by the rule $\nu-\mathsf{clo}_\mathsf{y}$ does not occur in the maximal subtree of ρ rooted at v. $\Box$

As an illustration of Lemma 4.3.9 consider the following Clo-derivation ρ_0.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\bar{p}^\mathsf{x}, p^\mathsf{y}}\,\mathsf{Ax1} \qquad \dfrac{\dfrac{\dfrac{B:\ \lceil \nu x.\varphi^\mathsf{x}, \nu y.\psi \rceil^\mathsf{x}}{\nu x.\varphi^\mathsf{x}, \nu y.\psi^\mathsf{x}}\,\mathsf{exp}}{\Box \nu x.\varphi^\mathsf{x}, \Diamond \nu y.\psi^\mathsf{x}, p^\mathsf{y}}\,\mathsf{weak},\Box}{\chi^\mathsf{x}, p^\mathsf{y}}\,\vee}{\bar{p}\wedge\chi^\mathsf{x}, p^\mathsf{y}}\,\wedge \qquad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{C:\ \varphi[\nu x.\varphi/x], \nu y.\psi^\mathsf{y}}{\varphi[\nu x.\varphi/x]^\mathsf{y}, \nu y.\psi^\mathsf{y}}\,\mathsf{exp}}{\nu x.\varphi^\mathsf{y}, \nu y.\psi^\mathsf{y}}\,\nu}{\bar{p}^\mathsf{x}, \Box \nu x.\varphi^\mathsf{y}, \Diamond \nu y.\psi^\mathsf{y}}\,\mathsf{weak},\Box}{\bar{p}^\mathsf{x}, \chi^\mathsf{y}}\,\vee \qquad \cdots}{\bar{p}\wedge\chi^\mathsf{x}, \chi^\mathsf{y}}\,\wedge}{\bar{p}\wedge\chi^\mathsf{x}, p\wedge\chi^\mathsf{y}}\,\wedge}{\dfrac{\varphi[\nu x.\varphi/x]^\mathsf{x}, \psi[\nu y.\psi/y]^\mathsf{y}}{\dfrac{\varphi[\nu x.\varphi/x]^\mathsf{x}, \nu y.\psi}{\nu x.\varphi, \nu y.\psi}\,\nu-\mathsf{clo}_\mathsf{x}}\,\nu-\mathsf{clo}_\mathsf{y}}\,\Box$$

Let $u = u'$ be the root of ρ_0 and v be the node C. Lemma 4.3.9 states that there can be no discharged assumption of $\nu-\mathsf{clo}_\mathsf{x}$ in the subproof rooted at C. This holds as the name x does not occur at C.

Similarly, there are no discharged assumptions of $\nu-\mathsf{clo}_\mathsf{y}$ in the subproof of ρ_0 rooted at B.

Clo-proofs of Θ

We show that every Clo-proof can be obtained from some NW-proof. On the other hand we see that every Clo-derivation of Θ obtained from an NW-proof is infinite, and thus not a Clo-proof.

4.3.10. LEMMA. *Every Clo-proof ρ of a sequent Γ can be obtained from some NW-proof π of Γ.*

Proof:
Let ρ be a Clo-proof and ρ^* its infinite unfolding. Replacing $\nu-\mathsf{clo}$ rules by ν rules in ρ^*, removing nodes labeled with exp and removing annotations yields an NW-proof π of Γ, from which ρ can be obtained. $\Box$

4.3.11. LEMMA. *There is no Clo-proof of Θ.*

Proof:
Let π be an NW-derivation of Θ and ρ be a Clo-derivation obtained from π. Let $\mathcal{U}_\rho$ be the unfolding tree of ρ. We want to show that $\mathcal{U}_\rho$ is infinite. This implies that ρ is infinite as well and therefore not a Clo-proof. Towards that aim we define the *height* of a node $u \in \mathcal{U}_\rho$ to be the number of ancestors of u in $\mathcal{U}_\rho$. We show by induction:

$\circledast$ There exists a root-like unfolding node in $\mathcal{U}_\rho$ of arbitrary height.

Recall that an unfolding node u is called root-like, if no node in the maximal subtree of ρ rooted at u is discharged by a $\nu-\mathsf{clo}$ rule at an ancestor-node of u. The induction base is trivial, as the root is an unfolding node, which does not have any ancestors. For the induction step assume that u is a root-like unfolding node. The unfolding node u is labeled with ν or $\nu-\mathsf{clo}$. Let u' be the lowest descendant of u in ρ, which is labeled with ν or $\nu-\mathsf{clo}$.

We make a case distinction on whether u and u' are labeled with ν or $\nu-\mathsf{clo}$. If both u and u' are labeled with ν, then any child of u in $\mathcal{U}_\rho$ has the same ancestors labeled with $\nu-\mathsf{clo}$ as u. Thus any child of u in $\mathcal{U}_\rho$ is root-like.

Assume that exactly one of u and u', say u_i, is labeled with $\nu-\mathsf{clo}$; without loss of generality assume that u_i is labeled with $\nu-\mathsf{clo}_\mathsf{y}$ with principal formula $\nu y.\psi$. Let v be the child of u in $\mathcal{U}_\rho$ that is an x-node given by Lemma 4.3.8. Then Lemma 4.3.9 states that none of the discharged assumptions of the $\nu-\mathsf{clo}_\mathsf{y}$ rule are in the maximal subtree rooted at v. Together with the induction hypothesis this proves that v is root-like.

If both u and u' are labeled with $\nu-\mathsf{clo}$ we may without loss of generality assume that u is labeled with $\nu-\mathsf{clo}_\mathsf{x}$ with principal formula $\nu x.\varphi$. Then u' is labeled with a $\nu-\mathsf{clo}_\mathsf{y}$ rule of the following form for some annotations a and b:

$$\frac{\varphi[\nu x.\varphi/x]^{a\mathsf{x}}, \psi[\nu y.\psi/y]^{b\mathsf{y}}}{\varphi[\nu x.\varphi/x]^{a\mathsf{x}}, \nu y.\psi^{b}}\ \nu-\mathsf{clo}_\mathsf{y}$$

Let v be the child of u in $\mathcal{U}_\rho$ that is a y-node, meaning that there are no traces from $\varphi[\nu x.\varphi/x]$ at u' to $\nu x.\varphi$ or $\nu y.\psi$ at v. Thus the name x does not occur in the maximal subtree rooted at v. In particular, no node in the maximal subtree rooted at v is labeled with $\varphi[\nu x.\varphi/x]^{a\mathsf{x}}, \nu y.\psi^{b\mathsf{y}}$ and therefore there is no discharged assumption of $\nu-\mathsf{clo}_\mathsf{y}$ in the maximal subtree rooted at v. As v is a y-node there is also no discharged assumption of $\nu-\mathsf{clo}_\mathsf{x}$ in the maximal subtree rooted at v because of Lemma 4.3.9. Thus v is root-like.

We have shown that ρ is infinite and hence not a Clo-proof. As every Clo-proof can be obtained from an NW-proof, there is no Clo-proof of Θ. $\Box$

4.3.12. THEOREM. Clo *is not complete.*

Proof:
Clo does not prove the valid sequent Θ. □

4.3.3 Variations of Clo

We mention that Clo is complete if we add the cut rule or if we restrict formulas to be adisjunctive. Informally, a formula φ is adisjunctive if in every subformula of φ of the form $\psi_0 \vee \psi_1$, any variable x does not occur in both ψ_0 and ψ_1.

Completeness of Clo + cut

4.3.13. Definition. Let $\mathsf{Clo} + \mathsf{cut}$ be the proof system expanding Clo with the cut rule

$$\mathsf{cut}\colon \frac{\Gamma, \varphi^{\varepsilon} \qquad \overline{\varphi}^{\varepsilon}, \Gamma}{\Gamma}$$

In order to show the completeness of $\mathsf{Clo} + \mathsf{cut}$, we first need the following technical lemma. A more general version together with a proof of it can be found in Lemma V.2. in [AL17].

4.3.14. Lemma. *Let $\varphi(x)$ be a formula with free variable x and let ψ, χ be sentences. Let σ, τ be annotations such that $\eta y.\xi \leq \sigma, \tau$ for all fixpoint formulas $\eta y.\xi \in \mathsf{Clos}(\varphi)$. Then*

$$\{\psi^{\sigma}, \chi^{\tau}\} \vdash_{\mathsf{Clo}} \varphi[\psi/x]^{\sigma}, \overline{\varphi}[\chi/x]^{\tau}.$$

4.3.15. Lemma. *The proof system $\mathsf{Clo} + \mathsf{cut}$ is complete.*

Proof:
We reduce the completeness of $\mathsf{Clo} + \mathsf{cut}$ to the completeness of Koz. It therefore suffices to show the admissibility of all Koz-rules in $\mathsf{Clo} + \mathsf{cut}$; this is obvious for all rules apart from the rule ind:

$$\mathsf{ind}\colon \frac{\varphi[\bigwedge \overline{\Gamma}/x], \Gamma}{\nu x.\varphi, \Gamma}$$

For showing the admissibility of ind consider the following $\mathsf{Clo} + \mathsf{cut}$ derivation

$$\dfrac{\dfrac{\dfrac{\lceil \nu x.\varphi^{\mathsf{x}}, \Gamma \rceil^{\mathsf{x}}}{\nu x.\varphi^{\mathsf{x}}, \bigvee \Gamma}\vee \\ \vdots \\ \pi_0 \\ \vdots \\ \varphi[\nu x.\varphi/x]^{\mathsf{x}}, \overline{\varphi}[\bigvee \Gamma/x] \qquad \varphi[\bigwedge \overline{\Gamma}/x], \Gamma}{\varphi[\nu x.\varphi/x]^{\mathsf{x}}, \Gamma}\,\mathsf{cut}}{\nu x.\varphi, \Gamma}\,\nu{-}\mathsf{clo}_{\mathsf{x}}$$

where the Clo-derivation π_0 is given by Lemma 4.3.14 $\square$

In the proof of Lemma 4.3.15 we assumed the completeness of Koz, which presupposes the completeness proof by Walukiewicz [Wal00]. Note that the other direction holds as well, meaning that any Clo + cut proof can be translated to a Koz proof: The translation given in [AL17] also works with the inclusion of cut. Therefore, if one could show the completeness of Clo + cut directly, it would yield an alternative proof for the completeness of Koz.

Completeness for the adisjunctive fragment

4.3.16. Definition. We call a sequent Γ *adisjunctive* if for every fixpoint formula of the form $\nu x.\varphi \in \mathsf{Clos}(\Gamma)$ and every disjunction $\psi_0 \vee \psi_1 \in \mathsf{Clos}(\nu x.\varphi)$, we have either $\nu x.\varphi \notin \mathsf{Clos}(\psi_0)$ or $\nu x.\varphi \notin \mathsf{Clos}(\psi_1)$.

The sequent Θ is not adisjunctive and this is crucial in showing that Θ is not provable in Clo. Observe that $\chi = \Box\nu x.\varphi \vee \Diamond\nu y.\psi \in \mathsf{Clos}(\nu x.\varphi)$ and $\nu x.\varphi \in \mathsf{Clos}(\Box\nu x.\varphi)$ as well as $\nu x.\varphi \in \mathsf{Clos}(\Diamond\nu y.\psi)$.

The notion of adisjunctivity is dual to the concept of aconjunctivity introduced by Kozen [Koz83]. This duality arises from the fact that we are proving validity in contrast to satisfiability as in [Koz83]. For the aconjunctive fragment of the modal μ-calculus Kozen proved the completeness of Koz in a relatively direct way. Similarly, Clo is complete for the adisjunctive fragment of the modal μ-calculus. This is justified by the fact that the translation from JS to Clo, as given in [AL17], holds for proofs of adisjunctive sequents.

4.3.17. Remark. In the setting of linear logic with fixpoints, similar proof systems were introduced raising analogous questions. The finitary proof system μMALL introduced in [Bae12] contains an induction rule similar to the one in Koz. Alternatively, $\mu\mathsf{MALL}^\infty$ is an infinitary proof system with a trace-based global validity condition similar to the one in NW; and the cyclic proof system $\mu\mathsf{MALL}^\omega$ corresponds to the regular fragment of $\mu\mathsf{MALL}^\infty$, see [BDS16]. Note that in the setting of linear logic the infinitary system $\mu\mathsf{MALL}^\infty$ proves more sequents than the corresponding cyclic system $\mu\mathsf{MALL}^\omega$.

A natural question to ask is if μMALL and $\mu\mathsf{MALL}^\omega$ prove the same set of sequents? A translation from finitary μMALL proofs to cyclic $\mu\mathsf{MALL}^\omega$ proofs is rather straight forward [Dou17]; the idea is to introduce cuts in a similar way as we did in the proof of Lemma 4.3.15.

In their attempt to translate $\mu\mathsf{MALL}^\omega$ proofs to finitary μMALL proofs, Nollet, Saurin and Tasson [Nol21] introduced the cyclic proof system $\mu\mathsf{MALL}^{\circlearrowright}_{lab}$. In spirit this system is very similar to Clo. A difference in presentation is that they annotate fixpoints inside formulas, whereas we combine all of those annotation to one annotation decorating the whole formula. They show that $\mu\mathsf{MALL}^{\circlearrowright}_{lab}$ proofs can be translated to $\mu\mathsf{MALL}$ proofs – resembling the translation from Clo to Koz. However, $\mu\mathsf{MALL}^{\circlearrowleft}_{lab}$ does not capture all $\mu\mathsf{MALL}^{\omega}$ proofs and it is an open problem if all $\mu\mathsf{MALL}^{\omega}$ proofs can be translated to finitary $\mu\mathsf{MALL}$ proofs.

In the modal μ-calculus, the analogous problem was overcome in Walukiewicz's completeness proof by first showing that every $\mathcal{L}_\mu$ formula is provably equivalent to an aconjunctive one. An analogous normal form is however not known for linear logic.

4.4 Conclusion

We defined a uniform construction that, given an infinitary proof system and a deterministic automaton checking its global soundness condition, yields an infinitary proof system with a simpler soundness condition. Applying this construction, we obtained the infinitary proof system BT^{∞} and the cyclic proof system BT for the modal μ-calculus.

Due to its uniformity, our method will apply to non-wellfounded derivation systems for many other logics as well. For instance, in the proof systems LKID^{ω} [Bro06] for first-order logic with inductive definitions, cyclic arithmetic CA [Sim17] and similar systems, the global soundness condition demands that on every infinite branch there is a term/variable which progresses infinitely often. This condition can be checked by a nondeterministic Büchi automaton, which one can determinize to employ our uniform construction. Using our determinization method for Büchi automata in Section 3.2.2 would yield an annotated proof system, where the annotations are binary strings, which label the terms/variables.

In the final section, we showed that the cyclic proof system Clo is incomplete. This result leaves some further questions open:

1. Can the completeness of Koz be proved without reducing the problem to the aconjunctive fragment as in Walukiewicz's proof [Wal00]? More recently, different completeness proofs of Koz have been presented in a automata-theoretic setting [ESV18] and in a proof-theoretic setting [ALM25]. Yet, on a high level both of those proofs follow the approach of Walukiewicz by first showing that every $\mathcal{L}_\mu$-formula is provable equivalent to an aconjunctive $\mathcal{L}_\mu$-formula.

2. Is there a finitary, wellfounded, cut-free, and annotation-free proof system that is sound and complete for the modal μ-calculus? A candidate would be the system $\mathsf{Koz}^{-}_{\mathsf{s}}$ introduced in [AL17], for which the completeness is unknown.

   Note that the finitized version of the infinitely branching system JKS falls in this category [JKS08]. Yet, as the number of premises in its proof rules is not bound, this system is rather unnatural and not the kind of system we are hoping for here.

3. Is the natural axiomatization for game logic complete? As the proof given in [EHKMV19] relies on the completeness of Clo, this is again an open question.

Chapter 5

Interpolation for the two-way modal μ-calculus

The language $\mathcal{L}^2_\mu$ of the *two-way modal μ-calculus* is obtained from $\mathcal{L}_\mu$ by adding, for each modality a, a modality $\breve{a}$, which in the semantics will be interpreted as the converse of the accessibility relation for a. At first glance, this seems like a minor change, yet it turns out that it complicates matters quite a bit. In fact, as mentioned in Subsection 2.4.1, only few results are known about $\mathcal{L}^2_\mu$. Most notably, it lacks the finite model property and its satisfiability problem is in exponential time. We contribute to the knowledge on the two-way μ-calculus by showing two important properties: Craig interpolation and Beth definability.

5.0.1. Definition. A logic has the *Craig Interpolation Property* if, for any pair of formulas φ and ψ such that $\varphi \vDash \psi$, there exists an *interpolant* – that is, a formula ι such that

1. $\varphi \vDash \iota$,
2. $\iota \vDash \psi$ and
3. $\mathsf{Voc}(\iota) \subseteq \mathsf{Voc}(\varphi) \cap \mathsf{Voc}(\psi)$.

Note that the consequence relation $\varphi \vDash \psi$ that we consider is the *local* one, meaning that $\varphi \vDash \psi$ is equivalent to the validity of the implication $\varphi \to \psi$.

The property was named after William Craig, who proved it for first-order logic [Cra57]. Since then it has been studied intensively in logic (see, for example, van Benthem [Ben08]), while it also has many computer science applications in, for instance, knowledge representation [JLPW21] and model checking [McM05].

Beth Definability is a related property; informally it states that the implicit definability of a concept in a logic implies the existence of an explicit definition. This notion originates from the work of Beth [Bet53], who discovered it as a property of classical first-order logic. Beth definability is studied intensively in description logics, where it is used to optimize reasoning [CFS13].

5.0.2. Definition. A logic has the *Beth Definability Property* if, for any proposition letters p, q and formulas $\varphi(p)$ such that[1] $\varphi(p), \varphi(q) \vDash p \leftrightarrow q$, there is a formula χ with $\mathsf{Voc}(\chi) \subseteq \mathsf{Voc}(\varphi) \setminus \{p\}$ and $\varphi(p) \vDash p \leftrightarrow \chi$.

The main result of this chapter is the following.

5.0.3. Theorem. *The two-way modal μ-calculus has the Craig interpolation property and, as a corollary, the Beth definability property.*

There are various ways to prove interpolation. In particular, it has been shown using model-theoretic [BCV15], automata-theoretic [DH00] and algebraic [GM05] approaches. In proof theory, the most common technique is *Maehara's method* [Mae61].

In this approach, in order to obtain an interpolant for $\varphi \to \psi$, one considers a finite proof π of $\varphi, \overline{\psi}$.[2] First, the proof π is translated to a so-called *split proof*, in which each sequent Σ in π is split into a *split sequent* $\Sigma_l \mid \Sigma_r$, where the left component Σ_l contains all descendants of φ, and the right component Σ_r contains all descendants of $\overline{\psi}$. In the next step, interpolants for each node of the derivation tree are computed by means of a leaf-to-root induction: For every split sequent $\Gamma \mid \Delta$, a formula ι in the common vocabulary of Γ and Δ is defined with accompanying proofs of $\Gamma \mid \iota$ and $\bar{\iota} \mid \Delta$. For axioms this is usually an easy task, and, by inductively dealing with all rules in π, one ultimately obtains an interpolant for $\varphi \mid \overline{\psi}$ at the root.

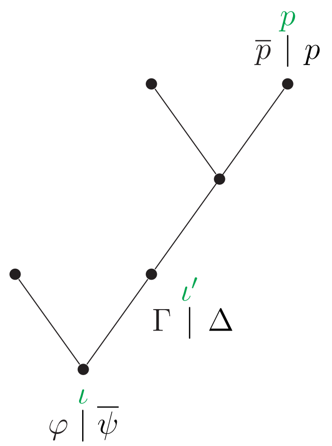


Figure 5.1: A finite split proof with some indicated sequents. The interpolants are written in green above the respective sequents.

[1] We use $\varphi(q)$ as an abbreviation of $\varphi[q/p]$.

[2] Note that our perspective in this chapter is partly tableaux-theoretic: A sequent Γ will be provable iff $\bigwedge \Gamma$ is unsatisfiable. Therefore, the formula $\varphi \to \psi$ is valid iff the sequent $\varphi, \overline{\psi}$ is provable.

In recent years, the scope of the method has been extended to include cyclic proofs. Shamkanov [Sha14], Afshari and Leigh [AL19] and Marti and Venema [MV21b] used it to prove interpolation properties for, respectively, Gödel-Löb logic, $\mathcal{L}_\mu$ and its alternation-free fragment.

The challenge for cyclic proofs is that some leaves are not axiomatic and hence fail to have a trivial interpolant. However, each such leaf is discharged at a companion node, closer to the root. The idea is to associate a fixpoint variable – as a kind of pre-interpolant – with each discharged leaf, and to bind this variable at the companion with a fixpoint operator. For this to work one relies on a cyclic proof system with a *path-based soundness condition* satisfying an extra condition: In split proofs the success condition on repeat paths depends solely on either the left or right components of the repeat path. Depending on whether the left or right components of the repeat path are successful, a greatest or least fixpoint operator is introduced at the companion node.

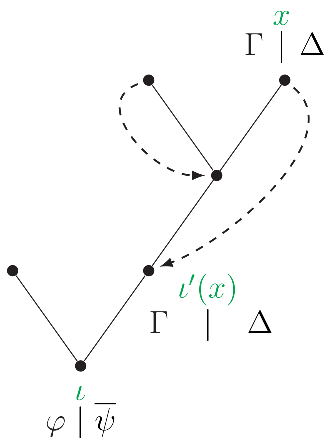


Figure 5.2: A cyclic split proof with some indicated sequents. We wrote the respective (pre-)interpolants above those sequents in green. As x and $\iota'(x)$ are interpolants for the same split sequent, we have $x = \iota'(x)$. Therefore, the interpolant for the companion node is a fixpoint solving this equation; hence it is either $\mu x.\iota'(x)$ or $\nu x.\iota'(x)$.

Sound and complete proof systems for $\mathcal{L}^2_\mu$ have already been proposed, yet none are adequate for adapting Maehara's method. Afshari, Jäger and Leigh proposed a sound, complete and cut-free derivation system, which features an infinitely branching proof rule [AJL19]. As formulas are finite, infinitely branching systems are not suitable for proving interpolation.

A finitary, cyclic proof calculus was given by Afshari et al. [AELMV23]; this calculus is not cut-free, but its restrictions on the cut-rule make the system suitable for proof search procedures. However, the restriction does not constrain enough for Maehara's method to be applicable.

Our goal is to obtain a path-based cyclic proof system for $\mathcal{L}_\mu$ suitable for proving interpolation. To that end, we first introduce the infinitary proof system NW_2, that generalizes the system NW for the one-way modal μ-calculus and is inspired by an infinitary system for the alternation-free two-way modal μ-calculus proposed by Rooduijn and Venema [RV23]. Using the uniform construction from Chapter 4 and the determinization method for parity automata with ε-transitions from Chapter 3 we obtain the cyclic proof system JS_2. The formulation of JS_2 is inspired by proof systems for $\mathcal{L}_\mu$ by Jungteerapanich [Jun10] and Stirling [Sti14].

In [KV25], the paper that this chapter is partly based on, a split version of the JS_2 system is used to show interpolation. Unfortunately, however, this split system is not sound and the interpolation proof given is incorrect. In this chapter, we correct this mistake and show that interpolation for $\mathcal{L}^2_\mu$ nonetheless holds.

We proceed by translating the JS_2 system into a different cyclic proof system Circ_2, which has simplified annotations and takes inspiration from work on the modal μ-calculus by Afshari and Leigh [AL16]. The Circ_2 system is path-based, and we can define a split version of it that makes Maehara's method applicable.

5.1 Trace-based proof system NW_2

We introduce the infinitary proof system NW_2, a generalization of the system NW for the one-way modal μ-calculus. Note that in NW_2, we are proving *unsatisfiability* of sequents compared to validity in NW. This shift in perspective is motivated by the definition of trace atoms, which becomes more natural in this setting. Consequently, we interpret sequents conjunctively, and infinite branches are considered successful if they carry a μ-trace. The extension with backward modalities causes two kinds of challenges.

The first complication is that, even without fixpoint operators, cut-free derivation systems for modal logics with backward modalities must go beyond simple sequent systems [Nis80]. One solution to this problem is to take resort to more structured sequents, such as the nested sequents of Kashima [Kas94]. However, this approach does not combine well with cyclic proofs, as there is no bound on the number of possible sequents. Alternatively, one may simply allow those applications of cut that are *analytic* – in the broad sense that the cut formula is taken from some bounded set of formulas. Kowalski & Ono [KO17] have shown that the presence of an analytic cut rule does not preclude the application of Maehara's method. Since analytic cuts combine naturally with cyclic proofs, we will adopt this approach.

The second and main challenge is to formulate adequate success conditions for infinite proof branches. The problem is that the combinatorics of the formula traces are more complicated than in the one-way setting, as traces may move both upward and downward through the proof tree. To address this issue, we follow Rooduijn & Venema [RV23], who enrich the syntax of their proof calculus

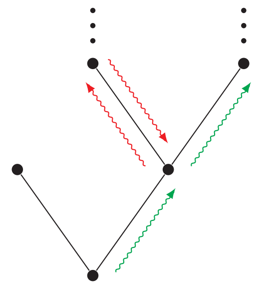


Figure 5.3: A depiction of an NW_2 proof. We indicated a trace following the right-most branch consisting of green upward traces and red detour traces.

with so-called *trace atoms*. Roughly speaking, trace atoms hardwire the ideas underlying Vardi's [Var98] two-way automata explicitly into the syntax. Rooduijn & Venema restricted their attention to the *alternation-free fragment* of $\mathcal{L}^2_\mu$, in which entanglement of least- and greatest fixpoint operators is basically avoided.

The intuitive idea about trace atoms is the following: Let π be an NW_2-derivation. Analogously to the one-way case, an infinite branch in π is successful if it carries a μ-trace. Yet, due to the presence of backward modalities, traces might go up and down the proof tree, meaning that they may also visit multiple branches. Since π is a tree, we can split up infinite traces into segments that return to the same node and those only going upwards. We refer to the former as *detour traces* and model them with trace atoms. In our setting, infinite traces consist of upward traces, as in NW, interleaved with detour traces modelled by trace atoms.

5.1.1 NW_2 sequents

Throughout this chapter we fix a finite set of $\mathcal{L}^2_\mu$-formulas Φ that is closed under $\to_C$ and negation, in other words, such that $\mathsf{Clos}^\neg(\Phi) = \Phi$. These notions were defined in Section 2.3. For a proof of a sequent Γ, the set Φ can be defined as $\mathsf{Clos}(\Gamma) \cup \mathsf{Clos}(\overline{\Gamma})$.

We also define a *priority function* $\Omega_{2\mu} : \mathsf{Fix} \to \mathbb{N}^+$ similarly as in Section 2.3, yet here μ-formulas have *even* priority. This reflects the shift in perspective, as we prove unsatisfiability compared to validity in Section 2.7. We define $\Omega_{2\mu}$ as the minimal-valued function such that

1. $\Omega_{2\mu}(\eta x.\varphi) \leq \Omega_{2\mu}(\lambda y.\psi)$ if $\eta x.\varphi \leq_d \lambda y \psi$ and
2. $\Omega_{2\mu}(\eta x.\varphi)$ is even iff $\eta = \mu$.

We extend $\Omega_{2\mu}$ to a function $\Omega_{2\mu} : \mathcal{L}^2_\mu \to \mathbb{N}^+$ by setting $\Omega_{2\mu}(\varphi) = 1$ if φ is not a fixpoint formula. We let m be the maximal *even* priority of $\Omega_{2\mu}$ on Φ, that is, the maximal even number in $\{\Omega_{2\mu}(\varphi) \mid \varphi \in \Phi\}$, and let m' be the maximal priority of $\Omega_{2\mu}$ on Φ.

5.1.1. Definition. For any pair of formulas φ, ψ and $k = 1, \dots, m'$ we define the *trace atom of priority* k, written $\varphi \rightsquigarrow_k \psi$, and the *negated trace atom of priority* k, written $\varphi \not\rightsquigarrow_k \psi$.

5.1.2. Definition. Given a strategy f for $\exists$ in $\mathcal{E}_\mu(\mathbb{S})$, we say that $\varphi \rightsquigarrow_k \psi$ is *satisfied* in $\mathbb{S}$ at s with respect to f, written $\mathbb{S}, s \Vdash_f \varphi \rightsquigarrow_k \psi$ if there is an f-guided match

$$(\varphi, s) = (\varphi_0, s_0) \cdots (\varphi_n, s_n) = (\psi, s), \qquad n > 0$$

such that $k = \max\{\Omega_{2\mu}(\varphi_i) \mid i = 0, \dots, n-1\}$.

A *pure sequent* is a finite set of formulas, a *trace sequent* is a finite set of trace atoms, and a *sequent* is a pure sequent together with a trace sequent. We will use letters $A, B, \dots$ as variables ranging over formulas and trace atoms and $\Gamma, \Delta, \Sigma, \dots$ for sequents. We will state explicitly if a sequent is a pure or a trace sequent, that is, if it only consists of formulas or trace atoms, respectively.

Given a sequent Γ, we define $\mathsf{Clos}(\Gamma) := \mathsf{Clos}(\{\varphi \in \mathcal{L}^2_\mu \mid \varphi \in \Gamma\})$ and analogously for $\mathsf{Clos}^\neg$. We say that a trace atom $\varphi \rightsquigarrow_k \psi$ is in the closure of Γ, written $\varphi \rightsquigarrow_k \psi \in \mathsf{Clos}^\neg(\Gamma)$, if $\varphi, \psi \in \mathsf{Clos}^\neg(\Gamma)$ and analogously for $\varphi \not\rightsquigarrow_k \psi$. We define Seq_Φ to be the set of sequents consisting of formulas and trace atoms in $\mathsf{Clos}^\neg(\Phi)$.

Note that in this chapter our perspective on sequents and derivations is partly tableau-theoretic: We read sequents *conjunctively* and aim to derive sequents that are *unsatisfiable.*

5.1.3. Definition. We say that $\mathbb{S}, s \Vdash_f \Gamma$ if $\mathbb{S}, s \Vdash_f \bigwedge \Gamma$. Similarly, we say that a sequent Γ is *satisfiable* if $\bigwedge \Gamma$ is satisfiable and *unsatisfiable* otherwise.

5.1.2 NW_2-proofs

The rules of the derivation system NW_2 are given in Figure 5.4. Apart from the inclusion of trace atoms these rules coincide with the rules of NW for the one-way μ-calculus[3], except that formulas have been added in the premise of $\langle a \rangle$. Trace atoms and the extra rules acut, tcut and trans are added to deal with converse modalities. In the rules acut and tcut we demand that $\varphi, \psi \in \mathsf{Clos}^\neg(\Gamma)$. The modal rule needs extra consideration.

5.1.4. Definition. Given a sequent Γ we define $[a]\Gamma := \{[a]\varphi \mid \varphi \in \Gamma\}$. In order to define the modal rule $\langle a \rangle$, let $\Psi = \langle a \rangle\varphi, [a]\Sigma, \Gamma$ be the conclusion of the modal rule. We demand that Σ is a pure sequent, and define $\langle \breve{a} \rangle\Gamma := \{\langle \breve{a} \rangle\gamma \in \mathsf{Clos}^\neg(\Psi) \mid$

[3] If presented dually, that is, where sequents are read conjunctively.

$\gamma \in \Gamma\}$ and

$$\begin{aligned}\Gamma^{\langle a\rangle\varphi} := &\{[\breve{a}]\chi \rightsquigarrow_k \varphi \mid \chi \rightsquigarrow_k \langle a\rangle\varphi \in \Gamma \text{ and } [\breve{a}]\chi \in \mathsf{Clos}^{\neg}(\Psi)\}\\ \cup\ &\{\varphi \not\rightsquigarrow_k [\breve{a}]\chi \mid \langle a\rangle\varphi \not\rightsquigarrow_k \chi \in \Gamma \text{ and } [\breve{a}]\chi \in \mathsf{Clos}^{\neg}(\Psi)\}\\ \cup\ &\{[\breve{a}]\chi \rightsquigarrow_k \psi \mid \chi \rightsquigarrow_k [a]\psi \in \Gamma \text{ and } [\breve{a}]\chi \in \mathsf{Clos}^{\neg}(\Psi)\}\\ \cup\ &\{\psi \not\rightsquigarrow_k [\breve{a}]\chi \mid [a]\psi \not\rightsquigarrow_k \chi \in \Gamma \text{ and } [\breve{a}]\chi \in \mathsf{Clos}^{\neg}(\Psi)\}\end{aligned}$$

Note that $\langle\breve{a}\rangle\Gamma$ and $\Gamma^{\langle a\rangle\varphi}$ also depend on Ψ, yet for simpler notation we do not write Ψ explicitly.

In order to motivate the definition of $\langle\breve{a}\rangle\Gamma$, we show the soundness of the modal rule.

5.1.5. Lemma. *The modal rule is sound. That is, if* $\Psi = \langle a\rangle\varphi, [a]\Sigma, \Gamma$ *is satisfiable, then* $\varphi, \Sigma, \langle\breve{a}\rangle\Gamma, \Gamma^{\langle a\rangle\varphi}$ *is satisfiable as well.*

Proof:
Let $\mathbb{S}, s$ be a pointed model and f be a positional strategy for $\exists$ in $\mathcal{E}_\mu(\mathbb{S})$ such that $\mathbb{S}, s \Vdash_f \langle a\rangle\varphi, [a]\Sigma, \Gamma$. Assume that $f(\langle a\rangle\varphi, s) = (\varphi, t)$. We want to show that $\mathbb{S}, t \Vdash_f \varphi, \Sigma, \langle\breve{a}\rangle\Gamma, \Gamma^{\langle a\rangle\varphi}$.

To start with, since f is winning, we have $\mathbb{S}, t \Vdash_f \varphi$. Moreover, $\mathbb{S}, t \Vdash_f \Sigma$ because $\mathbb{S}, s \Vdash_f [a]\Sigma$ and t is an a-successor of s. This implies that s is an $\breve{a}$-successor of t and therefore $\mathbb{S}, t \Vdash_f \langle\breve{a}\rangle\Gamma$ due to the fact that $\mathbb{S}, s \Vdash_f \Gamma$. It remains to show that $\mathbb{S}, t \Vdash_f \Gamma^{\langle a\rangle\varphi}$.

Assume that $\chi \rightsquigarrow_k \langle a\rangle\varphi \in \Gamma$. This means that there is an f-guided match

$$(\chi, s) \cdots (\langle a\rangle\varphi, s)$$

of priority k. But then there is also the following f-guided match of priority k:

$$([\breve{a}]\chi, t) \cdot (\chi, s) \cdots (\langle a\rangle\varphi, s) \cdot (\varphi, t).$$

Therefore, $\mathbb{S}, t \Vdash_f [\breve{a}]\chi \rightsquigarrow_k \varphi$.

Next assume that $\langle a\rangle\varphi \not\rightsquigarrow_k \chi \in \Gamma$. In order to show that $\mathbb{S}, t \Vdash_f \varphi \not\rightsquigarrow_k [\breve{a}]\chi$, we argue by contradiction. If there is an f-guided match of priority k of the form

$$(\varphi, t) \cdots ([\breve{a}]\chi, t),$$

then there is also the following f-guided match of priority k:

$$(\langle a\rangle\varphi, s) \cdot (\varphi, t) \cdots ([\breve{a}]\chi, t) \cdot (\chi, s).$$

This implies that $\mathbb{S}, s \Vdash_f \langle a\rangle\varphi \rightsquigarrow_k \chi$, which is the desired contradiction. The remaining two cases are analogous. □

$$\text{Ax1: } \frac{}{\varphi, \overline{\varphi}} \qquad \text{Ax2: } \frac{}{\bot} \qquad \text{Ax3: } \frac{}{\varphi \rightsquigarrow_k \psi, \varphi \not\rightsquigarrow_k \psi} \qquad \text{Ax4: } \frac{}{\varphi \rightsquigarrow_{2k} \varphi}$$

$$\wedge\text{: } \frac{\varphi, \psi, \varphi \wedge \psi \rightsquigarrow_1 \varphi, \varphi \wedge \psi \rightsquigarrow_1 \psi, \Gamma}{\varphi \wedge \psi, \Gamma} \qquad \langle a \rangle\text{: } \frac{\varphi, \Sigma, \langle \breve{a} \rangle \Gamma, \Gamma^{\langle a \rangle \varphi}}{\langle a \rangle \varphi, [a]\Sigma, \Gamma}$$

$$\vee\text{: } \frac{\varphi, \varphi \vee \psi \rightsquigarrow_1 \varphi, \Gamma \qquad \psi, \varphi \vee \psi \rightsquigarrow_1 \psi, \Gamma}{\varphi \vee \psi, \Gamma} \qquad \text{weak: } \frac{\Gamma}{A, \Gamma}$$

$$\eta\text{: } \frac{\varphi[\eta x.\varphi/x], \eta x.\varphi \rightsquigarrow_{\Omega_{2\mu}(\eta x.\varphi)} \varphi[\eta x.\varphi/x], \Gamma}{\eta x.\varphi, \Gamma}$$

$$\text{trans: } \frac{\varphi \rightsquigarrow_k \psi, \psi \rightsquigarrow_l \chi, \varphi \rightsquigarrow_{\max\{k,l\}} \chi, \Gamma}{\varphi \rightsquigarrow_k \psi, \psi \rightsquigarrow_l \chi, \Gamma}$$

$$\text{acut: } \frac{\varphi, \Gamma \qquad \overline{\varphi}, \Gamma}{\Gamma} \qquad \text{tcut: } \frac{\varphi \rightsquigarrow_k \psi, \Gamma \qquad \varphi \not\rightsquigarrow_k \psi, \Gamma}{\Gamma}$$

Figure 5.4: Rules of NW_2

5.1.6. Remark. An occurrence of a rule is usually called *analytic* if all formulas φ in the premise of the rule are subformulas of the conclusion Γ. In the context of fixpoint logics, this notion has to be extended, such that we demand that all formulas φ are in the *closure* of Γ. Because the rules acut, tcut and $\langle a \rangle$ are restricted, all rules in NW_2 are analytic.

The notions of active, principal and auxiliary formulas in a rule are defined as for NW rules. In particular, in the rules trans, acut and tcut, all occurring formulas are inactive. As for the NW system defined in Section 2.7, rules in NW_2 formally are pairs (R, ξ), where R is the name of the rule and ξ is either its principal formula or "nil" if R does not have a principal formula. Whenever it is clear from the context, we will omit the principal formula and just write R for the pair (R, ξ).

Similarly to the proof system NW, an infinite branch in an NW_2-derivation π is successful if it carries a μ-trace. Yet, due to the presence of backwards modalities, traces in π might go up and down the proof tree. We call those parts of a trace that go up and down and return to the same node *detour traces* and model them by trace atoms. Traces that only move upwards are called *upward traces* and are dealt with similarly as in NW. A trace in π is a sequence of upward traces interleaved with detour traces.

This necessitates a more careful definition of traces. We define the trace relation to consist of triples (φ, ψ, k), where ψ is a descendant of φ and the *weight* k keeps track of the priority of unfolded fixpoints along the trace from φ to ψ. For upward traces, ψ is a direct descendant of φ and for detour traces there is a trace atom $\varphi \rightsquigarrow_k \psi$ in the sequent.

5.1.7. DEFINITION (Traces). Let Γ be the conclusion and Γ' a premise of a rule R in Figure 5.4. The *upward trace relation* $\mathsf{T}_{\Gamma,\mathsf{R},\Gamma'} \subseteq \Gamma \times \Gamma' \times \mathbb{N}$ consists of triples (φ, ψ, k) such that φ and ψ are formulas, and either

(i) φ and ψ are inactive, $\varphi = \psi$ and $k = 1$, or

(ii) φ and ψ are active and either

  (a) φ is the principal formula, ψ is an auxiliary formula of R and $k = \Omega_{2\mu}(\varphi)$, or

  (b) $\mathsf{R} = \langle a \rangle$ such that $\varphi = [a]\psi$ and $k = 1$.

Let u, v be nodes in an NW_2-derivation such that either $u = v$ or such that v is a child of u. We define the *trace relation* $\mathsf{T}_{u,v} \subseteq \mathsf{S}_u \times \mathsf{S}_v \times \mathbb{N}$ as follows. If $u = v$ we define $\mathsf{T}_{u,u} := \{(\varphi, \psi, k) \mid \varphi \rightsquigarrow_k \psi \in \mathsf{S}_u\}$ and call (φ, ψ, k) a *detour trace*. Otherwise v is a child of u and we define $\mathsf{T}_{u,v} := \mathsf{T}_{\mathsf{S}(u),\mathsf{R}(u),\mathsf{S}(v)}$.

Let $\beta = (v_i)_{i<\kappa}$ be a path in an NW_2-derivation π. A *trace* on β is a sequence of upward traces with inserted detour traces. Due to the presence of cuts, traces do not necessarily start at the root, but could also start from a cut formula. Formally, a *trace* τ on β is a word in

$$\mathsf{T}_{v_N,v_{N+1}}(\mathsf{T}_{v_{N+1},v_{N+1}})^*\mathsf{T}_{v_{N+1},v_{N+2}}(\mathsf{T}_{v_{N+2},v_{N+2}})^* \cdots$$

for some $N \geq 0$ and such that for any subword $(\varphi, \psi, k)(\chi, \zeta, l)$ of τ it holds that $\psi = \chi$. An infinite trace τ is called a *μ-trace* if the maximum number in $\{k \mid k \text{ appears infinitely often on } \tau\}$ is even and a *ν-trace* otherwise. An infinite path β in an NW_2-derivation is called *successful* if there is a μ-trace on β.

5.1.8. DEFINITION. The infinitary proof system NW_2 is defined from the rules in Figure 5.4 together with all successful paths.

5.1.9. EXAMPLE. In Example 2.4.4 we saw that $\langle a \rangle p \vDash \nu x.\langle a \rangle \langle \breve{a} \rangle x$. Therefore, the sequent $\langle a \rangle p, \mu x.[a][\breve{a}]x$ is unsatisfiable and we can give an NW_2-proof π of it. For convenience, we define $\varphi := \mu x.[a][\breve{a}]x$. We have that $\Omega_{2\mu}(\varphi) = 2$. The proof π is given as follows.

$$\dfrac{\dfrac{\dfrac{}{p, [\breve{a}]\varphi, [\breve{a}]\varphi \rightsquigarrow_2 [\breve{a}]\varphi}\ \mathsf{Ax4}}{\langle a \rangle p, [a][\breve{a}]\varphi, \varphi \rightsquigarrow_2 [a][\breve{a}]\varphi}\ \langle a \rangle}{\langle a \rangle p, \mu x.[a][\breve{a}]x}\ \mu$$

Note that $[\breve{a}]\varphi \in \mathsf{Clos}^{\neg}([a][\breve{a}]\varphi)$ implying that the trace atom $[\breve{a}]\varphi \rightsquigarrow_2 [\breve{a}]\varphi$ is added in the premise of $\langle a \rangle$.

5.1.10. Example. To motivate the use of acut we consider another example. In Example 2.4.5 we saw that $q \to \nu x.[a]x \wedge \mu y.\langle \breve{a}\rangle y \vee q$ is valid. Therefore, the sequent $q, \mu x.\langle a\rangle x \vee \nu y.[\breve{a}]y \wedge \overline{q}$ is unsatisfiable and we can give an NW_2-proof π of it. Let us define

$$\begin{aligned}\varphi &:= \mu x.\langle a\rangle x \vee \nu y.[\breve{a}]y \wedge \overline{q},\\ \psi &:= \nu y.[\breve{a}]y \wedge \overline{q}.\end{aligned}$$

We have $\Omega_{2\mu}(\varphi) = 2$ and $\Omega_{2\mu}(\psi) = 1$. We omit the trace atoms as they are not relevant in this example, and apply weak implicitly. The proof π is defined as follows, where ρ is given below.

$$\dfrac{\dfrac{\dfrac{\dfrac{\rho}{\langle a\rangle\varphi, \overline{\psi}} \qquad \dfrac{\dfrac{\dfrac{}{[\breve{a}]\psi, \overline{q}, q}\,\mathsf{Ax1}}{[\breve{a}]\psi \wedge \overline{q}, q}\,\wedge}{\psi, q}\,\nu}{q, \langle a\rangle\varphi}\,\mathsf{acut} \qquad \dfrac{\dfrac{\dfrac{}{q, [\breve{a}]\psi, \overline{q}}\,\mathsf{Ax1}}{q, [\breve{a}]\psi \wedge \overline{q}}\,\wedge}{q, \nu y.[\breve{a}]y \wedge \overline{q}}\,\nu}{q, \langle a\rangle\varphi \vee \nu y.[\breve{a}]y \wedge \overline{q}}\,\vee}{q, \mu x.\langle a\rangle x \vee \nu y.[\breve{a}]y \wedge \overline{q}}\,\mu$$

The proof ρ is given as follows, where ρ' is isomorphic to ρ.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\rho'}{\langle a\rangle\varphi, \overline{\psi}} \qquad \dfrac{\dfrac{\dfrac{}{[\breve{a}]\psi, \overline{q}, \langle\breve{a}\rangle\overline{\psi}}\,\mathsf{Ax1}}{[\breve{a}]\psi \wedge \overline{q}, \langle\breve{a}\rangle\overline{\psi}}\,\wedge}{\psi, \langle\breve{a}\rangle\overline{\psi}}\,\nu}{\langle a\rangle\varphi, \langle\breve{a}\rangle\overline{\psi}}\,\mathsf{acut} \qquad \dfrac{\dfrac{\dfrac{}{[\breve{a}]\psi, \overline{q}, \langle\breve{a}\rangle\overline{\psi}}\,\mathsf{Ax1}}{[\breve{a}]\psi \wedge \overline{q}, \langle\breve{a}\rangle\overline{\psi}}\,\wedge}{\nu y.[\breve{a}]y \wedge \overline{q}, \langle\breve{a}\rangle\overline{\psi}}\,\nu}{\langle a\rangle\varphi \vee \nu y.[\breve{a}]y \wedge \overline{q}, \langle\breve{a}\rangle\overline{\psi}}\,\vee}{\varphi, \langle\breve{a}\rangle\overline{\psi}}\,\mu}{\langle a\rangle\varphi, \overline{\psi}}\,\langle a\rangle$$

Note that $\psi \in \mathsf{Clos}^{\neg}(\varphi)$ and therefore the applications of acut are analytic. The single infinite branch of π carries the following μ-trace:

$$\varphi \to_C \langle a\rangle\varphi \vee \nu y.[\breve{a}]y \wedge \overline{q} \to_C \langle a\rangle\varphi \to_C \varphi \to_C \langle a\rangle\varphi \vee \nu y.[\breve{a}]y \wedge \overline{q} \to_C \langle a\rangle\varphi \cdots$$

Therefore, π is an NW_2-proof of $q, \mu x.\langle a\rangle x \vee \nu y.[\breve{a}]y \wedge \overline{q}$.

5.1.3 Proof search game

We define the proof search game, where one player (named Prover) tries to prove a sequent Γ, while the other player (named Builder) aims to refute it. If Builder wins the game, we can define a model $\mathbb{S}$ from the game and relate the proof search game with the evaluation game $\mathcal{E}_\mu(\mathbb{S})$.

Let

$$\frac{\Delta_1 \quad \cdots \quad \Delta_n}{\Delta}\ \mathsf{R}$$

be a rule in NW_2. We let conc be the function mapping rules to their conclusions. A rule is *cumulative* if all premises are supersets of the conclusion and *productive* if all the premises are distinct from its conclusion.

We define the *proof search game* $\mathcal{G}(\Phi)$, here we call the two players Prover and Builder. Its positions are given by $\mathsf{Seq}_\Phi \cup \mathsf{Rules}_\Phi$, where Seq_Φ is the set of sequences and Rules_Φ the set of NW_2 rules containing only formulas and trace atoms in $\mathsf{Clos}^\neg(\Phi)$. The ownership function and admissible moves are given in Table 5.1.

Position	Owner	Admissible moves
Δ	Prover	$\{R \in \mathsf{Rules}_\Phi \mid \mathsf{conc}(R) = \Delta\}$
$\frac{\Delta_1 \quad \cdots \quad \Delta_n}{\Delta}\ \mathsf{R}$	Builder	$\{\Delta_i \mid i = 1, \dots, n\}$

Table 5.1: The proof search game $\mathcal{G}(\Phi)$

As usual, finite matches are lost by the player who gets stuck. An infinite match $\mathcal{M}$ corresponds to an infinite NW_2 branch $\beta_\mathcal{M}$ and is won by Prover iff $\beta_\mathcal{M}$ carries a μ-trace.

As we will see in Subsection 5.2.1, the game $\mathcal{G}(\Phi)$ is ω-regular. It follows from Proposition 2.2.10 that ω-regular games have finite-memory strategies. In particular, the strategy trees of Prover's winning strategies may be assumed to be regular trees.

Strategy trees of Prover's winning strategies in $\mathcal{G}(\Phi)@\Gamma$ can be identified with NW_2-proofs of Γ. Therefore, we may assume that NW_2-proofs are regular.

5.1.4 Soundness of NW_2

For proving soundness we need to show that, if NW_2 proves Γ, then Γ is unsatisfiable. By contradiction we assume that Γ is satisfiable and show that NW_2 does not prove Γ. To do so, we assume a pointed model $\mathbb{S}, s$ and a strategy f for $\exists$ in $\mathcal{E}_\mu(\mathbb{S})$ such that $\mathbb{S}, s \Vdash_f \Gamma$. Using f we will construct a winning strategy $\overline{f}$ for Builder in $\mathcal{G}(\Phi)@\Gamma$.

The following lemma deals with the local soundness of our rules.

5.1.11. Lemma. *Let*

$$\frac{\Delta_1 \quad \cdots \quad \Delta_n}{\Delta}\ \mathsf{R}$$

be a rule of NW_2*. If* Δ *is satisfiable, then there is an* $i = 1, ..., n$ *such that* Δ_i *is satisfiable.*

In particular, if $\mathsf{R} \neq \langle a \rangle$*, and given a pointed model* $\mathbb{S}, s$ *and positional strategy* f *for* $\exists$ *in* $\mathcal{E}_\mu(\mathbb{S})$ *such that* $\mathbb{S}, s \Vdash_f \Delta$*, then* $\mathbb{S}, s \Vdash_f \Delta_i$*.*

If $\mathsf{R} = \vee$ *with principal formula* $\varphi_0 \vee \varphi_1$ *such that* $\mathbb{S}, s \Vdash_f \varphi_0 \vee \varphi_1, \Gamma$*, then* $\mathbb{S}, s \Vdash_f \varphi_i, \varphi_0 \vee \varphi_1 \rightsquigarrow_1 \varphi_i, \Gamma$*, where* $f(\varphi_0 \vee \varphi_1, s) = (\varphi_i, s)$*.*

If $\mathsf{R} = \langle a \rangle$ *with principal formula* $\langle a \rangle \varphi$ *such that* $\mathbb{S}, s \Vdash_f \langle a \rangle \varphi, [a]\Sigma, \Gamma$*, then* $\mathbb{S}, t \Vdash_f \varphi, \Sigma, \langle \breve{a} \rangle \Gamma, \Gamma^{\langle a \rangle \varphi}$*, where* $f(\langle a \rangle \varphi, s) = (\varphi, t)$*.*

Proof:
The soundness of the modal rule is shown in Lemma 5.1.5. For the other rules the proof is straightforward and will be omitted. □

5.1.12. Theorem (Soundness). *If* $\mathsf{NW}_2 \vdash \Gamma$*, then* Γ *is unsatisfiable.*

Proof:
By contraposition we show that, if Γ is satisfiable, then Builder has a winning strategy in $\mathcal{G} := \mathcal{G}(\Phi)@\Gamma$. So assume that there is a pointed model $\mathbb{S}, s$ and a positional strategy f for $\exists$ in the game $\mathcal{E} := \mathcal{E}_\mu(\mathbb{S})$ such that $\mathbb{S}, s \Vdash_f \Gamma$. We will construct a winning strategy $\overline{f}$ for Builder in $\mathcal{G}$ and a function $s_f : PM(\Phi) \to \mathbb{S}$, mapping partial $\mathcal{G}$-matches to states of $\mathbb{S}$, such that $\mathbb{S}, s_f(\mathcal{M}) \Vdash_f \mathsf{last}(\mathcal{M})$ for every $\overline{f}$-guided $\mathcal{M} \in PM_{Prover}(\Phi)$.

The functions $\overline{f}$ and s_f can be defined inductively by a case distinction based on the rule. For the base case $|\mathcal{M}| = 1$ it holds that $\mathcal{M} = \Gamma$. We define $s_f(\mathcal{M}) := s$ and do not have to define $\overline{f}$ as this is a position owned by Prover. For the inductive case we follow the specifications of the rule. If the rule is $\langle a \rangle$, define s_f as given by f and let $\overline{f}$ choose the only premise. For any other rule, the definition of s_f remains the same and we invoke Lemma 5.1.11 for the definition of $\overline{f}$.

We need to show that $\overline{f}$ is a winning strategy for Builder in $\mathcal{G}$. Because of Lemma 5.1.11 we know that all finite matches are won by Builder. Thus, assume by contradiction that Prover wins an infinite $\overline{f}$-guided $\mathcal{G}$-match $\mathcal{M}$. Then there is a μ-trace $\tau = \tau_0\tau_1 \cdots$ on $\mathcal{M}$. Note that τ is not necessarily a trace starting from the root; it might also be starting from a cut formula. We will use τ to obtain an infinite f-guided $\mathcal{E}$-match $\mathcal{N}$ that is won by $\forall$.

Let $\tau_i = (\varphi_i, \psi_i, k_i)$ for $i \in \omega$, recall that $\varphi_{i+1} = \psi_i$ for all $i \in \omega$. For each i, the triple τ_i is in $\mathsf{T}_{u(i),v(i)}$ for some nodes $u(i)$ and $v(i)$. Let $\mathcal{M}_i$ be the initial partial match of $\mathcal{M}$, such that $\varphi_i \in \mathsf{S}_{u(i)}$. We will define f-guided partial $\mathcal{E}$-matches $\mathcal{N}_i$ starting at $(\varphi_i, s_f(\mathcal{M}_i))$ and ending at $(\varphi_{i+1}, s_f(\mathcal{M}_{i+1}))$ for every $i \in \omega$, such that $k_i = \max\{\Omega_{2\mu}(\varphi) \mid (\varphi, s) \text{ is a position in } \mathcal{N}_i \text{ for some } s\}$.

For an upward trace τ_i in $\mathsf{T}_{u,v}$ we can define $\mathcal{N}_i$ straightforwardly. Otherwise, τ_i is a detour trace in $\mathsf{T}_{u,u}$ for some u. Then $\varphi_i \rightsquigarrow_{k_i} \varphi_{i+1} \in \mathsf{last}(\mathcal{M}_i)$. As $\mathbb{S}, s_f(\mathcal{M}_i) \Vdash_f \mathsf{last}(\mathcal{M}_i)$, it holds that $\mathbb{S}, s_f(\mathcal{M}_i) \Vdash_f \varphi_i \rightsquigarrow_{k_i} \varphi_{i+1}$. This exactly means that there is an f-guided match $\mathcal{N}_i$ starting at $(\varphi_i, s_f(\mathcal{M}_i))$ and ending at $(\varphi_{i+1}, s_f(\mathcal{M}_i))$ as needed.

Glueing together the matches $\mathcal{N}_i$ we obtain an infinite f-guided $\mathcal{E}$-match $\mathcal{N} = \mathcal{N}_0\mathcal{N}_1\cdots$, such that $\max\{\Omega_{2\mu}(\varphi) \mid \varphi \text{ occurs infinitely often in } \mathcal{N}\} = \max\{k \mid k \text{ appears infinitely often on } \tau\}$. Thus we conclude that $\mathcal{N}$ is won by $\forall$ and therefore $\mathbb{S}, s_f(\mathcal{M}_0) \nVdash_f \mathsf{last}(\mathcal{M}_0)$, which contradicts that $\mathsf{last}(\mathcal{M}_0)$ is satisfiable. □

5.1.5 Completeness

To prove completeness for pure sequents Γ, we follow the same proof strategy as in [RV23]. We show that every unsatisfiable pure sequent is provable in NW_2. By contraposition, given a winning strategy f for Builder in $\mathcal{G}(\Phi)@\Gamma$, we construct a model $\mathbb{S}^f$ and a positional strategy $\underline{f}$ for $\exists$ in $\mathcal{E} := \mathcal{E}_\mu(\mathbb{S}^f)$ such that $\mathbb{S}^f \Vdash_{\underline{f}} \Gamma$.

Let $\mathcal{T}$ be the maximal subgraph of the game tree of $\mathcal{G}(\Phi)@\Gamma$, where Builder plays the strategy f and Prover picks rules according to the following priorities:

1. axioms $\mathsf{Ax1}$, $\mathsf{Ax2}$, $\mathsf{Ax3}$ and $\mathsf{Ax4}$ preceded by weak;
2. cumulative and productive rules $\vee$, $\wedge$, μ, ν, trans, acut and tcut;
3. modal rules $\langle a \rangle$.

We say that a trace atom $\varphi \rightsquigarrow_k \psi$ is *relevant* if (i) $\psi \in \mathsf{Clos}(\varphi)$ and (ii) φ, ψ contain fixpoints. We restrict the rules $\vee$, $\wedge$, tcut and $\langle a \rangle$ to only introduce relevant trace atoms. For the rules $\vee$, $\wedge$ and $\langle a \rangle$ this amounts to changing the rule to a variant, where only relevant trace atoms occur in the premise. This rule can easily shown to be admissible using weak. Because of these assumptions we may assume that all trace atoms in a constructed proof are relevant.

The model $\mathbb{S}^f := (S^f, \{R^f_a\}_{a \in \mathsf{Act}}, V^f)$ is defined as follows: The set S^f of states consists of all maximal upward paths ρ in $\mathcal{T}$ not containing a modal rule. In order to define the accessibility relations, we write $\rho_1 \xrightarrow{a} \rho_2$ if ρ_2 is directly above ρ_1 only separated by an application of $\langle a \rangle$. The relations R^f_a are defined as follows:

$$\rho_1 R^f_a \rho_2 \quad :\Leftrightarrow \quad \rho_1 \xrightarrow{a} \rho_2 \text{ or } \rho_2 \xrightarrow{\breve{a}} \rho_1.$$

The sequent $\mathsf{S}(\rho)$ of a path ρ is defined as $\mathsf{S}(\rho) := \bigcup\{\Delta \mid \Delta \text{ occurs in } \rho\}$ and we define the valuation $V^f(p)$ as $V^f(p) := \{\rho \mid p \in \mathsf{S}(\rho)\}$.

The strategy $\underline{f}$ for $\exists$ in $\mathcal{E}$ is defined as follows:

- At $(\varphi_0 \vee \varphi_1, \rho)$ pick a disjunct φ_i such that $\varphi_i \in \mathsf{S}(\rho)$.
- At $(\langle a \rangle\varphi, \rho)$ choose (φ, τ) for some τ such that $\rho \xrightarrow{a} \tau$, where the principal formula in the rule $\langle a \rangle$ between ρ and τ is $\langle a \rangle\varphi$.

Let ρ_0 be a state of $\mathbb{S}^f$ containing the root Γ of $\mathcal{T}$ and let $\varphi_0 \in \Gamma$. We will show that the strategy $\underline{f}$ is well-defined and winning for $\exists$ in $\mathcal{E}@(\varphi_0, \rho_0)$. From this completeness follows.

We will now formalize the outlined strategy. First, we need to gather information about the sequents $\mathsf{S}(\rho)$, where ρ is a local path in $\mathbb{S}^f$.

5.1.13. Lemma (Saturation). *For every state ρ in $\mathbb{S}^f$ the set $\mathsf{S}(\rho)$ is* saturated*, meaning that the following conditions are satisfied:*

1. *For all $\varphi \in \mathsf{Clos}^\neg(\mathsf{S}(\rho))$ it holds that $\varphi \in \mathsf{S}(\rho)$ iff $\overline{\varphi} \notin \mathsf{S}(\rho)$.*
2. *Never $\bot \in \mathsf{S}(\rho)$.*
3. *For all $\varphi \in \mathsf{Clos}^\neg(\mathsf{S}(\rho))$ and relevant trace atoms $\varphi \rightsquigarrow_k \psi$ it holds that $\varphi \rightsquigarrow_k \psi \in \mathsf{S}(\rho)$ or $\varphi \not\rightsquigarrow_k \psi \in \mathsf{S}(\rho)$.*
4. *For no trace atom $\varphi \rightsquigarrow_k \psi$ it holds that $\varphi \rightsquigarrow_k \psi \in \mathsf{S}(\rho)$ and $\varphi \not\rightsquigarrow_k \psi \in \mathsf{S}(\rho)$.*
5. *For no φ, k it holds that $\varphi \rightsquigarrow_{2k} \varphi \in \mathsf{S}(\rho)$.*
6. *If $\varphi_0 \wedge \varphi_1 \in \mathsf{S}(\rho)$, then for both $i = 0, 1$ it holds that $\varphi_i \in \mathsf{S}(\rho)$, and $\varphi_0 \wedge \varphi_1 \rightsquigarrow_1 \varphi_i \in \mathsf{S}(\rho)$ if the trace atom is relevant.*
7. *If $\varphi_0 \vee \varphi_1 \in \mathsf{S}(\rho)$, then for some $i = 0, 1$ it holds that $\varphi_i \in \mathsf{S}(\rho)$, and $\varphi_0 \vee \varphi_1 \rightsquigarrow_1 \varphi_i \in \mathsf{S}(\rho)$ if the trace atom is relevant.*
8. *If $\eta x.\varphi \in \mathsf{S}(\rho)$, then $\varphi[\eta x.\varphi/x], \eta x.\varphi \rightsquigarrow_{\Omega_{2\mu}(\eta x.\varphi)} \varphi[\eta x.\varphi/x] \in \mathsf{S}(\rho)$.*
9. *If $\varphi \rightsquigarrow_k \psi, \psi \rightsquigarrow_l \chi \in \mathsf{S}(\rho)$, then $\varphi \rightsquigarrow_{\max\{k,l\}} \chi \in \mathsf{S}(\rho)$.*

Proof:
The lemma follows from our restriction on the strategy of Prover. We only show the exemplary cases 2 and 3. If $\bot \in \mathsf{S}(\rho)$, then for some node $v \in \rho$, Prover can apply weak and $\mathsf{Ax2}$. Yet, this contradicts the fact that Builder's strategy is winning.

Regarding item 3, assume that $\varphi \in \mathsf{Clos}^\neg(\mathsf{S}(\rho))$ and that $\varphi \rightsquigarrow_k \psi$ is a relevant trace atom. If $\varphi \rightsquigarrow_k \psi \notin \mathsf{S}(\rho)$ and $\varphi \not\rightsquigarrow_k \psi \notin \mathsf{S}(\rho)$, then at any node v in ρ a cumulative and productive acut rule with cut-formula $\varphi \rightsquigarrow_k \psi$ is applicable. Because Prover prioritizes picking this rule over a modal rule, the claim follows. □

5.1.14. Lemma (Truth Lemma). *Let ρ_0 be a state of $\mathbb{S}^f$ containing the root Γ of $\mathcal{T}$ and let $\psi_0 \in \Gamma$. Let $\mathcal{M}$ be an $\underline{f}$-guided $\mathcal{E}$-match with starting position (ψ_0, ρ_0). Then for every position (ψ, ρ) in $\mathcal{M}$ it holds that $\psi \in \mathsf{S}(\rho)$.*

Proof:
We write (ψ_n, ρ_n) for the n-th position of $\mathcal{M}$ and prove the claim by strong induction on n. The base case is clear. For the induction step let $\psi_n \in \mathsf{S}(\rho_n)$, we have to show that $\psi_{n+1} \in \rho_{n+1}$. We proceed with a case distinction based on the shape of ψ. If ψ is not a modal formula, then $\rho_{n+1} = \rho_n$ and the claim follows from Lemma 5.1.13 and the definition of $\underline{f}$.

Assume that $\psi_n = [a]\chi$, then $\psi_{n+1} = \chi$. In this case $\rho_n R_a^f \rho_{n+1}$, so either $\rho_n \xrightarrow{a} \rho_{n+1}$ or $\rho_{n+1} \xrightarrow{\breve{a}} \rho_n$. If $\rho_n \xrightarrow{a} \rho_{n+1}$, then $[a]\chi$ is in the conclusion of $\langle a \rangle$, hence χ is in its premise and thus $\chi \in \rho_{n+1}$.

Next consider the case where $\rho_{n+1} \xrightarrow{\breve{a}} \rho_n$. Because $\mathbb{S}^f$ is a forest and $\rho_0 ... \rho_{n+1}$ forms a path in $\mathbb{S}^f$ starting at one of the roots, where the last step of the path is downwards, there has to be an $i \in \{0, ..., n-1\}$ with $\rho_i = \rho_{n+1}$. As $\mathcal{M}$ is a match with positions (ψ_i, ρ_i) for some ψ_i and $([a]\chi, \rho_n)$, it holds that $[a]\chi \in \mathsf{Clos}(\psi_i)$. As by induction hypothesis $\psi_i \in \mathsf{S}(\rho_i) = \mathsf{S}(\rho_{n+1})$, this yields $[a]\chi \in \mathsf{Clos}(\mathsf{S}(\rho_{n+1}))$ and also $\chi \in \mathsf{Clos}(\mathsf{S}(\rho_{n+1}))$.

Towards a contradiction assume that $\chi \notin \mathsf{S}(\rho_{n+1})$. Because $\chi \in \mathsf{Clos}^{\neg}(\mathsf{S}(\rho_{n+1}))$ it holds that $\overline{\chi} \in \mathsf{S}(\rho_{n+1})$ by Lemma 5.1.13. If $\overline{\chi}$ is in the conclusion of $\langle \breve{a} \rangle$, then $\langle a \rangle \overline{\chi}$ is in its premise as $\langle a \rangle \overline{\chi} \in \mathsf{Clos}^{\neg}(\mathsf{S}(\rho_{n+1}))$, therefore $\langle a \rangle \overline{\chi} \in \rho_n$. Again by Lemma 5.1.13 we conclude that $[a]\chi \notin \rho_n$, which is a contradiction.

Finally the case where $\psi_n = \langle a \rangle \chi$ is similar to the first direction of the previous case. □

5.1.15. Lemma (Truth Lemma for trace atoms). *Let $\rho \in S^f$, $\varphi \in \mathsf{S}(\rho)$ and $\varphi \rightsquigarrow_k \psi$ be a relevant trace atom. If $\mathbb{S}^f, \rho \Vdash_{\underline{f}} \varphi \rightsquigarrow_k \psi$, then $\varphi \rightsquigarrow_k \psi \in \mathsf{S}(\rho)$.*

Proof:
Let $\mathcal{N}$ be an $\underline{f}$-guided $\mathcal{E}$-match witnessing $\mathbb{S}^f, \rho \Vdash_{\underline{f}} \varphi \rightsquigarrow_k \psi$, meaning that $\mathcal{N}$ is of the form

$$(\varphi, \rho) = (\varphi_0, \rho_0) \cdots (\varphi_n, \rho_n) = (\psi, \rho), \qquad n > 0$$

such that $k = \max\{\Omega_{2\mu}(\varphi_i) \mid i = 0, \ldots, n-1\}$.

We prove the lemma by an induction on the number of distinct states occurring in $\mathcal{N}$. For the base case, where the only state occurring in $\mathcal{N}$ is ρ, we proceed with an inner induction on the length of $\mathcal{N}$. The claim then follows straightforwardly from the definition of $\underline{f}$ and the fact that $\mathsf{S}(\rho)$ is saturated (Lemma 5.1.13).

For the induction step let $\mathcal{N} = \mathcal{A}_1 \mathcal{B}_1 \cdots \mathcal{B}_{l-1} \mathcal{A}_l$, such that for any position (χ, τ) in $\mathcal{N}$ we have: If $\tau = \rho$, then $(\chi, \tau) \in \mathcal{A}_i$ for some $i = 1, ..., l$; and if $\tau \neq \rho$, then $(\chi, \tau) \in \mathcal{B}_i$ for some $i = 1, ..., l-1$. Because $\mathbb{S}^f$ is a forest it follows that there is τ_i such that $\mathsf{first}(\mathcal{B}_i) = (\beta_i, \tau_i)$ and $\mathsf{last}(\mathcal{B}_i) = (\delta_i, \tau_i)$ for $i = 1, ..., l-1$. We will fix the following notation for each i:

$$\mathsf{first}(\mathcal{A}_i) = (\alpha_i, \rho) \qquad \mathsf{first}(\mathcal{B}_i) = (\beta_i, \tau_i),$$
$$\mathsf{last}(\mathcal{A}_i) = (\gamma_i, \rho) \qquad \mathsf{last}(\mathcal{B}_i) = (\delta_i, \tau_i),$$

By the base case of the induction it holds that $\alpha_i \rightsquigarrow_{k_i} \gamma_i \in \mathsf{S}(\rho)$ for some $k_i \leq k$. For readability we will omit the subscripts in the trace atoms in the rest of the proof, for instance, we will write $\alpha_i \rightsquigarrow \gamma_i$ instead of $\alpha_i \rightsquigarrow_{k_i} \gamma_i$. As we closely follow the match $\mathcal{N}$ we will end up with a trace atom of the form $\varphi \rightsquigarrow_k \psi$, where k is as required. In the matches $\mathcal{B}_i$ less states occur as in $\mathcal{N}$, therefore we may apply the induction hypothesis to obtain $\beta_i \rightsquigarrow \delta_i \in \mathsf{S}(\tau_i)$ for $i = 1, ..., l-1$. We aim to show that $\gamma_i \rightsquigarrow \alpha_{i+1} \in \mathsf{S}(\rho)$ for $i = 1, ..., l-1$.

Since the match $\mathcal{N}$ transitions from ρ to τ_i it holds that $\rho R^f_a \tau_i$ for some action a. First consider the case that $\rho \xrightarrow{a} \tau_i$, then γ_i must be of the form $\langle a \rangle \beta_i$ or of the form $[a]\beta_i$. Looking at the transition from τ_i to ρ, because by the definition of $\underline{f}$ Eloise only moves upwards in $\mathbb{S}^f$, it must hold that $\delta_i = [\breve{a}]\alpha_{i+1}$. We then obtain:

$$
\begin{aligned}
& \beta_i \rightsquigarrow [\breve{a}]\alpha_{i+1} \in \mathsf{S}(\tau_i) && \text{(Induction hypothesis)} \\
\Rightarrow\ & \beta_i \not\rightsquigarrow [\breve{a}]\alpha_{i+1} \notin \mathsf{S}(\tau_i) && \text{(Saturation)} \\
\Rightarrow\ & \gamma_i \not\rightsquigarrow \alpha_{i+1} \notin \mathsf{S}(\rho) && \text{(Definition of } \langle a \rangle) \\
\Rightarrow\ & \gamma_i \rightsquigarrow \alpha_{i+1} \in \mathsf{S}(\rho) && \text{(Saturation)}
\end{aligned}
$$

Note that in the second implication we use the fact that $[\breve{a}]\alpha_{i+1} \in \mathsf{Clos}^{\neg}(\mathsf{S}(\rho))$. This follows as $\varphi \in \mathsf{S}(\rho)$ by assumption and $[\breve{a}]\alpha_{i+1} \in \mathsf{Clos}(\varphi)$. In the third implication we rely on $\gamma_i \in \mathsf{Clos}(\varphi) \subseteq \mathsf{Clos}^{\neg}(\mathsf{S}(\rho))$.

Now consider the case that $\tau_i \xrightarrow{\breve{a}} \rho$. Then $\gamma_i = [a]\beta_i$ for some action a and δ_i is of the form $\langle \breve{a} \rangle \alpha_{i+1}$ or $[\breve{a}]\alpha_{i+1}$. Here we find:

$$
\begin{aligned}
& \beta_i \rightsquigarrow \delta_i \in \mathsf{S}(\tau_i) && \text{(Induction hypothesis)} \\
\Rightarrow\ & [a]\beta_i \rightsquigarrow \alpha_{i+1} \in \mathsf{S}(\rho) && \text{(Definition of } \langle a \rangle)
\end{aligned}
$$

For this implication to hold it is required that $[a]\beta_i \in \mathsf{Clos}^{\neg}(\mathsf{S}(\tau_i))$. As $[a]\beta_i \in \mathsf{Clos}(\mathsf{S}(\rho))$ and all NW_2 rules are analytic, this implies $[a]\beta_i \in \mathsf{Clos}^{\neg}(\mathsf{S}(\tau_i))$ indeed.

In both cases $\gamma_i \rightsquigarrow \alpha_{i+1} \in \mathsf{S}(\rho)$ and, as also $\alpha_i \rightsquigarrow \gamma_i \in \mathsf{S}(\rho)$ for $i = 1, ..., l$, we can combine these statements using saturation to obtain $\alpha_1 \rightsquigarrow \gamma_l \in \mathsf{S}(\rho)$. Because we closely followed the match $\mathcal{N}$ this yields $\varphi \rightsquigarrow_k \psi \in \mathsf{S}(\rho)$ as required.
□

5.1.16. PROPOSITION. *Let ρ_0 be a state of $\mathbb{S}^f$ containing the root Γ of $\mathcal{T}$ and let $\psi_0 \in \Gamma$. Then the strategy $\underline{f}$ is winning for $\exists$ in $\mathcal{E}@(\psi_0, \rho_0)$.*

Proof:
Let $\mathcal{M}$ be an arbitrary $\underline{f}$-guided $\mathcal{E}@(\psi_0, \rho_0)$-match. If $\mathcal{M}$ is a finite match, then it is straightforward to check that it is won by $\exists$.

Suppose that $\mathcal{M} = (\psi_n, \rho_n)_{n \in \omega}$ is infinite, and to arrive at a contradiction assume that $\forall$ wins $\mathcal{M}$. By positional determinacy we may assume that his strategy is positional. We make a case distinction.

First assume that there is a state ρ that is visited infinitely often. Then there must be a segment $\mathcal{N}$ of $\mathcal{M}$ such that $\mathsf{first}(\mathcal{N}) = \mathsf{last}(\mathcal{N}) = (\psi, \rho)$ for some formula ψ. As the match is positional this means that $\mathcal{M} = \mathcal{K}\mathcal{N}^*$ for some initial segment $\mathcal{K}$ of $\mathcal{M}$, meaning that only finitely many states are visited. By our assumption the match $\mathcal{M}$ is winning for $\forall$, thus the most important fixpoint formula occurring infinitely often is of the form $\mu x.\psi$. Let $k = \Omega_{2\mu}(\mu x.\psi)$. Because only finitely many states are visited, there has to be a position $(\tau, \mu x.\psi)$ occurring infinitely often in $\mathcal{M}$ and thus $\mathbb{S}^f, \tau \Vdash_{\underline{f}} \mu x.\psi \rightsquigarrow_k \mu x.\psi$. Then Lemma 5.1.14 yields $\mu x.\psi \in \mathsf{S}(\tau)$ and Lemma 5.1.15 gives $\mu x.\psi \rightsquigarrow_k \mu x.\psi \in \mathsf{S}(\tau)$. But this contradicts Lemma 5.1.13 as k is even.

Now consider the case that $\mathcal{M} = (\psi_n, \rho_n)_{n\in\omega}$ visits each state at most finitely often. Then there are sequences of indices $(i(n))_{n\in\omega}, (j(n))_{n\in\omega} \in \omega^\omega$ such that $i(n)$ and $j(n)$ are the respective first and last index l such that ρ_l is the n-th distinct state in $\mathcal{M}$. Formally, the indices $i(n)$ and $j(n)$ for $n \in \omega$ satisfy

1. $i(n) \leq j(n)$ and $j(n) + 1 = i(n+1)$ for all $n \in \omega$,
2. $\rho_{i(n)} = \rho_{j(n)}$ for all $n \in \omega$ and
3. $\psi_{j(n)}$ is modal for every $n \in \omega$ and there is an action a_n such that $\rho_{j(n)} \xrightarrow{a_n} \rho_{i(n+1)}$.

These indices can be defined by induction rather straightforwardly.

Assume that $\mathcal{M}$ is winning for $\forall$, then there is $N \in \omega$ such that for some even k it holds that $\Omega_{2\mu}(\psi_n) \leq k$ for all $n \geq N$ and $\Omega_{2\mu}(\psi_n) = k$ for infinitely many $n \geq N$. It holds that $\mathbb{S}^f, \rho_{i(n)} \Vdash_{\underline{f}} \psi_{i(n)} \rightsquigarrow_{k(n)} \psi_{j(n)}$, where $k(n) \leq k$ for all $n \geq N$ and $k(n) = k$ for infinitely many $n > N$. Lemma 5.1.15 together with Lemma 5.1.14 yields that $\psi_{i(n)} \rightsquigarrow_{k(n)} \psi_{j(n)} \in \mathsf{S}(\rho_{i(n)})$ and because Prover only applies cumulative rules in $\mathcal{T}$ this implies $\psi_{i(n)} \rightsquigarrow_{k(n)} \psi_{j(n)} \in \mathsf{last}(\rho_{i(n)})$.

Clearly there is a trace τ_n from $\psi_{j(n)}$ at $\mathsf{last}(\rho_{i(n)})$ to $\psi_{i(n+1)}$ at $\mathsf{first}(\rho_{i(n+1)})$ of weight 1; there is only one modal rule applied. Again because Prover only applies cumulative rules in $\mathcal{T}$ there are traces τ'_n of weight 1 from $\psi_{i(n)}$ at $\mathsf{first}(\rho_{i(n)})$ to $\psi_{i(n)}$ at $\mathsf{last}(\rho_{i(n)})$.

Thus we obtain the weighted trace

$$\tau = (\psi_{i(0)}, \psi_{j(0)}, k(0)) \cdot \tau_0 \cdot \tau'_1 \cdot (\psi_{i(1)}, \psi_{j(1)}, k(1)) \cdot \tau_1 \cdots$$

where $\max\{l \mid l \text{ appears infinitely often on } \tau\} = k$ is even and therefore τ is a μ-trace. Yet this contradicts the fact that $\mathcal{G}(\Phi)@\Gamma$ is winning for Builder. □

5.1.17. Theorem (Completeness). *If a pure sequent* Γ *is unsatisfiable, then there is a regular* NW_2*-proof of* Γ.

Proof:
By contraposition, assume that there is no regular NW_2-proof of Γ. Then Builder has a winning strategy f in $\mathcal{G}(\Phi)@\Gamma$. Proposition 5.1.16 shows that $\mathbb{S}^f, \rho_0 \Vdash \Gamma$, contradicting the assumption that Γ is unsatisfiable. □

5.2 Annotated proof system JS_2

In order to define the cyclic proof system JS_2 we want to apply the uniform construction from Section 4.1.1 to the infinitary proof system NW_2. For that aim we first need to define an automaton checking the success condition on infinite NW_2-branches. As traces may not only go upwards in a proof branch but also pass through trace atoms, the natural automaton to check this condition is a parity automaton with ε-transitions $\mathbb{A}_{2\mu}$. We will then use the determinization method from Section 3.3 to obtain the deterministic automaton $\mathbb{A}^S_{2\mu}$. Up to some technicalities we obtain the proof system JS_2 by applying the uniform construction to NW_2 and $\mathbb{A}^S_{2\mu}$.

5.2.1 Tracking automaton for NW_2

To simplify the definition of the tracking automaton $\mathbb{A}_{2\mu}$ we will first define a certain normal form on NW_2-proofs. On those proof it suffices to only track traces of a certain kind – so called slim traces.

We call an NW_2-proof π *saturated*, if

(i) the rule trans is always applied when applicable and

(ii) all applications of η rules are cumulative.

Note that every NW_2-proof π can easily be transformed into a saturated proof π' of the same sequent.

We call an infinite trace τ *slim*, if

(i) there are no two consecutive detour traces on τ and

(ii) there is no upward trace of the form $(\eta x.\varphi, \varphi[\eta x.\varphi/x], k)$ on τ.

5.2.1. LEMMA. *Let π be a saturated NW_2-proof of Γ. On every infinite branch of π there is a slim μ-trace.*

Proof:
Let π be a saturated NW_2-proof of Γ. Let β be a branch of π and τ be a μ-trace on β. Regarding condition (ii), assume that there is an upward trace $(\eta x.\varphi, \varphi[\eta x.\varphi/x], k) \in \mathsf{T}_{u,v}$ on τ. The sequent S_v contains $\eta x.\varphi \rightsquigarrow_k \varphi[\eta x.\varphi/x]$

and $\eta x.\varphi$, because the application of η is cumulative. Thus we can replace the trace $(\eta x.\varphi, \varphi[\eta x.\varphi/x], k) \in \mathsf{T}_{u,v}$ in τ with $(\eta x.\varphi, \eta x.\varphi, 1)(\eta x.\varphi, \varphi[\eta x.\varphi/x], k) \in \mathsf{T}_{u,v}\mathsf{T}_{v,v}$.

Regarding condition (i), we assume that detour traces in τ occur only at nodes labeled by a rule different from trans. This is not a restriction as all upward trace relations for the rule trans are of the form $(\varphi, \varphi, 1)$, thus we can apply the same detour trace at its child. Assume that there is a subword of τ consisting of two detour traces $(\varphi, \psi, k)(\psi, \chi, l)$, where $(\varphi, \psi, k), (\psi, \chi, l) \in \mathsf{T}_{u,u}$. As u is not labeled by trans and trans is always applied if applicable, also $(\varphi, \chi, \max\{k, l\}) \in \mathsf{T}_{u,u}$. Hence we can replace $(\varphi, \psi, k), (\psi, \chi, l)$ with $(\varphi, \chi, \max\{k, l\})$. Doing this for all upward traces of the form $(\eta x.\varphi, \varphi[\eta x.\varphi/x], k)$ and subwords consisting of two consecutive detour traces results in a slim μ-trace τ'. □

We will define a nondeterministic parity automaton $\mathbb{A}_{2\mu}$, called the *tracking automaton for* NW_2, that checks if an infinite branch β of a saturated NW_2-proof π is successful. Conceptually, the automaton $\mathbb{A}_{2\mu}$ non-deterministically follows the trace relation on β. The states of the automaton will be formulas and trace atoms, where we add extra states for every fixpoint formula, in order to track the unfolding of fixpoints. Additionally, we have an extra initial state, which is always reachable, as traces may start at any node.

Upward traces on β correspond to basic transitions in $\mathbb{A}_{2\mu}$ and detour traces are modelled by ε-transitions going through a trace atom. In order to simplify the automaton, we do not allow consecutive ε-transitions and ε-transitions starting from the auxiliary formula $\varphi[\eta x.\varphi/x]$ of the rule η. Hence $\mathbb{A}_{2\mu}$ will not follow all infinite traces, but only those of a simple form, in particular all slim traces.

5.2.2. Definition. The alphabet Σ consists of all triples $(\Gamma, \mathsf{R}, \Gamma')$, where $\Gamma \in \mathsf{Seq}_\Phi$ describes the conclusion, and $\Gamma' \in \mathsf{Seq}_\Phi$ describes a premise of a rule R in Figure 5.4. We define the following nondeterministic ε-parity automaton $\mathbb{A}_{2\mu} = (A, \Delta, a_I, \Omega_A)$ over Σ:

- $A := \{a_I\} \cup \Phi \cup \{\eta x.\psi^* \mid \eta x.\psi \in \Phi\} \cup \{\varphi \rightsquigarrow_k \psi \mid \varphi \rightsquigarrow_k \psi \text{ a trace atom}\}$.
- For each $\chi \in A$ and $(\Gamma, \mathsf{R}, \Gamma') \in \Sigma$ we define Δ_b as follows.
    1. if $\chi = a_I$, then $\Delta_b(\chi, (\Gamma, \mathsf{R}, \Gamma')) := \Gamma' \cup \{a_I\}$,
    2. if $\chi = \eta x.\psi$ is the principal formula of R, then $\Delta_b(\chi, (\Gamma, \mathsf{R}, \Gamma')) := \{\eta x.\psi^*\}$,
    3. else if $\chi = \varphi \in \Phi$ then $\Delta_b(\varphi, (\Gamma, \mathsf{R}, \Gamma')) := \{\varphi' \mid (\varphi, \varphi', 1) \in \mathsf{T}_{\Gamma,\mathsf{R},\Gamma'}\} \cup \{\varphi' \rightsquigarrow_k \psi' \mid (\varphi, \varphi', 1) \in \mathsf{T}_{\Gamma,\mathsf{R},\Gamma'} \;\&\; \psi', \varphi' \rightsquigarrow_k \psi' \in \Gamma'\}$,
    4. else $\Delta_b(\chi, (\Gamma, \mathsf{R}, \Gamma')) := \varnothing$.
- For each $\chi \in A$ we define Δ_ε as follows.

1. if $\chi = \eta x.\psi^*$, then $\Delta_\varepsilon(\eta x.\psi^*) := \{\eta x.\psi\}$,
2. if $\chi = \varphi \rightsquigarrow_k \psi$ then $\Delta_\varepsilon(\chi) := \{\psi\}$,
3. else $\Delta_\varepsilon(\chi) := \varnothing$.

- For states of the form $\eta x.\psi^*$ let $\Omega_A(\eta x.\psi^*) := \Omega_{2\mu}(\eta x.\psi)$. For states of the form $\varphi \rightsquigarrow_k \psi$ let $\Omega_A(\varphi \rightsquigarrow_k \psi) := k$. For all other states χ let $\Omega_A(\chi) := 1$.

We call $\mathbb{A}_{2\mu}$ the *tracking automaton* for NW_2.

Let $\beta = (v_n)_{n\in\omega}$ be an infinite branch in an NW_2-proof π. We define $w(\beta) \in \Sigma^\omega$ to be the stream $(\mathsf{S}(v_0), \mathsf{R}(v_0), \mathsf{S}(v_0))(\mathsf{S}(v_0), \mathsf{R}(v_0), \mathsf{S}(v_1))(\mathsf{S}(v_1), \mathsf{R}(v_1), \mathsf{S}(v_2))....$

The following lemma states the adequacy of the tracking automaton.

5.2.3. Lemma. *Let β be an infinite branch in a saturated NW_2-proof. Then β is successful iff $w(\beta) \in \mathcal{L}(\mathbb{A}_{2\mu})$.*

Proof:
If β is successful, then β carries a slim μ-trace due to Lemma 5.2.1 and therefore $w(\beta) \in \mathcal{L}(\mathbb{A}_{2\mu})$. Conversely, $w(\beta) \in \mathcal{L}(\mathbb{A}_{2\mu})$ implies that there is a μ-trace on β. □

Analogously to Chapter 4 we want to apply Lemma 4.1.2 to the infinitary proof system NW_2 using the deterministic Rabin automaton $\mathbb{A}^S_{2\mu}$. To do so, we define the proof system $\mathsf{NW}_2^S := \mathsf{NW}_2{}^{\mathbb{A}^S_{2\mu}}$.

Note that the set of Safra-states A^S of $\mathbb{A}^S_{2\mu}$ consists of set of pairs a^σ, where $a \in A$ and σ is a stack, that is, a sequence of names. Given a sequent Γ we define $\Gamma^\varepsilon := \{\varphi^\varepsilon \mid \varphi \in \Gamma\} \in A^S$.

5.2.4. Lemma. *For any pure sequent Γ there is a saturated NW_2-proof π of Γ iff there is a NW_2^S proof ρ of $\Gamma^\varepsilon \cup \{a_I^\varepsilon\}$. The proof ρ is regular iff π is so.*

Proof:
Lemma 5.2.3 states that the tracking automaton $\mathbb{A}_{2\mu}$ exactly captures the successful infinite paths in NW_2-proofs. Therefore, the correctness of the Safra construction for ε-parity automata, Theorem 3.3.5, implies that an infinite NW_2-branch γ carries a μ-trace iff $w(\gamma) \in \mathcal{L}(\mathbb{A}^S_{2\mu})$. It remains to define the functions f and g in a suitable way. We define

$$\begin{aligned} f : A^S &\to \mathsf{Seq}_\Phi \\ Y &\mapsto \{\varphi \in \mathcal{L}_\mu \mid \varphi^\sigma \in Y\} \cup \\ &\quad\ \{\varphi \rightsquigarrow_k \psi \mid (\varphi \rightsquigarrow_k \psi)^\sigma \in Y\} \end{aligned} \qquad \begin{aligned} g : \mathsf{Seq}_\Phi &\to A^S \\ \Gamma &\mapsto \Gamma^\varepsilon \cup \{a_I^\varepsilon\} \cup \\ &\quad\ \{(\varphi \rightsquigarrow_k \psi)^\varepsilon \mid \varphi \rightsquigarrow_k \psi \in \Gamma\} \end{aligned}$$

Clearly the desired conditions are fulfilled. Thus the lemma directly follows from Lemma 4.1.2 applied to NW_2 and the automaton $\mathbb{A}^S_{2\mu}$. □

We proceed by defining the proof system JS_2. Up to some minor changes this system will coincide with NW^S_2. The completeness of JS_2 will follow directly from the completeness of NW^S_2.

5.2.2 Definition of JS_2-proofs

The construction of the JS_2-proof systems relies on the Safra construction for parity automata with ε-transitions defined in Section 3.3. We will use notations introduced to define this determinization method and shortly recite the most important notions.

Recall that we fixed a finite set of formulas Φ and we defined m as the maximal even number in $\{\Omega_{2\mu}(\varphi) \mid \varphi \in \Phi\}$. Note that the range of the priority function $\Omega_{2\mu}$ is $\mathbb{N}^+$ and we can therefore omit the priority 0. For each even number $k = 2, \ldots, m$ we fix a set of *k-names* X_k and let the set of *names* be $X := X_2 \uplus \cdots \uplus X_m$. We use the symbols $\mathsf{x}, \mathsf{y}, \mathsf{z}, ...$ for names in X and write $\Omega(\mathsf{x}) = k$ if $\mathsf{x} \in X_k$. We call a non-repeating sequence of k-names τ_k a *k-stack* and let T_k be the set of all k-stacks. The empty sequence will be denoted by ε. We define the set of all stacks T to be $T_m \cdots T_2$. In case $\tau_i = \varepsilon$ for all $i < k$ we may write $\tau_m \cdots \tau_k$ rather than $\tau_m \cdots \tau_k \cdots \tau_2$. For a stack τ we define $\tau \downharpoonleft l$ to be the stack obtained from τ by removing all k-names, where $k < l$.

An *annotated formula* is a pair (φ, σ), written as φ^σ, where φ is a formula and σ is a stack such that $\sigma = \sigma \downharpoonleft \Omega(\varphi)$. We call σ the *annotation* of φ. An *annotated sequent* consists of a finite set of annotated formulas $\{\varphi_1^{\sigma_1}, \ldots, \varphi_n^{\sigma_n}\}$, a set of trace atoms $\mathcal{A}$ and a finite, non-repeating sequence of names θ, called the *control*, such that θ contains all names that occur in $\sigma_1, \ldots, \sigma_n$. We denote annotated sequents as $\theta : \varphi_1^{\sigma_1}, \ldots, \varphi_n^{\sigma_n}, \mathcal{A}$. The control can be seen as a linear order on the names occurring in a sequent; it keeps track of when a name is added to a sequent. If it is clear from the context we call annotated sequents just sequents. We use $A, B, \ldots$ as variables ranging over annotated formulas and trace atoms, and use the symbols $\Gamma, \Delta, \Sigma, \ldots$ to denote sets consisting of annotated formulas and trace atoms. For a set of formulas Γ we define $\Gamma^\varepsilon := \{\varphi^\varepsilon \mid \varphi \in \Gamma\}$ and for an annotated sequent Γ we define $\Gamma^\varepsilon := \{\varphi^\varepsilon \mid \varphi^\sigma \in \Gamma \text{ for some } \sigma\}$. Given an annotated sequent Γ we define $\mathsf{Clos}(\Gamma) := \mathsf{Clos}(\{\varphi \in \mathcal{L}^2_\mu \mid \varphi^\sigma \in \Gamma \text{ for some } \sigma\})$ and analogously for $\mathsf{Clos}^\neg$.

In Figure 5.5 the rules of the JS_2-derivation system are given. If one ignores the control and the annotations, the axioms and the rules $\wedge$, $\vee$, η, $\langle a \rangle$, trans, weak, acut and tcut coincide with the rules of NW_2. As before, in the rules acut and tcut we demand that $\varphi, \psi \in \mathsf{Clos}^\neg(\Gamma)$. Annotated sequents correspond to Safra-states of $\mathbb{A}^S_{2\mu}$, where $\mathbb{A}_{2\mu}$ is the tracking automaton checking the success

$$\mathsf{Ax1}: \frac{}{\theta : \varphi^{\sigma}, \overline{\varphi}^{\tau}} \qquad \mathsf{Ax2}: \frac{}{\theta : \bot^{\sigma}} \qquad \mathsf{Ax3}: \frac{}{\theta : \varphi \rightsquigarrow_k \psi, \varphi \not\rightsquigarrow_k \psi} \qquad \mathsf{Ax4}: \frac{}{\theta : \varphi \rightsquigarrow_{2k} \varphi}$$

$$\wedge: \frac{\theta : \varphi^{\sigma}, \psi^{\sigma}, \varphi \wedge \psi \rightsquigarrow_1 \varphi, \varphi \wedge \psi \rightsquigarrow_1 \psi, \Gamma}{\theta : (\varphi \wedge \psi)^{\sigma}, \Gamma} \qquad \langle a \rangle: \frac{\theta : \varphi^{\sigma}, \Sigma, \langle \breve{a} \rangle \Gamma^{\varepsilon}, \Gamma^{\langle a \rangle \varphi}}{\theta : \langle a \rangle \varphi^{\sigma}, [a]\Sigma, \Gamma}$$

$$\vee: \frac{\theta : \varphi^{\sigma}, \varphi \vee \psi \rightsquigarrow_1 \varphi, \Gamma \qquad \theta : \psi^{\sigma}, \varphi \vee \psi \rightsquigarrow_1 \psi, \Gamma}{\theta : (\varphi \vee \psi)^{\sigma}, \Gamma} \qquad \mathsf{weak}: \frac{\theta : \Gamma}{\theta : A, \Gamma}$$

$$\mu: \frac{\theta \cdot \mathsf{x} : \varphi[\mu x.\varphi/x]^{\sigma \lfloor k \cdot \mathsf{x}}, \mu x.\varphi \rightsquigarrow_k \varphi[\mu x.\varphi/x], \Gamma}{\theta : \mu x.\varphi^{\sigma}, \Gamma} \quad k = \Omega_{2\mu}(\mu x.\varphi) \text{ and } \mathsf{x} \text{ is a fresh } k\text{-name}$$

$$\nu: \frac{\theta : \varphi[\nu x.\varphi/x]^{\sigma \lfloor k}, \nu x.\varphi \rightsquigarrow_k \varphi[\nu x.\varphi/x], \Gamma}{\theta : \nu x.\varphi^{\sigma}, \Gamma} \quad k = \Omega_{2\mu}(\nu x.\varphi)$$

$$\mathsf{trans}: \frac{\theta : \varphi \rightsquigarrow_k \psi, \psi \rightsquigarrow_l \chi, \varphi \rightsquigarrow_{\max\{k,l\}} \chi, \Gamma}{\theta : \varphi \rightsquigarrow_k \psi, \psi \rightsquigarrow_l \chi, \Gamma}$$

$$\mathsf{jump}_o: \frac{\theta : \varphi^{\sigma}, \psi^{\sigma \lfloor 2k+1}, \psi^{\tau}, \varphi \rightsquigarrow_{2k+1} \psi, \Gamma}{\theta : \varphi^{\sigma}, \psi^{\tau}, \varphi \rightsquigarrow_{2k+1} \psi, \Gamma}$$

$$\mathsf{jump}_e: \frac{\theta \cdot \mathsf{x} : \varphi^{\sigma}, \psi^{\sigma \lfloor 2k \cdot \mathsf{x}}, \psi^{\tau}, \varphi \rightsquigarrow_{2k} \psi, \Gamma}{\theta : \varphi^{\sigma}, \psi^{\tau}, \varphi \rightsquigarrow_{2k} \psi, \Gamma} \quad \mathsf{x} \text{ is a fresh } 2k\text{-name}$$

$$\mathsf{acut}: \frac{\theta : \varphi^{\varepsilon}, \Gamma \qquad \theta : \overline{\varphi}^{\varepsilon}, \Gamma}{\theta : \Gamma} \qquad \mathsf{tcut}: \frac{\theta : \varphi \rightsquigarrow_k \psi, \Gamma \qquad \theta : \varphi \not\rightsquigarrow_k \psi, \Gamma}{\theta : \Gamma}$$

$$\mathsf{Reset}_{\mathsf{x}}: \frac{\theta : \varphi_1^{\sigma \mathsf{x}}, \ldots, \varphi_n^{\sigma \mathsf{x}}, \Gamma}{\theta : \varphi_1^{\sigma \mathsf{x} \mathsf{x}_1 \tau_1}, \ldots, \varphi_n^{\sigma \mathsf{x} \mathsf{x}_n \tau_n}, \Gamma} \quad \mathsf{x}, \mathsf{x}_1, \ldots, \mathsf{x}_n \text{ are } k\text{-names, } \mathsf{x} \text{ not in } \Gamma$$

$$\mathsf{exp}: \frac{\theta' : \varphi^{\tau}, \Gamma}{\theta : \varphi^{\sigma}, \Gamma} \quad \theta' \sqsubseteq \theta \text{ and } \tau \sqsubseteq \sigma \qquad \qquad \begin{array}{c} [\theta : \Gamma]^{\dagger} \\ \vdots \\ \mathsf{D}_{\dagger}: \dfrac{\theta : \Gamma}{\theta : \Gamma} \end{array}$$

Figure 5.5: Rules of JS_2

condition on infinite NW_2 paths. The transition function δ_A is split up between multiple rules: Step 1 is carried out in every rule; Step 2 adds a fresh name in μ; Step 3 corresponds to the jump rules; Step 4 is a special instance of weak and Step 5 corresponds to Reset. We also add a weakening rule for names, called exp. In order to obtain a cyclic system we add the discharge rule D.

5.2.5. Definition (Successful paths). Let π be a JS_2-derivation. We call an infinite path β in π *successful*, if there is a name x such that

1. x occurs in the control of cofinitely many sequents on β and
2. there are infinitely many applications of $\mathsf{Reset}_\mathsf{x}$ on β.

We call a finite path β in π *successful* if there is a name x such that

1. x occurs in the control of every sequent on β and
2. there is an application of $\mathsf{Reset}_\mathsf{x}$ on β.

In both cases we say that the path β is successful via the name x.

5.2.6. Definition. The *infinitary proof system* JS_2^∞ is defined from the rules in Figure 5.5 together with all infinite successful paths.

The *cyclic proof system* JS_2 is *path-based* and defined from the rules in Figure 5.5 together with all finite successful paths. This means, that a JS_2-proof is a finite JS_2-derivation, where every every leaf is closed and every repeat path is successful.

We say that JS_2 proves a set of formulas Γ, written $\mathsf{JS}_2 \vdash \Gamma$, if there is a JS_2-proof of $\varepsilon : \Gamma^\varepsilon$. Analogously for JS_2^∞.

5.2.7. Example. Consider the following formulas:

$$\varphi := \mu x.\langle a\rangle x \vee \nu y.[\breve{a}]y \wedge \overline{q},$$
$$\psi := \nu y.[\breve{a}]y \wedge \overline{q}.$$

In Example 5.1.10, an NW_2 proof of the sequent $q, \mu x.\langle a\rangle x \vee \nu y.[\breve{a}]y \wedge \overline{q}$ is given. We now define a JS_2-proof π of the same sequent.

Note that $\Omega_{2\mu}(\varphi) = 2$ and $\Omega_{2\mu}(\psi) = 1$. In this example we omit trace atoms and apply weak implicitly. Names are colored blue, and annotations with ε are omitted.

$$\dfrac{\dfrac{\dfrac{\overset{\rho}{\mathsf{x} : \langle a\rangle\varphi^\mathsf{x}, \overline{\psi}} \qquad \dfrac{\dfrac{\dfrac{}{\varepsilon : [\breve{a}]\psi, \overline{q}, q}\,\mathsf{Ax1}}{\varepsilon : [\breve{a}]\psi \wedge \overline{q}, q}\,\wedge}{\varepsilon : \psi, q}\,\nu}{\mathsf{x} : q, \langle a\rangle\varphi^\mathsf{x}}\,\mathsf{acut} \qquad \dfrac{\dfrac{\dfrac{}{\mathsf{x} : q, [\breve{a}]\psi^\mathsf{x}, \overline{q}^\mathsf{x}}\,\mathsf{Ax1}}{\mathsf{x} : q, [\breve{a}]\psi \wedge \overline{q}^\mathsf{x}}\,\wedge}{\mathsf{x} : q, \nu y.[\breve{a}]y \wedge \overline{q}^\mathsf{x}}\,\nu}{\mathsf{x} : q, \langle a\rangle\varphi \vee \nu y.[\breve{a}]y \wedge \overline{q}^\mathsf{x}}\,\vee}{\varepsilon : q, \mu x.\langle a\rangle x \vee \nu y.[\breve{a}]y \wedge \overline{q}}\,\mu$$

The JS_2-proof ρ is given as follows.

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          [\mathsf{x} : \langle a\rangle\varphi^{\mathsf{x}}, \overline{\psi}]^{\dagger}
          \qquad
          \dfrac{\dfrac{\dfrac{}{\varepsilon : [\breve{a}]\psi, \overline{q}, \langle\breve{a}\rangle\overline{\psi}}\;\mathsf{Ax1}}{\varepsilon : [\breve{a}]\psi \wedge \overline{q}, \langle\breve{a}\rangle\overline{\psi}}\;\wedge}{\varepsilon : \psi, \langle\breve{a}\rangle\overline{\psi}}\;\nu
        }{\mathsf{x} : \langle a\rangle\varphi^{\mathsf{x}}, \langle\breve{a}\rangle\overline{\psi}}\;\mathsf{acut}
        \qquad
        \dfrac{\dfrac{\dfrac{}{\mathsf{x} : [\breve{a}]\psi^{\mathsf{x}}, \overline{q}^{\mathsf{x}}, \langle\breve{a}\rangle\overline{\psi}}\;\mathsf{Ax1}}{\mathsf{x} : [\breve{a}]\psi \wedge \overline{q}^{\mathsf{x}}, \langle\breve{a}\rangle\overline{\psi}}\;\wedge}{\mathsf{x} : \nu y.[\breve{a}]y \wedge \overline{q}^{\mathsf{x}}, \langle\breve{a}\rangle\overline{\psi}}\;\nu
      }{\mathsf{x} : \langle a\rangle\varphi \vee \nu y.[\breve{a}]y \wedge \overline{q}^{\mathsf{x}}, \langle\breve{a}\rangle\overline{\psi}}\;\vee
    }{\dfrac{\mathsf{xz} : \langle a\rangle\varphi \vee \nu y.[\breve{a}]y \wedge \overline{q}^{\mathsf{xz}}, \langle\breve{a}\rangle\overline{\psi}}{\mathsf{x} : \varphi^{\mathsf{x}}, \langle\breve{a}\rangle\overline{\psi}}\;\mu}\;\mathsf{Reset}_{\mathsf{x}}
  }{\mathsf{x} : \langle a\rangle\varphi^{\mathsf{x}}, \overline{\psi}}\;\langle a\rangle
}{\mathsf{x} : \langle a\rangle\varphi^{\mathsf{x}}, \overline{\psi}}\;\mathsf{D}_{\dagger}
$$

The repeat path β of the leaf discharged by † is successful, because the name x occurs in the control of every sequent on β and there is an application of $\mathsf{Reset}_{\mathsf{x}}$ on β. Therefore π is a JS_2-proof.

5.2.3 Infinitary proof system JS_2^{∞}

The proof system NW_2^S is constructed by first applying the Safra construction from Section 3.3 to the tracking automaton $\mathbb{A}_{2\mu}$ for NW_2, and then using the uniform construction from Section 4.1.1. Compared to the system NW_2^S, the main distinction of JS_2^{∞} is that the transition function δ_A of $\mathbb{A}_{2\mu}^S$ is split up into multiple rules. In particular, step 4 corresponds to a specific shape of weak, which we call thin.

$$\mathsf{thin}\colon \dfrac{\theta : \varphi^{\sigma}, \Gamma}{\theta : \varphi^{\sigma}, \varphi^{\tau}, \Gamma}\quad \sigma <_{\theta} \tau$$

For the definition of this rule, recall from Section 3.3 that each non-repeating sequence of names θ defines a *linear order* $<_{\theta}$ on names by setting $\mathsf{x} <_{\theta} \mathsf{y}$ if x occurs before y in θ. This order extends to an order on stacks as follows: $\sigma <_{\theta} \tau$ if either

- $\sigma \downharpoonright k$ is a proper extension of $\tau \downharpoonright k$ for some $k \leq m$, or
- σ is lexicographically $<_{\theta}$-smaller than τ, meaning that σ and τ can be written as $\sigma = \rho \cdot \mathsf{x} \cdot \sigma'$ and $\tau = \rho \cdot \mathsf{y} \cdot \tau'$ with $\mathsf{x} <_{\theta} \mathsf{y}$.

We also need a particular instance of exp, that only removes names from θ which do not occur in Γ:

$$\mathsf{exp}'\colon \dfrac{\theta' : \Gamma}{\theta : \Gamma}\quad \theta' \sqsubseteq \theta$$

Using those rules we can translate NW_2^S-proofs to JS_2^{∞}-proofs.

5.2.8. LEMMA. *If there is an* NW_2^S*-proof* ρ *of* $\Gamma^\varepsilon \cup \{a_I^\varepsilon\}$*, then there is a* JS_2^∞*-proof* π *of* Γ^ε*. If* ρ *is regular, then so is* π*.*

Proof:
Recall that $\mathsf{NW}_2^S := \mathsf{NW}_2{}^{\mathbb{A}_{2\mu}^S}$. Let ρ be an NW_2^S-proof of $\Gamma^\varepsilon \cup \{a_I^\varepsilon\}$. We obtain a JS_2^∞-proof π by making the following adaptions to ρ:

1. In every sequent the element a_I^ε is removed.
2. Every element $(\eta x.\psi^*)^\sigma$ occurring in a sequent Y in ρ is replaced by $\eta x.\psi^\sigma$.
3. Every rule $\mathsf{R}^{\mathbb{A}_{2\mu}^S}$ is split up in the rule R and multiple occurrences of jump, thin, Reset and exp'.

This results in a JS_2^∞ derivation π. The global soundness condition of NW_2^S coincides with the success condition on infinite paths in JS_2^∞ and therefore π is a JS_2^∞-proof of Γ^ε. □

In the converse direction we cannot translate JS_2^∞-proofs to NW_2^S-proofs directly, as in ρ rules do not have to be applied in a specific order.

Yet, we show how one can reuse the proof of $\mathcal{L}(\mathbb{A}_{2\mu}^S) \subseteq \mathcal{L}(\mathbb{A}_{2\mu})$ (Converse direction of Theorem 3.3.5) with only minor adaptions, to translate JS_2^∞-proofs to NW_2-proofs.

5.2.9. LEMMA. *If* $\mathsf{JS}_2^\infty \vdash \Gamma$*, then* $\mathsf{NW}_2 \vdash \Gamma$*.*

Proof:
Let ρ be a JS_2^∞-proof of Γ. We let π be the NW_2-derivation defined from ρ by omitting the rules exp, jump and Reset and reducing all other rules to their corresponding NW_2 rules by removing annotations. To show that π is actually a proof, take an arbitrary branch $\beta = (\beta_i)_{i\in\omega}$; we have to prove that β is successful.

Let $\gamma = (\gamma_j)_{j\in\omega}$ be the corresponding infinite branch in ρ. As γ is successful, there is a k-name x that occurs in the control of cofinitely many sequents on γ and such that there are infinitely many applications of $\mathsf{Reset}_\mathsf{x}$ on γ. We can define minimal indices $t(0) < t(1) < \cdots$ such that x occurs in the control of γ_j for $j \geq t(0)$ and such that in $\gamma_{t(i)}$ the rule $\mathsf{Reset}_\mathsf{x}$ is applied for $i \in \omega$. The nodes $\gamma_{t(i)}$ correspond to nodes $\beta_{s(i)}$ on β for $i \in \omega$. As in the proof of Theorem 3.3.5 we can find traces τ_i from $\beta_{s(i)}$ to $\beta_{s(i+1)}$ with maximal weight k. Using König's Lemma we then glue together those traces and obtain an infinite μ-trace on β, which means that β is successful indeed. □

5.2.4 Cyclic proof system JS_2

The correspondence between cyclic JS_2-proofs and regular JS_2^∞-proofs follows similar lines as for the proof system BT in Chapter 4. The main difficulty is that the soundness condition of JS_2 is *path-based*, which requires some extra argumentation in the translation from cyclic to infinite proofs. Whereas the states visited infinitely often by an infinite path through a JS_2-proof form a strongly connected subgraph A, the soundness condition for JS_2-proofs is formulated in terms of repeat paths. We therefore need to show that the success condition of one such repeat path in A is preserved on the strongly connected subgraph A. Intuitively, this holds because the control – the linear order on names in a sequent – induces an order on the repeats in A. In order to make this precise, we first introduce some notations.

Let π be a proof. A *repeat* R of π is a pair (l, c), such that $c(l) = l$. The *repeat path* β_R of a repeat $R = (l, c)$ is defined as the path in $\mathcal{T}_\pi$ from c to l.

Given two strings σ, τ we write $\sigma \preccurlyeq \tau$, if σ is an *initial substring* of τ, that is, if there exists a string θ such that $\tau = \sigma\theta$.

5.2.10. Definition. Let π be a JS_2-proof and let A be a set of nodes in π. We define the *invariant* $\mathrm{inv}(A)$ of A to be the longest word which occurs as an initial segment of the control of each sequent on A. That is,

$$\mathrm{inv}(A) := \bigsqcap\{\theta \mid \theta : \Gamma \text{ in } A\},$$

where $\bigsqcap$ denotes the infimum on strings with respect to the substring order $\preccurlyeq$.

The *invariant* of a path β is the invariant of the set of nodes occurring in β. The *invariant* $\mathrm{inv}(R)$ of a repeat R is defined as the invariant of the repeat path β_R.

Let π be a proof. Recall that a *strongly connected subgraph* A of π is a strongly connected subgraph of $\mathcal{T}_\pi^C$. We say that a repeat $R = (l, c)$ is in A, if l and c are in A. Alternatively, a strongly connected subgraph might also be seen as a set of repeats, consisting of all nodes occurring in one of the repeat paths.

5.2.11. Lemma. *Let A be a strongly connected subgraph of a proof π. Then there is a repeat R in A such that* $\mathrm{inv}(R) = \mathrm{inv}(A)$.

Proof:
We first show that for every path $\beta = v_0 \dots v_n$ in π with $n \geq 1$ there is $i = 0, ..., n-1$ such that $\mathrm{inv}(\beta) = \mathrm{inv}(v_i v_{i+1})$. This is shown by induction on n.

The base case $n = 1$ is trivial. For the induction step let $\beta' = v_0 \dots v_n$ and $\beta = \beta' v_{n+1}$. We consider the controls θ_n and θ_{n+1} at the nodes v_n and v_{n+1}, respectively. If $\mathrm{inv}(\beta) = \mathrm{inv}(\beta')$, the claim follows from the induction hypothesis. Otherwise, $\mathrm{inv}(\beta') \not\preccurlyeq \theta_{n+1}$, which implies that $\mathrm{inv}(\beta') \sqcap \theta_{n+1} = \mathrm{inv}(\beta')\tau \sqcap \theta_{n+1}$ for any string τ. Therefore,

$$\mathrm{inv}(\beta) = \mathrm{inv}(\beta') \sqcap \theta_{n+1} = \theta_n \sqcap \theta_{n+1} = \mathrm{inv}(v_n v_{n+1}).$$

In order to show the lemma, let β be a path in π that visits exactly the nodes in A. Then it follows from the above argumentation that $\mathrm{inv}(\beta) = \mathrm{inv}(vv')$ for some nodes v, v' in A such that v' is a child of v. Because v' is a child of v, there is a repeat R in A such that v and v' are on the repeat path of R. It follows that

$$\mathrm{inv}(A) \preccurlyeq \mathrm{inv}(R) \preccurlyeq \mathrm{inv}(vv') = \mathrm{inv}(A),$$

and therefore that $\mathrm{inv}(R) = \mathrm{inv}(A)$. □

5.2.12. Lemma. $\mathsf{JS}_2 \vdash \Gamma$ *iff there is a regular* JS_2^∞*-proof of* Γ.

Proof:
Let π be a JS_2-proof of Γ. Let π^* be the infinite unfolding of π. Clearly, π^* is regular. Every infinite path α in π^* corresponds to an infinite path γ in π, where the nodes visited infinitely often by γ form a strongly connected subgraph A. Using Lemma 5.2.11, let R be a repeat in A such that $\mathrm{inv}(R) = \mathrm{inv}(A)$. Because the repeat path β_R is successful, there is a name x such that x occurs in the control of every sequent on β_R with an application of $\mathsf{Reset}_\mathsf{x}$ on β_R. In particular, x is in $\mathrm{inv}(R) = \mathrm{inv}(A)$ and therefore the name x occurs in cofinitely many controls on α. Because γ passes the repeat path β_R infinitely often, there are infinitely many applications of $\mathsf{Reset}_\mathsf{x}$ rules on γ and thus also on α. This implies that the infinite path α is successful via x.

Conversely, let ρ be a regular JS_2^∞-proof. For a node $v \in \rho$ let ρ_v be the maximal subtree of ρ rooted at v. For every infinite path $\beta = (\beta(i))_{i\in\omega}$ define minimal indices $j < k$ such that

1. $\rho_{\beta(j)} \sim \rho_{\beta(k)}$ and
2. the path $\beta(j)\cdots\beta(k)$ is successful.

Because ρ is regular and every infinite path is successful, such indices always exist. For each such infinite path we introduce a $\mathsf{D}_\dagger$ node at $\beta(j)$ and let $\beta(k)$ be a leaf discharged by $\dagger$. Using König's Lemma we can show that this procedure results in a finite JS_2-proof π of Γ. □

5.2.13. Theorem (Soundness and Completeness). *A pure sequent* Γ *is unsatisfiable iff there is a regular* JS_2^∞*-proof of* Γ *iff* $\mathsf{JS}_2 \vdash \Gamma$.

Proof:
From Theorem 5.1.12 and Theorem 5.1.17 it follows that Γ is unsatisfiable iff there is a regular NW_2-proof of Γ.

Combining Lemma 5.2.4 and Lemma 5.2.8 shows that there is a regular JS_2^∞-proof of Γ if Γ is unsatisfiable. Conversely, Lemma 5.2.9 proves that Γ is unsatisfiable if $\mathsf{JS}_2^\infty \vdash \Gamma$. The equivalence between JS_2-proofs and regular JS_2^∞-proofs follows from Lemma 5.2.12. □

5.2.14. Remark. The cyclic proof system JS_2 is path-based compared to the subgraph-based system BT from Chapter 4. The more local soundness condition in JS_2 is possible, because JS_2-sequents contain a control; a linear order on the occurring names. Those controls give an order on repeats and allow us find most important repeats in strongly connected subgraphs of a proof, as witnessed by Lemma 5.2.11.

5.2.5 Clean repeats

As mentioned in the introduction to this chapter, applying Maehara's method requires working with a split system – that is, a proof system in which sequents are divided into two components of the form $\Gamma \mid \Delta$. For JS_2-sequents this presents a problem: it is unclear whether the control θ should be split across both components as well. We argue that both approaches lead to difficulties.

If the control is split, so that sequents take the form $\theta : \Gamma \mid \kappa : \Delta$, then a path-based soundness condition would no longer suffice to guarantee that all infinite paths through a proof are successful. To see this, consider two interleaving repeats, where one is successful via a name x in θ and the other is successful via a name y in κ. Because the controls do not give an order on the names x and y, there is no guarantee that one of the names is preserved on both repeats.

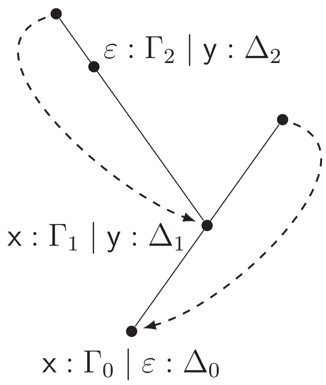


Figure 5.6: Example of a split proof where the right repeat is successful via x and the left repeat is successful via y. However, neither x nor y are preserved on both repeats.

Note that in [KV25] a path-based split JS_2 system is used. Yet, this system is not sound and the interpolation proof given is incorrect. We correct this mistake and show that interpolation for $\mathcal{L}^2_\mu$ nonetheless holds. The system given in [KV25] would be sound if a subgraph-based soundness condition is imposed on cyclic proofs. However, such a subgraph-based soundness condition does not combine well with Maehara's method.

Alternatively, we could retain a single control, so that sequents take the form $\theta : \Gamma \mid \Delta$, where θ is a sequence of all names occurring in Γ and Δ. However, in this case the interpolation proof becomes quite tricky, as the control is very fragile to changes in the components. For example, it becomes difficult to get a handle on the control when transforming proofs of $\Gamma \mid \Delta$ to proofs of $\Gamma \mid \iota$, where ι is the interpolant of Γ and Δ.

We therefore opt for a different solution: inspired by [AL16] we simplify the cyclic system JS_2 into a new system Circ_2 that completely *lacks* the control. For this translation to work, we first need to transform JS_2-proofs into a certain normal form. This involves two steps: first, we ensure that all repeats are *clean*; then, we translate the proof to a *monotone* one.

5.2.15. Definition. Let π be a JS_2-derivation. We call a repeat in π *clean*, if it is of the form

$$\dfrac{\dfrac{\lceil \theta : \varphi_1^{\sigma\mathsf{x}}, \ldots, \varphi_n^{\sigma\mathsf{x}}, \Gamma \rceil^\dagger}{\theta : \varphi_1^{\sigma\mathsf{x}\mathsf{x}_1\tau_1}, \ldots, \varphi_n^{\sigma\mathsf{x}\mathsf{x}_n\tau_n}, \Gamma}\ \mathsf{Reset}_\mathsf{x}}{\vdots} \qquad \dfrac{\theta : \varphi_1^{\sigma\mathsf{x}}, \ldots, \varphi_n^{\sigma\mathsf{x}}, \Gamma}{\theta : \varphi_1^{\sigma\mathsf{x}}, \ldots, \varphi_n^{\sigma\mathsf{x}}, \Gamma}\ \mathsf{D}_\dagger$$

and the repeat path is successful via the name x. Given such a clean repeat, we label the discharge rule with the name x and write $\mathsf{D}_\dagger(\mathsf{x})$.

We say that π has *clean repeats*, if all repeats in π are clean.

5.2.16. Lemma. *If there is a regular* JS_2^∞*-proof of* Γ*, then there is a* JS_2*-proof of* Γ *with clean repeats.*

Proof:
Let ρ be a regular JS_2^∞-proof of Γ. We can follow the same lines as in the converse direction of the proof of Lemma 5.2.12. For every infinite path $(\beta(i))_{i\in\omega}$ we define minimal indices $j < k$ satisfying the conditions

1. $\rho_{\beta(j)} \sim \rho_{\beta(k)}$,
2. the parent node of k is labeled with $\mathsf{Reset}_\mathsf{x}$ and
3. the path $\beta(j) \cdots \beta(k)$ is successful via x.

Such indices always exist because ρ is regular and on every infinite path there is a name x such that x occurs in the control of cofinitely many sequents on β and such that there are infinitely many applications of $\mathsf{Reset}_\mathsf{x}$ on β. Following the rest of the proof of Lemma 5.2.12 yields a JS_2-proof π of Γ with clean repeats. □

5.2.6 Monotone proofs

In the interpolation proof we need our proofs to satisfy a certain monotonicity condition: Names witnessing the success of repeats also should occur in repeats further up in the proof tree, as formally expressed in Lemma 5.2.18. One way to provide such a monotone proof is to unfold a cyclic proof, see [SD03]. We will rephrase their approach in our setting and refer to their proof. A similar transformation is also given in [AL17].

Recall that the *strongly connected subtree* $\mathrm{scst}(u)$ of a companion node u in π is the maximal strongly connected subgraph A of π such that u is the root of A.

5.2.17. DEFINITION. Let π be a JS_2-proof with clean repeats. We call π *monotone* if for every companion node c labeled with $\mathsf{D}(\mathsf{x})$ the name x occurs in the control of every node in $\mathrm{scst}(c)$.

5.2.18. LEMMA. *Let π be JS_2-proof with clean repeats. Then π can be transformed to a monotone JS_2-proof π' with clean repeats of the same sequent.*

The proof of this lemma will follow from Sprenger & Dam [SD03]. Before we can use their results, we have to define some of their notions in our setting. In particular, we define orders on the repeats of a cyclic derivation: the *structural dependency order* and *induction orders*.

Recall that a repeat is a pair $R = (l, v)$ where l is a repeat leaf and $c(l) = v$. The repeat path β_R of a repeat $R = (l, v)$ is defined as the path in $\mathcal{T}_\pi$ from v to l.

5.2.19. DEFINITION. Let π be a JS_2-derivation. Let $R_1 = (l_1, c_1)$ and $R_2 = (l_2, c_2)$ be repeats of π. We define $R_1 \preccurlyeq_\pi R_2$ if $c_1 \in \mathrm{scst}(c_2)$ and call $\preccurlyeq_\pi$ the *structural dependency order* on π.

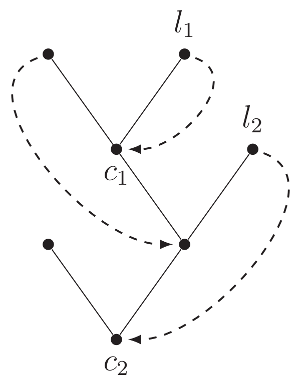


Figure 5.7: A proof π with two indicated repeats $R_1 = (l_1, c_1)$ and $R_2 = (l_2, c_2)$ such that $R_1 \preccurlyeq_\pi R_2$.

Intuitively, $R_1 \preccurlyeq_\pi R_2$ if the repeats R_1 and R_2 are in the same cluster and R_1 is further up in the proof tree compared to R_2. Thus, the repeat R_1 is structurally dependent on R_2.

In [SD03] the order $\preccurlyeq_\pi$ is defined as the transitive closure of $\leq_\pi$, where $R_1 \leq_\pi R_2$ if $c_2 \leq c_1 < l_2$, that is, c_1 lies on β_{R_2}. It can readily be seen that these notions coincide.

5.2.20. Definition. Let π a JS_2-derivation and let $\mathcal{R}$ be the set of repeats of π. An *induction order* of π is a partial order $\leq_I$ on $\mathcal{R}$ such that every weakly $\preccurlyeq_\pi$-connected set of repeats $\mathcal{R}_0 \subseteq \mathcal{R}$ has a $\leq_I$-greatest element.

Note that $\mathcal{R}_0 \subseteq \mathcal{R}$ is weakly $\preccurlyeq_\pi$-connected if the nodes occurring in the repeats $\mathcal{R}_0$ form a strongly connected subgraph of π. Therefore, $\leq_I$ is an induction order iff in each strongly connected subgraph A of π there is a $\leq_I$-greatest element.

In [SD03] induction orders are assumed to be tree-like. As already mentioned in their paper, this restriction is only for convenience and not necessary. In fact, one can easily obtain a tree-like induction order from any induction-order by coarsening the relation.

5.2.21. Example. Let π a JS_2-derivation. Then the structural dependency order $\preccurlyeq_\pi$ is an induction order on π.

If π has clean repeats, then we can assign names to repeats. This allows us to define an alternative notion of JS_2-proofs.

5.2.22. Definition. Let π be a JS_2-derivation with clean repeats and let $\mathcal{R}$ be the set of repeats of π. We define a map δ on $\mathcal{R}$, where for a repeat $R = (l, c)$ we define $\delta(R) = \mathsf{x}$ if c is labeled with $\mathsf{D}(\mathsf{x})$.

We say that an induction order $\leq_I$ *discharges* π, if for all $R \in \mathcal{R}$

1. $\delta(R)$ occurs in the control of every sequent on $\beta_{R'}$ whenever $R' \leq_I R$, and
2. there is an application of $\mathsf{Reset}_{\delta(\mathsf{R})}$ on β_R.

5.2.23. Definition. A JS_2-*preproof* is a finite JS_2-derivation where every leaf is either axiomatic or a repeat leaf.

Note that a JS_2-proof is simply a JS_2-preproof where every repeat leaf is discharged.

5.2.24. Lemma. *For every JS_2-preproof π with clean repeats, the following are equivalent:*

(i) there is an induction order $\leq_I$ discharging π, and

(ii) π is a JS_2-proof.

Proof:
Let π be a JS_2-preproof. If there is an induction order $\leq_I$ discharging π, then, in particular, every repeat path β_R is successful. Therefore, π is a JS_2-proof.

Conversely, if π is a JS_2-proof, then we can define an induction order $\leq_I$ as follows. Let $R' \leq_I R$ if $\delta(R)$ occurs in the control of every sequent on $\beta_{R'}$. Note that for every repeat R the name $\delta(R)$ occurs in the control of every sequent on β_R; in particular, $\delta(R)$ is in the invariant $\mathrm{inv}(R)$ of R. It thus follows from Lemma 5.2.11 that every weakly $\preccurlyeq_\pi$-connected set of repeats in π has a $\leq_I$-greatest element implying that $\leq_I$ is an induction order. By definition, $\leq_I$ discharges π.
□

5.2.25. Definition. Let π be JS_2-derivation with clean repeats. We say that an induction order $\leq_I$ on π is *tree-compatible* if for all repeats R and R' in π we have

$$R \preccurlyeq_\pi R' \quad \Rightarrow \quad R \leq_I R'.$$

5.2.26. Theorem ([SD03], Theorem 5). *Let π be a JS_2-preproof with clean repeats and with an induction order discharging π. Then π can be transformed to a JS_2-preproof ρ of the same sequent with clean repeats and with a tree-compatible induction order discharging ρ.*

Although in [SD03] a different proof system is covered, the content of Theorem 5.2.26 does not depend on the specifics of the system, but only on the structure of trees with back edges and a suitable definition of induction orders discharging derivations. Therefore, the proof of Theorem 5 in [SD03] works in our setting as well. The only extra requirement is that we must preserve clean repeats. Yet it can be easily observed in Algorithm 1 of [SD03] that preproofs with clean repeats are translated into preproofs with clean repeats. We can therefore use Theorem 5.2.26 to show that we can transform JS_2-proofs into monotone ones.

Proof of Lemma 5.2.18:
Let π be a JS_2-proof of a sequent Γ with clean repeats. Lemma 5.2.24 shows that there is an induction order discharging π. Using Theorem 5.2.26 we let ρ be a JS_2-preproof of Γ with clean repeats and with a tree-compatible induction order $\leq_I$ discharging ρ. Because of Lemma 5.2.24, ρ is a JS_2-proof.

It remains to show that ρ is monotone. Let c be a companion node labeled with $\mathsf{D}(\mathsf{x})$ and let $R = (l, c)$ be a repeat of ρ. Let $v \in \mathrm{scst}(c)$, we need to show that x occurs in the control of v. Because $v \in \mathrm{scst}(c)$, the node v lies on the repeat path $\beta_{R'}$ of a repeat R' where $R' \preccurlyeq_\rho R$. As the induction order $\leq_I$ is tree-compatible we have $R' \leq_I R$. By definition this means that $\mathsf{x} = \delta(R)$ occurs in the control of every sequent on $\beta_{R'}$. In particular, x occurs in the control of v. Therefore ρ is monotone. □

5.3 Circ$_2$-proof system

The control in the JS$_2$ system serves two purposes. First, it ensures the correctness of the automata determinization method used to define the system. In JS$_2$, the use of the control is reflected in the thin rule, a specific instance of the weak rule. Since we are only concerned with whether a sequent Γ is provable – and not whether every proof search for Γ results in a valid proof – we may assume that weak rules are applied in the correct manner. Therefore, this function of the control is not necessary in the proof system.

The second purpose of the control is to support a path-based soundness condition for cyclic JS$_2$-proofs. It ensures that infinite paths through a cyclic proof are successful. An infinite path through a cyclic proof corresponds to a path in a strongly connected subgraph A. As witnessed in the proof of Lemma 5.2.12, the control provides a most important repeat R on such A, where the successful name in R is preserved throughout A. To obtain a sound cyclic proof system without the control, this function must be fulfilled by other means.

We take inspiration from the cyclic proof system Circ introduced by Afshari & Leigh [AL16]. Influenced by the system JS, their approach is to partition the set of names into *variable names* and *assumption names*. The former track ν-unfoldings, as in JS, while the latter are additional and provide invariants on repeat paths. Since assumption names are unique, they naturally impose an order on repeats, enforcing monotonicity and enabling a path-based soundness condition on cyclic proofs. Our formulation of the system Circ$_2$ differs slightly from Circ: we eliminate the control *entirely*.

For every even $k = 2, \dots, m$ we partition the set of names X_k into two disjoint infinite sets: The set of *assumption names* X_k^A and the set of *variable names* X_k^V. We will use the symbols $\hat{\mathsf{x}}, \hat{\mathsf{y}}, \hat{\mathsf{z}}$... for assumption names and the symbols $\mathsf{x}, \mathsf{y}, \mathsf{z}, \dots$ for variable names. A *k-name* is a name in $X_k := X_k^A \cup X_k^V$. As before, we call a non-repeating sequence of k-names a *k-stack*, we let T_k be the set of all k-stacks and define the set of all Circ$_2$*-stacks* T to be $T_m \cdots T_2$. When it is clear from the context, we call Circ$_2$-stacks just stacks.

An *annotated* Circ$_2$*-formula* is a pair (φ, σ), written as φ^σ, where φ is a $\mathcal{L}_\mu^2$-formula and σ is a Circ$_2$-stack such that $\sigma = \sigma \downharpoonright \Omega_{2\mu}(\varphi)$. When it is clear from the context, we call annotated Circ$_2$-formulas just (annotated) formulas. A *sequent* in the Circ$_2$ system consists of a set of annotated formulas and a set of trace atoms. Note that, compared to JS$_2$ sequents, Circ$_2$ sequents do not contain a control.

5.3.1. Definition. The derivation system Circ$_2$ is defined from the rules in Figure 5.5, where all controls are removed, the Reset rule is removed and the discharge rule D is replaced by

$$\begin{array}{c}\ulcorner \varphi_1^{\sigma\hat{\mathsf{x}}\mathsf{x}_1}, \ldots, \varphi_n^{\sigma\hat{\mathsf{x}}\mathsf{x}_n}, \Gamma \urcorner^{\hat{\mathsf{x}}} \\ \vdots \\ \mathsf{D}_{\hat{\mathsf{x}}}\colon \dfrac{\varphi_1^{\sigma\hat{\mathsf{x}}}, \ldots, \varphi_n^{\sigma\hat{\mathsf{x}}}, \Gamma}{\varphi_1^{\sigma}, \ldots, \varphi_n^{\sigma}, \Gamma} \quad \hat{\mathsf{x}} \in X_k^A,\ \mathsf{x}_1, \ldots, \mathsf{x}_n \in X_k^V \text{ and } \mathsf{last}(\sigma) \in X_k\end{array}$$

Importantly, discharge rules are labeled with *unique assumption names*. Names introduced in the rules μ and jump_e are *variable names*.

5.3.2. Definition. The circular proof system Circ_2 is path-based and defined from all Circ_2-rules together with all finite paths.

In an Circ_2-proof π, for every assumption name $\hat{\mathsf{x}}$, there is at most one application of $\mathsf{D}_{\hat{\mathsf{x}}}$. Therefore, if v is labeled with $\mathsf{D}_{\hat{\mathsf{x}}}$, then $\hat{\mathsf{x}}$ occurs in the sequent of every node on a repeat path with companion v. Even more, $\hat{\mathsf{x}}$ occurs in the sequent of every node w in the *strongly connected subtree* of v.

5.3.3. Example. We continue Example 5.2.7 and give a Circ_2-proof of the same sequent. Consider the following formulas:

$$\begin{aligned}\varphi &:= \mu x.\langle a\rangle x \vee \nu y.[\breve{a}]y \wedge \overline{q}, \\ \psi &:= \nu y.[\breve{a}]y \wedge \overline{q}.\end{aligned}$$

Note that $\Omega_{2\mu}(\varphi) = 2$ and $\Omega_{2\mu}(\psi) = 1$. We now define a Circ_2-proof π of q, φ. In this example we omit trace atoms and apply weak implicitly. Names are colored blue, and annotations with ε are omitted.

$$\dfrac{\dfrac{\dfrac{\overset{\rho}{\langle a\rangle\varphi^{\times}, \overline{\psi}} \qquad \dfrac{\dfrac{\dfrac{}{[\breve{a}]\psi, \overline{q}, q}\,\mathsf{Ax1}}{[\breve{a}]\psi \wedge \overline{q}, q}\,\wedge}{\psi, q}\,\nu}{q, \langle a\rangle\varphi^{\times}}\,\mathsf{acut} \qquad \dfrac{\dfrac{\dfrac{}{q, [\breve{a}]\psi^{\times}, \overline{q}^{\times}}\,\mathsf{Ax1}}{q, [\breve{a}]\psi \wedge \overline{q}^{\times}}\,\wedge}{q, \nu y.[\breve{a}]y \wedge \overline{q}^{\times}}\,\nu}{q, \langle a\rangle\varphi \vee \nu y.[\breve{a}]y \wedge \overline{q}^{\times}}\,\vee}{q, \mu x.\langle a\rangle x \vee \nu y.[\breve{a}]y \wedge \overline{q}}\,\mu$$

The Circ$_2$-proof ρ is given as follows.

$$
\dfrac{
  \dfrac{
    \lceil\langle a\rangle\varphi^{\mathsf{x}\hat{\mathsf{x}}\mathsf{z}}, \overline{\psi}\,]^{\hat{\mathsf{x}}}
    \qquad
    \dfrac{\dfrac{\dfrac{}{[\breve{a}]\psi, \overline{q}, \langle\breve{a}\rangle\overline{\psi}}\,\mathsf{Ax1}}{[\breve{a}]\psi \wedge \overline{q}, \langle\breve{a}\rangle\overline{\psi}}\,\wedge}{\psi, \langle\breve{a}\rangle\overline{\psi}}\,\nu
  }{
    \langle a\rangle\varphi^{\mathsf{x}\hat{\mathsf{x}}\mathsf{z}}, \langle\breve{a}\rangle\overline{\psi}
  }\,\mathsf{acut}
  \qquad
  \dfrac{\dfrac{\dfrac{}{[\breve{a}]\psi^{\mathsf{x}}, \overline{q}^{\mathsf{x}\hat{\mathsf{x}}\mathsf{z}}, \langle\breve{a}\rangle\overline{\psi}}\,\mathsf{Ax1}}{[\breve{a}]\psi \wedge \overline{q}^{\mathsf{x}\hat{\mathsf{x}}\mathsf{z}}, \langle\breve{a}\rangle\overline{\psi}}\,\wedge}{\nu y.[\breve{a}]y \wedge \overline{q}^{\mathsf{x}\hat{\mathsf{x}}\mathsf{z}}, \langle\breve{a}\rangle\overline{\psi}}\,\nu
}{
  \dfrac{
    \dfrac{
      \dfrac{\langle a\rangle\varphi \vee \nu y.[\breve{a}]y \wedge \overline{q}^{\mathsf{x}\hat{\mathsf{x}}\mathsf{z}}, \langle\breve{a}\rangle\overline{\psi}}{\varphi^{\mathsf{x}\hat{\mathsf{x}}}, \langle\breve{a}\rangle\overline{\psi}}\,\mu
    }{\langle a\rangle\varphi^{\mathsf{x}\hat{\mathsf{x}}}, \overline{\psi}}\,\langle a\rangle
  }{\langle a\rangle\varphi^{\mathsf{x}}, \overline{\psi}}\,\mathsf{D}_{\hat{\mathsf{x}}}
}\,\vee
$$

In Circ$_2$ we witness success of assumption variables at the discharged leaves by implicitly applying a Reset rule. We can therefore omit explicit Reset rules in the Circ$_2$ system. However, Reset rules may also be seen as specific instances of exp and those will still be of importance when translating between JS$_2$ and Circ$_2$-proofs. We will denote the following specific instance of exp with Reset$_{\hat{\mathsf{x}}}$:

$$\mathsf{Reset}_{\hat{\mathsf{x}}}\colon\ \dfrac{\varphi_1^{\sigma\hat{\mathsf{x}}}, \ldots, \varphi_n^{\sigma\hat{\mathsf{x}}}, \Gamma}{\varphi_1^{\sigma\hat{\mathsf{x}}\mathsf{x}_1}, \ldots, \varphi_n^{\sigma\hat{\mathsf{x}}\mathsf{x}_n}, \Gamma} \quad \hat{\mathsf{x}} \in X_k^A \text{ and } \mathsf{x}_1, \ldots, \mathsf{x}_n \in X_k^V$$

In the discharge rule D$_{\hat{\mathsf{x}}}$ in Circ$_2$, discharged leaves are labeled with different sequents than their companions. Accordingly, the notion of an infinite unfolding of a proof must be adapted for Circ$_2$. Recall that we call a repeat leaf l *outermost*, if its companion $c(l)$ is the root of some proper cluster.

5.3.4. Definition. Let ρ be a Circ$_2$-proof. For an outermost repeat leaf l in ρ labeled with $\varphi_1^{\sigma\hat{\mathsf{x}}\mathsf{x}_1}, \ldots, \varphi_n^{\sigma\hat{\mathsf{x}}\mathsf{x}_n}$, we define the Circ$_2$-derivation ρ_l as

$$\dfrac{\begin{array}{c}\rho_v\\ \varphi_1^{\sigma\hat{\mathsf{x}}}, \ldots, \varphi_n^{\sigma\hat{\mathsf{x}}}, \Gamma\end{array}}{\varphi_1^{\sigma\hat{\mathsf{x}}\mathsf{x}_1}, \ldots, \varphi_n^{\sigma\hat{\mathsf{x}}\mathsf{x}_n}, \Gamma}\,\mathsf{Reset}_{\hat{\mathsf{x}}}$$

where ρ_v is the maximal subderivation of ρ rooted at the child node of $c(l)$.

The *infinite unfolding* ρ^* of ρ is defined as the Circ$_2$-derivation obtained from ρ by recursively replacing outermost leaves l with ρ_l.[4]

5.3.5. Lemma. *If* JS$_2 \vdash \Gamma$, *then* Circ$_2 \vdash \Gamma$.

[4]Note that in Circ$_2$-proofs discharge rules D are labeled with *unique* assumption names. In order to satisfy this requirement, for each assumption name $\hat{\mathsf{y}}$ that is labeling a discharge rule D$_{\hat{\mathsf{y}}}$ in ρ_l, we substitute $\hat{\mathsf{y}}$ by a fresh assumption name $\hat{\mathsf{z}}$ in ρ_l. This guarantees that in ρ^* each discharge rule is labeled with a unique assumption name.

Proof:
Let π be a JS_2-proof of $\varepsilon : \Gamma^\varepsilon$. Because of Lemma 5.2.16 and Lemma 5.2.18 we may assume that π is monotone and has clean repeats. In particular, all discharge rules $\mathsf{D}(\mathsf{x})$ are labeled with a name x.

We translate π to a Circ_2-proof ρ. Informally, we want to translate repeats in π of the form

$$\dfrac{\dfrac{\lceil \theta : \varphi_1^{\sigma\mathsf{x}}, \dots, \varphi_n^{\sigma\mathsf{x}}, \Gamma \rceil^\dagger}{\theta : \varphi_1^{\sigma\mathsf{x}\mathsf{x}_1\tau_1}, \dots, \varphi_n^{\sigma\mathsf{x}\mathsf{x}_n\tau_n}, \Gamma}\ \mathsf{Reset}_\mathsf{x}}{\begin{array}{c}\vdots\\ \dfrac{\theta : \varphi_1^{\sigma\mathsf{x}}, \dots, \varphi_n^{\sigma\mathsf{x}}, \Gamma}{\theta : \varphi_1^{\sigma\mathsf{x}}, \dots, \varphi_n^{\sigma\mathsf{x}}, \Gamma}\ \mathsf{D}_\dagger(\mathsf{x})\end{array}}$$

to repeats in ρ of the form

$$\dfrac{\dfrac{\lceil \varphi_1^{\sigma\mathsf{x}\hat{\mathsf{x}}\mathsf{x}_1}, \dots, \varphi_n^{\sigma\mathsf{x}\hat{\mathsf{x}}\mathsf{x}_n}, \Gamma \rceil^{\hat{\mathsf{x}}}}{\varphi_1^{\sigma\mathsf{x}\hat{\mathsf{x}}\mathsf{x}_1\tau_1}, \dots, \varphi_n^{\sigma\mathsf{x}\hat{\mathsf{x}}\mathsf{x}_n\tau_n}, \Gamma}\ \mathsf{exp}}{\begin{array}{c}\vdots\\ \dfrac{\varphi_1^{\sigma\mathsf{x}\hat{\mathsf{x}}}, \dots, \varphi_n^{\sigma\mathsf{x}\hat{\mathsf{x}}}, \Gamma}{\varphi_1^{\sigma\mathsf{x}}, \dots, \varphi_n^{\sigma\mathsf{x}}, \Gamma}\ \mathsf{D}_{\hat{\mathsf{x}}}\end{array}}$$

Because the cyclic structure of π might be more complicated, we we have to proceed in a more structured way.

Let ρ be obtained from π by removing all controls. Then π is almost a Circ_2-derivation, the only exception are companion nodes c labeled with $\mathsf{D}_\dagger(\mathsf{x})$ and leaves l discharged by $\dagger$.[5] We will transform those JS_2-repeats $R = (l, c)$ to Circ_2-repeats R' one by one starting from the leaves.

Let $R = (l, c)$ be a JS_2-repeat in ρ, such that all repeats above c are already transformed to Circ_2-repeats. Assume that c is labeled with

$$\dfrac{\varphi_1^{\sigma\mathsf{x}}, \dots, \varphi_n^{\sigma\mathsf{x}}, \Gamma}{\varphi_1^{\sigma\mathsf{x}}, \dots, \varphi_n^{\sigma\mathsf{x}}, \Gamma}\ \mathsf{D}_\dagger(\mathsf{x})$$

Because π is monotone, the name x occurs in every sequent in $\mathrm{scst}(c)$. Therefore, x is never introduced in $\mathrm{scst}(c)$.

Then we let c be labeled with $\mathsf{D}_{\hat{\mathsf{x}}}$, where $\hat{\mathsf{x}}$ is a fresh assumption name with $\Omega(\hat{\mathsf{x}}) = \Omega(\mathsf{x})$.

$$\dfrac{\varphi_1^{\sigma\mathsf{x}\hat{\mathsf{x}}}, \dots, \varphi_n^{\sigma\mathsf{x}\hat{\mathsf{x}}}, \Gamma}{\varphi_1^{\sigma\mathsf{x}}, \dots, \varphi_n^{\sigma\mathsf{x}}, \Gamma}\ \mathsf{D}_{\hat{\mathsf{x}}}$$

In the strongly connected subtree $\mathrm{scst}(c)$ of c in ρ we substitute x with $\mathsf{x}\hat{\mathsf{x}}$. Consequently, we remove the name $\hat{\mathsf{x}}$ with exp rules outside of $\mathrm{scst}(c)$, and in $\mathrm{scst}(c)$ we

[5] Here $\mathsf{Reset}_\mathsf{x}$ rules are just seen as specific instances of exp rules.

replace $\mathsf{Reset}_{\mathsf{x}}$ rules with $\mathsf{Reset}_{\hat{\mathsf{x}}}$ rules. All other rules in $\mathrm{scst}(c)$ remain applicable as before. This is possible because x is never introduced in $\mathrm{scst}(c)$, implying that in ρ the assumption name $\hat{\mathsf{x}}$ always occurs to the right of x and the names x and $\hat{\mathsf{x}}$ have the same priority.

Every leaf l in ρ discharged by $\dagger$ is not in $\mathrm{scst}(c)$ for any companion node v above c. Therefore, l remains unchanged by transformations of repeats above c. Thus every leaf l discharged by $\dagger$ is of the form

$$\dfrac{\lceil \varphi_1^{\sigma\mathsf{x}\hat{\mathsf{x}}}, \dots, \varphi_n^{\sigma\mathsf{x}\hat{\mathsf{x}}}, \Gamma \rceil^{\dagger}}{\varphi_1^{\sigma\mathsf{x}\hat{\mathsf{x}}\mathsf{x}_1\tau_1}, \dots, \varphi_n^{\sigma\mathsf{x}\hat{\mathsf{x}}\mathsf{x}_n\tau_n}, \Gamma}\ \mathsf{Reset}_{\hat{\mathsf{x}}}$$

We let l be a discharged leaf of the following form

$$\dfrac{\lceil \varphi_1^{\sigma\mathsf{x}\hat{\mathsf{x}}\mathsf{x}_1}, \dots, \varphi_n^{\sigma\mathsf{x}\hat{\mathsf{x}}\mathsf{x}_n}, \Gamma \rceil^{\hat{\mathsf{x}}}}{\varphi_1^{\sigma\mathsf{x}\hat{\mathsf{x}}\mathsf{x}_1\tau_1}, \dots, \varphi_n^{\sigma\mathsf{x}\hat{\mathsf{x}}\mathsf{x}_n\tau_n}, \Gamma}\ \mathsf{exp}$$

Doing this transformation for all repeats results in a well-formed Circ_2-proof. $\square$

For the converse direction, we will translate Circ_2-proofs into infinitary JS_2^{∞}-proofs. The idea is to unfold a Circ_2-proof ρ, remove all assumption names, and translate $\mathsf{Reset}_{\hat{\mathsf{x}}}$ rules to $\mathsf{Reset}_{\mathsf{x}}$ rules, where x is a variable name of the same priority as $\hat{\mathsf{x}}$.

5.3.6. LEMMA. *If* $\mathsf{Circ}_2 \vdash \Gamma$*, then* $\mathsf{JS}_2^{\infty} \vdash \Gamma$.

Proof:
Let ρ be a Circ_2-proof of Γ. We will translate ρ to a JS_2^{∞}-proof π of Γ, where the set of names in π consists of all variable names in ρ.

Because every assumption name $\hat{\mathsf{x}}$ is unique in ρ, there is a unique node in ρ labeled with $\mathsf{D}_{\hat{\mathsf{x}}}$ of the form

$$\dfrac{\varphi_1^{\sigma\hat{\mathsf{x}}}, \dots, \varphi_n^{\sigma\hat{\mathsf{x}}}, \Gamma}{\varphi_1^{\sigma}, \dots, \varphi_n^{\sigma}, \Gamma}\ \mathsf{D}_{\hat{\mathsf{x}}}$$

We define $\mathsf{origin}(\hat{\mathsf{x}})$ as the last variable name in σ. By definition of the $\mathsf{D}_{\hat{\mathsf{x}}}$ rule, it holds that $\Omega(\hat{\mathsf{x}}) = \Omega(\mathsf{origin}(\hat{\mathsf{x}}))$.

Let ρ^* be the infinite unfolding of ρ. We will translate ρ^* to a JS_2^{∞}-proof π such that every sequent in ρ^* of the form

$$\varphi_1^{\sigma_1}, \dots, \varphi_n^{\sigma_n}, \mathcal{A},$$

where $\mathcal{A}$ is a set of trace atoms, is translated to a JS_2 sequent in π of the form

$$\theta : \varphi_1^{\sigma_1'}, \dots, \varphi_n^{\sigma_n'}, \mathcal{A}.$$

Here σ_i' is obtained from σ_i by removing all assumption names for $i = 1, ..., n$ and θ is some non-repeating sequence of all names in $\sigma_1', ..., \sigma_n'$.

We define π inductively starting from the root. The root of π will be labeled with $\varepsilon : \Gamma^\varepsilon$. We proceed inductively with a case distinction on the applied rule. D rules are removed; $\mathsf{Reset}_{\hat{\mathsf{x}}}$ rules for assumption names $\hat{\mathsf{x}}$ are replaced by $\mathsf{Reset}_{\mathsf{origin}(\hat{\mathsf{x}})}$ rules; and every other rule in ρ^* is translated to their respective JS_2 rule. This results in a well-formed JS_2-derivation because $\hat{\mathsf{x}}$ and $\mathsf{origin}(\hat{\mathsf{x}})$ have the same priority.

It remains to show that every infinite branch β in π is successful. Towards that aim let γ be the corresponding infinite path in ρ^* and γ' be the corresponding infinite path through ρ. Let v be the root-most node in ρ that occurs infinitely often in γ'. Then v is the premise of a $\mathsf{D}_{\hat{\mathsf{x}}}$ rule and it holds that $\hat{\mathsf{x}}$ occurs in cofinitely many sequents in γ'. Consequently, infinitely many $\mathsf{Reset}_{\hat{\mathsf{x}}}$ rules are applied in γ. Therefore, $\mathsf{origin}(\hat{\mathsf{x}})$ occurs cofinitely often in β and infinitely many $\mathsf{Reset}_{\mathsf{origin}(\hat{\mathsf{x}})}$ rules are applied in β. This shows that π is a JS_2^∞-proof of Γ. □

5.3.7. THEOREM. *A pure sequent Γ is unsatisfiable iff $\mathsf{Circ}_2 \vdash \Gamma$.*

Proof:
Soundness follows from Lemma 5.3.6 together with Theorem 5.2.13. Completeness follows from Lemma 5.3.5 together with Theorem 5.2.13. □

5.4 Split proof system sCirc_2

Our overall strategy to prove interpolation is as follows: Given a Circ_2-proof π of φ, ψ we define a formula ι in the common vocabulary of φ and ψ and construct proofs π^l of φ, ι and π^r of $\bar{\iota}, \psi$. This is done by structural induction on π, where, roughly speaking, π^l contains those rules of π concerning descendants of φ and π^r contains those rules of π concerning descendants of ψ. In order to make that formal, we have to separate, in every sequent, those parts originating from φ and those originating from ψ. Sequents of this kind will be called split sequents.

5.4.1 sCirc_2-proofs

A *split sequent* is a pair (Γ, Δ), usually written as $\Gamma \mid \Delta$, where Γ and Δ are Circ_2 sequents and the sets of names occurring respectively in Γ and Δ are disjoint. Note that we do not require that Γ and Δ are disjoint. Given a split sequent $\Gamma \mid \Delta$ we call Γ the left and Δ the right component of the split sequent. We will write Ψ^l and Ψ^r for the left and right component of the split sequent Ψ, respectively, and use d as a variable ranging over the set $\{l, r\}$.

We will define sCirc$_2$-*proofs* consisting of split sequents, where Circ$_2$ rules are applied to either the left or the right component of a split sequent. Importantly, if Ψ^l is the left component of the conclusion of a rule R, then all formulas in the left component of a premise of R will be in $\mathsf{Clos}^\neg(\Psi^l)$.

5.4.1. DEFINITION. For any Circ$_2$ rule R we define a *left* sCirc$_2$ *rule* R^l as follows. If $\mathsf{R} \neq \langle a \rangle$ is of the form

$$\mathsf{R}\colon \frac{\Gamma_1 \quad \cdots \quad \Gamma_n}{\Gamma_0}$$

then R^l is of the form

$$\mathsf{R}^l\colon \frac{\Gamma_1 \mid \Delta \quad \cdots \quad \Gamma_n \mid \Delta}{\Gamma_0 \mid \Delta}$$

The rule $\langle a \rangle^l$ is of the form

$$\langle a \rangle^l\colon \frac{\varphi^\sigma, \Sigma, \langle \breve{a} \rangle^l \Gamma^\varepsilon, \Gamma^{\langle a \rangle^l \varphi} \mid \Pi, \langle \breve{a} \rangle^r \Delta^\varepsilon, \Delta^{\langle a \rangle^r \varphi}}{\langle a \rangle \varphi^\sigma, [a]\Sigma, \Gamma \mid [a]\Pi, \Delta}$$

Let Ψ^l and Ψ^r be the respective left and right component of the split sequent of its conclusion. Then we define

$$\langle \breve{a} \rangle^l \Gamma := \{\langle \breve{a} \rangle \gamma^\sigma \mid \gamma^\sigma \in \Gamma \text{ and } \langle \breve{a} \rangle \gamma \in \mathsf{Clos}^\neg(\Psi^l)\}.$$

The conditions in $\Gamma^{\langle a \rangle \varphi}$ are adapted, such that $\Gamma^{\langle a \rangle^l \varphi}$ is defined as

$$\begin{aligned} &\{\varphi \not\rightsquigarrow_k [\breve{a}]\chi \mid \langle a \rangle \varphi \not\rightsquigarrow_k \chi \in \Gamma \text{ and } [\breve{a}]\chi \in \mathsf{Clos}^\neg(\Psi^l)\} \\ \cup\ &\{[\breve{a}]\chi \rightsquigarrow_k \varphi \mid \chi \rightsquigarrow_k \langle a \rangle \varphi \in \Gamma \text{ and } [\breve{a}]\chi \in \mathsf{Clos}^\neg(\Psi^l)\} \\ \cup\ &\{\psi \not\rightsquigarrow_k [\breve{a}]\chi \mid [a]\psi \not\rightsquigarrow_k \chi \in \Gamma \text{ and } [\breve{a}]\chi \in \mathsf{Clos}^\neg(\Psi^l)\} \\ \cup\ &\{[\breve{a}]\chi \rightsquigarrow_k \psi \mid \chi \rightsquigarrow_k [a]\psi \in \Gamma \text{ and } [\breve{a}]\chi \in \mathsf{Clos}^\neg(\Psi^l)\} \end{aligned}$$

Analogously for $\langle \breve{a} \rangle^r \Delta$ and $\Delta^{\langle a \rangle^r \varphi}$.

Right sJS$_2$ *rules* are defined analogously. Additionally we allow so-called *split axioms* of the form

$$\mathsf{Ax1}'\colon \frac{}{\varphi^\sigma \mid \overline{\varphi}^\tau}$$

Split rules are either left rules, right rules or split axioms.

For most split rules the left and the right component of the split do not interact. The only exceptions are the modal rule $\langle a \rangle$ and the split axiom Ax1$'$. Note that for trace atoms there is no interaction between the left and the right component at all, and even the axiom Ax3 may only be applied if both a trace atom and its negated trace atom occur in the same component.

5.4.2. DEFINITION. The cyclic proof system sCirc$_2$ is defined from all split rules together with all finite paths.

Given sets of formulas Γ and Δ we say that there is a sCirc$_2$ proof of $\Gamma \mid \Delta$, if there is an sCirc$_2$-proof of which the root is labeled with $\Gamma^\varepsilon \mid \Delta^\varepsilon$.

5.4.2 Soundness and completeness of split proofs

For proving soundness of the split system, we translate sCirc_2-proofs to Circ_2-proofs.

5.4.3. Lemma. *If* $\mathsf{sCirc}_2 \vdash \Gamma \mid \Delta$*, then* $\mathsf{Circ}_2 \vdash \Gamma, \Delta$.

Proof:
Let π be an sCirc_2-proof of $\Gamma \mid \Delta$. We inductively translate π to a Circ_2-derivation ρ of Γ, Δ, such that every node labeled with $\Sigma \mid \Pi$ in π is translated to a node in ρ labeled with Σ, Π. This can be achieved by translating all rules of the form R^l and R^r to the corresponding rule R and split axioms to axioms $\mathsf{Ax1}$. If $\mathsf{R} = \langle a \rangle$, we may need to add extra weak rules, as the premise might contain more formulas as the corresponding premise of the split rule. It is easy to see that this results in a Circ_2-proof. □

In the soundness proof, it sufficed to translate sCirc_2-proofs to Circ_2 proofs. The converse translation from Circ_2-proofs to sCirc_2-proofs is more tricky, as we have to choose in which component formulas are put. In this translation, repeats in Circ_2-proofs are not necessarily translated to repeats in sCirc_2, we therefore opt for a detour via infinitary Circ_2 and sCirc_2-derivations: Given a Circ_2-proof ρ, we consider its infinite unfolding ρ^*. This infinite Circ_2-derivation is then translated to an sCirc_2-derivation π. Finally, π is folded into a cyclic sCirc_2-derivation π' completing the process.

5.4.4. Lemma. *If* $\mathsf{Circ}_2 \vdash \Gamma, \Delta$ *for a pure sequent* Γ, Δ*, then* $\mathsf{sCirc}_2 \vdash \Gamma \mid \Delta$.

Proof:
Let ρ be a Circ_2-proof of Γ, Δ and let ρ^* be the infinite unfolding of ρ. We first translate ρ^* to an sCirc_2-derivation π of $\Gamma \mid \Delta$. For the time being we assume that the bound variables in Γ and Δ are disjoint. Therefore, all formulas in $\mathsf{Clos}^\neg(\Gamma) \cap \mathsf{Clos}^\neg(\Delta)$ are fixpoint-free.

In the completeness proof of NW_2, an NW_2 proof ρ' was constructed such that all trace atoms $\varphi \rightsquigarrow_k \psi$ in ρ' are relevant, meaning that (i) $\psi \in \mathsf{Clos}^\neg(\varphi)$ and (ii) φ and ψ contain fixpoints. In the completeness proof of Circ_2 we translated ρ' to a JS_2-proof and further to a Circ_2-proof ρ without adding extra trace atoms. Thus, we may assume for every trace atom $\varphi \rightsquigarrow_k \psi$ in ρ that $\varphi, \psi \notin \mathsf{Clos}^\neg(\Gamma) \cap \mathsf{Clos}^\neg(\Delta)$ and either $\varphi, \psi \in \mathsf{Clos}^\neg(\Gamma)$ or $\varphi, \psi \in \mathsf{Clos}^\neg(\Delta)$. For simplicity, we write $\varphi \rightsquigarrow_k \psi \in \mathsf{Clos}^\neg(\Sigma)$ in the case that $\varphi, \psi \in \mathsf{Clos}^\neg(\Sigma)$.

We inductively translate ρ^* to an sCirc_2-derivation π of $\Gamma \mid \Delta$, where every node u labeled with Σ in ρ^* is translated to a node v (possibly with some additional nodes) in π labeled with $\Sigma^l \mid \Sigma^r$ such that

1. $\Sigma = \Sigma^l \cup \Sigma^r$,

2. $\Sigma^l \subseteq \mathsf{Clos}^\neg(\Gamma)$ and $\Sigma^r \subseteq \mathsf{Clos}^\neg(\Delta)$,

3. $\Sigma^l \cap \Sigma^r = \varnothing$,

The root Γ, Δ is translated to

$$\dfrac{\Gamma \mid \Delta \setminus \Gamma}{\Gamma \mid \Delta}\ \mathsf{weak}^r$$

For every rule in ρ^* we apply a corresponding left or right rule in π. By a case distinction on the applied rule we show how to satisfy the conditions 1 and 2.

- Ax1 can either be translated to $\mathsf{Ax1}^l$, $\mathsf{Ax1}^r$ or to a split axiom $\mathsf{Ax1}'$, depending on in which components the formulas φ and $\overline{\varphi}$ are located.

- Assume that in ρ^* the following $\langle a \rangle$ rule is applied:

$$\dfrac{\varphi^\sigma, \Sigma, \langle \breve{a} \rangle \Lambda^\varepsilon, \Lambda^{\langle a \rangle \varphi}, \Pi, \langle \breve{a} \rangle \Theta^\varepsilon, \Theta^{\langle a \rangle \varphi}}{\langle a \rangle \varphi^\sigma, [a]\Sigma, \Lambda, [a]\Pi, \Theta}\ \langle a \rangle$$

  Without loss of generality let the split of the translation of the conclusion in π be
$$\Psi = \langle a \rangle \varphi^\sigma, [a]\Sigma, \Lambda \mid [a]\Pi, \Theta.$$
  Let Ψ^l be the left, and Ψ^r be the right component of Ψ. If we just try to apply a $\langle a \rangle^l$ rule to Ψ this will not work: It could be that there is $\gamma^\tau \in \Lambda \setminus \Theta$ and $\langle \breve{a} \rangle \gamma^\varepsilon \in \langle \breve{a} \rangle \Lambda^\varepsilon$ such that $\langle \breve{a} \rangle \gamma \in \mathsf{Clos}^\neg(\Psi)$ but $\langle \breve{a} \rangle \gamma \notin \mathsf{Clos}^\neg(\Psi^l)$. Hence, $\langle \breve{a} \rangle \gamma^\varepsilon$ would be added neither in the left nor the right component of the premise of the $\langle a \rangle^l$ rule, yet in ρ^* the formula is added to the premise of $\langle a \rangle$.

  In this case we must have $\langle \breve{a} \rangle \gamma \in \mathsf{Clos}^\neg(\Psi^r)$. Thus $\gamma \in \mathsf{Clos}^\neg(\Psi^r)$ as well, and thence $\gamma \in \mathsf{Clos}^\neg(\Psi^l) \cap \mathsf{Clos}^\neg(\Psi^r)$. This yields that γ is fixpoint-free. For any such γ we apply an acut^r rule with cut-formula γ, where $\Lambda = \Lambda', \gamma$:[6]

$$\dfrac{\dfrac{}{\gamma^\tau \mid \overline{\gamma}^\varepsilon}\ \mathsf{Ax1}' \qquad \langle a \rangle \varphi^\sigma, [a]\Sigma, \Lambda', \gamma^\tau \mid [a]\Pi, \Theta, \gamma^\varepsilon}{\langle a \rangle \varphi^\sigma, [a]\Sigma, \Lambda', \gamma^\tau \mid [a]\Pi, \Theta}\ \mathsf{acut}^r$$

  Applying the modal rule will now make $\langle \breve{a} \rangle \gamma$ land in the proper (right) component of the premise. Likewise, applying a acut^l rule for every $\langle \breve{a} \rangle \delta^\varepsilon \in \langle \breve{a} \rangle \Theta^\varepsilon$, where $\langle \breve{a} \rangle \delta \in \mathsf{Clos}^\neg(\Psi^l) \setminus \mathsf{Clos}^\neg(\Psi^r)$ yields a split sequent, where we may apply a $\langle a \rangle^l$ rule and satisfy the conditions 1 – 2.

  For trace atoms $\gamma \rightsquigarrow_k \chi$ (and negated trace atoms $\gamma \not\rightsquigarrow_k \chi$) occurring in $\Lambda^{\langle a \rangle \varphi}$, this is not a problem, as there are no trace atoms where γ is fixpoint-free.

[6] In addition we implicitly weakened all unimportant side-formulas in the left premise.

- Assume that in ρ^* a $\mathsf{D}_{\hat{\mathsf{x}}}$ rule of the following form is applied:

$$\frac{\varphi_1^{\sigma\hat{\mathsf{x}}}, \ldots, \varphi_n^{\sigma\hat{\mathsf{x}}}, \Sigma}{\varphi_1^{\sigma}, \ldots, \varphi_n^{\sigma}, \Sigma} \; \mathsf{D}_{\hat{\mathsf{x}}}$$

  By induction on π we can show that in every annotated sequent no name occurs in both the left and the right component. This holds as only fresh names are introduced and in no rule do names cross the split.

  We have that $\sigma \neq \varepsilon$ and thus the formulas $\varphi_1^\sigma, \ldots, \varphi_n^\sigma$ either all belong to the left or the right component in π. Consequently, we can translate $\mathsf{D}_{\hat{\mathsf{x}}}$ to either $\mathsf{D}^l_{\hat{\mathsf{x}}}$ or $\mathsf{D}^r_{\hat{\mathsf{x}}}$.

- exp rules: Because no name occurs in both the left and the right component, Reset can always be translated to either Reset^l or Reset^r. Other exp rules can be split up into exp^l and exp^r rules.

- If the applied rule is acut (or tcut), add φ and $\overline{\varphi}$ (or $\varphi \rightsquigarrow_k \psi$ and $\varphi \not\rightsquigarrow_k \psi$) to the respective left components, if φ is in $\mathsf{Clos}^\neg$ of the left component of the conclusion, and to the respective right components otherwise.

- In the jump rules it holds that $\varphi^\sigma, \psi^\tau, \varphi \rightsquigarrow_k \psi$ are all either in $\mathsf{Clos}^\neg(\Gamma)$ or in $\mathsf{Clos}^\neg(\Delta)$, since all trace atoms are relevant. Similarly, for trans and $\mathsf{Ax3}$ all explicitly written formulas in its conclusion belong to the same component of the sequent.

- All other rules have only one explicitly written formula in the conclusion and thus can easily be translated to a left or right rule.

Condition 3 can be satisfied by applying weak^r if necessary. Thus we obtain an sCirc_2-derivation satisfying the specified conditions. The derivation ρ^* is regular as it is the infinite unfolding of the finite proof ρ. Therefore, π is regular as well.

Next we want to fold the sCirc_2-derivation π into an sCirc_2-proof π'. To do so the following claim is crucial:

`Claim 1`: Let β be an infinite branch in π. Then there is an assumption name $\hat{\mathsf{x}}$ and $d = l, r$ such that

1. $\hat{\mathsf{x}}$ occurs in Σ^d on cofinitely many split sequents Σ in β and
2. there are infinitely many applications of $\mathsf{Reset}^d_{\hat{\mathsf{x}}}$ on β.

`Proof of Claim 1`: Let γ be the corresponding infinite path of β in ρ^* and γ' be the corresponding infinite path through ρ. Let v be the root-most node in ρ that occurs infinitely often in γ'. Then v is the premise of a $\mathsf{D}_{\hat{\mathsf{x}}}$ rule and it holds that $\hat{\mathsf{x}}$ occurs in cofinitely many sequents in γ'. Consequently, infinitely many $\mathsf{Reset}_{\hat{\mathsf{x}}}$ rules are applied in γ. Therefore,

1. $\hat{\mathsf{x}}$ occurs in the control of Σ on cofinitely many sequents Σ on γ and
2. there are infinitely many applications of $\mathsf{Reset}_{\hat{\mathsf{x}}}$ on γ.

We can show inductively that $\hat{\mathsf{x}}$ either only occurs in left or only occurs in right components of β. If $\hat{\mathsf{x}}$ only occurs in left components of β, then infinitely many $\mathsf{Reset}_{\hat{\mathsf{x}}}$ rules are translated to $\mathsf{Reset}^{l}_{\hat{\mathsf{x}}}$ rules in π. Analogously, if $\hat{\mathsf{x}}$ occurs in cofinitely many right components of β. ⊣

Using Claim 1 we can fold π into an sCirc_2-proof using a similar argument as in the proof of Lemma 5.2.12. Let P_π be the set of infinite paths in π. For every infinite path $\beta = (\beta(i))_{i\in\omega}$ in P_π define minimal indices $j < k$ such that

1. $\pi_{\beta(j)} \sim \pi_{\beta(k)}$,
2. the parent nodes of j and k are labeled with $\mathsf{Reset}^{d}_{\hat{\mathsf{x}}}$ for some $d = l, r$.

Because π is regular and Claim 1, such indices always exist. For each path $\beta \in P_\pi$ we define two nodes $c_\beta := \beta(j)$ and $l_\beta := \beta(k)$.

We will transform the sCirc_2-derivation π into an sCirc_2-proof π' through the following steps:

(i) Choose an infinite path β such that c_β is minimal, meaning that there is no $\gamma \in P_\pi$ with $c_\gamma < c_\beta$;

(ii) Assume that the parent nodes of c_β and l_β are labeled with[7]

$$\frac{\Lambda \mid \Pi, \varphi_1^{\sigma\hat{\mathsf{x}}}, \ldots, \varphi_n^{\sigma\hat{\mathsf{x}}}}{\Lambda \mid \Pi, \varphi_1^{\sigma\hat{\mathsf{x}}\mathsf{x}_1}, \ldots, \varphi_n^{\sigma\hat{\mathsf{x}}\mathsf{x}_n}}\ \mathsf{Reset}^{r}_{\hat{\mathsf{x}}}$$

Introduce a $\mathsf{D}^{r}_{\hat{\mathsf{y}}}$ node at c_β with $\Omega(\hat{\mathsf{y}}) = \Omega(\hat{\mathsf{x}})$;

(iii) In the maximal subtree of π rooted at the child of c_β, substitute the assumption name $\hat{\mathsf{x}}$ with $\hat{\mathsf{x}}\hat{\mathsf{y}}$ and replace $\mathsf{Reset}^{r}_{\hat{\mathsf{x}}}$ rules with $\mathsf{Reset}^{r}_{\hat{\mathsf{y}}}$ rules;

(iv) We let the parent of l_β – which is labeled with $\mathsf{Reset}^{r}_{\hat{\mathsf{y}}}$ – be discharged by $\hat{\mathsf{y}}$;

(v) Remove all infinite paths from P_π that contain l_β;

(vi) Repeat until π' is finite.

Because we choose paths β such that c_β is minimal, for all other infinite paths γ, the nodes c_γ and l_γ still satisfy the conditions 1 and 2 stated above. König's Lemma shows that this procedure terminates and thus it results in an sCirc_2-proof π' of $\Gamma \mid \Delta$.

[7]The case where $d = l$ is analogous.

Lastly, we deal with the general case, where Γ and Δ may share bound variables. Let Γ' be an α-equivalent sequent of Γ, where all bound variables in Γ' and Δ are disjoint; for example replace every bound variable in Γ by a fresh new variable not occurring in either Γ or Δ. By the above reasoning we obtain an sCirc_2-proof π' of $\Gamma' \mid \Delta$. In π' we can translate back all newly introduced bound variables. This yields an sCirc_2-proof π of $\Gamma \mid \Delta$. □

5.4.5. Theorem. *A pure sequent* Γ, Δ *is unsatisfiable iff* $\mathsf{sCirc}_2 \vdash \Gamma \mid \Delta$.

Proof:
Theorem 5.3.7 yields that Γ, Δ is unsatisfiable iff there is a $\mathsf{Circ}_2 \vdash \Gamma, \Delta$. Therefore the soundness of sCirc_2 follows from Lemma 5.4.3 and the completeness from Lemma 5.4.4. □

5.5 Interpolation

In the previous section we saw that a pure sequent Γ, Δ is unsatisfiable iff $\mathsf{sCirc}_2 \vdash \Gamma \mid \Delta$. We can now use this proof system to show the main theorem of this chapter.

5.5.1. Theorem (Craig interpolation). *Let* φ *and* ψ *be two* $\mathcal{L}^2_\mu$*-formulas such that* $\varphi \vDash \psi$. *Then there is an interpolant for* φ *and* ψ.

Proof:
Follows from Lemma 5.5.3. □

As an immediate consequence of Craig interpolation we obtain Beth definability. Where $\varphi(p)$ is a $\mathcal{L}^2_\mu$-formula, we use $\varphi(q)$ as an abbreviation of $\varphi[q/p]$.

5.5.2. Corollary (Beth definability). *Let* $p, q \in \mathsf{Prop}$ *and let* $\varphi(p)$ *be a* $\mathcal{L}^2_\mu$ *formula. If* $\varphi(p), \varphi(q) \vDash p \leftrightarrow q$, *then there is a formula* χ *with* $\mathsf{Voc}(\chi) \subseteq \mathsf{Voc}(\varphi) \setminus \{p\}$ *and* $\varphi(p) \vDash p \leftrightarrow \chi$.

Proof:
Apply Craig interpolation to $\varphi(p), p \vDash \varphi(q) \rightarrow q$. □

The remainder of this chapter is devoted to the proof of Theorem 5.5.1. We will use the split sequent system sCirc_2 to find Craig interpolants for $\mathcal{L}^2_\mu$. We first transfer the concept of interpolation from formulas to split sequents, calling a formula ι an *interpolant* for an unsatisfiable split sequent $\Gamma \mid \Delta$ if $\mathsf{Voc}(\iota) \subseteq \mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Delta)$, and both sequents $\Gamma \mid \iota$ and $\bar{\iota} \mid \Delta$ are unsatisfiable. Since we have $\varphi \vDash \psi$ iff $\varphi \mid \overline{\psi}$ is unsatisfiable, it is easy to see that a formula ι is an interpolant for the formulas φ and ψ iff it is an interpolant for the split sequent $\varphi \mid \overline{\psi}$. By the soundness and completeness of the split system $\mathsf{sCirc_2}$ it therefore suffices to prove the following result.

5.5.3. Lemma. *Let π be an $\mathsf{sCirc_2}$ proof of $\Gamma \mid \Delta$. Then there is a formula ι such that $\mathsf{Voc}(\iota) \subseteq \mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Delta)$ and for which there are $\mathsf{sCirc_2}$ proofs π^l of $\Gamma \mid \iota$ and π^r of $\overline{\iota} \mid \Delta$.*

Proof:
We let L_π and K_π denote, respectively, the sets of discharged leaves and companions of π, in particular $\mathsf{K}_\pi := \{c(t) \mid t \in \mathsf{L}_\pi\}$. Recall that $<$ and $\leq$ denote, respectively, the transitive and reflexive-transitive closure of the parent relation $\lessdot$ of π. Given $u \in \pi$ we set

$$\mathsf{L}_{<u} := \{t \in \mathsf{L}_\pi \mid c(t) < u \leq t\} \quad \text{and} \quad \mathsf{K}_{<u} := \{c(t) \mid t \in \mathsf{L}_{<u}\}.$$

Let $V_\pi := \{x_c \mid c \in \mathsf{K}_\pi\}$ be a set of fresh new variables and let $V_u := \{x_c \in V_\pi \mid c \in \mathsf{K}_{<u}\}$. Our interpolant for $\Gamma \mid \Delta$ will be a formula with bound variables in V_π.

For each node $u \in \pi$ labeled with $\Gamma_u \mid \Delta_u$ we define

1. a formula ι_u with $\mathsf{FV}(\iota_u) \subseteq V_u$ and $\mathsf{Voc}(\iota_u) \subseteq \mathsf{Voc}(\Gamma_u) \cap \mathsf{Voc}(\Delta_u)$,
2. a derivation π^l_u of $\Gamma_u \mid \iota_u$ such that all open assumptions in π^l_u are labeled with $\Gamma_t \mid x_{c(t)}$ for some $t \in \mathsf{L}_{<u}$ and
3. a derivation π^r_u of $\overline{\iota_u} \mid \Delta_u$ such that all open assumptions in π^r_u are labeled with $x_{c(t)} \mid \Delta_t$ for some $t \in \mathsf{L}_{<u}$.

We define $\iota_u, \pi^l_u, \pi^r_u$ by induction on the proof tree of π, starting from the leaves. For the root r of π this will yield $\iota := \iota_r$ such that $\mathsf{Voc}(\iota) \subseteq \mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Delta)$ and proofs π^l of $\Gamma \mid \iota$ and π^r of $\overline{\iota} \mid \Delta$. The construction is defined by a case distinction on the last applied rule.

- *Axioms*: If u is labeled with an axiom of the form $\varphi^\sigma \mid \overline{\varphi}^\tau$, then $\iota_u := \overline{\varphi}$ and dually $\iota_u := \varphi$ if φ and $\overline{\varphi}$ are swapped. Otherwise an axiom is applied, where either $\varphi, \overline{\varphi}$; $\bot$; $\varphi \rightsquigarrow_k \psi, \varphi \not\rightsquigarrow_k \psi$ or $\varphi \rightsquigarrow_{2k} \varphi$ is on the left or the right side of the split. If it is on the left, let $\iota_u := \top$ and otherwise $\iota_u := \bot$. It is straightforward to check the conditions 1–3.
- *Discharged leaves:* For every discharged leaf u labeled with $\Gamma_u \mid \Delta_u$ with companion node c, let $\iota_u := x_c$. Define π^l_u and π^r_u to be the derivations consisting of one open assumption $\Gamma_u \mid x_c$ and $x_c \mid \Delta_u$, respectively. Clearly, the conditions 1–3 hold.

- *Companion nodes:* Let u be labeled with $\mathsf{D}^r_{\hat{\mathsf{x}}}$ and let v be its child. By induction hypothesis there is a formula ι_v and derivations π^l_v and π^r_v satisfying conditions 1–3. We define $\iota_u := \mu x_u.\iota_v$. In order to define π^l_u we transform the derivation π^l_v of $\Gamma_u \mid \iota_u$. Let O be the set of open assumptions in π^l_v labeled with $\Gamma_t \mid x_u$ for some discharged leaf t such that $c(t) = u$ and let P be the set of all other open assumptions.

  By uniformly substituting every occurrence of x_u in π^l_v with $\mu x_u.\iota_v$ we obtain a derivation ρ_v of $\Gamma_u \mid \iota_v[\mu x_u.\iota_v/x_u]$, where all open assumptions are either in P or labeled with $\Gamma_t \mid \mu x_u.\iota_v$ for some t as above.

  Let $\mathsf{y} \in X^V_k$ be a fresh variable name and $\hat{\mathsf{y}} \in X^A_k$ be a fresh assumption name such that $k = \Omega_{2\mu}(\mu x_u.\iota_v)$. We let $\rho^{\mathsf{y}\hat{\mathsf{y}}}_v$ be obtained from ρ_v by replacing every node w in the strongly connected subtree $\mathrm{scst}(u)$ of u labeled with $\Gamma_w \mid \iota_w^{\sigma(w)}$ with w', where w' is labeled with $\Gamma_w \mid \iota_w^{\mathsf{y}\hat{\mathsf{y}}\sigma(w)}$. If a node w is not in $\mathrm{scst}(u)$, but its parent is, then add an exp rule to remove the names y and $\hat{\mathsf{y}}$. This results in a well-formed derivation because y and $\hat{\mathsf{y}}$ are higher-ranking names than all names in $\sigma(w)$ for all w.

  We define the following derivation π^l_u, where all open assumptions are in P and all assumptions in O are discharged as follows.

$$\dfrac{\dfrac{\dfrac{\lceil \Gamma_t \mid \iota_v[\mu x_u.\iota_v/x_u]^{\mathsf{y}\hat{\mathsf{y}}\mathsf{z}}\rceil^{\hat{\mathsf{y}}}}{\Gamma_t \mid \mu x_u.\iota_v^{\mathsf{y}\hat{\mathsf{y}}}}\,\mu^r \\ \vdots\ \rho^{\mathsf{y}\hat{\mathsf{y}}}_v \\ \Gamma_u \mid \iota_v[\mu x_u.\iota_v/x_u]^{\mathsf{y}\hat{\mathsf{y}}}}{\Gamma_u \mid \iota_v[\mu x_u.\iota_v/x_u]^{\mathsf{y}}}\,\mathsf{D}^r_{\hat{\mathsf{y}}}}{\Gamma_u \mid \mu x_u.\iota_v}\,\mu^r$$

  For the definition of π^r_u, we let ρ^r_v be obtained from π^r_v by uniformly substituting every occurrence of x_u in π^r_v with $\nu x_u.\overline{\iota_v}$. We let π^r_u be the following derivation[8]

$$\dfrac{\dfrac{\lceil \nu x_u.\overline{\iota_v} \mid \Delta_t\rceil^{\hat{\mathsf{x}}} \\ \vdots\ \rho^r_v \\ \overline{\iota_v}[\nu x_u.\overline{\iota_v}/x_u] \mid \Delta_v}{\nu x_u.\overline{\iota_v} \mid \Delta_v}\,\nu^l}{\nu x_u.\overline{\iota_v} \mid \Delta_u}\,\mathsf{D}^r_{\hat{\mathsf{x}}}$$

  It holds that $\mathsf{FV}(\iota_u) = \mathsf{FV}(\iota_v) \setminus \{x_u\} \subseteq V_u$ and $\mathsf{Voc}(\iota_u) = \mathsf{Voc}(\iota_v)$, thus the conditions 1–3 are satisfied. The case where a D^l rule is applied is dual with $\iota_u := \nu x_u.\iota_v$.

[8]For readability, we omit annotations of ε.

- *Modal rules:* Let u be labeled with $\langle a\rangle^r$ and let v be its child. If the left component of v is empty, define $\iota_u := \bot$, then π^l_u is an instance of $\mathsf{Ax2}$ and π^r_u is obtained from π_v by applications of $\langle a\rangle$ and weak. The conditions 1–3 are clearly satisfied.

  Otherwise both components of the premise of $\langle a\rangle$ in π_u are non-empty. Then it follows that the action a belongs to the vocabulary of both Γ_u and Δ_u. To see that for the left component, let $\Gamma_v = \Sigma, \langle\breve{a}\rangle\Pi^\varepsilon_u$ be non-empty. If Σ is non-empty, then clearly $a \in \mathsf{Voc}(\Gamma_u)$. Otherwise, there is $\langle\breve{a}\rangle\gamma^\varepsilon \in \langle\breve{a}\rangle\Pi^\varepsilon_u$, yet this is only the case if $\langle\breve{a}\rangle\gamma \in \mathsf{Clos}^\neg(\Gamma_u)$, which implies $a \in \mathsf{Voc}(\Gamma_u)$ indeed.

  We define $\iota_u := \langle a\rangle\iota_v$, The proofs π^d_u are obtained from π^d_v by applying a $\langle a\rangle^r$ rule for $d = l, r$. It holds that $\mathsf{Voc}(\iota_u) = \mathsf{Voc}(\iota_v) \cup \{a\} \subseteq \mathsf{Voc}(\Gamma_u) \cap \mathsf{Voc}(\Delta_u)$ and therefore the conditions 1–3 are satisfied. The case of a $\langle a\rangle^l$ rule is dual with $\iota_u := \top$ or $\iota_u := [a]\iota_v$.

- *Unary rules:* If u is the conclusion of a unary rule different from D and $\langle a\rangle$ with premise v, define $\iota_u := \iota_v$. The proofs π^l_u and π^r_u are defined straightforwardly.

- *Binary rules:* If u is the conclusion of a binary rule R with premises v and w, than $\iota_u := \iota_v \wedge \iota_u$ or $\iota_u := \iota_v \vee \iota_w$, depending on whether R is a left or right rule. The proofs π^l_u and π^r_u are defined straightforwardly. For example, if $\mathsf{R} = \mathsf{acut}^l$, then π^l_u is the following proof. Note that we apply weak^r rules implicitly on both branches and omit annotations of ε.

$$\dfrac{\dfrac{\begin{matrix}\pi^l_v\\ \varphi, \Gamma_u \mid \iota_v\end{matrix}}{\varphi, \Gamma_u \mid \iota_v \wedge \iota_w}\wedge^r \qquad \dfrac{\begin{matrix}\pi^l_w\\ \overline{\varphi}, \Gamma_u \mid \iota_w\end{matrix}}{\overline{\varphi}, \Gamma_u \mid \iota_v \wedge \iota_w}\wedge^r}{\Gamma_u \mid \iota_v \wedge \iota_w}\mathsf{acut}^l$$

  The proof π^r_u is defined as follows.

$$\dfrac{\begin{matrix}\pi^r_v\\ \overline{\iota_v} \mid \Delta_u\end{matrix} \qquad \begin{matrix}\pi^r_w\\ \overline{\iota_w} \mid \Delta_u\end{matrix}}{\overline{\iota_v} \vee \overline{\iota_w} \mid \Delta_u}\vee^l$$

  As every application of acut is analytic it holds that $\mathsf{FV}(\varphi) \subseteq \mathsf{FV}(\Gamma_u)$. Therefore $\mathsf{Voc}(\iota_v \wedge \iota_w) \subseteq \mathsf{Voc}(\Gamma_u, \varphi) \cap \mathsf{Voc}(\Delta_u) = \mathsf{Voc}(\Gamma_u) \cap \mathsf{Voc}(\Delta_u)$, hence conditions 1–3 are satisfied.

□

5.6 Conclusion

We presented three sound and complete proof systems for the two-way modal μ-calculus. The first one, NW_2, is infinitary, whereas the latter two, JS_2 and Circ_2, are cyclic with a path-based soundness condition. We used the system Circ_2 to show that the logic enjoys Craig interpolation Property and the Beth definability property. Below we mention some questions for further research.

5.6.1. Question. *Uniform interpolation* is a strengthening of Craig interpolation: A logic has the uniform interpolation property if interpolants for $\varphi \vDash \psi$ can be defined *uniformly* in ψ. That is, given a formula φ, one can find a formula ι that is an interpolant of $\varphi \vDash \psi$ for any ψ whose vocabulary is restricted to a specified subset of the vocabulary of φ. Cyclic proofs have been used to show uniform interpolation for the one-way modal μ-calculus [ALM21]. It would be interesting to see whether a similar approach could be applied to the two-way modal μ-calculus using our cyclic proof system Circ_2.

5.6.2. Question. Benedikt and collaborators show interpolation for *guarded fixpoint logics* [BCV15; BBV19], formalisms that *extend* $\mathcal{L}^2_\mu$ in expressive power. Their approach is model-theoretic in nature. It would be interesting to compare these results to ours, and to see whether our approach could lead to proof systems for their logics, or whether their model-theoretic approach would also work for the two-way modal μ-calculus. A similar question applies to the work of French on modal logics extended with bisimulation quantifiers [Fre06; Fre07]. Given the connection between uniform interpolation and bisimulation quantifiers [DH00], French's results might even lead to an (indirect) proof that the two-way μ-calculus has uniform interpolation.

Chapter 6

Interpolation for Converse PDL

The language of Converse PDL (in short, CPDL) is obtained from PDL by adding converse modalities. For program logics such as PDL, the inclusion of converse modalities is both natural and fruitful [LPZ85]. Informally, converse modalities express strongest postconditions whereas forward modalities express weakest preconditions [HKQ03]. In this chapter, we introduce a sound and complete cyclic proof system CPDL_f for CPDL and use it to show that the logic has interpolation.

The proof system CPDL_f takes inspiration from the proof systems introduced in Chapter 5 for the two-way modal μ-calculus. Yet, due to the simpler shape of fixpoints in CPDL, trace atoms may be omitted. The cyclic system CPDL is path-based and employs focus-style annotations inspired by [MV21a]. Due to the simple form of annotations, we prove completeness of the system directly without referring to automata theory.

For PDL, the question whether the logic has interpolation has been a long-standing problem, see [BGHRDV25] for a historic survey on attempted proofs. Recently, the property has been shown by Borzechowski et al. [BGHRDV25] using ideas from Borzechowski [Bor88]. Our work extends [BGHRDV25] to include converse modalities. Notable differences are that our proof system features an analytic cut rule and a more involved modal rule, while, on the other hand, our rules for the program constructors are simpler than those in [BGHRDV25]. Additionally, our proof of correctness of the interpolant is purely proof-theoretic.

In order to motivate our interpolation proof we recall Maehara's method as introduced in Chapter 5. We will present this technique in a slightly different light: Given a split proof π one can define equations relating interpolants for the nodes in π. For instance, assume that the following $\vee^r$ rule appears in π:

$$\frac{\Gamma \mid \Delta, \varphi_0 \qquad \Gamma \mid \Delta, \varphi_0}{\Gamma \mid \Delta, \varphi_0 \vee \varphi_1} \vee^r$$

Let ι_0 and ι_1 be the respective interpolants of the premises of $\vee^r$ and ι_2 be the interpolant of its conclusion. Then we have the equation $\iota_2 = \iota_0 \vee \iota_1$. Combining those equations for all rules in π results in a *system of equations*. By solving this system of equations we obtain an interpolant for all nodes in π, in particular an interpolant for its root.

In the context of cyclic proofs, variables can appear on both sides of the equations, leading to a *system of fixpoint equations*. The two-way modal μ-calculus is expressive enough to define arbitrary fixpoints, allowing us to solve arbitrary fixpoint equations. One might accredit the success of our interpolation proof in Chapter 5 to the fact that the system of equations defined by Maehara's method can be solved within $\mathcal{L}^2_\mu$.

In CPDL, only certain kinds of fixpoints can be expressed, so this approach does not always yield interpolants. While every system of fixpoint equations S has a solution in the more expressive logic $\mathcal{L}^2_\mu$, that solution may not lie within CPDL. This is where the main idea from Borzechowsky [Bor88] comes in. By introducing a certain equivalence relation on the proof tree, we can define a different system of fixpoint equations S', that has the same solution as S. Importantly, S' is solvable within CPDL, and its solution therefore yields an interpolant.

6.1 Proof system CPDL_f

We introduce the path-based cyclic proof system CPDL_f. In this chapter we will simply write *formulas* for CPDL-formulas and fix a finite set of CPDL-formulas Φ that is closed under $\mathsf{Clos}^\neg$.

An *annotated formula* is a pair (φ, b), usually denoted as φ^b, where φ is a formula and b is either u (*out of focus*) or f (*in focus*). An *annotated sequent* Γ is a set of annotated formulas, such that at most one formula in Γ is in focus. We call a sequent *focused* if it has a formula in focus and *unfocused* otherwise. Given a set of formulas Δ, we define the annotated sequent $\Delta^u := \{\varphi^u \mid \varphi \in \Delta\}$. For an annotated sequent Γ we define

$$\Gamma^- := \{\varphi \mid \varphi^b \in \Gamma\},$$
$$\Gamma^u := \{\varphi^u \mid \varphi^b \in \Gamma\},$$
$$[a]\Gamma := \{[a]\varphi^b \mid \varphi^b \in \Gamma\}.$$

We read annotated sequents *conjunctively* and say that Γ is *satisfiable* if $\bigwedge \Gamma^-$ is satisfiable and call Γ *unsatisfiable* otherwise. If no confusion is likely, we will call annotated sequents just sequents.

The rules of the derivation system CPDL_f are given in Figure 6.1. Note that the calculus aims to derive sequents that are *unsatisfiable*. Apart from the annotations, the rules are as expected. In the rules $\langle a \rangle$, $\wedge$, $\vee$, $\langle ; \rangle$, $[;]$, $\langle \cup \rangle$, $[\cup]$, $\langle ? \rangle$, $[?]$, $\langle * \rangle$ and $[*]$ we call the single explicitly written formula in its conclusion the *principal formula* of the rule. We only allow applications of rules R with a principal formula different from $\langle \alpha \rangle \varphi$ if the principal formula is out of focus.

(Hence, to apply such a rule to a formula in focus, first a u rule has to be applied.) The modal rule $\langle a\rangle$ is only allowed if its principal formula $\langle a\rangle\varphi$ is in focus.

In applications of acut we demand that $\varphi \in \mathsf{Clos}^{\neg}(\Gamma)$. For a modal rule $\langle a\rangle$ with conclusion $\Theta = \langle a\rangle\varphi^f, [a]\Sigma, \Gamma$ the sequent $\langle\breve{a}\rangle\Gamma$ is defined as $\langle\breve{a}\rangle\Gamma := \{\langle\breve{a}\rangle\chi^u \mid \chi^u \in \Gamma \text{ and } \langle\breve{a}\rangle\chi \in \mathsf{Clos}^{\neg}(\Theta)\}$. This ensures that all rules are analytic.

$$\mathsf{Ax1}: \frac{}{\varphi^u, \overline{\varphi}^u} \qquad \mathsf{Ax2}: \frac{}{\bot^u} \qquad \wedge: \frac{\varphi^u, \psi^u, \Gamma}{\varphi \wedge \psi^u, \Gamma} \qquad \vee: \frac{\varphi^u, \Gamma \qquad \psi^u, \Gamma}{\varphi \vee \psi^u, \Gamma}$$

$$\langle ; \rangle: \frac{\langle\alpha\rangle\langle\beta\rangle\varphi^b, \Gamma}{\langle\alpha;\beta\rangle\varphi^b, \Gamma} \qquad [;]: \frac{[\alpha][\beta]\varphi^u, \Gamma}{[\alpha;\beta]\varphi^u, \Gamma} \qquad \langle\cup\rangle: \frac{\langle\alpha\rangle\varphi^b, \Gamma \qquad \langle\beta\rangle\varphi^b, \Gamma}{\langle\alpha\cup\beta\rangle\varphi^b, \Gamma}$$

$$\langle * \rangle: \frac{\langle\alpha\rangle\langle\alpha^*\rangle\varphi^b, \Gamma \qquad \varphi^b, \Gamma}{\langle\alpha^*\rangle\varphi^b, \Gamma} \qquad [*]: \frac{[\alpha][\alpha^*]\varphi^u, \varphi^u, \Gamma}{[\alpha^*]\varphi^u, \Gamma} \qquad [\cup]: \frac{[\alpha]\varphi^u, [\beta]\varphi^u, \Gamma}{[\alpha\cup\beta]\varphi^u, \Gamma}$$

$$\langle ? \rangle: \frac{\psi^u, \varphi^b, \Gamma}{\langle\psi?\rangle\varphi^b, \Gamma} \qquad [?]: \frac{\overline{\psi}^u, \Gamma \qquad \varphi^u, \Gamma}{[\psi?]\varphi^u, \Gamma} \qquad \langle a \rangle: \frac{\varphi^f, \Sigma, \langle\breve{a}\rangle\Gamma}{\langle a\rangle\varphi^f, [a]\Sigma, \Gamma}$$

$$\mathsf{f}: \frac{\varphi^f, \Gamma}{\varphi^u, \Gamma} \qquad \mathsf{u}: \frac{\varphi^u, \Gamma}{\varphi^f, \Gamma} \qquad \mathsf{weak}: \frac{\Gamma}{\varphi^u, \Gamma} \qquad \mathsf{acut}: \frac{\varphi^u, \Gamma \qquad \overline{\varphi}^u, \Gamma}{\Gamma}$$

$$\mathsf{D}_\dagger: \begin{array}{c} [\Gamma]^\dagger \\ \vdots \\ \dfrac{\Gamma}{\Gamma} \end{array}$$

Figure 6.1: Rules of CPDL_f

6.1.1. Definition. A finite path β in a CPDL_f-derivation is *successful* if

1. every sequent on β has a formula in focus, and
2. there is a node on β where the formula in focus is principal.

6.1.2. Definition (Cyclic proof). The cyclic proof system CPDL_f is path-based and defined from the rules in Figure 6.1 together with all finite successful paths. For a set of unannotated formulas Δ we define $\vdash \Delta$ as $\vdash \Delta^u$.

6.1.3. Remark. As mentioned in Chapter 2, CPDL corresponds to a fragment of the two-way modal μ-calculus. For that reason, the proof system CPDL_f is heavily inspired by the proof systems NW_2 and JS_2 for $\mathcal{L}^2_\mu$. We deal with converse modalities in a similar way, as can be observed by looking at the rules $\langle a \rangle$ and acut, that are almost identical in both systems. Notably, we do not add trace atoms in CPDL_f. This can be explained as follows. CPDL corresponds to the *completely additive two-way μ-calculus $\mathcal{L}_\mu^{2ca}$*; recall that a $\mathcal{L}_\mu^2$-formula φ is in $\mathcal{L}_\mu^{2ca}$, if for any subformula $\mu x.\psi$ of φ, the variable x in ψ is not in the scope of a $\Box$-modality, an essential conjunction or a ν-operator; and dually for any subformula $\nu x.\psi$ of φ.

Consider an NW_2 proof π of an $\mathcal{L}_\mu^{2ca}$-sequent Γ. All successful traces in π pass through infinitely many μ-fixpoints. Therefore, cofinitely many formulas on such a successful trace are in the scope of a μ-formula. Consequently, because all formulas in π are in $\mathcal{L}_\mu^{2ca}$, successful traces, from some point onward, do not pass through box-formulas.

Let us now study the kind of trace atoms introduced in NW_2. We introduce *local* trace atoms in the rules $\wedge$, $\vee$ and η, combine them with trans and then transform them in $\langle a \rangle$. All trace atoms introduced in $\langle a \rangle$ contain a box-formula. As successful traces do not pass box-formulas in π, these trace atoms are not needed in π. Traces through local trace atoms – where traces do not pass through modalities – may be replaced by traces through the principal and auxiliary formulas of rules. Therefore, on successful paths in π, we can always find a successful trace that does not pass through trace atoms. This implies that trace atoms are not needed for NW_2-proofs of $\mathcal{L}_\mu^{2ca}$-sequents. As a consequence, trace atoms may be omitted in CPDL_f.

6.1.4. Remark. In Chapter 5, we first defined a trace-based proof system NW_2 for $\mathcal{L}_\mu^2$, and then used automata theory to obtain a path-based cyclic system JS_2. In this chapter on the other hand, we directly define the path-based proof system CPDL_f. The reason we choose this approach is that annotations in CPDL_f are much simpler, making a direct completeness proof more attainable. The simpler form of annotations in CPDL_f can be attributed to the specific fragment of the two-way modal μ-calculus that CPDL corresponds to. For the alternation-free modal μ-calculus, Marti and Venema [MV21a] introduced a focus-style proof system, where formulas are annotated by one bit of information – they are in focus or out of focus. We will study this system in Chapter 7. Because CPDL corresponds to a fragment of the alternation-free two-way modal μ-calculus, we can use the same form of annotations. More specifically, CPDL corresponds to the completely additive two-way μ-calculus, where μ-fixpoint variables are not in the scope of essential conjunctions. In essential conjunctions, both conjuncts may be put in focus. As those are avoided in CPDL, one formula in focus suffices in any sequent.

The following theorem states the soundness and completeness of CPDL_f. This result follows as a special case of the soundness and completeness of the split proof system that we introduce in the next section.

6.1.5. Theorem (Soundness and Completeness). *A sequent* Γ *is unsatisfiable iff* $\mathsf{CPDL}_f \vdash \Gamma$.

6.1.6. Example. In Example 2.5.16 we saw that $\langle a^*\rangle p \vDash q \to \langle a^*; p?; \breve{a}^*\rangle q$. Therefore, the sequent

$$\Gamma := \langle a^*\rangle p, q, [a^*; p?; \breve{a}^*]\overline{q}$$

is unsatisfiable and we can give a CPDL_f-proof π of Γ. For convenience, we define $\varphi := [a^*][p?][\breve{a}^*]\overline{q}$. The proof π is given as follows. Note that we imply weak rules implicitly and omit annotations of u for readability.

$$\pi : \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{[\breve{a}][\breve{a}^*]\overline{q}, \overline{q}, q}\,\mathsf{Ax1}}{[\breve{a}^*]\overline{q}, q}\,[*] \qquad \dfrac{\rho}{\langle \breve{a}^*\rangle q, \langle a\rangle\langle a^*\rangle p, [a]\varphi}}{\langle a\rangle\langle a^*\rangle p, q, [a]\varphi}\,\mathsf{acut} \qquad \dfrac{\dfrac{}{p, \overline{p}}\,\mathsf{Ax1} \qquad \dfrac{\dfrac{}{q, [\breve{a}][\breve{a}^*]\overline{q}, \overline{q}}\,\mathsf{Ax1}}{q, [\breve{a}^*]\overline{q}}\,[*]}{p, q, [p?][\breve{a}^*]\overline{q}}\,[?]}{\langle a^*\rangle p, q, [a]\varphi, [p?][\breve{a}^*]\overline{q}}\,\langle * \rangle}{\langle a^*\rangle p, q, [a^*][p?][\breve{a}^*]\overline{q}}\,[*]}{\langle a^*\rangle p, q, [a^*; p?][\breve{a}^*]\overline{q}}\,[;]}{\langle a^*\rangle p, q, [a^*; p?; \breve{a}^*]\overline{q}}\,[;]}{}$$

Note that $[\breve{a}^*]\overline{q} \in \mathsf{Clos}^\neg(\varphi)$ and therefore the application of acut is analytic. The CPDL-proof ρ is given as follows.

$$\rho : \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{[\breve{a}][\breve{a}^*]\overline{q}, \langle \breve{a}\rangle\langle \breve{a}^*\rangle q}\,\mathsf{Ax1}}{[\breve{a}^*]\overline{q}, \langle \breve{a}\rangle\langle \breve{a}^*\rangle q}\,[*] \qquad [\langle \breve{a}^*\rangle q, \langle a\rangle\langle a^*\rangle p^{f}, [a]\varphi]^\dagger}{\langle \breve{a}\rangle\langle \breve{a}^*\rangle q, \langle a\rangle\langle a^*\rangle p^{f}, [a]\varphi}\,\mathsf{acut} \qquad \dfrac{\dfrac{}{p, \overline{p}}\,\mathsf{Ax1} \qquad \dfrac{\dfrac{}{\langle \breve{a}\rangle\langle \breve{a}^*\rangle q, [\breve{a}][\breve{a}^*]\overline{q}}\,\mathsf{Ax1}}{\langle \breve{a}\rangle\langle \breve{a}^*\rangle q, [\breve{a}^*]\overline{q}}\,[*]}{\langle \breve{a}\rangle\langle \breve{a}^*\rangle q, p, [p?][\breve{a}^*]\overline{q}}\,[?]}{\langle \breve{a}\rangle\langle \breve{a}^*\rangle q, \langle a^*\rangle p^{f}, [a]\varphi, [p?][\breve{a}^*]\overline{q}}\,\langle * \rangle}{\langle \breve{a}\rangle\langle \breve{a}^*\rangle q, \langle a^*\rangle p^{f}, \varphi}\,[*]}{\langle \breve{a}^*\rangle q, \langle a\rangle\langle a^*\rangle p^{f}, [a]\varphi}\,\langle a\rangle}{\dfrac{\langle \breve{a}^*\rangle q, \langle a\rangle\langle a^*\rangle p^{f}, [a]\varphi}{\langle \breve{a}^*\rangle q, \langle a\rangle\langle a^*\rangle p, [a]\varphi}\,\mathsf{f}}\,\mathsf{D}_\dagger$$

Take a look at the application of the $\langle a\rangle$ rule in ρ. The formula $\langle \breve{a}\rangle\langle \breve{a}^*\rangle q$ is in $\mathsf{Clos}^\neg(\varphi)$ and therefore $\langle \breve{a}\rangle\langle \breve{a}^*\rangle q \in \langle \breve{a}\rangle\Delta$ for $\Delta = \langle \breve{a}^*\rangle q, \langle a\rangle\langle a^*\rangle p^{f}, [a]\varphi$.

All sequents on the repeat of the leaf discharged by † have a formula in focus, and the principal formula is principal, namely in the rules $\langle a\rangle$ and $\langle * \rangle$. Therefore, the repeat path is successful.

6.2 Split proof system sCPDL_f

One of the core ideas underlying Maehara's proof-theoretic approach towards Craig interpolation is to work with a version of the derivation system that operates on so-called *split sequents*. Here a *split sequent* (Γ, Δ), usually written as $\Gamma \mid \Delta$, is a pair of annotated sequents, of which at most one is focused. Note that we do not require Γ and Δ to be disjoint. Given a split sequent $\Sigma = \Gamma \mid \Delta$, we will write Σ^l for its *left component* Γ and Σ^r for its *right component* Δ. We will use d as a variable ranging over the set $\{l, r\}$. If Γ (respectively Δ) contains a formula in focus, we call Γ (respectively Δ) the *focused component* of $\Gamma \mid \Delta$.

We say that a split sequent $\Gamma \mid \Delta$ is satisfiable (unsatisfiable), if Γ^-, Δ^- is satisfiable (unsatisfiable).

The rules of the split proof system sCPDL_f are obtained from the rules of CPDL_f by applying the rules to one of the components, similarly as we did for the split system sJS_2. Importantly all CPDL_f rules are *analytic* respecting the components. That is, if Ψ^l is the left component of the conclusion of a rule R, then all formulas in the left component of the premise of R are in $\mathsf{Clos}^{\neg}(\Psi^l)$, and analogously for the right component.

6.2.1. Definition. For any CPDL_f rule R in Figure 6.1 we define a *left* sCPDL_f *rule* R^l as follows. If $\mathsf{R} \neq \langle a \rangle$ is of the form

$$\mathsf{R}\colon \frac{\Gamma_1 \quad \cdots \quad \Gamma_n}{\Gamma_0}$$

then R^l is of the form

$$\mathsf{R}^l\colon \frac{\Gamma_1 \mid \Delta \quad \cdots \quad \Gamma_n \mid \Delta}{\Gamma_0 \mid \Delta}$$

The rule $\langle a \rangle^l$ is of the form

$$\langle a \rangle^l : \frac{\varphi^f, \Sigma, \langle \breve{a} \rangle \Lambda \mid \Pi, \langle \breve{a} \rangle \Theta}{\langle a \rangle \varphi^f, [a]\Sigma, \Lambda \mid [a]\Pi, \Theta}$$

where

$$\langle \breve{a} \rangle \Lambda := \{\langle \breve{a} \rangle \chi^u \mid \chi^u \in \Lambda \text{ and } \langle \breve{a} \rangle \chi \in \mathsf{Clos}^{\neg}(\langle a \rangle \varphi, [a]\Sigma, \Lambda)\} \text{ and}$$
$$\langle \breve{a} \rangle \Theta := \{\langle \breve{a} \rangle \chi^u \mid \chi^u \in \Theta \text{ and } \langle \breve{a} \rangle \chi \in \mathsf{Clos}^{\neg}([a]\Pi, \Theta)\}$$

Note that $\langle \breve{a} \rangle \Lambda$ depends on $\langle a \rangle \varphi^f, [a]\Sigma$ as well.

Right sCPDL_f *rules* are defined analogously. Additionally, we also allow so-called *split axioms* of the form

$$\mathsf{Ax1}'\colon \frac{}{\varphi^u \mid \overline{\varphi}^u}$$

Split rules are either left rules, right rules or split axioms.

Note that the only split rules with interactions between the components are split axioms and modal rules. Notions defined in 6.1 translate straightforwardly to sCPDL_f derivations.

6.2.2. Definition. The derivation system sCPDL_f is defined from all split rules. As for CPDL_f, a finite path τ in an sCPDL_f derivation is *successful* if every node on τ features a formula in focus and there is a node on τ where the formula in focus is principal. The cyclic proof system sCPDL_f is path-based and defined by all split rules together with all successful paths.

6.2.3. Example. Consider the proof ρ of the sequent $\langle \breve{a}^* \rangle q, \langle a \rangle \langle a^* \rangle p, [a]\varphi$ in Example 6.1.6, where $\varphi := [a^*][p?][\breve{a}^*]\overline{q}$. We give a sCPDL_f proof ρ' of the split sequent

$$\Gamma \mid \Xi := \langle \breve{a}^* \rangle q, [a]\varphi \mid \langle a \rangle \langle a^* \rangle p.$$

Again we apply weak rules implicitly and omit annotations of u. Because of space issues, we omit the (simple) proof ρ_0 of the split sequent $\langle \breve{a} \rangle \langle \breve{a}^* \rangle q, [p?][\breve{a}^*]\overline{q} \mid p$.

$$\rho' : \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{[\breve{a}][\breve{a}^*]\overline{q}, \langle \breve{a} \rangle \langle \breve{a}^* \rangle q \mid}\,\mathsf{Ax1}^l}{[\breve{a}^*]\overline{q}, \langle \breve{a} \rangle \langle \breve{a}^* \rangle q \mid}\,[*]^l \qquad \lceil \langle \breve{a}^* \rangle q, [a]\varphi \mid \langle a \rangle \langle a^* \rangle p^f \rceil^\dagger}{\langle \breve{a} \rangle \langle \breve{a}^* \rangle q, [a]\varphi \mid \langle a \rangle \langle a^* \rangle p^f}\,\mathsf{acut}^l \qquad \begin{matrix}\rho_0 \\ \langle \breve{a} \rangle \langle \breve{a}^* \rangle q, [p?][\breve{a}^*]\overline{q} \mid p\end{matrix}}{\langle \breve{a} \rangle \langle \breve{a}^* \rangle q, [a]\varphi, [p?][\breve{a}^*]\overline{q} \mid \langle a^* \rangle p^f}\,\langle * \rangle^r}{\langle \breve{a} \rangle \langle \breve{a}^* \rangle q, \varphi \mid \langle a^* \rangle p^f}\,[*]^l}{\langle \breve{a}^* \rangle q, [a]\varphi \mid \langle a \rangle \langle a^* \rangle p^f}\,\langle a \rangle^r}{\langle \breve{a}^* \rangle q, [a]\varphi \mid \langle a \rangle \langle a^* \rangle p^f}\,\mathsf{D}^r_\dagger}{\langle \breve{a}^* \rangle q, [a]\varphi \mid \langle a \rangle \langle a^* \rangle p}\,\mathsf{f}^r$$

Notably, we have $\langle \breve{a} \rangle \langle \breve{a}^* \rangle q \in \mathsf{Clos}^\neg(\varphi)$ and therefore the occurrences of $\langle a \rangle^l$ and cut^l are analytic with respect to the left component.

6.3 Soundness and completeness of split proofs

We prove the soundness and completeness of sCPDL_f by *game-theoretic* means, similarly as for NW_2 in Chapter 5. Given a split sequent Σ, we define a proof-search game $\mathcal{G}(\Phi)$ in which one player (Prover) aims to find a proof of Σ, while the other player (Builder) aims to construct a model where Σ is satisfied. Winning strategies for Prover and Builders then correspond to, respectively, proofs and models for Σ. To tighten the correspondence between winning strategies for Prover and proofs we will work with the infinitary proof system sCPDL_f^∞.

6.3.1 Infinite sCPDL_f^∞-proofs

6.3.1. Definition. An infinite path β in an sCPDL_f derivation is *successful* if

1. on cofinitely many sequents on β there is a formula in focus, and
2. there are infinitely many nodes on β where the formula in focus is principal.

6.3.2. Definition. The infinitary proof system sCPDL_f^∞ is defined from all split rules together with all infinite successful paths.

The correspondence between regular sCPDL_f^∞-proofs and sCPDL_f proofs is as usual.

6.3.3. Lemma. *There is a regular sCPDL_f^∞-proof of Σ iff $\mathsf{sCPDL}_f \vdash \Sigma$.*

Proof:
Let π be a sCPDL_f-proof of Γ and let π^* be its infinite unfolding. Recall that the infinite unfolding π^* of π is the sCPDL_f-derivation obtained from π by recursively unfolding outermost leaves, and removing all discharge rules. We need to show that every infinite path β in π^* is successful. Let γ be the corresponding path of β in π. Then cofinitely nodes of γ are in some proper cluster of π^*. Therefore cofinitely many sequents on γ are focused and on infinitely many nodes on γ the principal formula is in focus. The same holds for the path β implying that it is successful.

Conversely, let ρ be a regular sCPDL_f^∞-proof. For a node v in ρ let ρ_v be the maximal subderivation of ρ rooted at v. For every infinite path $\gamma = (\gamma(i))_{i\in\omega}$ define minimal indices $j < k$ such that

1. $\rho_{\gamma(j)} \sim \rho_{\gamma(k)}$ and
2. the path $\gamma(j) \cdots \gamma(k)$ is successful.

Because ρ is regular and every infinite path in ρ is successful, such indices always exist. For each such infinite path let $\gamma(k)$ be a leaf discharged with companion $\gamma(j)$. Using König's Lemma we can show that this procedure results in a finite sCPDL_f proof π of Σ. $\square$

6.3.2 Proof search game

Let $\dfrac{\Sigma_1 \quad \cdots \quad \Sigma_n}{\Sigma}\ \mathsf{R}$ be a sCPDL_f-rule. We let conc be the function mapping rules to their conclusions. Similarly to Chapter 5, a rule is *cumulative* if all premises are component-wise supersets of the conclusion and *productive* if each premise is distinct from the conclusion. Consequently, a u rule is cumulative and productive, if it is of the form $\dfrac{\varphi^u, \varphi^f, \Gamma}{\varphi^f, \Gamma}\ \mathsf{u}$ with $\varphi^u \notin \Gamma$. In a non-cumulative u rule, all formulas in the premise are out of focus.

We define the *proof search game* $\mathcal{G}(\Phi)$ as in Chapter 5 – but now for sCPDL^∞_f proofs. Its positions are given by $\mathsf{Seq}_\Phi \cup \mathsf{Rules}_\Phi$, where Seq_Φ is the set of split sequents and Rules_Φ the set of sCPDL^∞_f rules containing formulas in Φ. The ownership function and admissible moves are given in the table below. As always, finite matches are lost by the player who gets stuck. An infinite match is won by Prover iff the resulting infinite path is successful.

Position	Owner	Admissible moves
Σ	Prover	$\{R \in \mathsf{Rules}_\Phi \mid \mathsf{conc}(R) = \Sigma\}$
$\dfrac{\Sigma_1 \quad \cdots \quad \Sigma_n}{\Sigma}\ \mathsf{R}$	Builder	$\{\Sigma_i \mid i = 1, \cdots, n\}$

Table 6.1: The proof search game $\mathcal{G}(\Phi)$

Interestingly, we may assume that winning strategies in $\mathcal{G}(\Phi)$ are positional, that is, only depend on the current position of the game, and not on the history of the play leading up to this position. The key observations here is that $\mathcal{G}(\Phi)$ can be formulated as a parity game. As shown in Theorem 2.2.8, parity games enjoy positional determinacy. To see that $\mathcal{G}(\Phi)$ is a parity game, we may assign the following priorities to its positions: $\Omega(\Sigma) := 1$ for any sequent position, and for the other positions we put

$$\Omega\left(\frac{\Sigma_1 \quad \cdots \quad \Sigma_n}{\Sigma}\ \mathsf{R}\right) := \begin{cases} 3 & \text{if } \Sigma \text{ has no formula in focus} \\ 2 & \text{if in } \Sigma \text{ the principal formula of } \mathsf{R} \text{ is in focus} \\ 1 & \text{otherwise.} \end{cases}$$

An sCPDL^∞_f-proof of a split sequent Σ may be identified with the strategy tree of a winning strategy for Prover in $\mathcal{G}(\Phi)@\Sigma$. Thus, as a consequence of positional determinacy, we may assume that CPDL^∞_f-proofs are regular.

6.3.3 Soundness

We show soundness of sCPDL^∞_f along the same lines as soundness for NW_2 in Chapter 5. Given a satisfiable split sequent Σ we define a winning strategy for Builder in $\mathcal{G}(\Phi)@\Sigma$. The following lemma deals with the local soundness of our rules. For simplicity, we present it for CPDL_f rules, the statement transfers straightforwardly to split rules in sCPDL_f.

Recall that we work with the game semantics of CPDL. We say that a pointed model $\mathbb{S}, s$ satisfies a formula φ if there is a winning strategy g for $\exists$ in the game $\mathcal{E}_{\mathsf{PDL}}@(\varphi, s)$; this we denote as $\mathbb{S}, s \Vdash_g \varphi$. Because the game $\mathcal{E}_{\mathsf{PDL}}$ is a parity game, we may assume that both players play positional strategies. A sequent Γ is satisfiable if $\mathbb{S}, s \Vdash_g \bigwedge \Gamma^-$ for some pointed model $\mathbb{S}, s$ and strategy g for $\exists$.

6.3.4. Lemma. *Let*

$$\frac{\Delta_1 \quad \cdots \quad \Delta_n}{\Gamma} \mathsf{R}$$

be a CPDL_f *rule. If* Γ *is satisfiable, then* Δ_i *is satisfiable for some* $i = 1, ..., n$.

More concretely, let $\mathbb{S}, s$ *be a pointed model and* g *a positional strategy for* $\exists$ *in* $\mathcal{E}_{\mathsf{PDL}}(\mathbb{S})$ *such that* $\mathbb{S}, s \Vdash_g \Gamma$. *Then,*

1. *if* $\mathsf{R} \neq \langle a \rangle$ *then* $\mathbb{S}, s \Vdash_g \Delta_i$ *for some* $i = 1, ..., n$,
2. *if* $\mathsf{R} = \vee$ *with principal formula* $\varphi_0 \vee \varphi_1^u$, *then* $\mathbb{S}, s \Vdash_g \varphi_j^u, \Gamma$, *where* $g(\varphi_0 \vee \varphi_1, s) = (\varphi_j, s)$,
3. *if* $\mathsf{R} = \langle \cup \rangle$ *with principal formula* $\langle \alpha_0 \cup \alpha_1 \rangle \varphi$, *then* $\mathbb{S}, s \Vdash_g \langle \alpha_j \rangle \varphi, \Gamma$, *where* $g(\langle \alpha_0 \cup \alpha_1 \rangle \varphi, s) = (\langle \alpha_j \rangle \varphi, s)$,
4. *if* $\mathsf{R} = \langle * \rangle$ *with principal formula* $\langle \alpha^* \rangle \varphi^b$, *then* $\mathbb{S}, s \Vdash_g \langle \alpha \rangle \langle \alpha^* \rangle \varphi^b, \Gamma$ *if* $g(\langle \alpha^* \rangle \varphi, s) = (\langle \alpha \rangle \langle \alpha^* \rangle \varphi, s)$, *and* $\mathbb{S}, s \Vdash_g \varphi^b, \Gamma$ *else,*
5. *if* $\mathsf{R} = [?]$ *with principal formula* $[\psi?]\varphi^u$, *then* $\mathbb{S}, s \Vdash_g \overline{\psi}^u, \Gamma$ *if* $g([\psi?]\varphi, s) = (\overline{\psi}, s)$, *and* $\mathbb{S}, s \Vdash_g \varphi^u, \Gamma$ *else,*
6. *if* $\mathsf{R} = \langle a \rangle$ *with principal formula* $\langle a \rangle \varphi^f$ *and* $\Gamma = \langle a \rangle \varphi^f, [a]\Sigma, \Pi$, *then* $\mathbb{S}, t \Vdash_g \varphi^f, \Sigma, \langle \breve{a} \rangle \Pi$, *where* $g(\langle a \rangle \varphi, s) = (\varphi, t)$.

Proof:
The soundness of the modal rule is a simpler version of the soundness of the modal rule in NW_2 (Lemma 5.1.5). The soundness of all other rules is straightforward and will be omitted. □

Given a CPDL_f^∞-proof of a split sequent $\Sigma = \Gamma \mid \Delta$ we want to show that Γ, Δ is unsatisfiable. By contraposition, given a pointed model $\mathbb{S}, s$ that satisfies Γ, Δ we provide a winning strategy for Builder in $\mathcal{G}(\Phi)@(\Sigma)$.

6.3.5. Theorem (Soundness). *If* CPDL_f^∞ *proves a split sequent* Σ, *then* Σ *is unsatisfiable.*

Proof:
By contraposition we show that, if Σ is satisfiable, then Builder has a winning strategy in $\mathcal{G} := \mathcal{G}(\Phi)@\Sigma$. So assume that there is a pointed model $\mathbb{S}, s$ and a positional strategy g for $\exists$ in the game $\mathcal{E} := \mathcal{E}_{\mathsf{PDL}}(\mathbb{S})$ such that $\mathbb{S}, s \Vdash_g \Sigma$. We will construct a winning strategy $\overline{g}$ for Builder in $\mathcal{G}$ and a function $s_g : PM(\mathcal{G}) \to \mathbb{S}$, mapping partial $\mathcal{G}$-matches to states of $\mathbb{S}$, such that $\mathbb{S}, s_g(\mathcal{M}) \Vdash_g \mathsf{last}(\mathcal{M})$ for every $\overline{g}$-guided partial match $\mathcal{M} \in PM_{Prover}(\mathcal{G})$.

The functions $\overline{g}$ and s_g can be defined inductively. For the base case $|\mathcal{M}| = 1$ the partial match $\mathcal{M}$ consists of the single position Γ. We define $s_g(\mathcal{M}) := s$ and do not have to define $\overline{g}$ as this is a position owned by Prover.

For the induction step we follow the specifications of the rule. If the rule is $\langle a \rangle$, define s_g as given by g and let $\overline{g}$ choose the only premise. For any other rule s_g remains the same and we invoke Lemma 6.3.4 to choose a premise for Builder.

We need to show that $\overline{g}$ is a winning strategy for Builder in $\mathcal{G}$. Because of Lemma 6.3.4 we know that all finite matches are won by Builder. Thus, assume by contradiction that Builder loses an infinite $\overline{g}$-guided $\mathcal{G}$-match $\mathcal{M}$. We write $\mathcal{M} = \Sigma_1 R_1 \Sigma_2 R_2 ...$ and let $\mathcal{M}_n = \Sigma_1 R_1 ... R_{n-1} \Sigma_n$ be the partial match up to position Σ_n. Then cofinitely positions in $\mathcal{M}$ of the form Σ_n contain a formula in focus, where infinitely often Prover picks a $\langle * \rangle$ rule with the principal formula in focus. We will use $\mathcal{M}$ to obtain an infinite g-guided $\mathcal{E}$-match that is won by $\forall$. Let $N \in \omega$ be such that Σ_n has a formula in focus for all $n \geq N$ and let $\psi_n^f \in \Sigma_n$ be this formula in focus.

We claim that there is an g-guided $\mathcal{E}@(\psi_N, s_g(\mathcal{M}_N))$-match $\mathcal{P} = P_1 P_2 ...$ that is won by $\forall$ and such that $P_j = (\psi_n, s_g(\mathcal{M}_n))$ for some $n \in \omega$. We define $P_1 = (\psi_N, s_g(\mathcal{M}_N))$. Given $P_j = (\psi_n, s_g(\mathcal{M}_n))$ let $k \geq n$ be minimal such that Prover picks a rule with principal formula ψ_n^f. This always exists as infinitely often Prover picks a $\langle * \rangle$ rule where the principal formula is in focus. Then define $P_{j+1} = (\psi_{k+1}, s_g(\mathcal{M}_{k+1}))$. The match $\mathcal{P}$ is well-defined and g-guided by the definition of s_g and $\overline{g}$. Moreover, ψ_n is a diamond fixpoint formula infinitely often and thus $\mathcal{P}$ is won by $\forall$. This implies $\mathbb{S}, s_g(\mathcal{M}_N) \not\Vdash_g \psi_N$ contradicting $\mathbb{S}, s_g(\mathcal{M}_N) \Vdash_g \Sigma_N$. □

6.3.4 Completeness

For any unsatisfiable split sequent Σ we have to find an sCPDL_f^∞-proof of Σ, in other words, a winning strategy for Prover in $\mathcal{G}(\Phi)@\Sigma$. We will show an even stronger statement: By restricting the strategy for Prover we show that for every unsatisfiable split sequent we obtain an sCPDL_f^∞-proof in a certain normal form. These *uniform* proofs will be instrumental in our interpolation proof.

6.3.6. Definition (Uniform Split Derivation). A set of formulas Γ is called *saturated*, if no axiom or cumulative and productive CPDL_f rule may be applied to Γ^u. This definition is equivalent to Γ^u being a saturated set in the usual sense. A split derivation π is *uniform* if it satisfies the following conditions:

U0. If possible an axiom is applied.

U1. Else if possible a cumulative and productive rule is applied to a formula in an unfocused component.

U2. Let t_1 and t_2 be nodes in π, labeled with split sequents $\Gamma_1 \mid \Delta$ and $\Gamma_2 \mid \Delta$, respectively, and with rules different from D. Assume that their common component Δ is focused, while their unfocused components Γ_1 and Γ_2 are both saturated. Then at both t_1 and t_2 the same rule with principal formula in Δ is applied. If possible, this rule is cumulative and productive with an out of focus principal formula, and else it is productive with its principal formula in focus.

U3. The analogous condition to (U2) for split sequents $\Gamma \mid \Delta_1$ and $\Gamma \mid \Delta_2$.

We thus aim to show the following completeness theorem.

6.3.7. Theorem (Completeness). *If a split sequent Σ is unsatisfiable, then there is a uniform, regular* $\mathsf{sCPDL}^{\infty}_{f}$ *proof of* Σ.

We start with presenting a proof sketch; a full proof will be presented afterwards. Completeness will be shown along the same lines as for the system NW_2 for $\mathcal{L}^2_{\mu}$. Because we show it directly for the split system, which is also annotated, there are a few additional complications that we need to address. We show the completeness of sCPDL_f by contraposition; given a winning strategy for Builder in $\mathcal{G}(\Phi)@\Sigma$, we find a pointed model $\mathbb{S}, s$ satisfying Σ. Let g be a *positional* winning strategy for Builder in $\mathcal{G}(\Phi)@\Sigma$, we construct a pointed model $\mathbb{S}^g, s$ and a strategy $\underline{g}$ for $\exists$ in $\mathcal{E}_{\mathsf{PDL}}(\mathbb{S}^g)$ such that $\mathbb{S}^g, s \Vdash_{\underline{g}} \Sigma$.

Let $\mathcal{T}$ be the maximal subgraph of the game tree of $\mathcal{G}(\Phi)@\Sigma$, where Builder plays the strategy g and Prover picks rules such that the uniformity conditions are satisfied. We want to define a model $\mathbb{S}^g$ from $\mathcal{T}$. We call a maximal path ρ in $\mathcal{T}$ not containing the rules $\langle a \rangle$, f and non-cumulative rules u a *local path.* It will turn out that every local path is finite.

The model $\mathbb{S}^g$ will consist of all local paths. The accesability relation R_a for local paths ρ, τ is defined as follows: $\rho R_a \tau$ if either

(i) τ is above ρ in $\mathcal{T}$ only separated by a $\langle a \rangle$ rule and (possibly) f rules and non-cumulative u rules or

(ii) ρ is above τ only separated by a $\langle \breve{a} \rangle$ rule and (possibly) f rules and non-cumulative u rules.

The sequent $\mathsf{S}(\rho)$ at a local path ρ is defined as $\bigcup\{\Sigma^l \cup \Sigma^r \mid \Sigma \text{ occurs in } \rho\}$.

For the definition of the strategy $\underline{g}$ for $\exists$ in $\mathcal{E}_{\mathsf{PDL}}(\mathbb{S}^g)$ we use the fact that $\mathsf{S}(\rho)$ is a saturated set for every local path ρ. For instance, at position $(\varphi \vee \psi, \rho)$ the formula φ or ψ is in $\mathsf{S}(\rho)^-$ and $\underline{g}$ picks one that is in $\mathsf{S}(\rho)^-$. At position $(\langle a \rangle\varphi, \rho)$ the strategy $\underline{g}$ picks some τ such that τ is above ρ in $\mathcal{T}$.

Now let $\psi_0 \in \Sigma$ and let ρ_0 be a local path containing Σ. Let $\mathcal{M}$ be an $\underline{g}$-guided $\mathcal{E}_{\mathsf{PDL}}(\mathbb{S}^g)$-match starting at (ψ_0, ρ_0). Then we can prove that for every position

(ψ, ρ) in $\mathcal{M}$ it holds that $\psi \in \mathsf{S}(\rho)$ and consequently that $\exists$ wins $\mathcal{M}$. This shows that $\mathbb{S}^g, \rho_0 \Vdash_{\underline{g}} \Sigma$.

We will now make this proof sketch formal. Let g be a *positional* winning strategy for Builder in $\mathcal{G}(\Phi)@\Sigma$. We construct a pointed model $\mathbb{S}^g, s$ and a strategy $\underline{g}$ for $\exists$ in $\mathcal{E}_{\mathsf{PDL}}(\mathbb{S}^g)$ such that $\mathbb{S}^g, s \Vdash_{\underline{g}} \Sigma$. We start by defining the pointed model $\mathbb{S}^g, s$.

Let $\mathcal{T}$ be the maximal subgraph of the game tree of $\mathcal{G}(\Phi)@\Sigma$, where Builder plays the strategy g and Prover picks rules according to the following priorities:

1. axioms $\mathsf{Ax1}$ or $\mathsf{Ax2}$ preceded by weak;
2. cumulative and productive rules $\wedge$, $\vee$, $\langle ; \rangle$, $[;]$, $\langle \cup \rangle$, $[\cup]$, $\langle ? \rangle$, $[?]$, $\langle * \rangle$, $[*]$ or acut, where the principal formula is in an unfocused component;
3. cumulative and productive rules u;
4. cumulative and productive rules $\wedge$, $\vee$, $\langle ; \rangle$, $[;]$, $\langle \cup \rangle$, $[\cup]$, $\langle ? \rangle$, $[?]$, $\langle * \rangle$, $[*]$ or acut, where the principal formula is out of focus but in the focused component;
5. productive rules $\langle ; \rangle$, $\langle \cup \rangle$, $\langle ? \rangle$ or $\langle * \rangle$, where the principal formula is in focus;
6. rules $\langle a \rangle$, cumulative rules f or non-cumulative rules u.

Additionally, at any two positions Σ_0 and Σ_1, where the focused components of Σ_0 and Σ_1 coincide and no rule of type $1-3$ is applicable, Prover picks the same rule of type 4, if possible.

Any winning strategy for Prover, where she picks rules according to those requirements, results in a uniform sCPDL^∞_f-proof.

6.3.8. Definition. We call a maximal path in $\mathcal{T}$ of rules of type $1-5$ a *local path*. Let ρ, τ be local paths in $\mathcal{T}$. We define $\rho \overset{a}{\to} \tau$ if τ is above ρ in $\mathcal{T}$, only separated by a $\langle a \rangle$ rule and possible f rules and non-cumulative u rules. We let $\mathsf{S}^d(\rho) := \bigcup \{\Sigma^d \mid \Sigma \text{ occurs in } \rho\}$ for $d = l, r$ and $\mathsf{S}(\rho) := \mathsf{S}^l(\rho) \cup \mathsf{S}^r(\rho)$.

Let ρ be a local path in $\mathcal{T}$. Note that $\mathsf{S}^d(\rho)$ is not necessarily an annotated sequent as it may contain multiple formulas in focus. Because of the restriction on Prover's strategy $\mathsf{S}^d(\rho)$ is a *saturated set*, meaning that no rule of type $1-4$ is applicable to $\mathsf{S}^d(\rho)^u$. Note that this definition conforms with the usual notion of a saturated set. In particular, for every formula $\varphi \in \mathsf{Clos}^\neg(\mathsf{S}^d(\rho)^-)$ it holds that either $\varphi \in \mathsf{S}^d(\rho)^-$ or $\overline{\varphi} \in \mathsf{S}^d(\rho)^-$ and not both.

6.3.9. Lemma. *All local paths ρ in $\mathcal{T}$ are finite and $\mathsf{S}^d(\rho)^- = \mathsf{S}^d(\mathsf{last}(\rho))^-$.*

Proof:
As there are only finitely many formulas in Φ, and rules of type 2 – 4 are cumulative and productive, all paths consisting of rules of type 1 – 4 are finite. In rules of type 5 the principal formula is in focus. Therefore, if a local path τ is infinite it contains infinitely many rules of type 5 and no application of f. This implies that τ is successful, which is a contradiction, as we assumed that Builder plays a winning strategy.

All rules $\dfrac{\Sigma_1 \quad \cdots \quad \Sigma_n}{\Sigma}\,\mathsf{R}$ of type 2 – 5 are cumulative regarding the unannotated sequent, meaning that Σ^- is a component-wise subset of Σ_i^- for all $i = 1, .., n$. For all cumulative rules this is clear. The only non-cumulative ones are of type 5: rules where the principal formula is in focus. Yet, we may assume that, if a d-rule with principal formula φ^f is chosen, then φ^u is in the sequent Σ^d as well. Otherwise, a cumulative and productive u rule would be applicable. Hence, even though φ^f may not be in a premise of the rule, φ^u is. Therefore, we inductively obtain $\mathsf{S}^d(\rho)^- = \mathsf{S}^d(\mathsf{last}(\rho))^-$. □

We can now define the model $\mathbb{S}^g = (S^g, R^g, V^g)$. We let S^g be the set of local paths in $\mathcal{T}$. We define $R^g = \{R^g_a\}_{a \in \mathsf{Act}}$ as follows:

$$\rho R_a \tau \quad :\Leftrightarrow \quad \rho \overset{a}{\to} \tau \text{ or } \tau \overset{\breve{a}}{\to} \rho$$

The valuation is defined as $V^g(p) := \{\rho \in S^g \mid p \in \mathsf{S}(\rho)^-\}$.

Next we define a strategy $\underline{g}$ for $\exists$ in $\mathcal{E} := \mathcal{E}_{\mathsf{PDL}}(\mathbb{S}^g)$. This is done as follows:

1. At $(\varphi \vee \psi, \rho)$ pick φ if $\varphi \in \mathsf{S}(\rho)^-$ and ψ else.

2. At $(\langle a \rangle \varphi, \rho)$ choose some τ such that $\rho \overset{a}{\to} \tau$ by virtue of a $\langle a \rangle$ rule with principal formula $\langle a \rangle \varphi^f$ and as few applications of f as possible.

3. At $(\langle \alpha \cup \beta \rangle \varphi, \rho)$ we make a case distinction:

   (a) If $\langle \alpha \rangle \varphi$ is not in $\mathsf{S}(\rho)^-$, pick $\langle \beta \rangle \varphi$,

   (b) If $\langle \beta \rangle \varphi$ is not in $\mathsf{S}(\rho)^-$, pick $\langle \alpha \rangle \varphi$,

   (c) Otherwise both $\langle \alpha \rangle \varphi$ and $\langle \beta \rangle \varphi$ are in $\mathsf{S}(\rho)^-$. If $\langle \alpha \cup \beta \rangle \varphi \in \mathsf{S}^l(\rho)^-$ and at the rule

   $$\frac{\langle \alpha \rangle \varphi^f, \mathsf{S}^l(\rho)^u \mid \mathsf{S}^r(\rho)^u \qquad \langle \beta \rangle \varphi^f, \mathsf{S}^l(\rho)^u \mid \mathsf{S}^r(\rho)^u}{\langle \alpha \cup \beta \rangle \varphi^f, \mathsf{S}^l(\rho)^u \mid \mathsf{S}^r(\rho)}\,\langle \cup \rangle$$

   Builder chooses the left premise, then pick $\langle \alpha \rangle \varphi$ and if he chooses the right premise, pick $\langle \beta \rangle \varphi$.

   Else if at the rule

   $$\frac{\mathsf{S}^l(\rho)^u \mid \langle \alpha \rangle \varphi^f, \mathsf{S}^r(\rho)^u \qquad \mathsf{S}^l(\rho)^u \mid \langle \beta \rangle \varphi^f, \mathsf{S}^r(\rho)^u}{\mathsf{S}^l(\rho)^u \mid \langle \alpha \cup \beta \rangle \varphi^f, \mathsf{S}^r(\rho)^u}\,\langle \cup \rangle$$

Builder chooses the left premise, then pick $\langle\alpha\rangle\varphi$ and if he chooses the right premise, pick $\langle\beta\rangle\varphi$.

4. Analogously for $(\langle\alpha^*\rangle\varphi, \rho)$.

5. At $([\psi?]\varphi, \rho)$ pick φ if it is in $\mathsf{S}(\rho)^-$ and $\overline{\psi}$ else.

6.3.10. Remark. To give an intuitive explanation of the somewhat exotic definition of $\underline{g}$ we have to think about our proof strategy. The aim of this definition is to ensure that in an $\underline{g}$-guided $\mathcal{E}$-match $\mathcal{M} = (\psi_n, \rho_n)_{n<\kappa}$ it holds that $\psi_n \in \mathsf{S}(\rho_n)^-$. This already guarantees that all such finite matches are won by $\exists$.

For infinite matches we also have to take the annotations into account. In order to show that all infinite matches are won by $\exists$ we argue by contraposition. Given an $\underline{g}$-guided infinite $\mathcal{E}$-match $\mathcal{M} = (\psi_n, \rho_n)_{n\in\omega}$ that is won by $\forall$ we have to find an infinite successful path in $\mathcal{T}$. Because $\mathcal{M}$ is won by $\forall$, there is $N \in \omega$ such that ψ_n is a diamond formula for all $n \geq N$. We aim to find a path in $\mathcal{T}$ where ψ_n is in focus in $\mathsf{S}(\rho_n)$ for all $n \geq N$. We have to be very careful, ψ_n might be in focus in the left or the right component of $\mathsf{S}(\rho_n)$ and Builder's strategy g might differ in the cases where ψ_n^f is principal in the left or the right component. This explains our complicated definition of $\underline{g}$ in case 3(c), in which we give priority to the left component. If $\psi_M \in \mathsf{S}^l(\rho_M)$ for some $M \geq N$ (guaranteeing that $\psi_n \in \mathsf{S}^l(\rho_n)$ for all $n \geq M$) we may then find a path in $\mathcal{T}$ where ψ_n is in focus in the left component for all $n \geq N$. If on the other hand $\psi_n \notin \mathsf{S}^l(\rho_n)$ for all $n \geq N$ (guaranteeing that $\psi_n \in \mathsf{S}^r(\rho_n)$ for all $n \geq N$) we find a path in $\mathcal{T}$ where ψ_n is in focus in the right component for all $n \geq N$. Note that this definition is only possible because we assume that the strategy g for Builder is positional.

In order to show that the strategy $\underline{g}$ is well-defined we have to guarantee that at any position $(\langle a\rangle\varphi, \rho)$ in a match it holds that $\langle a\rangle\varphi \in \mathsf{S}(\rho)^-$. Then, at $\mathsf{last}(\rho)$ in $\mathcal{T}$ Prover might put $\langle a\rangle\varphi$ in focus and apply an $\langle a\rangle$ rule. For matches starting at the root of $\mathcal{T}$, the next lemma guarantees that for any positions $(\langle a\rangle\varphi, \rho)$ indeed we have $\langle a\rangle\varphi \in \mathsf{S}(\rho)^-$.

6.3.11. Lemma (Truth Lemma). *Let $\psi_0 \in \Sigma^d$ and let ρ_0 be a local path containing Σ. Let $\mathcal{M} = (\psi_n, \rho_n)_{n<\kappa}$ be an $\underline{g}$-guided $\mathcal{E}$-match starting at (ψ_0, ρ_0). Then for every $n < \kappa$ it holds that $\psi_n \in \mathsf{S}^d(\rho_n)^-$.*

Proof:
We prove the claim by strong induction on n. The base case holds by assumption. For the induction step let $\psi_n \in \mathsf{S}^d(\rho_n)^-$. We need to show that $\psi_{n+1} \in \mathsf{S}^d(\rho_{n+1})^-$. We proceed by a case distinction on the shape of ψ_n. If ψ_n is not of the form $[a]\chi$ or $\langle a\rangle\chi$ for some action a, then $\rho_{n+1} = \rho_n$ and the claim easily follows from the fact that $\mathsf{S}^d(\rho_n)$ is a saturated set and the definition of $\underline{g}$.

Next assume that $\psi_n = [a]\chi$, then $\psi_{n+1} = \chi$. In this case either $\rho_n \overset{a}{\to} \rho_{n+1}$ or $\rho_{n+1} \overset{\breve{a}}{\to} \rho_n$. First assume that $\rho_n \overset{a}{\to} \rho_{n+1}$. Then $[a]\chi^u$ is in the conclusion of the $\langle a \rangle$ rule between ρ_n and ρ_{n+1}, hence χ^u is in its premise and therefore $\chi \in \mathsf{S}^d(\rho_{n+1})^-$.

Now consider the case $\rho_{n+1} \overset{\breve{a}}{\to} \rho_n$. We first show that $[a]\chi \in \mathsf{Clos}^\neg(\mathsf{S}^d(\rho_{n+1})^-)$. As $\mathbb{S}^g$ is a forest and $\rho_0 ... \rho_{n+1}$ forms a path in $\mathbb{S}^g$ starting at one of the roots, where the last step of the path is downwards, there has to be an $i \in \{0, ..., n-1\}$ with $\rho_i = \rho_{n+1}$. In the match $\mathcal{M}$ there are positions (ψ_i, ρ_i) and $([a]\chi, \rho_n)$, hence $[a]\chi \in \mathsf{Clos}(\psi_i)$. By induction hypothesis $\psi_i \in \mathsf{S}^d(\rho_i)^- = \mathsf{S}^d(\rho_{n+1})^-$ and thus $[a]\chi \in \mathsf{Clos}(\mathsf{S}^d(\rho_{n+1})^-)$.

Towards a contradiction assume that $\chi \notin \mathsf{S}^d(\rho_{n+1})^-$. Then, because $\mathsf{S}^d(\rho_{n+1})$ is a saturated set and $\chi \in \mathsf{Clos}^\neg(\mathsf{S}^d(\rho_{n+1})^-)$ it holds that $\overline{\chi} \in \mathsf{S}^d(\rho_{n+1})^-$. Let R be the $\langle \breve{a} \rangle$ rule between ρ_{n+1} and ρ_n. The formula $\overline{\chi}^u$ is in the conclusion of R, therefore $\langle \breve{\breve{a}} \rangle \overline{\chi}^u = \langle a \rangle \overline{\chi}^u$ is in its premise as $\langle a \rangle \overline{\chi} \in \mathsf{Clos}^\neg(\mathsf{S}^d(\rho_{n+1})^-)$. This implies $\langle a \rangle \overline{\chi} \in \mathsf{S}^d(\rho_n)^-$, contradicting the fact that $\mathsf{S}^d(\rho_n)$ is a saturated set and $[a]\chi \in \mathsf{S}^d(\rho_n)^-$.

Lastly, assume that $\psi_n = \langle a \rangle \chi$ and $\psi_n \in \mathsf{S}^d(\rho_n)^-$. Then $\psi_n^b \in \mathsf{last}(\rho_n)^d$ and by the definition of the strategy $\underline{g}$ it holds that $\rho_n \overset{a}{\to} \rho_{n+1}$. If $\psi_n^f \in \mathsf{S}^d(\rho_n)$, then a $\langle a \rangle^d$ rule with principal formula ρ_n^f is applied in $\mathcal{T}$ and therefore $\psi_{n+1}^f \in \mathsf{S}^d(\rho_{n+1})$. Otherwise, Prover may first put ψ_n in focus and then apply the rule $\langle a \rangle^d$, yielding the same sequent. □

6.3.12. Lemma. *Let Ψ be a split sequent. Let $\psi_0^f \in \Psi^d$ and let ρ_0 be a local path containing Ψ. Let $\mathcal{M} = (\psi_n, \rho_n)_{n \in \omega}$ be an $\underline{g}$-guided $\mathcal{E}$-match starting at (ψ_0, ρ_0), such that for all $n \in \omega$ it holds that ψ_n is a diamond formula. Moreover, if $\psi_n = \langle \varphi ? \rangle \chi$, then $\psi_{n+1} = \chi$.*

If either $d = l$, or $d = r$ and $\psi_n \notin \mathsf{S}^l(\rho_n)^-$ for all $n \in \omega$, then $\psi_n^f \in \mathsf{S}^d(\rho_n)$ for all $n \in \omega$. If additionally ψ_n is of the form $\psi_n = \langle a \rangle \chi$, then $\psi_n^f \in \mathsf{last}(\rho_n)^d$.

Proof:
We prove the claim by induction on n. The base case holds by assumption. For the induction step let $\psi_n^f \in \mathsf{S}^d(\rho_n)$. We need to show that $\psi_{n+1}^f \in \mathsf{S}^d(\rho_{n+1})$. We proceed by a case distinction on the shape of ψ_n. If ψ_n is not of the form $\langle a \rangle \chi$ for some action a, then $\rho_{n+1} = \rho_n$.

The cases $\psi_n = \langle \alpha \cup \beta \rangle \chi$ and $\psi_n = \langle \alpha^* \rangle \chi$ follow by the definition of the strategy $\underline{g}$. In the case $d = r$ we need the extra condition that $\psi_n \notin \mathsf{S}^l(\rho_n)^-$ for all $n \in \omega$ to guarantee that the choice of $\exists$ coincides with the choice of Builder in the proof search game. The case where $\psi_n = \langle \alpha ; \beta \rangle \chi$ is clear and the case $\psi_n = \langle \varphi ? \rangle \chi$ follows by assumption.

Lastly, assume that $\psi_n = \langle a \rangle \chi$ and $\psi_n^f \in \mathsf{S}^d(\rho_n)$. Then $\psi_n^f \in \mathsf{last}(\rho_n)^d$ and by the definition of the strategy $\underline{g}$ it holds that $\rho_n \overset{a}{\to} \rho_{n+1}$. Hence, Prover applies a

$\langle a\rangle^d$ rule in $\mathcal{T}$ with principal formula ψ_n^f and therefore $\psi_{n+1}^f \in \mathsf{S}^d(\rho_{n+1})$. □

6.3.13. Lemma. *Let $\psi_0 \in \Sigma$ and let ρ_0 be a local path containing Σ. Then the strategy $\underline{g}$ is winning for $\exists$ in $\mathcal{E}@(\psi_0, \rho_0)$.*

Proof:
Let $\mathcal{M}$ be an $\underline{g}$-guided $\mathcal{E}@(\psi_0, \rho_0)$-match. If $\mathcal{M}$ is finite it is straightforward to check that it is winning for $\exists$. Thus we consider the case where $\mathcal{M} = (\psi_n, \rho_n)_{n\in\omega}$ is infinite and assume that it is winning for $\forall$. Then there is $N \in \omega$ such that ψ_n is a diamond formula for all $n \geq N$. Without loss of generality we may assume that N is big enough such that for all formulas ψ_n of the form $\langle\varphi?\rangle\chi$ (with $n \geq N$) it holds that $\psi_{n+1} = \chi$.

Let $M \geq N$ be such that $\psi_M \in \mathsf{S}^l(\rho_M)^-$, or if such an M does not exist (meaning that $\psi_n \notin \mathsf{S}^l(\rho_n)$ for all $n \geq N$), let $M := N$. In the first case let $d := l$ and in the second let $d := r$. In both cases it holds that $\psi_n \in \mathsf{S}(\rho_n)^-$ for all $n \in \omega$ by Lemma 6.3.11.

We first assume that for infinitely many n the formula ψ_n is of the form $\langle a_n\rangle\chi$ for some (atomic) program a_n. Let $K \geq M$ such that ψ_K is of the form $\langle a\rangle\chi$, then $\psi_{K+1} = \chi$ and $\chi^f \in \mathsf{S}^d(\rho_{K+1})$. By Lemma 6.3.12 we have $\psi_n^f \in \mathsf{S}^d(\rho_n)$ for all $n > K$. Additionally, for every $n > K$ there is some $m \geq n$ such that $\psi_m = \langle a_m\rangle\chi$ and $\psi_m^f \in \mathsf{last}(\rho_n)^d$. But if the last sequent on ρ_n has a formula in focus, then all sequents in ρ_n have a formula in focus for all $n > K$, as no f rule is applied on local paths. Between the local paths ρ_n and ρ_{n+1} with $\rho_n \neq \rho_{n+1}$ no f rule is applied either, as by the definition of $\underline{g}$ applications of f are minimized. Thus there is an infinite path in $\mathcal{T}$ where cofinitely many sequents have a formula in focus and infinitely many rules with principal formula in focus are applied. This contradicts the assumption that Builder plays a winning strategy.

Now assume that for some $K \geq M$ there is no $n \geq K$ such that the formula ψ_n is of the form $\langle a\rangle\chi$. As a consequence $\rho_n = \rho_N$ for all $n \geq K$.

Let $\mathsf{last}(\rho_K) = \Pi \mid \Xi$, we show how to obtain an infinite successful path in $\mathcal{T}$ starting at $\Pi \mid \Xi$ contradicting the assumption that Builder plays a winning strategy. Assume that $d = l$, the case where $d = r$ is analogous. At $\Pi \mid \Xi$ no rule of type 1 – 5 is applicable, thus Prover may put ψ_K in focus in $\mathcal{T}$ to obtain a node $\psi_K^f, \Pi^u \mid \Xi^u$. We will show that there is an infinite path $\Sigma_0 R_0 \Sigma_1 R_1 ...$ in $\mathcal{T}$, where for all $n \in \omega$ it holds $\Sigma_n = \psi_{K+n}^f, \Pi^u \mid \Xi^u$ and R_n is a rule of type 5.

Inductively assume that the position $\psi_{K+n}^f, \Pi^u \mid \Xi^u$ is in $\mathcal{T}$. Following her strategy in $\mathcal{T}$, Prover may pick a rule of type 5 with principal formula ψ_{K+n}^f. We proceed by a case distinction on the shape of ψ_{K+n}.

If $\psi_{K+n} = \langle\alpha \cup \beta\rangle\chi$, then $\langle\alpha \cup \beta\rangle\chi \in \mathsf{S}^l(\rho_N)^-$ and thus $\exists$ picks the correct formula according to Builder's strategy $\underline{g}$. Analogously for $\psi_{K+n} = \langle\alpha^*\rangle\chi$. The case where $\psi_{K+n} = \langle\alpha;\beta\rangle\chi$ is clear. If $\psi_{K+n} = \langle\varphi?\rangle\chi$, then by the definition of

N, and the fact that $K \geq N$, it follows $\psi_{K+n+1} = \chi$. The rule R_n only has one premise $\varphi^u, \chi^f, \Pi^u \mid \Xi^u$. Since Π is saturated, either φ or $\overline{\varphi}$ is in Π. If $\varphi \in \Pi$ we have shown the induction step. If on the other hand $\overline{\varphi} \in \Pi$, then an axiom would be applicable, contradicting the fact that Builder's strategy is winning.

Thus there is an infinite path in $\mathcal{T}$ where cofinitely many sequents have a formula in focus and infinitely often a rule with principal formula in focus is applied, contradicting that Builder's strategy g is winning. □

Proof of Theorem 6.3.7:
Assume that g is a winning strategy for Builder in $\mathcal{G}(\Phi)@(\Sigma)$. Let ρ_0 be a local path in $\mathcal{T}$ containing Σ. Then Lemma 6.3.13 shows that $\mathbb{S}^g, \rho_0 \Vdash_{\underline{g}} \Sigma$, which implies that Σ is satisfiable. By contraposition this means that for every unsatisfiable sequent Σ Prover has a winning strategy in $\mathcal{G}(\Phi)@(\Sigma)$ and thus there is a $\mathsf{sCPDL}^{\infty}_f$-proof π of Σ, which may be assumed to be regular. By our restriction on the strategy for Prover, π is actually a uniform proof.

□

6.3.14. THEOREM. *A split sequent Σ is unsatisfiable iff there is a uniform sCPDL_f-proof of Σ.*

Proof:
Combining the soundness of $\mathsf{sCPDL}^{\infty}_f$ (Theorem 6.3.5) and the completeness of $\mathsf{sCPDL}^{\infty}_f$(Theorem 6.3.7) with Lemma 6.3.3. □

6.4 Interpolation

The following theorem is the main contribution of this chapter.

6.4.1. THEOREM (Craig Interpolation). *Let φ and ψ be* CPDL*-formulas such that $\varphi \vDash \psi$. Then there is an* interpolant *for φ and ψ – that is, a* CPDL*-formula ι such that*

1. $\mathsf{Voc}(\iota) \subseteq \mathsf{Voc}(\varphi) \cap \mathsf{Voc}(\psi)$,
2. $\varphi \vDash \iota$ *and*
3. $\iota \vDash \psi$.

As an immediate consequence of this we obtain Beth definability. Where $\varphi(p)$ is a formula, we use $\varphi(q)$ as an abbreviation of $\varphi[q/p]$.

6.4.2. Corollary (Beth Definability). *Let $p, q \in \mathsf{Prop}$ and let $\varphi(p)$ be a* CPDL-*formula. If $\varphi(p), \varphi(q) \models p \leftrightarrow q$, then there is a* CPDL-*formula χ with $\mathsf{Voc}(\chi) \subseteq \mathsf{Voc}(\varphi) \setminus \{p\}$ and $\varphi(p) \models p \leftrightarrow \chi$.*

Proof:
Apply Craig interpolation to $\varphi(p), p \models \varphi(q) \to q$. □

6.4.1 Proof setup

In the remainder of this chapter we will prove Theorem 6.4.1. We will use the split sequent system sCPDL_f to find Craig interpolants for CPDL. As in Chapter 5, we first transfer the concept of interpolation from formulas to split sequents, calling a formula θ an *interpolant* for an unsatisfiable split sequent $\Gamma \mid \Xi$ if $\mathsf{Voc}(\theta) \subseteq \mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Xi)$, and both sequents $\Gamma \mid \theta$ and $\overline{\theta} \mid \Xi$ are unsatisfiable. Since we have $\varphi \models \psi$ iff $\varphi \mid \overline{\psi}$ is unsatisfiable, it is easy to see that a formula θ is an interpolant for the formulas φ and ψ iff it is an interpolant for the split sequent $\varphi \mid \overline{\psi}$. Because the system sCPDL_f is complete, it suffices to prove the following result.

6.4.3. Theorem. *If $\vdash \Gamma \mid \Xi$, then the split sequent $\Gamma \mid \Xi$ has an interpolant.*

The proof of Theorem 6.4.3 easily follows from the Lemmas 6.4.4 and 6.4.5 below. The key notion in the proof is that of a *cluster*; recall that a cluster of a proof π is a maximal strongly connected subgraph of π. A cluster is called *trivial* if it consists of one node and *proper* otherwise.

Every proper cluster of π is in fact a *subtree* of the underlying tree of π, in the sense that the structure $(C, \lessdot\restriction_C)$ is a tree itself (here $\lessdot\restriction_C$ denotes the parent-child relation of π, restricted to C). In particular, every proper cluster has a root. We refer to the children of C-nodes which lie outside of C as the *exit nodes* of C — in the case of a trivial cluster these are just the children of the cluster's unique member.

We prove Theorem 6.4.3 by induction on the size of the derivation of $\Gamma \mid \Xi$. Lemma 6.4.4 takes care of leaves and of the induction step in the case where the root of the derivation forms a trivial cluster. We omit its proof, which is a straightforward adaptation of Maehara's method for wellfounded proofs.

6.4.4. Lemma. *Let π be an sCPDL_f proof of $\Gamma \mid \Xi$. Assume that the root r of π forms a trivial cluster, and that for every child v of r we have an interpolant θ_v for the split sequent $\mathsf{S}^l_v \mid \mathsf{S}^r_v$. Then $\Gamma \mid \Xi$ has an interpolant.*

The key task is to obtain interpolants for the roots of proper clusters.

6.4.5. Lemma. *Let π be a uniform sCPDL_f proof of $\Gamma \mid \Xi$, assume that the root r of π belongs to a proper cluster C, and that for every exit node v of C we have an interpolant θ_v for the split sequent $\mathsf{S}^l_v \mid \mathsf{S}^r_v$. Then $\Gamma \mid \Xi$ has an interpolant.*

Fix π, C, r, Γ, Ξ for the remainder of this chapter.

For reasons of symmetry we may confine our attention to the case where Ξ is focused. Furthermore, we will assume that Γ is non-empty, since if $\Gamma = \varnothing$ we may simply define the interpolant to be the formula $\bot$.

6.4.6. Remark. At this point, one might try to apply Maehara's method to prove interpolation for CPDL as in Chapter 5. It is instructive to see why this would fail. As mentioned in the introduction to this chapter, Maehara's method for cyclic proof systems requires solving certain systems of fixpoint equations.

The two-way modal μ-calculus is expressive enough to define arbitrary fixpoints, allowing us to solve all systems of fixpoint equations. In contrast, in CPDL only certain kinds of fixpoints can be expressed, so not every system can be solved within it.

To analyze which kinds of fixpoints are needed in the definition of the interpolant, let π be a sCPDL_f proof, where for simplicity all right components are focused. Let $\tau = c(l) \dots l$ be a successful repeat path, where every right component in τ contains a formula in focus. Following the strategy from Chapter 5, we introduce a pre-interpolant x at l, propagate the interpolant along τ and end up with a fixpoint equation

$$x = \theta(x)$$

at the companion node $c(l)$. As τ is successful in the right components, the called for solution in $\mathcal{L}^2_\mu$ of this equation is

$$\mu x.\theta(x).$$

Recall from Chapter 2 that CPDL corresponds to the *completely additive two-way μ-calculus* $\mathcal{L}^{2ca}_\mu$: recall that a $\mathcal{L}^2_\mu$-formula φ is in $\mathcal{L}^{2ca}_\mu$, if for any subformula $\mu x.\psi$ of φ, the variable x in ψ is not in the scope of a $\Box$-modality, an essential conjunction or a ν-operator; and dually for any subformula $\nu x.\psi$.

Therefore, we can find an equivalent formula to $\mu x.\theta(x)$ in CPDL iff $\mu x.\theta(x)$ is in $\mathcal{L}^{2ca}_\mu$. That is, if x is not in the scope of a $\Box$-modality, an essential conjunction or a ν-operator. Because all right components of τ have a formula in focus, all modal rules are of the form $\langle a \rangle^r$, and therefore x is not in the scope of a $\Box$-modality. On all companion nodes on τ above $c(l)$, μ-fixpoints are introduced. This is the case as there are no formulas on focus in the left components, and therefore all repeat paths are successful on the right. Consequently, x is not in the scope of a ν-operator.

The crux lies in the conjunctions in θ; to see this, assume that the following split rule is applied on τ:

$$\frac{\varphi_0, \Gamma \mid \Delta \qquad \varphi_1, \Gamma \mid \Delta}{\varphi_0 \lor \varphi_1, \Gamma \mid \Delta} \lor^l$$

Given interpolants ι_0 and ι_1 for the respective premises of $\vee^l$ and ι_2 for its conclusion, we have $\iota_2 = \iota_0 \wedge \iota_1$. If x occurs freely in ι_0 and in ι_1, then x is in the scope of the essential conjunction $\iota_0 \wedge \iota_1$ in $\mu x.\theta(x)$.

As a consequence of this argument, if we want to find a system of fixpoint equations that is solvable within CPDL, we have to avoid rules with the principal formula in the unfocused component, and with multiple premises. This is exactly the motivation behind quasi-proofs, that we will introduce in Subsection 6.4.3: a *quasi-proof* Q consists of the focused components of a CPDL_f-proof π, where nodes with the same focused component are unified. In a way, the quasi-proof Q is obtained from π by "forgetting" the unfocused components. Consequently, nodes in a quasi-proof will be labeled with a focused sequent Δ – the focused component of the corresponding node in π. Because we assume proofs to be uniform, there is only one right rule applied at nodes in π with right component Δ. We thus may relate nodes in the quasi-proof by right rules and get an almost proof-like structure Q. On this quasi-proof Q we can then apply Maehara's method to define the interpolant. Because in Q we only apply rules with principal formula in the focused component, this results in a fixpoint equation that is solvable within CPDL. Importantly, an interpolant for the root of Q will also be an interpolant for the root of π. We will make these ideas precise below.

6.4.2 Proper clusters

We first discuss proper clusters in some more detail. Let

$$C^+ := C \cup \{v \in \pi \mid w \lessdot v \text{ for some } w \in C\}$$

be the set of nodes that either belong to C or are the child of a node in C. Then $C^+ \setminus C$ is the set of exit nodes of C. The following lemma will be used implicitly. Its proof is straightforward and will be omitted.

6.4.7. LEMMA. *For all $v \in C$, the following hold: (1) S^r_v and S^l_v are both non-empty; (2) S^r_v is focused; (3) all children of v are in C^+, and at least one is in C; (4) if a right rule, other than D, is applied at v, then $\mathsf{S}^r_v \neq \mathsf{S}^r_w$, for every child w of v.*

6.4.8. DEFINITION. We let $\mathcal{F}_C$ denote the sets of sequents occurring as a right component in C, namely $\mathcal{F}_C := \{\mathsf{S}^r_v \mid v \in C\}$, and likewise for $\mathcal{F}_{C^+}$. Given a sequent $\Delta \in \mathcal{F}_{C^+}$, we define $C_\Delta := \{v \in C \mid \mathsf{S}^r_v = \Delta\}$, $C^+_\Delta := \{v \in C^+ \mid \mathsf{S}^r_v = \Delta\}$ and we let C^l_Δ (C^r_Δ, respectively), denote the set of nodes in C_Δ where a left rule (a right rule other than D, respectively) is applied.

6.4.9. LEMMA. *For all sequents Δ, the following hold:*

1. *If $v \in C^l_\Delta$, then the rule applied at v is not the modal rule.*

2. *If* $v \in C^l_\Delta$*, then all of its children belong to* C^+_Δ*, and at least one to* C_Δ*.*

3. *If* C_Δ *is not empty, then* C^r_Δ *is not empty.*

Proof:
Obvious by the the definitions. □

By uniformity of π, for each $\Delta \in \mathcal{F}_C$ there is a unique right rule R_Δ that is applied at each $v \in C^r_\Delta$ (provided $C^r_\Delta \neq \varnothing$). If R_Δ is the modal rule we call Δ a *modal sequent*. If $\Delta \in \mathcal{F}_{C^+} \setminus \mathcal{F}_C$, we call Δ an *exit sequent*.

6.4.10. Lemma. *If* Δ *is neither a modal nor an exit sequent then there are sequents* $\Pi_1, \dots, \Pi_n$ *such that* $\bigwedge \Delta \equiv \bigvee_i \bigwedge \Pi_i$*, and, for all* $v \in C^r_\Delta$*, the children of* v *can be listed as* $w_1, \dots, w_n$ *such that* $\mathsf{S}_{w_i} = \mathsf{S}^l_v \mid \Pi_i$*, for all* $i = 1, ..., n$*.*

Proof:
Let R_Δ be the unique right rule that is applied at each $v \in C^r_\Delta$. As an exemplary case assume that $\mathsf{R}_\Delta = \vee^r$. Then at each such $v \in C^r_\Delta$ the rule looks as follows.

$$\frac{\Sigma \mid \Pi, \varphi^u \qquad \Sigma \mid \Pi, \psi^u}{\Sigma \mid \Pi, \varphi \vee \psi^u} \vee^r$$

The claim of the lemma is immediate. By assumption $\mathsf{R}_\Delta \neq \langle a \rangle^r$. The case of all other right rules is analogous. □

6.4.3 Quasi-proofs

We can now introduce the pivotal structure in our interpolation proof: the *quasi-proof* $\mathsf{Q} = (Q, \lessdot_{\mathsf{Q}}, k, \Psi)$ associated with the cluster C. Roughly, Q is a finite labeled tree that represents the focused part of C. In particular, its labeling is a map $\Psi : Q \to \mathcal{F}_{C^+}$ that respects the labeling of C as suggested by Lemma 6.4.10; also, any node labeled with an exit sequent is a leaf of Q. To ensure that Q is based on a *finite* tree, we make sure that every repeat node is a leaf.

To explain the role of the typing map k in Q, note that the purpose of Q is to help find an interpolant for the root r of C. We will do this by inductively associating with each node in Q an auxiliary formula that we will call a *pre-interpolant*. To facilitate this definition, we construct Q in such a way that its internal nodes come in *triples*. The subsequent nodes of such a triple are all labeled with the same sequent in $\mathcal{F}_{C^+}$, but they have a different *type* (respectively, 1, 2 and 3). This typing will play a role in the actual definition of the pre-interpolants.

6.4.11. Definition. Given the cluster C we construct a structure $\mathsf{Q} = (Q, \lessdot_{\mathsf{Q}}, k, \Psi)$, called a *quasi-proof*, step by step. Here $(Q, \lessdot_{\mathsf{Q}})$ will be a finite tree, $k : Q \to \{1, 2, 3\}$ types the nodes of Q and $\Psi : Q \to \mathcal{F}_{C^+}$ is a labeling.

To start the construction, we put a root node r_{Q} in Q and let r_{Q} have type 1 and label Ξ. Inductively, given a node $x \in Q$, define the children of x as follows:

Case $k(x) = 1$. If x is a *repeat* in Q (that is, there exists $y \in Q$ such that y is an ancestor of x and $\Psi_x = \Psi_y$) or an *exit* (that is, Ψ_x is an exit sequent), then x is a leaf. Otherwise, x has a unique child with type 2 and label Ψ_x.

Case $k(x) = 2$. Then x has a unique child with type 3 and label Ψ_x.

Case $k(x) = 3$. In this case[1] $C^r_{\Psi_x} \neq \varnothing$. If Ψ_x is modal, say, it is of the form $\langle a\rangle\varphi^f, [a]\Sigma, \Pi$ then x has a unique child y with type 1 and label $\varphi^f, \Sigma, \langle\breve{a}\rangle\Pi$.

If Ψ_x is not modal then by 6.4.10 there exist $\Pi_1, \dots, \Pi_n$ such that $\bigwedge \Psi_x \equiv \bigvee_i \bigwedge \Pi_i$ and for all $v \in C^r_{\Psi_x}$, the children of v can be listed as $w_1, \dots, w_n$ with $\mathsf{S}_{w_i} = \mathsf{S}^l_v \mid \Pi_i$, for all i. We define the children of x in Q as $y_1, \dots, y_n$, where each y_i has type 1 and label Π_i.

Given the repeat condition it is fairly easy to check that Q is a finite tree.

6.4.12. Example. Consider the sCPDL_f-proof ρ' of $\langle\breve{a}^*\rangle q, [a]\varphi \mid \langle a\rangle\langle a^*\rangle p^f$ given in Example 6.2.3, where $\varphi := [a^*][p?][\breve{a}^*]\overline{q}$. We recall the proper cluster C in ρ'. Note that the nodes v_0 and v_1 labeled with respectively $[\breve{a}^*]\overline{q}, \langle\breve{a}\rangle\langle\breve{a}^*\rangle q \mid$ and $\langle\breve{a}\rangle\langle\breve{a}^*\rangle q, [p?][\breve{a}^*]\overline{q} \mid p$ are exit nodes in $C^+ \setminus C$.

$$C^+ : \quad \dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{[\breve{a}^*]\overline{q}, \langle\breve{a}\rangle\langle\breve{a}^*\rangle q \mid \qquad [\langle\breve{a}^*\rangle q, [a]\varphi \mid \langle a\rangle\langle a^*\rangle p^f]^{\dagger}}{\langle\breve{a}\rangle\langle\breve{a}^*\rangle q, [a]\varphi \mid \langle a\rangle\langle a^*\rangle p^f}\,\mathsf{acut}^l \qquad \langle\breve{a}\rangle\langle\breve{a}^*\rangle q, [p?][\breve{a}^*]\overline{q} \mid p}{\langle\breve{a}\rangle\langle\breve{a}^*\rangle q, [a]\varphi, [p?][\breve{a}^*]\overline{q} \mid \langle a^*\rangle p^f}\,\langle * \rangle^r}{\langle\breve{a}\rangle\langle\breve{a}^*\rangle q, \varphi \mid \langle a^*\rangle p^f}\,[*]^l}{\langle\breve{a}^*\rangle q, [a]\varphi \mid \langle a\rangle\langle a^*\rangle p^f}\,\langle a\rangle^r}{\langle\breve{a}^*\rangle q, [a]\varphi \mid \langle a\rangle\langle a^*\rangle p^f}\,\mathsf{D}^r_{\dagger}}{}$$

The sequents at the exit nodes v_0 and v_1 have interpolants $\top$ and p, respectively. In order to find an interpolant of the root sequent, we give the quasi-proof Q associated with the cluster C. We write the type of a node in Q to its left, and label nodes $x \in \mathsf{Q}$ of type 3 with the unique rule that is applied to split sequents in C where the right component is Ψ_x.

[1] Every type-3 node is the grandchild of a type-1 node with the same label, but any type-1 node z such that $C^r_{\Psi_z} = \varnothing$ is a leaf.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{1 : \langle a\rangle\langle a^*\rangle p^f \qquad 1 : p}{3 : \langle a^*\rangle p^f}\ \langle * \rangle}{2 : \langle a^*\rangle p^f}}{1 : \langle a^*\rangle p^f}}{3 : \langle a\rangle\langle a^*\rangle p^f}\ \langle a \rangle}{2 : \langle a\rangle\langle a^*\rangle p^f}}{1 : \langle a\rangle\langle a^*\rangle p^f}$$

Note that the left leaf of Q is a *repeat* and the right leaf is an *exit*.

6.4.13. Definition. We let $<_{\mathsf{Q}}$ and $\leq_{\mathsf{Q}}$ denote, respectively, the transitive and the reflexive-transitive closure of $\lessdot_{\mathsf{Q}}$. For a repeat leaf $z \in Q$, we let $c(z)$ be its *companion*, defined as the unique node x such that $x <_{\mathsf{Q}} z$, $k(x) = k(z) = 1$ and $\Psi_x = \Psi_z$. We let L_{Q} and K_{Q} denote, respectively, the sets of repeats and companions of Q, in particular $\mathsf{K}_{\mathsf{Q}} := \{c(z) \mid z \in \mathsf{L}_{\mathsf{Q}}\}$. Given $x \in Q$ we set

$$\mathsf{L}_{<x} := \{z \in \mathsf{L}_{\mathsf{Q}} \mid c(z) <_{\mathsf{Q}} x \leq_{\mathsf{Q}} z\} \text{ and } \mathsf{K}_{<x} := \{c(z) \mid z \in \mathsf{L}_{<x}\}.$$

A *repeat path* in Q is a sequence of the form $(x_k)_{0 \leq k \leq n}$ such that for some leaf z we have $x_0 = c(z)$, $x_n = z$ and $x_i \lessdot_{\mathsf{Q}} x_{i+1}$ for all $i < n$.

6.4.14. Lemma. *Let* Q *be a quasi-proof and let $x \in Q$. Then the following hold:*

1. *Ψ_x is focused.*
2. *If x is a leaf or a companion node in* Q*, then $k(x) = 1$.*
3. *$\mathsf{L}_{<r_{\mathsf{Q}}} = \mathsf{K}_{<r_{\mathsf{Q}}} = \varnothing$.*
4. *$x \notin \mathsf{K}_{<x}$.*
5. *If x is not a companion, then $\mathsf{L}_{<x} = \bigcup_{x \lessdot_{\mathsf{Q}} y} \mathsf{L}_{<y}$ and $\mathsf{K}_{<x} = \bigcup_{x \lessdot_{\mathsf{Q}} y} \mathsf{K}_{<y}$.*
6. *If x only has one child y, then $\mathsf{L}_{<x} \subseteq \mathsf{L}_{<y}$ and $\mathsf{K}_{<x} \subseteq \mathsf{K}_{<y}$.*
7. *Every repeat path features nodes with distinct formulas in focus.*

Proof:
The items 1 – 6 follow immediately from the definitions. For item 7 let ρ be a repeat path. We write $\rho = x_0 y_0 z_0 x_1 y_1 \cdots z_{n-1} x_n$, where the x, y and z-nodes are, respectively, of type 1, 2 and 3. Note that $\Psi_{x_i} = \Psi_{y_i} = \Psi_{z_i}$ for all $i < n$, and

that $\Psi_{x_0} = \Psi_{x_n}$. We will simply write Ψ_i for Ψ_{x_i} and let ξ_i denote the formula in focus in Ψ_i. The key claim in the proof is the following:[2]

$$\text{if } \xi_i = \xi_{i+1} \text{ then } \Psi_i \subset \Psi_{i+1}. \tag{6.1}$$

To see this, first note that by definition of Q there must be some $v \in C^r_{\Psi_i}$ which has a successor $u \in C^r_{\Psi_{i+1}}$. Hence by Lemma 6.4.7(4) the sets $\Psi_i = \mathsf{S}^r_v$ and $\Psi_{i+1} = \mathsf{S}^r_u$ must be distinct. Now assume $\xi_i = \xi_{i+1}$; it follows that the principal formula at v must be out of focus. But by uniformity the rule applied to this formula is cumulative and productive, which implies that Ψ_i is a proper subset of Ψ_{i+1}. This proves (6.1).

Now assume for contradiction that $\xi_i = \xi_{i+1}$ for all $i < n$. Then $(\Psi_i)_{0 \leq i \leq n}$ is a strictly increasing sequence of sets, which clearly contradicts the assumption that $\Psi_0 = \Psi_n$. □

As mentioned before, each node x in Q represents a certain (not necessarily connected) subset R_x of C^+, which we call its *region*:

$$R_x := \begin{cases} C^+_{\Psi_x} & \text{if } k(x) = 1, 2 \\ C^r_{\Psi_x} & \text{if } k(x) = 3. \end{cases}$$

6.4.15. LEMMA. *Let $x \in \mathsf{Q}$ with $k(x) = 3$. If we list the children of $x \in \mathsf{Q}$ as $z_1, \ldots, z_n$ then for all $v \in R_x$, the children of v may be listed as $w_1, \ldots, w_n$ so that $w_i \in R_{z_i}$ for all $i = 1, ..., n$.*

Proof:
Follows from the definition of Q. □

6.4.4 Pre-interpolants and the interpolant

We are now ready to define the interpolant θ_r for the root r of C. The key idea underlying this definition is to first associate with each node x in the quasi-proof Q a so-called *pre-interpolant* ι_x. These pre-interpolants are auxiliary formulas that will be defined by a leaf-to-root induction on the tree $(Q, \lessdot_{\mathsf{Q}})$; once we have arrived at the root r_{Q} of Q we simply define the interpolant θ_r as $\theta_r := \iota_{r_{\mathsf{Q}}}$. For the definition of these pre-interpolants we extend the language with a set $\{q_x \mid x \in \mathsf{K}_{\mathsf{Q}}\}$ of internal variables, and with every set $\Delta \in \mathcal{F}_{C^+}$ we associate an *exit interpolant*:

$$\theta_\Delta := \bigwedge \{\theta_v \mid v \in C^+_\Delta \setminus C_\Delta\}.$$

[2]Here $X \subset Y$ denotes that X is a *proper* subset of Y, where $X \neq Y$.

6.4.16. DEFINITION. By a leaf-to-root induction, we define, for all nodes $x \in Q$, a formula ψ_x and a family of programs $\{\alpha_{x,y} \mid y \in \mathsf{K}_{<x}\}$. In all applicable cases, z denotes the unique successor of x, and in the case where x is a modal node of type 3, a denotes the leading atomic program of the formula in focus in Ψ_x. Note that for exit nodes x, we have $\mathsf{K}_{<x} = \varnothing$, so no definition of $\alpha_{x,y}$ is required.

Case	ψ_x	$\alpha_{x,y}$
x is a repeat	$\bot$	$\top?$
x is an exit	θ_{Ψ_x}	–
x is a companion	$\langle \alpha^*_{z,x} \rangle \psi_z$	$\alpha^*_{z,x}; \alpha_{z,y}$
x is otherwise of type 1	ψ_z	$\alpha_{z,y}$
x is of type 2	$\langle \theta_{\Psi_x}? \rangle \psi_z$	$\theta_{\Psi_x}?; \alpha_{z,y}$
x is of type 3, not modal	$\bigvee \{\psi_z \mid x \lessdot_{\mathsf{Q}} z\}$	$\bigcup \{\alpha_{z,y} \mid x \lessdot_{\mathsf{Q}} z, y \in \mathsf{K}_{<z}\}$
x is of type 3, modal	$\langle a \rangle \psi_z$	$a; \alpha_{z,y}$

Based on these expressions the *pre-interpolant* ι_x of a node $x \in Q$ is defined as:

$$\iota_x \;:=\; \psi_x \vee \bigvee_{y \in \mathsf{K}_{<x}} \langle \alpha_{x,y} \rangle q_y.$$

Note that the programs $\alpha_{x,y}$ and formulas ψ_x do not contain internal variables.

6.4.17. DEFINITION. We define the interpolant θ_r of the root r of the cluster C as

$$\theta_r := \iota_{r_{\mathsf{Q}}}.$$

6.4.18. EXAMPLE. We continue Example 6.4.12 in which a proper cluster C with root sequent $\langle \breve{a}^* \rangle q, [a]\varphi \mid \langle a \rangle \langle a^* \rangle p^f$ is given, where $\varphi := [a^*][p?][\breve{a}^*]\overline{q}$. In this example we defined the associated quasi-proof Q as follows:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{1 : \langle a \rangle \langle a^* \rangle p^f \qquad 1 : p}{3 : \langle a^* \rangle p^f}\langle * \rangle}{2 : \langle a^* \rangle p^f}}{1 : \langle a^* \rangle p^f}}{3 : \langle a \rangle \langle a^* \rangle p^f}\langle a \rangle}{2 : \langle a \rangle \langle a^* \rangle p^f}}{1 : \langle a \rangle \langle a^* \rangle p^f}$$

Note that this quasi-proof contains one repeat (the left leaf) with companion y (the root). We inductively define the formula ψ_x and the program $\alpha_{x,y}$ for every node x in Q by the following table:

Node x	ψ_x	$\alpha_{x,y}$
$1 : \langle a\rangle\langle a^*\rangle p^f$	$\bot$	$\top?$
$1 : p$	p	$-$
$3 : \langle a^*\rangle p^f$	$\bot \vee p$	$\top?$
$2 : \langle a^*\rangle p^f$	$\bot \vee p$	$\top?$
$1 : \langle a^*\rangle p^f$	$\bot \vee p$	$\top?$
$3 : \langle a\rangle\langle a^*\rangle p^f$	$\langle a\rangle(\bot \vee p)$	$a; \top?$
$2 : \langle a\rangle\langle a^*\rangle p^f$	$\langle \top?\rangle\langle a\rangle(\bot \vee p)$	$\top?; a; \top?$
$1 : \langle a\rangle\langle a^*\rangle p^f$	$\langle(\top?; a; \top?)^*\rangle\langle \top?\rangle\langle a\rangle(\bot \vee p)$	$-$

Note that in the last line $\alpha_{x,y}$ is undefined because $x = y$. For the root r of C we obtain that

$$\theta_r = \iota_y = \langle(\top?; a; \top?)^*\rangle\langle \top?\rangle\langle a\rangle(\bot \vee p).$$

This formula can be simplified to the equivalent formula $\langle a^*\rangle\langle a\rangle p$ and we can check that θ_r is indeed an interpolant of $\langle \breve{a}^*\rangle q, [a][a^*][p?][\breve{a}^*]\overline{q} \mid \langle a\rangle\langle a^*\rangle p^f$.

6.5 Correctness of the interpolant

To prove Lemma 6.4.5 and thereby establish the Craig interpolation property for CPDL, we verify that the formula θ_r from Definition 6.4.17 satisfies the three conditions of an interpolant for $\Gamma \mid \Xi$: first, the vocabulary condition $\mathsf{Voc}(\theta_r) \subseteq \mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Xi)$ in Lemma 6.5.3; second, that $\Gamma \mid \theta_r$ is unsatisfiable in Lemma 6.5.6; and third, that $\overline{\theta_r} \mid \Xi$ is unsatisfiable in Lemma 6.5.9.

To establish the base cases of those three lemmas, we first state the following auxiliary lemma that follows from the assumptions of Lemma 6.4.5.

6.5.1. LEMMA. *For any sequent $\Delta \in \mathcal{F}_{C^+}$ the following hold:*

1. $\vdash \overline{\theta_\Delta} \mid \Delta$.

2. $\vdash \mathsf{S}^l_v \mid \theta_\Delta$ *for all* $v \in C^+_\Delta \setminus C_\Delta$.

3. $\mathsf{Voc}(\theta_\Delta) \subseteq \mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Xi)$.

6.5.1 Proof of vocabulary condition

The following lemma is used to show the vocabulary condition.

6.5.2. LEMMA. *For all $v \in C$ we have that:*

$$\mathsf{Voc}(\mathsf{S}^l_v) \subseteq \mathsf{Voc}(\Gamma) \quad \textit{and} \quad \mathsf{Voc}(\mathsf{S}^r_v) \subseteq \mathsf{Voc}(\Xi).$$

Proof:
By root-to-leaf induction on C using the fact that all our proof rules are analytic. □

We now state and show the vocabulary condition.

6.5.3. Lemma. *For all nodes $x \in Q$, we have*

$$\mathsf{Voc}(\iota_x) \subseteq \big(\mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Xi)\big) \cup \{q_y \mid y \in \mathsf{K}_{<x}\}.$$

As an immediate corollary, we have: $\mathsf{Voc}(\theta_r) \subseteq \mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Xi)$.

Proof:
The statement of the lemma is proved by a leaf-to-root induction on the structure of the quasi-proof Q.

Case $k(x) = 1$, x is a repeat. We have $\iota_x = q_{c(x)}$, so the claim follows from the fact that $c(x) \in \mathsf{K}_{<x}$.

Case $k(x) = 1$, x is an exit. We have $\iota_x = \theta_{\Psi_x}$, so the claim holds by Lemma 6.5.1.

Case $k(x) = 1$, x is a companion. In this case x has a unique child z with $\mathsf{K}_{<z} = \mathsf{K}_{<x} \cup \{x\}$. By definition the interpolants ι_x and ι_z are of the forms

$$\iota_x = \bigvee_{y \in \mathsf{K}_{<x}} \langle \alpha^*_{z,x}; \alpha_{z,y} \rangle q_y \vee \langle \alpha^*_{z,x} \rangle \psi_z,$$
$$\iota_z = \langle \alpha_{z,x} \rangle q_x \vee \bigvee_{y \in \mathsf{K}_{<x}} \langle \alpha_{z,y} \rangle q_y \vee \psi_z.$$

By the inductive hypothesis and the definition of ι_x, it is straightforward to calculate that:

$$\mathsf{Voc}(\iota_x) \subseteq \big(\mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Xi)\big) \cup \big(\{q_y \mid y \in \mathsf{K}_{<z}\} \setminus \{q_x\}\big).$$

Case $k(x) = 1$, x is neither a leaf nor a companion. In this case, x has a unique child z, and $\iota_x = \iota_z$. By Lemma 6.4.14, we have $\mathsf{K}_{<x} = \mathsf{K}_{<z}$. By the induction hypothesis for z the claim follows.

Case $k(x) = 2$. In this case x has a unique child z, and

$$\iota_x = \langle \theta_{\Psi_x}? \rangle \iota_z.$$

By the induction hypothesis,

$$\mathsf{Voc}(\iota_z) \subseteq \big(\mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Xi)\big) \cup \{q_y \mid y \in \mathsf{K}_{<z}\}.$$

Notice that x is not a companion by Lemma 6.4.14, and thus by the same lemma we get that $\mathsf{K}_{<z} = \mathsf{K}_{<x}$. Also, by Lemma 6.5.1, we find $\mathsf{Voc}(\theta_{\Psi_x}) \subseteq \mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Xi)$. The claim follows from this.

Case $k(x) = 3$, Ψ_x is not modal. Here we have

$$\iota_x = \bigvee_{x \lessdot_{\mathsf{Q}} z} \psi_z \vee \bigvee_{y \in \mathsf{K}_{<x}} \langle \bigcup \{\alpha_{z,y} \mid x \lessdot_{\mathsf{Q}} z, y \in \mathsf{K}_{<z}\} \rangle q_y,$$

where $\bigcup\{\alpha_{z,y} \mid x \lessdot_{\mathsf{Q}} z, y \in \mathsf{K}_{<z}\}$ represents the choice program formed by combining all the programs $\alpha_{z,y}$ such that $x \lessdot_{\mathsf{Q}} z$ and $y \in \mathsf{K}_{<z}$. By the induction hypothesis, $\mathsf{Voc}(\iota_z)$ is a subset of

$$\big(\mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Xi)\big) \cup \{q_y \mid y \in \mathsf{K}_{<z}\}.$$

Since x is not a companion by Lemma 6.4.14, by the same lemma we find $\mathsf{K}_{<x} = \bigcup_{x \lessdot_{\mathsf{Q}} z} \mathsf{K}_{<z}$. The claim follows from this.

Case $k(x) = 3$, Ψ_x is modal. In this case, x has a unique child z, and $\iota_x = \langle a \rangle \iota_z$ where a is the leading atomic program of the formula in focus in Ψ_x. By the induction hypothesis,

$$\mathsf{Voc}(\iota_z) \subseteq \big(\mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Xi)\big) \cup \{q_y \mid y \in \mathsf{K}_{<z}\}.$$

Since x is not a companion by Lemma 6.4.14, the same lemma gives $\mathsf{K}_{<z} = \mathsf{K}_{<x}$. Clearly $a \in \mathsf{Voc}(\Psi_x) \subseteq \mathsf{Voc}(\Xi)$. It remains to show that also $a \in \mathsf{Voc}(\Gamma)$. Recall that an action a is in the vocabulary of Γ if a or $\breve{a}$ occur in Γ. Arguing towards a contradiction assume that $a \notin \mathsf{Voc}(\Gamma)$. Then by Lemma 6.5.2 we get that $a \notin \mathsf{Voc}(\mathsf{S}^l(v))$ for any $v \in C$. At any node $v \in R_x$ the rule $\langle a \rangle^r$ is applied. Let $v \in R_x$ with child w. Then $\mathsf{S}^l_w = \langle \breve{a} \rangle \mathsf{S}^l_v$. But $\langle \breve{a} \rangle \mathsf{S}^l_v$ only contains formulas $\langle \breve{a} \rangle \chi$, such that $\langle \breve{a} \rangle \chi \in \mathsf{Clos}^{\neg}(\mathsf{S}^l_v)$ and if $\breve{a} \notin \mathsf{Voc}(\mathsf{S}^l_v)$ this implies that S^l_w is empty. This contradicts Lemma 6.4.7.

Finally, the corollary that $\mathsf{Voc}(\theta_r) \subseteq \mathsf{Voc}(\Gamma) \cap \mathsf{Voc}(\Xi)$ is immediate since $\theta_r = \iota_{r_{\mathsf{Q}}}$ and $\mathsf{K}_{<r_{\mathsf{Q}}} = \varnothing$. □

6.5.2 Proof of second condition: $\Gamma \mid \theta_r$ is unsatisfiable

In the proof of $\vdash \Gamma \mid \theta_r$ we will need the following definition and lemma.

6.5.4. Definition. Let π be some split proof, possibly with assumptions. We call π *right-focused* if the path from the root of π to any of its assumptions is right-focused (that is, the right component of each node on such a path is focused).

For the proof of the following lemma, note that assumption-free proofs are automatically right-focused, and that $\rho_\Sigma : \mathcal{A} \vdash \Sigma \mid \chi^u$ can only be right-focused if $\mathcal{A} = \varnothing$.

6.5.5. LEMMA. *Let $\mathcal{S}$ and $\mathcal{A}$ be finite sets of respectively (unfocused) sequents and split sequents, let α be some program, q a proposition letter not occurring in either $\mathcal{S}, \mathcal{A}$ or α, and χ some formula. Assume that for every $\Sigma \in \mathcal{S}$ there are right-focused proofs $\pi_\Sigma : \{\Pi \mid q^f : \Pi \in \mathcal{S}\} \vdash \Sigma \mid \langle\alpha\rangle q^f$ and $\rho_\Sigma : \mathcal{A} \vdash \Sigma \mid \chi^b$. Then we may construct a right-focused proof witnessing that $\mathcal{A} \vdash \Sigma \mid \langle\alpha^*\rangle\chi^b$.*

Proof:
We first consider the case where $b = f$. Abbreviate $\mathcal{S}' := \{\Pi \mid \langle\alpha^*\rangle\chi^f : \Pi \in \mathcal{S}\}$, then for any $\Sigma \mid \langle\alpha^*\rangle\chi^f$ in $\mathcal{S}'$, we may consider the following derivation π'_Σ : $\mathcal{A} \cup \mathcal{S}' \vdash \Sigma \mid \langle\alpha^*\rangle\chi^f$:

$$\dfrac{\begin{matrix}\mathcal{A}\\ \vdots\ \rho_\Sigma \\ \Sigma \mid \chi^f\end{matrix} \qquad \begin{matrix}\mathcal{S}'\\ \vdots\ \pi_\Sigma[\langle\alpha^*\rangle\chi/q] \\ \Sigma \mid \langle\alpha\rangle\langle\alpha^*\rangle\chi^f\end{matrix}}{\Sigma \mid \langle\alpha^*\rangle\chi^f}\ \langle * \rangle^r$$

Here, $\pi_\Sigma[\langle\alpha^*\rangle\chi/q]$ denotes the derivation obtained from π_Σ by substituting every occurrence of the proposition letter q with the formula $\langle\alpha^*\rangle\chi$. It is straightforward to check that $\pi_\Sigma[\langle\alpha^*\rangle\chi/q]$ is right-focused. Note that $\langle\alpha^*\rangle\chi^f$, the formula in focus at the root of π'_Σ is actually principal there.

The main claim in the proof is the following:

<u>`Claim 1`</u>: For all $n \geq 1$ and for every $\Sigma \mid \langle\alpha^*\rangle\chi^f$ in $\mathcal{S}'$ there is a right-focused proof $\pi_n : \mathcal{A} \cup \mathcal{S}' \vdash \Sigma \mid \langle\alpha^*\rangle\chi^f$, such that for every open leaf ℓ, which is labeled with some $\mathcal{S}'$ assumption, there are at least n distinct nodes on the path from the root of π_n to ℓ that are labeled with split sequents from $\mathcal{S}'$ and where the formula in focus is principal.

`Proof of Claim 1`: We prove this claim by induction on n. In the base step, where $n = 1$, we can take the proof π'_Σ of the split sequent $\Sigma \mid \langle\alpha^*\rangle\chi^f$. In the inductive step we assume a proof π_k satisfying the above constraints for $n = k$. Now consider an arbitrary non-repeat leaf ℓ of π_n which is labeled with some split sequent in $\mathcal{S}'$, say, with $\Sigma_\ell \mid \langle\alpha^*\rangle\chi^f$. If we replace each such ℓ with the derivation π'_{Σ_ℓ}, it is easily verified that the resulting derivation π_{k+1} satisfies the constraints for $n = k + 1$. ⊣

Finally, we consider the derivation π_N, with $N-1$ being the number of all sequents in $\mathcal{S}'$. Obviously then, for every path from the root of π_N to an open leaf labeled with some $\mathcal{S}'$ assumption, there are two nodes labeled with the same sequent in $\mathcal{S}'$. For any such path β let c_β and v_β be the minimal such nodes. We introduce a $\mathsf{D}_\dagger$ rule at c_β and let v_β be discharged by $\dagger$. Because π_N is right-focused and the formula in focus is principal at c_β, the path from c_β to v_β is successful. This finishes the proof for the case where $b = f$.

For the case where $b = u$ we first note that this implies $\mathcal{A} = \varnothing$ by definition of a right-focused proof. Because Σ is unfocused and $\pi_\Sigma \vdash \Sigma \mid \chi^u$ we easily obtain proofs ρ_Σ of $\Sigma \mid \chi^f$ by applying a u rule. Each ρ_Σ is without assumptions and hence right-focused. We can thus apply the focused-case of the Lemma (with $\mathcal{A} = \varnothing$) and obtain proofs of $\Sigma \mid \langle \alpha^* \rangle \chi^f$. Applying one f rule then concludes the proof. □

With this definition and lemma, we are now ready to establish the second condition of the interpolant θ_r that states that $\Gamma \mid \theta_r$ is unsatisfiable. In particular, we establish this by providing a sCPDL_f-proof of $\Gamma \mid \theta_r$.

6.5.6. LEMMA. $\vdash \Gamma \mid \theta_r$.

Proof:
The intuition underlying the proof is to show, by means of a leaf-to-root induction, that for each node x of Q and for each $v \in R_x$ we can find a right-focused proof of $\mathcal{C}_x \vdash \mathsf{S}^l_v \mid \iota^f_x$, where $\mathcal{C}_x$ is some suitable set of assumptions. In the case where x is a companion, the idea is to *discharge* some of the assumptions, so that, when we arrive at the root r_Q of Q we obtain an assumption-free proof of the split sequent $\Lambda^l_v \mid \iota^f_{r_\mathsf{Q}} = \Gamma \mid \theta^f_r$. For a proper proof-theoretic execution of this elimination procedure we need to prove a somewhat stronger claim, which involves separate statements on the constituting parts of the pre-interpolants.

<u>`Claim 2`</u>: For all $x \in Q$, for all $y \in \mathsf{K}_{<x}$ and all $v \in R_x$ we have that (1) $\vdash \mathsf{S}^l_v \mid \psi^u_x$ and (2) there is a right-focused proof of $\mathsf{S}^l_v \mid \langle \alpha_{x,y} \rangle q^f_y$ with assumptions $\mathcal{A}_y := \{(\mathsf{S}^l_w \mid q^f_y) \mid w \in R_y\}$.

`Proof of Claim 2:` We prove the Claim by a leaf-to-root induction on x.

Case $k(x) = 1$, x is a repeat. In this case, we have $\mathsf{K}_{<x} = \{c(x)\}$, which implies $y = c(x)$ and consequently $R_x = R_y$. From this it follows that $v \in R_y$ and so we find that $\mathsf{S}^l_v \mid q^f_y \in \mathcal{A}_y$. Note as well that in this case we have $\langle \alpha_{x,y} \rangle q_y = \langle \top? \rangle q_y$ and $\psi_x = \bot$. It is then easy to show that $\vdash \mathsf{S}^l_v \mid \bot^u$, and to find a right-focused proof witnessing $\mathcal{A}_y \vdash \mathsf{S}^l_v \mid \langle \top? \rangle q^f_y$.

Case $k(x) = 1$, x is an exit. Since $\mathsf{K}_{<x} = \varnothing$ we only need to show that $\vdash \mathsf{S}^l_v \mid \psi^u_x$. Observe that by definition we have $\psi_x = \theta_{\Psi_x}$; but then by Lemma 6.5.1 we obtain $\vdash \mathsf{S}^l_v \mid \theta^u_{\Psi_x}$ for any $v \in R_x$ as required.

Case $k(x) = 1$, x is a companion. First we show (2). Notice that in this case x has a unique child z and $\mathsf{K}_{<z} = \mathsf{K}_{<x} \cup \{x\}$ and $\alpha_{x,y} = \alpha^*_{z,x}; \alpha_{z,y}$. Let y and v be as in the claim, and note that $y \neq x$ since $x \notin \mathsf{K}_{<x}$ (Lemma 6.4.14). By the induction hypothesis on z we have right-focused proofs witnessing $\{\mathsf{S}^l_w \mid q^f_x : w \in R_x\} \vdash \mathsf{S}^l_v \mid \langle \alpha_{z,x} \rangle q^f_x$ and $\mathcal{A}_y \vdash \mathsf{S}^l \mid \langle \alpha_{z,y} \rangle q^f_y$. Then we may apply Lemma 6.5.5, with $\mathcal{A} = \mathcal{A}_y$, $\mathcal{S} = \{\mathsf{S}^l_w : w \in R_x\}$, $\alpha = \alpha_{z,x}$, $q = q_x$ and $\chi = \langle \alpha_{z,y} \rangle q_y$. This yields a right-focused proof witnessing

$\mathcal{A}_y \vdash \mathsf{S}^l \mid \langle \alpha^*_{z,x} \rangle \langle \alpha_{z,y} \rangle q^f_y$, so that with one right application of $\langle ; \rangle$ we find that $\mathcal{A}_y \vdash \mathsf{S}^l \mid \langle \alpha^*_{z,x} ; \alpha_{z,y} \rangle q^f_y$. This suffices, since we have $\alpha_{x,y} = \alpha^*_{z,x} ; \alpha_{z,y}$.

We now show (1). Recall that $\psi_x = \langle \alpha^*_{z,x} \rangle \psi_z$. Notice that in this case $R_x = R_z$. Write $R := R_x$.

Note that by the inductive hypothesis on z there exists, for every $v \in R$, a proof of $\vdash \mathsf{S}^l_v \mid \psi^u_z$ and a right-focused proof of $\mathsf{S}^l_v \mid \langle \alpha_{z,x} \rangle q^f_x$ with assumptions from the set $\{(\mathsf{S}^l_w \mid \langle \alpha_{z,x} \rangle q^f_x) \mid w \in R\}$. Then by Lemma 6.5.5 with $\mathcal{S} = \{\mathsf{S}^l_w : w \in R_z\}$, $\mathcal{A} = \varnothing$, $\alpha = \alpha_{z,x}$, $q = q_x$ and $\chi = \psi_z$ we get that $\vdash \mathsf{S}^l_v \mid \langle \alpha_{z,x} \rangle \psi^u_z$, as required.

Case $k(x) = 1$, x is neither a leaf nor a companion. Then x has a unique child z and we have that $\alpha_{x,y} := \alpha_{z,y}$ and $\psi_x := \psi_z$. Since $\mathcal{A}_x = \mathcal{A}_z$ and $R_x = R_z$, the claim is immediate by the inductive hypothesis.

Case $k(x) = 2$.

Then x has a unique child z and we have that $\alpha_{x,y} = \theta_{\Psi_z}?; \alpha_{z,y}$ and $\psi_x = \langle \theta_{\Psi_z}? \rangle \psi_z$. Since $\Psi_x = \Psi_z$, write $\Psi := \Psi_z = \Psi_x$, and since $\mathcal{A}_x = \mathcal{A}_z$, write $\mathcal{A} := \mathcal{A}_x$. We will prove the case by establishing the following claim through a leaf-to-root inner induction on $v \in R_x$.

`Claim 3`: For all $v \in R_x$, we have that $\vdash \mathsf{S}^l_v \mid \langle \theta_\Psi? \rangle \psi^u_z$ and there exists a right-focused proof of $\mathcal{A} \vdash \mathsf{S}^l_v \mid \langle \theta_\Psi?; \alpha_{z,y} \rangle q^f_y$.

`Proof of Claim 3`: We distinguish the following three subcases:

Subcase 1 If $v \notin C$, then v is an exit, meaning that $v \in C^+ \setminus C$, and thus we have that $\vdash \mathsf{S}^l_v \mid \theta^u_\Psi$ by Lemma 6.5.1. From this, we can obtain the required proofs:

$$\dfrac{\dfrac{\mathsf{S}^l_v \mid \theta^u_\Psi}{\mathsf{S}^l_v \mid \theta^u_\Psi, \psi^u_z}\ \mathsf{weak}^r}{\mathsf{S}^l_v \mid \langle \theta_\Psi? \rangle \psi^u_z}\ \langle ? \rangle^r \qquad \dfrac{\dfrac{\dfrac{\dfrac{\mathsf{S}^l_v \mid \theta^u_\Psi}{\mathsf{S}^l_v \mid \theta^u_\Psi, \langle \alpha_{z,y} \rangle q^u_y}\ \mathsf{weak}^r}{\mathsf{S}^l_v \mid \theta^u_\Psi, \langle \alpha_{z,y} \rangle q^f_y}\ \mathsf{u}^r}{\mathsf{S}^l_v \mid \langle \theta_\Psi? \rangle \langle \alpha_{z,y} \rangle q^f_y}\ \langle ? \rangle^r}{\mathsf{S}^l_v \mid \langle \theta_\Psi?; \alpha_{z,y} \rangle q^f_y}\ \langle ; \rangle^r$$

Notice that the proof of $\vdash \mathsf{S}^l_v \mid \langle \theta_\Psi?; \alpha_{z,y} \rangle q^f_y$ is right-focused, as it does not have any assumptions.

Subcase 2 If $v \in C^r_\Psi$, then by the outer inductive hypothesis, we get that $\vdash \mathsf{S}^l_v \mid \psi^u_z$ and there exists a right-focused proof of $\mathcal{A} \vdash \mathsf{S}^l_v \mid \langle \alpha_{z,y} \rangle q^f_y$. From this, we can construct the required proofs as follows:

$$\dfrac{\dfrac{\mathsf{S}^l_v \mid \psi^u_z}{\mathsf{S}^l_v \mid \theta^u_\Psi, \psi^u_z}\ \mathsf{weak}^r}{\mathsf{S}^l_v \mid \langle \theta_\Psi ? \rangle \psi^u_z}\ \langle ? \rangle^r \qquad \dfrac{\dfrac{\dfrac{\begin{matrix}\mathcal{A}\\ \vdots \\ \mathsf{S}^l_v \mid \langle \alpha_{z,y} \rangle q^f_y\end{matrix}}{\mathsf{S}^l_v \mid \theta^u_\Psi, \langle \alpha_{z,y} \rangle q^f_y}\ \mathsf{weak}^r}{\mathsf{S}^l_v \mid \langle \theta_\Psi ? \rangle \langle \alpha_{z,y} \rangle q^f_y}\ \langle ? \rangle^r}{\mathsf{S}^l_v \mid \langle \theta_\Psi ?; \alpha_{z,y} \rangle q^f_y}\ \langle ; \rangle^r$$

Notice that, since the proof of $\mathcal{A} \vdash \mathsf{S}^l_v \mid \langle \alpha_{z,y} \rangle q^f_y$ is right-focused, the proof of $\mathcal{A} \vdash \mathsf{S}^l_v \mid \langle \theta_\Psi ?; \alpha_{z,y} \rangle q^f_y$ is right-focused as well.

Subcase 3 If $v \in C^l_\Psi$, let $u_1, \ldots, u_n$ list the children of v in C^+. Since a left rule was applied, we have that $\Psi_{u_i} = \Psi$ for all $u_i \in C^+_\Psi = R_x$. Hence, by the inner inductive hypothesis, it holds for $i = 1, ..., n$ that

$$\vdash \mathsf{S}^l_{u_i} \mid \langle \theta_\Psi ? \rangle \psi^u_z, \quad \text{and} \quad \mathcal{A} \vdash \mathsf{S}^l_{u_i} \mid \langle \theta_\Psi ?; \alpha_{z,y} \rangle q^f_y.$$

By an application of the same left rule that was applied at v, the claim follows.

This finishes the proof of Claim 3 and, hence, that of the case. ⊣

Case $k(x) = 3$, Ψ_x is not modal. In this case, x has $n > 0$ children $z_1, \ldots, z_n$ in Q. We need to show that $\vdash \mathsf{S}^l_v \mid \psi_{z_1} \vee \ldots \vee \psi^u_{z_n}$, and that there exists a right-focused proof of

$$\mathcal{A}_y \vdash \mathsf{S}^l_v \mid \langle \bigcup \{\alpha_{z,y} \mid x \lessdot_{\mathsf{Q}} z, y \in \mathsf{K}_{<z}\} \rangle q^f_y.$$

Here, recall that $\bigcup \{\alpha_{z,y} \mid x \lessdot_{\mathsf{Q}} z, y \in \mathsf{K}_{<z}\}$ represents the choice program formed by combining all the programs $\alpha_{z,y}$ such that $x \lessdot_{\mathsf{Q}} z$ and $y \in \mathsf{K}_{<z}$.

By Lemma 6.4.15, the children of v can be listed as $t_1, \ldots, t_n$ such that $t_i \in R_{z_i}$ for all $i = 1, ..., n$. Since $\mathsf{S}^l_v = \mathsf{S}^l_{t_i}$, we will write $\mathsf{S}^l := \mathsf{S}^l_v$.

First, we will show that $\vdash \mathsf{S}^l \mid \psi_{z_1} \vee \ldots \vee \psi^u_{z_n}$. By the inductive hypothesis, for each i, we have that $\vdash \mathsf{S}^l \mid \psi^u_{z_i}$. Using the $\vee^r$ rule repeatedly, we can construct the following proof:

$$\dfrac{\begin{matrix}\dfrac{\mathsf{S}^l \mid \psi^u_{z_1} \qquad \mathsf{S}^l \mid \psi^u_{z_2}}{\mathsf{S}^l \mid \psi_{z_1} \vee \psi^u_{z_2}}\ \vee^r \\ \vdots \\ \mathsf{S}^l \mid \psi_{z_1} \vee \ldots \vee \psi^u_{z_{n-1}}\end{matrix} \qquad \mathsf{S}^l \mid \psi^u_{z_n}}{\mathsf{S}^l \mid \psi_{z_1} \vee \ldots \vee \psi^u_{z_n}}\ \vee^r$$

Now we construct a right-focused proof of $\vdash \mathsf{S}^l \mid \langle \bigcup\{\alpha_{z,y} \mid x \lessdot_{\mathsf{Q}} z, y \in \mathsf{K}_{<z}\}\rangle q_y^f$

Notice that for all children z of x, whenever $y \in \mathsf{K}_{<z}$, there exists a right-focused proof $\pi_{z,y}$ of $\mathcal{A}_z \vdash \mathsf{S}^l \mid \langle \alpha_{z,y}\rangle q_y^f$ by the inductive hypothesis. By a repeated application of $\langle \cup \rangle^l$ to every $\pi_{z,y}$ such that $y \in \mathsf{K}_{<z}$, we can obtain a proof ρ of $\mathsf{S}^l \mid \langle \bigcup_{z \in \mathsf{K}_{<x}} \alpha_{z,y}\rangle q_y^f$ with assumptions $\bigcup\{\mathcal{A}_z \mid x \lessdot_{\mathsf{Q}} z, y \in \mathsf{K}_{<z}\}$.

By Lemma 6.4.14, we know that $\bigcup\{\mathsf{K}_{<z} \mid x \lessdot_{\mathsf{Q}} z\} = \mathsf{K}_{<x}$. From this it is straightforward to see that $\bigcup\{\mathcal{A}_z \mid x \lessdot_{\mathsf{Q}} z, y \in \mathsf{K}_{<z}\} = \mathcal{A}_y$. Therefore we can conclude that ρ is a proof of $\mathsf{S}^l \mid \langle \bigcup_{z \in \mathsf{K}_{<x}} \alpha_{z,y}\rangle q_y^f$ with assumptions $\mathcal{A}_y$. This completes the proof of this case.

Case $k(x) = 3$, Ψ_x is modal. Then x has a unique child z and we have $\alpha_{x,y} = a; \alpha_{z,y}$ and $\psi_x = \langle a\rangle\psi_z$, for some atomic program a. By Lemma 6.4.14 we have that $\mathsf{K}_{<x} = \mathsf{K}_{<z}$, and thus $\mathcal{A}_x = \mathcal{A}_z$. Write $\mathcal{A} := \mathcal{A}_x$.

Let $v \in C^r_{\Psi_x}$. By Lemma 6.4.15, the unique child v' of v satisfies $v' \in R_z$. By the inductive hypothesis we have $\mathcal{A} \vdash \mathsf{S}^l_{v'} \mid \langle \alpha_{z,y}\rangle q_y^f$ and $\vdash \mathsf{S}^l_{v'} \mid \psi_z^u$. Then we can construct the required proofs as follows:

$$\dfrac{\dfrac{\dfrac{\mathsf{S}^l_{v'} \mid \psi_z^u}{\mathsf{S}^l_{v'} \mid \psi_z^f}\,\mathsf{u}}{\mathsf{S}^l_v \mid \langle a\rangle\psi_z^f}\,\langle a\rangle^r}{\mathsf{S}^l_v \mid \langle a\rangle\psi_z^u}\,\mathsf{f} \qquad \dfrac{\dfrac{\begin{matrix}\mathcal{A}\\ \vdots \\ \mathsf{S}^l_{v'} \mid \langle \alpha_{z,y}\rangle q_y^f\end{matrix}}{\mathsf{S}^l_v \mid \langle a\rangle\langle \alpha_{z,y}\rangle q_y^f}\,\langle a\rangle^r}{\mathsf{S}^l_v \mid \langle a; \alpha_{z,y}\rangle q_y^f}\,\langle ;\rangle^r$$

Observe that the proof of $\mathsf{S}^l_v \mid \langle a; \alpha_{z,y}\rangle q_y^f$ is right-focused. This follows directly from the inductive hypothesis, as the proof of $\mathsf{S}^l_{v'} \mid \langle \alpha_{z,y}\rangle q_y^f$ is right-focused, and the applied rules preserve that property.

This finishes the proof of Claim 2. $\dashv$

Now consider the root r_{Q} of Q. Since we have $\mathsf{K}_{<r_{\mathsf{Q}}} = \varnothing$, Claim 2 yields that $\vdash \mathsf{S}^l_v \mid \psi^u_{r_{\mathsf{Q}}}$ for all $v \in R_{r_{\mathsf{Q}}}$. In particular, the root r of the cluster C belongs to $R_{r_{\mathsf{Q}}}$ and, since $\mathsf{S}^l_r = \Gamma$, we find that $\vdash \Gamma \mid \psi^u_{r_{\mathsf{Q}}}$. Finally, unravelling the definitions we find that $\theta_r = \iota_{r_{\mathsf{Q}}}$, and, again since $\mathsf{K}_{<r_{\mathsf{Q}}} = \varnothing$, that $\iota_{r_{\mathsf{Q}}} = \psi_{r_{\mathsf{Q}}} \vee \bot$. But then from $\vdash \Gamma \mid \psi^u_{r_{\mathsf{Q}}}$ we easily obtain $\vdash \Gamma \mid \theta_r$, as required.

$\square$

6.5.3 Proof of third condition: $\overline{\theta_r} \mid \Xi$ is unsatisfiable

The third interpolation condition states that the split sequent $\overline{\theta_r} \mid \Xi$ is unsatisfiable. We will show this by providing an actual derivation as well, but here we use the unrestricted cut rule. We let $\vdash^c$ denote derivability in the version of sCPDL_f where we allow the unrestricted cut rules cut^l and cut^r.

Before providing the proof that $\vdash^c \overline{\theta_r} \mid \Xi$, we first state the following definition and lemma, with the purpose of simplifying the proofs.

6.5.7. Definition. Let x be a node in Q, and let $\rho : \mathcal{A} \vdash^c \Sigma^l \mid \Sigma^r$ be a proof with assumptions. We say that ρ is (Q, x)-*shaped* if $\mathcal{A} = \{(\overline{q_y}^u \mid \Psi_y) \mid y \in \mathsf{K}_{<x}\}$, $\Sigma^r = \Psi_x$ and for every open leaf ℓ of ρ labeled with an assumption $\overline{q_y}^u \mid \Psi_y$ there is a repeat z in Q with $c(z) = y$, and such that the list of formulas in focus on the Q-path from x to z is, up to repetitions, equal to the list of formulas in focus on the path in ρ from the root to ℓ.

6.5.8. Lemma. *Let φ and ψ be equivalent* CPDL *formulas. Then we can transform any (Q, x)-shaped proof $\rho : \mathcal{A} \vdash^c \varphi^u \mid \Delta$ into a (Q, x)-shaped proof $\rho' : \mathcal{A} \vdash^c \psi^u \mid \Delta$.*

Proof:
By completeness, there is a proof σ of the split sequent $\overline{\varphi}^u, \psi^u \mid \Delta$. Using this, we construct the desired proof ρ' as follows:

$$\dfrac{\begin{matrix}\mathcal{A}\\ \vdots\,\rho \\ \varphi^u \mid \Delta\end{matrix} \qquad \begin{matrix}\vdots\,\sigma \\ \overline{\varphi}^u, \psi^u \mid \Delta\end{matrix}}{\psi^u \mid \Delta}\ \mathsf{cut}^l$$

It is easy to verify that ρ' is (Q, x)-shaped if ρ is so. This concludes the proof. □

6.5.9. Lemma. $\vdash^c \overline{\theta_r} \mid \Xi$.

Proof:
By a leaf-to-root induction on Q we will prove the following claim, where we write $\mathcal{B}_x := \{(\overline{q_y}^u \mid \Psi_y) \mid y \in \mathsf{K}_{<x}\}$.

`Claim 4`: For every $x \in Q$ there is a (Q, x)-shaped proof $\pi_x : \mathcal{B}_x \vdash^c \overline{\iota_x}^u \mid \Psi_x$.

`Proof of Claim 4:`

Case $k(x) = 1$, x is a repeat. Here we have $\mathsf{K}_{<x} = \varnothing$ and thus $\mathcal{B}_x = \{(\overline{q_{c(x)}}^u \mid \Psi_x)\}$. By definition, $\iota_x = \psi_x \vee \langle \alpha_{x,c(x)} \rangle q_{c(x)}$, where $\alpha_{x,c(x)} = \top?$ and $\psi_x = \bot$. It is then straightforward to find a (cut-free) (Q, x)-shaped proof $\pi_x : \mathcal{B}_x \vdash \overline{\iota_x}^u \mid \Psi_x$.

Case $k(x) = 1$, x an exit. By definition we have $\iota_x = \theta_{\Psi_x}$ and by Lemma 6.5.1 that $\vdash \overline{\theta_{\Psi_x}}^u \mid \Psi_x$.

Case $k(x) = 1$, x is a companion.

Here x has a unique child z, for which we have $\Psi_x = \Psi_z$; we will write $\Psi := \Psi_x$. Furthermore recall that by the definition of pre-interpolants we have $\iota_x \equiv \iota_z[\iota_x/q_x]$. By completeness this means that there is some proof $\rho : \vdash \overline{\iota_x}^u, \iota_z[\iota_x/q_x]^u \mid \Psi$.

By the inductive hypothesis we have a (Q, z)-shaped proof $\pi_z : \mathcal{B}_x \cup \{\overline{q_x}^u \mid \Psi\} \vdash^c \overline{\iota_z}^u \mid \Psi$. Substituting q_x with ι_x everywhere in π_z we obtain a proof $\pi_z[\iota_x/q_x]$ which we may cut with ρ to obtain the following proof π_x with assumptions:

$$\dfrac{\begin{array}{c}\mathcal{B}_x \cup \{\overline{\iota_x}^u \mid \Psi\} \\ \vdots\ \pi_z[\iota_x/q_x] \\ \overline{\iota_z}[\iota_x/q_x]^u \mid \Psi\end{array} \qquad \begin{array}{c}\vdots\ \rho \\ \iota_z[\iota_x/q_x]^u, \overline{\iota_x}^u \mid \Psi\end{array}}{\overline{\iota_x}^u \mid \Psi}\ \mathsf{cut}^l$$

Note that all paths from the root of π_x to assumptions of the form $\overline{\iota_x}^u \mid \Psi$ are successful, because π_z is (Q, x)-shaped and Lemma 6.4.14(7). Therefore all assumptions $\overline{\iota_x}^u \mid \Psi$ in π_x are discharged. This implies that the proof $\pi_x : \mathcal{B}_x \vdash^c \overline{\iota_x}^u \mid \Psi$ is (Q, x)-shaped.

Case $k(x) = 1$, x is neither a leaf nor a companion. In this case x has a unique child z, for which we have $\alpha_{x,y} = \alpha_{z,y}$, $\psi_x = \psi_z$ and thus $\iota_x = \iota_z$. Furthermore we have $\Psi_z = \Psi_x$ and $\mathcal{B}_z = \mathcal{B}_x$. Since $R_x = R_z$, the inductive hypothesis directly applies to z, providing us with a (Q, z)-shaped proof $\pi_z : \mathcal{B}_z \vdash^c \overline{\iota_z}^u \mid \Psi_z$. We may now simply take $\pi_x := \pi_z$.

Case $k(x) = 2$. In this case x has a unique child z, for which we have $\Psi_x = \Psi_z$ and $\mathcal{B}_x = \mathcal{B}_z$. Write $\mathcal{B} := \mathcal{B}_x$ and $\Psi := \Psi_x$. By the inductive hypothesis we have a (Q, z)-shaped proof $\pi_z : \mathcal{B} \vdash^c \overline{\iota_z}^u \mid \Psi$, and by Lemma 6.5.1 we have $\vdash \overline{\theta_\Psi}^u \mid \Psi$.

Applying the $[?]^l$ rule we obtain a (Q, z)-shaped proof witnessing that $\mathcal{B} \vdash^c [\theta_\Psi ?]\overline{\iota_z}^u \mid \Delta$. It is straightforward to verify that $[\theta_\Psi ?]\overline{\iota_z} \equiv \overline{\iota_x}$. But then by Lemma 6.5.8 we obtain the desired proof $\pi_x : \mathcal{B} \vdash^c \overline{\iota_x}^u \mid \Psi$.

Case $k(x) = 3$, Ψ_x is not modal. In this case x has $n > 0$ children $z_1, \ldots, z_n$ in Q. Recall that by the uniformity of π, the same right rule R is applied at each $v \in C^r_{\Psi_x} = R_x$.

By the inductive hypothesis, for each z_i, we have a (Q, z_i)-shaped proof of $\mathcal{B}_{z_i} \vdash^c \overline{\iota_{z_i}}^u \mid \Psi_{z_i}$. By repeated applications of the rules weak^l and $\wedge^l$, we obtain, for each z_i, a (Q, z_i)-shaped proof of $\pi_{z_i} : \mathcal{B}_{z_i} \vdash^c \bigwedge_{1\leq i\leq n} \overline{\iota_{z_i}}^u \mid \Psi_{z_i}$. By an application of the rule R we obtain a (Q, x)-shaped proof π_x:

$$\bigcup_{1\leq i\leq n} \mathcal{B}_{z_i} \vdash^c \bigwedge_{1\leq i\leq n} \overline{\iota_{z_i}}^u \mid \Psi_x$$

By Lemma 6.4.14, it follows that $\bigcup_{1\leq i\leq n} \mathcal{B}_{z_i} = \mathcal{B}_x$ and thus, that $\pi_x : \mathcal{B}_x \vdash^c \bigwedge_{1\leq i\leq n} \overline{\iota_{z_i}}^u \mid \Psi_x$.

It is easy to see that $\bigwedge_{1\leq i\leq n} \overline{\iota_{z_i}} \equiv \overline{\iota_x}$, and therefore Lemma 6.5.8, concludes the case.

Case $k(x) = 3$, Ψ_x is modal. In this case x has a unique child z, for which we have $\iota_x \equiv \langle a\rangle\iota_z$ and $\mathcal{B}_x = \mathcal{B}_z$. Write $\mathcal{B} = \mathcal{B}_x$. By the inductive hypothesis there exists a (Q, z)-shaped proof of $\mathcal{B} \vdash^c \overline{\iota_z}^u \mid \Psi_z$. By an application of the rule $\langle a\rangle^r$ to the formula in focus in Ψ_z we obtain a (Q, x)-shaped proof $\mathcal{B} \vdash^c [a]\overline{\iota_z}^u \mid \Psi_x$. But since we have $[a]\overline{\iota_z} \equiv \overline{\iota_x}$, we may use Lemma 6.5.8 to transform this proof into a (Q, x)-shaped proof of $\mathcal{B} \vdash^c \overline{\iota_x}^u \mid \Psi_x$, as required.

This finishes the proof of the Claim. ⊣

To finish the proof of Lemma 6.5.9, for the root r_{Q} of Q, Claim 4 yields that $\mathcal{B}_{r_{\mathsf{Q}}} \vdash^c \overline{\iota_{r_{\mathsf{Q}}}}^u \mid \Psi_{r_{\mathsf{Q}}}$. But since $\mathsf{K}_{<r_{\mathsf{Q}}} = \varnothing$, we find $\mathcal{B}_{r_{\mathsf{Q}}} = \varnothing$, and as $\Psi_{r_{\mathsf{Q}}} = \Xi$ and $\theta_r = \iota_{r_{\mathsf{Q}}}$, we may conclude that $\vdash^c \overline{\theta_r} \mid \Xi$, as required. □

6.6 Conclusion

We presented a sound and complete cyclic proof system for CPDL and used it to show that the logic enjoys the Craig interpolation property. As a corollary, we established that CPDL also has the Beth definability property.

We sketch how this approach can be adapted to show interpolation for PDL as well. First, we define the proof system PDL_f for PDL. The cyclic system PDL_f is defined as CPDL_f where the acut-rule is removed and the modal rule is replaced by the standard rule for (one-way) modal logic:

$$\langle a\rangle\colon \frac{\varphi^f, \Sigma}{\langle a\rangle\varphi^f, [a]\Sigma, \Gamma}$$

The split system for PDL_f can be defined analogously as for CPDL_f. We then verify that, up to minor adaptations (that are in fact simplifications), the soundness and completeness proof still holds. In particular, the use of the acut-rule in the completeness proof is required only for handling backwards modalities and can thus be omitted for PDL_f. Finally, we observe that the current definition of the interpolant will not involve the use of the converse modality, and check that the correctness proofs for the interpolant can be adapted to PDL_f.

In Chapter 7 we show cut elimination for a cyclic proof system for the alternation-free modal μ-calculus. In the conclusion to that chapter, we sketch how this method can be adapted to also apply to the system PDL_f sketched above. As we included the cut-rule to our system in the proof of correctness of the interpolant, we could use this cut-elimination result to obtain a purely proof-theoretic proof of correctness inside PDL_f without adding the cut-rule.

It would be interesting whether for CPDL the correctness of the interpolant can be proved inside CPDL_f as well; this, however, seems more challenging. One possibility would be to extend CPDL_f with admissible rules so that the proof remains inside the system, without requiring the addition of unrestricted cut.

Another open question is whether our method can be extended to other variants of PDL such as PDL with intersection [Lut05] or deterministic PDL [BHP82].

Chapter 7

Cut elimination for the alternation-free modal μ-calculus

Since the introduction of sequent calculi, cut elimination has been the backbone of proof theory; or as Girard [Gir95] puts it: "A sequent calculus without cut elimination is like a car without engine". However, for cyclic proof systems cut-elimination methods are underdeveloped. In this chapter, we show how to utilize annotations to prove cut elimination for a cyclic proof system for the alternation-free modal μ-calculus.

In the context of finitary systems, cut elimination is usually proved following the approach of Gentzen's seminal proof for first-order logic [Gen35]: First, an application of cut is pushed upwards by permuting rules until the cut formula is principal in both premises. Then, a cut reduction is applied, reducing the complexity of the cut formula. This process is continued inductively until both premises of the cut are instances of axioms, in which case the cut can be omitted.

With the growing popularity of non-wellfounded proofs, it is not surprising that cut elimination has been investigated across a range of infinitary proof systems [FS13; SS20; Sha25; ACG24; MSZ24; BDS16; BDKS22; Sau23; BS25; ALM25; DP18]. At a high level, these approaches follow a similar method: first, cuts are pushed upwards as in the finitary case. Because proof branches may be infinite, it is shown that a cut-free derivation can be obtained as the limit of this procedure. In the second step, this derivation is shown to satisfy the global soundness condition and therefore constitutes an actual proof.

When it comes to cyclic proof systems, cut-elimination procedures that directly produce cut-free cyclic proofs are rare. Although it is possible to unfold cyclic proofs into infinitary ones – thereby allowing the application of the aforementioned two-step approach – the resulting structure may not necessarily be a regular tree. Consequently, it may not be possible to readily obtain a cyclic proof from it. Only for those calculi whose cyclic fragment does exhaust all validities one may invoke other machinery, such as automata, to find a cut-free cyclic proof. To the best of our knowledge, only the cut-elimination method of [AK24] operates directly on cyclic proofs.

In this chapter, we extend the method of [AK24] to apply to the cyclic proof system Focus for the alternation-free modal μ-calculus $\mathcal{L}_\mu^{af}$ introduced by Marti and Venema [MV21a]. The Focus system is path-based and formulas are annotated with very simple kinds of annotations: formulas are either in focus or out of focus. Compared to existing work, our result is noteworthy for the following reasons.

1. **Directness** Our method applies to a cyclic proof and outputs a cyclic cut-free proof without appealing to intermediate machinery for regularising the end proof. Working on the cyclic proof allows us to employ induction invariants utilising the structure of cyclic proof trees and to eliminate cuts depending on where they are located.

2. **Expressiveness** Many of the studies on cut elimination for non-wellfounded proofs [FS13; DP18; ACG24; HSS25] deal with systems with very simple forms of global soundness conditions. Regarding fragments of the modal μ-calculus, such methods have been developed for Grz [SS20; MSZ24] and modal logic with transitive closure [AK24; Sha25]. Here we address a system with a more complex global soundness condition for a larger fragment of the modal μ-calculus.

3. **Transparency** Cut-elimination procedures for systems with complex global soundness conditions have so far been developed primarily in the context of linear logic [BDS16; BDKS22; Sau23]. In [BS25], Bauer and Saurin extend this line of work to the modal μ-calculus by encoding modalities in linear logic via super-exponentials. In contrast, our approach avoids any detours through other proof systems. This is preferable from a practical, as well as from a theoretical point of view, as it provides a more transparent explanation of why cut elimination holds in a certain system.

Our proof strategy may be presented as an extension of reductive cut elimination for cyclic proofs. As in the finitary case, we push the cuts upwards until the premises of the cuts are leaves. Yet, for cyclic proofs, those leaves may not be axiomatic, but may instead be discharged leaves. The main question we need to address is what happens when the premise of a cut is a discharged leaf, and correspondingly, what happens when a premise of a cut is a companion node.

In order to tackle these questions, we make essential use of the structure of cyclic proofs and the annotations on formulas. We distinguish between cuts inside cycles, which we call *unimportant*, and cuts outside cycles, referred to as *important*, and handle them differently. We show that the cut formulas of unimportant cuts do not interfere with the global soundness condition. Consequently, such cuts can be pushed upwards, away from the root, allowing successful repeats to be identified below them.

The treatment of important cuts is more intricate. Our approach builds on the strategy developed for modal logic with the eventually operator[1] in [AK24]: Let π_L and π_R be the left and right subproofs rooted at the respective left and right premise of an important cut. The key idea is to push the important cut upwards while retaining annotations on formulas in π_L and removing annotations on formulas in π_R. Progress on repeat paths in π_L is preserved in the resulting proof, enabling the identification of successful repeats below the cuts.

For the alternation-free μ-calculus this procedure becomes more involved, particularly because conjunctions and disjunctions may occur in the scope of fixpoints. To handle this complexity, we introduce *multicuts*. This, however, further complicates the elimination of important cuts, as it requires determining which premises of a multicut should retain annotations on their formulas.

The introduction of multicuts requires working in a system where sequents are defined over multisets of formulas. Consequently, an explicit *contraction* rule becomes necessary, which in turn poses additional challenges for the elimination of important cuts. To overcome these challenges, we first eliminate contractions from cut-free proofs, where we establish termination of this procedure using known results on well-quasi-orders. This reduces the problem to eliminating important cuts in proofs without contractions.

Overview of the chapter In Section 7.1 we state some facts about multisets and well-quasi-orders that are used later on. The cyclic proof system Focus is defined in Section 7.2. In Section 7.3, we lay the groundwork for the cut-elimination procedure: we provide a high-level overview of the setup, introduce the notions of important and unimportant cuts, and define a normal form for Focus proofs. We deal with important cuts in Section 7.4, with unimportant cuts in Section 7.5 and eliminate contractions in Section 7.6. Results of these sections are combined in the proof of the cut-elimination theorem in Section 7.7. In Section 7.8 we discuss possible directions for further work.

7.1 Mathematical preliminaries

Before we introduce the proof system, we need some definitions. Differently to all other proof systems in this thesis, sequents in our proof system will consist of *multisets* of formulas. As we define various procedures manipulating sequents, they require a clear definition. Termination of our procedure for eliminating contractions from proofs relies on known results about *well-quasi-orders*, which we introduce in Subsection 7.1.2.

[1]This operator is equivalent to the master modality, see for instance [Roo21].

7.1.1 Multisets

We define multisets slightly differently than usual. However, all intuitions about multisets remain the same.

Let X be a set. A *multiset over* X is a set of indexed elements, meaning that it consists of pairs (x, n), where $x \in X$ and $n \geq 1$, such that $(x, n) \in A$ implies $(x, m) \in A$ for all $m = 1, ..., n$. We write $\mathcal{M}_X$ for the set of all finite multisets over X. We only mention indices if they are of importance and otherwise denote a multiset $A = \{(x_1, n_1), ..., (x_k, n_k)\}$ by $[x_1, ..., x_k]$. For simplicity we also sometimes omit the brackets and write $A = x_1, ..., x_k$.

If A is a multiset, we define the *multiplicity* $\sigma_A(x)$ of x in A as the maximal n such that $(x, n) \in A$ and define it to be 0 if no such n exists. This definition agrees with the number of occurrences of x in A. For example, we denote the multiset $A = \{(x, 1), (x, 2), (y, 1)\}$ by $[x, x, y]$ and have that $\sigma_A(x) = 2$, $\sigma_A(y) = 1$ and $\sigma_A(z) = 0$ for all $z \notin \{x, y\}$. Note that a multiset A over X is uniquely defined by the function σ_A on all elements in X.

For two multisets A and B over a set X we say that A is a *submultiset* of B, written as $A \subseteq B$, if $\sigma_A(x) \leq \sigma_B(x)$ for all $x \in X$. We define $A_{\mathsf{Set}} := \{a \mid (a, n) \in A \text{ for some } n\}$ for the *underlying set of* A and write $A =_{\mathsf{Set}} B$ if $A_{\mathsf{Set}} = B_{\mathsf{Set}}$. We write A, B for the union of the multisets A and B, defined as expected.

The reason for this choice of definition lies in the need to talk about specific elements x of a multiset A. As A is a set of indexed elements we can then choose (x, n) for some specific n.

Let $(X, <_X)$ be a well-ordered set and $\mathcal{M}_X$ be the set of all finite multisets over X. We define the *Dershowitz-Manna ordering* $<_{\mathrm{DM}}$ on $\mathcal{M}_X$ as follows: Let A, B be in $\mathcal{M}_X$, then $A <_{\mathrm{DM}} B$ iff there exists $x \in X$ such that

1. $\sigma_A(x) < \sigma_B(x)$ and
2. for all $y >_X x$ it holds $\sigma_A(y) = \sigma_B(y)$.

The Dershowitz-Manna ordering was introduced in [DM79], where it was also shown to be wellfounded.

7.1.1. Proposition. *Let $(X, <_X)$ be a well-ordered set and $\mathcal{M}_X$ be the set of all finite multisets over X. Then $(\mathcal{M}_X, <_{\mathrm{DM}})$ is a well-order.*

7.1.2 Well-quasi-orders

We shortly introduce well-quasi-orders to the extent used in this chapter; for a more extensive treatment we refer to [SSW20] and for examples of applications of well-quasi-orders to proof-theory we refer to [GGRJ25]. We will use well-quasi-orders in Section 7.6 as a tool for showing that the procedure of eliminating contractions terminates. This resembles their use in showing termination of proof search algorithms in substructural logics [GGRJ25].

Let $\mathcal{Q} = (Q, \leq_Q)$ be a *quasi-order*, meaning that $\leq_Q$ is a reflexive and transitive relation on a non-empty set Q. Let $\kappa \leq \omega$. A *bad sequence* of length κ over a quasi-order $\mathcal{Q}$ is a sequence $(q_n)_{n<\kappa}$ such that $q_m \not\leq_Q q_n$ for all $m < n$, . A quasi-order $\mathcal{Q}$ is a *well-quasi-order*, in short *wqo*, if every bad sequence over $\mathcal{Q}$ is finite.

Let $(\mathbb{N}^k, \leq)$ be the set of k-tuples of natural numbers ordered with the natural product order: $(m_1, ..., m_k) \leq (n_1, ..., n_k) :\Leftrightarrow m_i \leq n_i$ for all $i = 1, ..., k$. Clearly, $(\mathbb{N}^k, \leq)$ is a quasi-order. Dicksons's Lemma [Dic13] states that it is in fact a wqo.

7.1.2. Lemma (Dickson's Lemma). *For every $k \in \mathbb{N}$, $(\mathbb{N}^k, \leq)$ is a well-quasi-order.*

Proof:
By induction on k. The base case is trivial. For the inductive step assume that $(\mathbb{N}^k, \leq)$ is a wqo. We need to show that $(\mathbb{N}^{k+1}, \leq)$ is a wqo. Towards a contradiction assume that $(a_n)_{n\in\omega}$ is an infinite sequence of $(k+1)$-ary tuples such that for all $m < n$ it holds $a_m \not\leq a_n$. For $n \in \omega$ let $a_n = (a_n^1, ..., a_n^k, a_n^{k+1})$ and define $b_n := (a_n^1, ..., a_n^k)$ and $s_n := a_n^{k+1}$. Because $(s_n)_{n\in\omega}$ is an infinite sequence of natural numbers there are increasing indices $(n(i))_{i\in\omega}$ such that $s_{n(i)} \leq s_{n(j)}$ for all $i < j$. But then $(b_{n(i)})_{i\in\omega}$ is an infinite sequence of k-tuples such that $b_{n(i)} \not\leq b_{n(j)}$ for all $i < j$. This contradicts the fact that $(\mathbb{N}^k, \leq)$ is a wqo. □

We do not only need the non-existence of infinite bad sequences, but moreover a bound on the length of finite bad sequences. Such a bound may not always be found for wqos; for example consider the wqo $(\mathbb{N}, \leq)$, where we can easily find bad sequences of arbitrary length. We therefore move to the concepts of *normed well-quasi-orders* and *controlled bad sequences.*

7.1.3. Definition. A *normed well-quasi-order*, in short *nwqo*, is a triple $\mathcal{Q} = (Q, \leq_Q, [\![\cdot]\!])$, where $(Q, \leq_Q)$ is a wqo and $[\![\cdot]\!] : Q \to \mathbb{N}$ is a *proper norm*, meaning that for every $n \in \mathbb{N}$, the set $\{q \in Q \mid [\![q]\!] \leq n\}$ is finite.

7.1.4. Definition. A *control function* is a map $f : \mathbb{N} \to \mathbb{N}$ that is strictly increasing, that is, $f(m) > f(n)$ for all $m > n$.

Given an nwqo $\mathcal{Q}$, a control function f and $t \in \mathbb{N}$, an *(f,t)-controlled bad sequence over $\mathcal{Q}$* is a bad sequence $(q_n)_{n<\kappa}$ over $\mathcal{Q}$ where[2] $[\![q_n]\!] \leq f^n(t)$ for all $n < \kappa$.

7.1.5. Lemma. *Let $\mathcal{Q}$ be a nwqo, f a control function and $t \in \mathbb{N}$. Then there is a bound on the length of (f,t)-controlled bad sequences over $\mathcal{Q}$.*

[2]Here $f^n(t) = f(\cdots(f(t))\cdots)$ stands for the n-th iterate of f applied to t.

Proof:
The idea is to construct a finitely branching tree T of all possible (f,t)-controlled bad sequences over $\mathcal{Q}$ and then use König's Lemma. The root of T will be unlabeled, and at level 1 we add all elements $q \in Q$ such that $[\![q]\!] \leq t$. Now let $q_1, ..., q_i$ be a path to a node q_i at level i. As children of q_i we add all elements $q_{i+1} \in Q$ such that $[\![q_{i+1}]\!] \leq f^{i+1}(t)$ and such that $q_1, ..., q_i, q_{i+1}$ is a bad sequence over $\mathcal{Q}$. This constructs a tree T of all possible (f,t)-controlled bad sequences over $\mathcal{Q}$. As $[\![\cdot]\!]$ is a proper norm, T is finitely branching. Because Q is a wqo, it does not have an infinite branch. Therefore König's Lemma yields that T is finite. In particular, T has a maximal depth which corresponds to the maximal length of an (f,t)-controlled bad sequences over $\mathcal{Q}$. □

7.1.6. Definition. Given an nwqo $\mathcal{Q}$ and a control function f we define the *length function* $L[\mathcal{Q}, f] : \mathbb{N} \to \mathbb{N}$ that maps each $t \in \mathbb{N}$ to the maximal length of (f,t)-controlled bad sequences over $\mathcal{Q}$.

It is easily verified that the *infinity norm* $[\![\cdot]\!]_\infty : \mathbb{N}_k \to \mathbb{N}$ given by $[\![(n_1, ...n_k)]\!]_\infty := \max\{n_i \mid i = 1, ..., k\}$ is a proper norm on $\mathbb{N}$. We define the nwqo $\mathsf{N}^k := (\mathbb{N}^k, \leq, [\![\cdot]\!]_\infty)$. A thorough investigation of the complexity of $L[\mathsf{N}^k, f]$ can be found in [FFSS11]. We will not go into more detail as we are not dealing with complexity issues in this paper. Let us note though, that for a primitive recursive f and fixed k the function $L[\mathsf{N}^k, f]$ is primitive recursive as well. If k is added as a part of the input, the function $(k,t) \mapsto L[\mathsf{N}^k, f](t)$ is not primitive recursive and its growth is comparable to that of the Ackermann function.

In this chapter we will be working with the following nwqo.

7.1.7. Definition. Let X be a finite set. We let $\mathsf{M}_X := (\mathcal{M}_X, \subseteq, [\![\cdot]\!]_\infty)$ be the nwqo consisting of the set of all multisets over X ordered by inclusion, together with the infinity norm $[\![A]\!]_\infty := \max\{\sigma_A(x) \mid x \in X\}$.

7.1.8. Lemma. *Let $X = \{x_1, ..., x_k\}$. Then M_X is isomorphic to N^k. In particular, M_X is an nwqo.*

Proof:
Consider the map

$$g : \mathcal{M}_X \to \mathbb{N}^k$$
$$A \mapsto (\sigma_A(x_1), ..., \sigma_A(x_k)).$$

Clearly, g is an isomorphism between M_X and N^k. Therefore, $(\mathcal{M}_X, \subseteq)$ is an wqo due to Lemma 7.1.2. Because the infinity norm $[\![A]\!]_\infty$ is a proper norm, M_X is indeed a nwqo. □

7.2 The Focus system

We are now ready to define the cyclic proof system Focus. In this chapter we will simply write *formulas* for guarded, closed and alternation-free formulas in $\mathcal{L}_\mu$. For simplicity, we will not have $\bot$ and $\top$ as primitives in the language. If needed, they can be defined as usual by $p \wedge \overline{p}$ and $p \vee \overline{p}$, respectively. Moreover, we assume that the set of actions Act is a singleton and we denote modalities by $\Box$ and $\Diamond$. This is done solely to avoid syntactic clutter and does not change anything fundamental.

An *annotated formula* is a pair (φ, b), usually denoted as φ^b, where φ is a formula and $b \in \{f, u\}$. We call annotated formulas of the form φ^f *in focus* and of the form φ^u *out of focus*. In this chapter a *sequent* is a finite *multiset* of annotated formulas. We define the following operations on sequents Γ:

$$\Gamma^u := \{\varphi^u \mid \varphi^b \in \Gamma\} \qquad \Gamma^- := \{\varphi \mid \varphi^b \in \Gamma\}$$
$$\Gamma^f := \{\varphi^f \mid \varphi^b \in \Gamma\} \qquad \Diamond\Gamma \ := \{\Diamond\varphi^b \mid \varphi^b \in \Gamma\}$$

We call a sequent *focused* if it contains a formula in focus and *unfocused* otherwise. We read sequents *disjunctively* and aim to prove *validity*; we say that a sequent Γ is valid, if $\bigvee \Gamma^-$ is valid.

Figure 7.1 depicts the rules of the Focus derivation system. Apart from annotations, the axiom Ax1 and the rules $\vee$,$\wedge$,$\Box$, μ, ν and weak are as in NW. Note that in the Focus system sequents are *multisets* of formulas and we therefore have to adjust the precise formulation of the rules to be in accordance with the definition of multisets. For instance, if the premise of a $\wedge$ rule is the multiset $((\varphi \wedge \psi)^b, n), \Gamma$ then its left premise is $(\varphi^b, k), \Gamma'$; where k is the minimal number such that (φ^b, k) is not in Γ, and Γ' is obtained from Γ by replacing $((\varphi \wedge \psi)^b, m)$ with $((\varphi \wedge \psi)^b, m-1)$ for all $m > n$. This ensures that the premise is a well-formed multiset. Analogously for the right premise of $\wedge$ and other rules.

We add the contraction rule contr because, in this setting, sequents are *multisets* rather than *sets* of formulas. In the *focus rules* f and u we can put formulas in focus and out of focus, together with μ these are the only formulas changing annotations. The notions of principal, auxiliary and active formulas are defined as in NW. In the rules f, u and contr there are no principal, auxiliary and active formulas.

The rule D marks repeats as usual; note that a repeat leaf and its companion are labeled with the same *multiset* of formulas. Every occurrence D rule is labeled with a unique discharge token taken from a fixed infinite set Tokens $= \{\dagger, \ddagger, ‡, ...\}$. The rule cut is formulated in a multiplicative way, that is, its context is split across the two premises. Notably, the cut formula is always out of focus. It will be the goal of this chapter to show that we can eliminate cut rules.

7.2.1. Definition (Successful path). A path β in a Focus-derivation is called *successful* if

$$\mathsf{Ax1}\colon \frac{}{p^u, \overline{p}^u} \qquad \vee\colon \frac{\varphi^b, \psi^b, \Gamma}{(\varphi \vee \psi)^b, \Gamma} \qquad \mu\colon \frac{\varphi[\mu x.\varphi/x]^u, \Gamma}{\mu x.\varphi^b, \Gamma} \qquad \mathsf{weak}\colon \frac{\Gamma}{\varphi^b, \Gamma}$$

$$\Box\colon \frac{\varphi^b, \Gamma}{\Box\varphi^b, \Diamond\Gamma} \qquad \wedge\colon \frac{\varphi^b, \Gamma \qquad \psi^b, \Gamma}{(\varphi \wedge \psi)^b, \Gamma} \qquad \nu\colon \frac{\varphi[\nu x.\varphi/x]^b, \Gamma}{\nu x.\varphi^b, \Gamma} \qquad \mathsf{contr}\colon \frac{\varphi^b, \varphi^b, \Gamma}{\varphi^b, \Gamma}$$

$$\begin{array}{c}[\Gamma]^\dagger \\ \vdots \\ \mathsf{D}_\dagger\colon \dfrac{\Gamma}{\Gamma}\end{array} \qquad \mathsf{f}\colon \frac{\Delta^f, \Gamma}{\Delta^u, \Gamma} \qquad \mathsf{u}\colon \frac{\Delta^u, \Gamma}{\Delta^f, \Gamma} \qquad \mathsf{cut}\colon \frac{\Gamma_0, \varphi^u \qquad \overline{\varphi}^u, \Gamma_1}{\Gamma_0, \Gamma_1}$$

Figure 7.1: Rules of Focus

1. every sequent on β has a formula in focus,
2. there is no application of f on β and
3. β passes through an application of $\Box$.

7.2.2. Definition (Proof). The *cyclic proof system* Focus is path-based and defined from the rules in Figure 7.1 together with all successful paths.

7.2.3. Remark. In spirit, the Focus system is similar to the annotated proof systems for the full modal μ-calculus BT and JS defined in Chapter 4. Recall that these system were obtained by determinizing the tracking automaton for NW, which checks the success of infinite branches. Due to the absence of fixpoint-alternations, the tracking automaton becomes much simpler on NW-proofs of $\mathcal{L}_\mu^{af}$-formulas. The Focus system can be obtained by determinizing this simpler form of tracking automaton.

Note that we make some adaptions to the Focus system compared to the presentation in [MV21a]:

1. Sequents are *multisets* of formulas, compared to sets in [MV21a]. Therefore, we add the contraction rule contr.
2. We change the focus rules f and u to apply to multisets of formulas compared to single formulas.
3. On successful paths we allow u rules.

It can be easily seen that the adaptions 1 and 2 are harmless. Proposition 7.2.4 deals with the third adaption and thus shows the equivalence of the two presentations. Consequently, we obtain Soundness and Completeness from [MV21a].

7.2.4. Proposition. *Let π be a* Focus*-proof. By only adding and deleting focus rules we can obtain a* Focus*-proof π' from π which has no applications of* u *rules on repeat paths.*

Proof:
Let π be a Focus-proof, where u rules might be applied on repeat paths β_v for discharged leaves v. Let π' be the Focus-proof, where all u rules on repeat paths β_v are deleted and focus annotations are inductively propagated upwards in $\mathcal{T}_\pi^C$. As we are only putting formulas in focus, that is, changing formulas φ^u to φ^f, this terminates. It remains to adjust nodes that do not lie on a repeat path β_v for any v, by putting formulas in focus and adjusting f and u rules. This results in a Focus-proof of the same sequent without u rules on repeat paths. □

7.2.5. Theorem (Soundness and Completeness, [MV21a]). *Let Γ be a sequent. Then* Focus $\vdash \Gamma$ *iff* Γ *is valid.*

We end this section with a few definitions that will be of importance later on.

7.2.6. Definition. Let rank be the minimal-valued function from the set of $\mathcal{L}_\mu$-formulas to $\mathbb{N}$, such that

1. $\mathsf{rank}(p) = \mathsf{rank}(\overline{p}) = 1$,
2. $\mathsf{rank}(\varphi) = \mathsf{rank}(\psi)$ if $\varphi \equiv_C \psi$ and
3. $\mathsf{rank}(\varphi) > \mathsf{rank}(\psi)$ if $\varphi \twoheadrightarrow_C \psi$ and $\psi \not\twoheadrightarrow_C \varphi$.

7.2.7. Definition. The *rank* of a cut with cut formula φ is $\mathsf{rank}(\varphi)$. The *cut rank* of a Focus-derivation π is the maximal rank of a cut in π and is 0 if there is no cut in π.

Let π be a Focus-proof. Recall that we call a leaf l in π *outermost*, if $c(l)$ is the root of some proper cluster in π. The *unfolding* of an outermost leaf l in π is the derivation obtained from π by replacing l with the maximal subderivation $\pi_{c(l)}$ of π rooted at $c(l)$.[3]

7.2.8. Definition. Let π be a Focus-proof. Let the root r of π be labeled with $\mathsf{D}_\dagger$. The *unfolding of* π is obtained from π by replacing every discharged leaf labeled with $\dagger$ with π, and removing the node r.[4]

Recall that the *strongly connected subtree* $\mathrm{scst}(u)$ of a companion node u in π is the maximal strongly connected subgraph A of π such that u is the root of A.

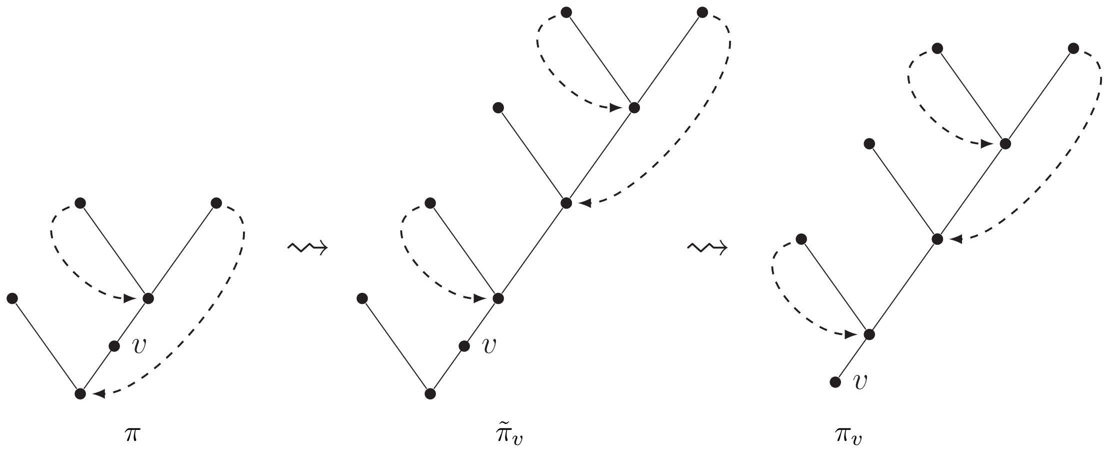
[3] In order to guarantee that D rules are labeled with unique discharge tokens, in $\pi_{c(l)}$ discharge tokens may be replaced by fresh discharge tokens not occurring in π.

[4] We replace discharge tokens $\ddagger$ with fresh discharge tokens, whenever a $\mathsf{D}_\ddagger$ rule is duplicated.

7.2.9. DEFINITION. Let v be a node in a Focus-proof π. We let $\tilde{\pi}_v$ be the Focus-proof obtained from π by recursively unfolding outermost leaves l, where $v \in \mathrm{scst}(c(l))$. The *generated proof from* π *rooted at* v, written as π_v, is the maximal subderivation of $\tilde{\pi}_v$ rooted at v.

Note that $\tilde{\pi}_v$ and π_v are well-defined and that π_v is a Focus-proof. This holds, as for every v the set $\mathsf{L}_{<v} := \{l \text{ leaf in } \pi \mid v \in \mathrm{scst}(c(l))\}$ is finite; one of those leaves $l \in \mathsf{L}_{<v}$ is outermost; and after unfolding l, the size of $\mathsf{L}_{<v}$ gets reduced.

7.2.10. EXAMPLE. Consider the following depiction of a Focus-proof π with indicated node v in the figure on the left. The middle figure shows the proof $\tilde{\pi}_v$ and the figure on the right the generated proof π_v from π rooted at v.



7.3 Cut-elimination strategy

We present a cut-elimination procedure for the Focus-proof system. Our approach builds on the strategy developed for the GKe proof system for modal logic with the eventually operator[5] presented in [AK24]. The method is based on reductive cut elimination adjusted to cyclic proofs. We start with an informal explanation of the cut-elimination strategy.

7.3.1 Main ideas

One way to prove cut elimination for finitary proofs is by first proving *cut admissibility*, in other words eliminating a cut at the root of a proof. In the context of cyclic proofs the notion of cut admissibility has to be extended, such that we first eliminate cuts that are *in the root cluster* – those nodes from which there

[5]This operator is equivalent to the master modality, see for instance [Roo21].

is a path to the root in the proof tree with back edges. If the root cluster only consists of one node we retrieve the usual notion.

Cut admissibility is shown by an induction on the *rank* of the cut formulas, which is a linearisation of the trace relation $\twoheadrightarrow_C$. Importantly $\mathsf{rank}(\varphi) > \mathsf{rank}(\psi)$ if $\varphi \twoheadrightarrow_C \psi$ and $\psi \not\twoheadrightarrow_C \varphi$.

At the core of our strategy is the need to isolate the applications of cut that present the greatest challenges. We thus split applications of cut into two categories: Cuts that are located inside a cycle are called *unimportant* and cuts that are not are called *important*. We reduce unimportant cuts to important ones of the same rank and reduce the rank of important cuts.

As the name suggests, unimportant cuts are easier to deal with. Cut reductions on unimportant cuts do not affect formulas in focus, hence those can be pushed upwards and we can find successful repeats *below* the cuts. All remaining cuts will be important and of the same rank.

The treatment of important cuts is more complicated, as descendants of the cut formula might be in focus. Pushing up those cuts might put formulas out of focus and consequently undermine successful paths. In order to still find successful repeats we use a property of $\mathcal{L}_\mu^{af}$: given any formula $\varphi \in \mathcal{L}_\mu^{af}$, at most one of φ and $\overline{\varphi}$ is navy. (Recall that φ is navy if $\varphi \equiv_C \nu x.\chi$ for some ν-formula $\nu x.\chi$.) Assume that φ is not navy and consider the following important cut:

$$\dfrac{\begin{array}{cc}\pi_0 & \pi_1 \\ \Gamma_0, \varphi & \overline{\varphi}, \Gamma_1\end{array}}{\Gamma_0, \Gamma_1}\ \mathsf{cut}$$

Since φ is not navy, no descendant of φ in π_0 of the same rank is a ν-formula. As we may assume that only navy formulas are in focus, all descendants of φ in π_0 of the same rank are out of focus. We carry on by deleting all descendants of φ of the same rank in π_0 and all descendants of $\overline{\varphi}$ of the same rank in π_1 and "merge" those two proofs. This process is similar to pushing cuts upwards, unfolding cycles whenever necessary and introducing cuts for descendants of φ of lower rank. In the resulting proof ρ we can find successful repeats, as all formulas in focus in π_0 are carried over and therefore successful paths in π_0 are projected to successful paths in ρ.

The main difficulty compared to the system GKe for modal logic with the eventually operator [AK24] are occurrences of conjunctions and disjunctions in the scope of fixpoint operators. Applying cut reductions leads to multiple cut formulas in sequents and multiple sequents connected by cuts. To deal with these situations we employ a multi-cut rule. Because the multi-cut may increase in size one extra difficulty in the termination proof is to show that pushing up multi-cuts is productive.

As it is often the case, *contractions* pose one of the main problems to cut elimination. For finitary proof systems there are two approaches to deal with contractions: In the first approach a generalization of the cut rule is added to the system – the *mix rule*. This rule allows to introduce the cut formula multiple times in the premises of its rule and therefore functions as a combination of cut and contraction. All cut rules can then be replaced by mix rules and henceforth all mix rules are eliminated. In the second approach the contraction rule is first shown to be admissible in the proof system without an explicit contraction rule and then cut rules are eliminated from the system without contractions.

We take inspiration from both of these approaches. In order to eliminate unimportant cuts we introduce a mix rule. The proof is then partitioned into subproofs not containing modal rules – on these finitary subproofs we can eliminate the mix rules as for finitary proofs. Before eliminating important cuts, first the subproofs rooted at the premises of the cut-rule are pre-processed, such that those subproofs do not contain contractions anymore – this elimination of contractions is done in Section 7.6.

In the next subsection we introduce the necessary notions to make the definitions of important and unimportant cuts formal.

7.3.2 Important and unimportant cuts

Let π be a Focus-proof. Recall that $\mathcal{T}_\pi^C$ is the proof tree of π with back-edges. A *cluster* of π is a maximal strongly connected subgraph of $\mathcal{T}_\pi^C$. We call a cluster *trivial* if it consists of only one node and *proper* otherwise. Let $\mathcal{S}_\pi$ be the set of *proper clusters* of π. We define a relation $\twoheadrightarrow_\pi$ on $\mathcal{S}_\pi$ as follows: $S_1 \twoheadrightarrow_\pi S_2$ if $S_1 \neq S_2$ and there are nodes $v_1 \in S_1, v_2 \in S_2$ such that there is a path from v_1 to v_2 in $\mathcal{T}_\pi^C$. The relation $\twoheadrightarrow_\pi$ is a strict partial order. We write $\mathsf{depth}(S)$ for the length of the longest path in $(\mathcal{S}_\pi, \twoheadrightarrow_\pi)$ starting from the cluster S.

For a node v in a proof π, we define the *depth* of v to be

$$\mathsf{depth}(v) = \max\{\mathsf{depth}(S) \mid S \in \mathcal{S}_\pi \text{ and there is a path from } v \text{ to some } u \in S\}$$

where $\max \varnothing = 0$. In words, $\mathsf{depth}(v)$ is the maximal n such that there is a path $S_1 \twoheadrightarrow_\pi \cdots \twoheadrightarrow_\pi S_n$ in $\mathcal{S}_\pi$ for which S_1 is reachable from v. The *depth* of a proof is defined as the depth of its root.

Recall that two nodes u and v are connected in a graph (G, E), if there is a $E \cup \breve{E}$-path from u to v, where $\breve{E}$ is the converse relation of E. A *component* of π is a maximal connected set of nodes in $\mathcal{T}_\pi^C$ of the same depth. Note that all nodes in the same cluster are connected and have the same depth. Therefore, all nodes of one cluster belong to the same component. It follows that a component of π can be partitioned into clusters of π of the same depth.

The proof π itself can be partitioned into its components. Thus every node v in π belongs to a unique component, which we denote by $\mathsf{comp}(v)$. The component of the root is called the *root component* and the cluster of the root is called the *root cluster*. We call a descendant w of v a *component descendant*, if $w \in \mathsf{comp}(v)$.

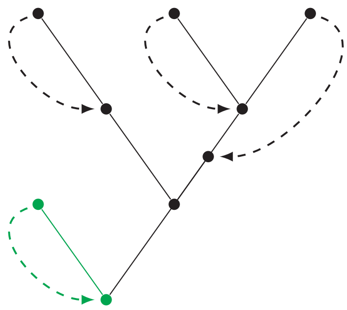


Figure 7.3: A proof of depth 2. It can be partitioned into two components: the root component (colored green) which coincides with the root cluster and has depth 2; and another component (the rest of the proof) of depth 1 that contains two proper clusters.

7.3.1. Definition (Important cut). Let C be an occurrence of a cut rule at a node v in a Focus-proof π. We call C *important* if v is in a trivial cluster of π and *unimportant* otherwise.

7.3.3 Minimally focused proofs

In the operations we perform on proof-trees we need a good handle on the shape of the proof-trees we are dealing with. We therefore introduce a normal form on proofs that aligns proper clusters with sequents that have formulas in focus.

Any node v in a proper cluster of a Focus-proof π has formulas in focus, as it is on the path β_l of a discharged leaf l to its companion. For nodes in a trivial cluster this is not necessarily the case. We can rearrange f and u rules in a certain way to minimize the number of nodes with formulas in focus. By doing so, nodes with formulas in focus resemble the proper clusters of the proof tree with back edges: Any node with formulas in focus either belongs to a proper cluster, or it is the child of a node in a proper cluster and itself labeled with u.

Moreover, we can minimize the number of focused formulas at every node in a cluster. Without loss of generality, we may also assume that all formulas in focus are navy and of the same rank. This can be ensured by only focusing navy formulas of the same rank when an f rule is applied, and applying a u rule whenever a focused formula of lower rank appears.

7.3.2. Definition (Minimally focused). A Focus-proof is called *minimally focused* if the following conditions are satisfied:

1. if v is labeled with f, then its child is labeled with D;

2. if $\mathsf{depth}(v') < \mathsf{depth}(v)$ for a child v' of v, and v' contains formulas in focus, then v' is labeled with u, and all formulas in its premise are out of focus. These are the only applications of u in trivial clusters;

3. in any rule application of f all formulas in Δ are navy formulas with the same rank;

4. for any node v in a proper cluster S, where k is the maximal rank of a formula in focus in S: If v is labeled with Γ, φ^f where $\mathsf{rank}(\varphi) < k$, then v is labeled with u with premise Γ, φ^u. These are the only applications of u in proper clusters.

7.3.3. LEMMA. *Let π be a* Focus*-proof. Then we can obtain a minimally focused* Focus*-proof π' from π by only rearranging focus rules and changing annotations.*

Proof:
Annotations only matter on repeat paths. Therefore we may employ focus rules in such a way, that as few nodes in trivial clusters as possible are focused – hence satisfying conditions 1 and 2. Now assume that there is a proper cluster S that does not satisfy 3 or 4. Because of condition 1, we may assume that the parent of the root of S is labeled with an f rule[6], say $\dfrac{\Delta^f, \Gamma}{\Delta^u, \Gamma}\,\mathsf{f}$. Let Δ_m be the submultiset of Δ consisting of all navy formulas in Δ of maximal rank k and let $\Delta_r = \Delta \setminus \Delta_m$. We change the f rule to $\dfrac{\Delta_m^f, \Delta_r, \Gamma}{\Delta_m^u, \Delta_r, \Gamma}\,\mathsf{f}$ and propagate the annotations upwards accordingly, where we apply u rules, whenever formulas of rank lower than k are in focus. It remains to show that discharged leaves remain discharged leaves. A formula in focus of rank k can only originate from a formula in focus of the same rank, as there are no applications of f on repeat paths and k is the maximal rank of formulas in focus. Therefore, all navy formulas of maximal rank in focus in the original proof remain in focus in the adapted proof. Thus it holds that all discharged leaves are translated to discharged leaves, meaning that they are still repeat leaves and that on every sequent on the repeat path there is a formula of rank k in focus. Doing so we satisfy conditions 3 and 4. □

As every proof can be transformed to a minimally focused proof of the same sequent by only rearranging focus rules and changing annotations, we may always implicitly transform Focus-proofs to ones that are minimally focused.

We assume that every Focus-proof is minimally focused.

7.3.4. PROPOSITION. *If π is minimally focused, then an occurrence of* cut *in π is important iff all formulas in the conclusion of the cut are out of focus.*

[6] If S is the root cluster, we may add a u and an f rule at the root.

7.3.4 Cut Reductions

To finish this section, we state the cut reductions that we employ in the following sections. A cut reduction transforms a proof π into another proof such that the *complexity* of the cut at the root of π is reduced. Cut elimination is then proved by an induction on the complexity of cuts. The exact definition of the complexity of a cut depends on the specific proof system and is usually a combination of multiple notions. For finitary proofs, it commonly refers to a combination of the complexity of the cut formula and the height of the proof tree above the cut.

For cyclic proofs, one needs to come up with a different definition. First, because principal reductions with fixpoint formulas increase the complexity of the cut formula (as for instance witnessed in the reduction for η). Second, because the height of a proof tree increases when a discharged leaf is unfolded. We therefore do not aim to decrease the height of the proof above the cut, but to increase the height of the proof *below* the cut: informally, we say that the complexity of a cut is reduced, if

(i) the height of the proof below the cut is increased (this is advantageous because we want to find repeats below the cut), or

(ii) the number of modal rules above the cut is increased (reductions with modal rules do then increase the height below the cut), or

(iii) the number of non-modal rules above the cut is decreased and the number of modal rules above the cut is not decreased (therefore we get closer to a reduction with a modal rule).

In this sense all presented cut reductions reduce the complexity of the cut, where we leave it to the reader to check that this is the case. Note that unfolding discharged leaves always increases the number of modal rules, as successful paths contain modal nodes.

The reader may prefer to skip this part and revisit specific reductions as they arise. For readability, we omit the annotations whenever they are not affected by the cut reductions. Note that the cut formula is always out of focus.

Principal cut reductions

$$\dfrac{\dfrac{\overset{\pi_0}{\Gamma, \varphi} \qquad \overset{\pi_1}{\Gamma, \psi}}{\Gamma, \varphi \wedge \psi}\,\wedge \qquad \dfrac{\overset{\pi_2}{\overline{\varphi}, \overline{\psi}, \Gamma_2}}{\overline{\varphi} \vee \overline{\psi}, \Gamma_2}\,\vee}{\Gamma, \Gamma_2}\,\mathsf{cut} \qquad \longrightarrow \qquad \dfrac{\dfrac{\overset{\pi_0}{\Gamma, \varphi} \qquad \dfrac{\overset{\pi_1}{\Gamma, \psi} \qquad \overset{\pi_2}{\overline{\varphi}, \overline{\psi}, \Gamma_2}}{\overline{\varphi}, \Gamma, \Gamma_2}\,\mathsf{cut}}{\Gamma, \Gamma, \Gamma_2}\,\mathsf{cut}}{\Gamma, \Gamma_2}\,\mathsf{contr}$$

$$\dfrac{\dfrac{\overset{\pi_0}{\Gamma_0, \varphi[\mu x.\varphi/x]}}{\Gamma_0, \mu x.\varphi}\,\mu \qquad \dfrac{\overset{\pi_1}{\overline{\varphi}[\nu x.\overline{\varphi}/x], \Gamma_1}}{\nu x.\overline{\varphi}, \Gamma_1}\,\nu}{\Gamma_0, \Gamma_1}\,\mathsf{cut} \quad\longrightarrow\quad \dfrac{\overset{\pi_0}{\Gamma_0, \varphi[\mu x.\varphi/x]} \qquad \overset{\pi_1}{\overline{\varphi}[\nu x.\overline{\varphi}/x], \Gamma_1}}{\Gamma_0, \Gamma_1}\,\mathsf{cut}$$

$$\dfrac{\dfrac{\overset{\pi_0}{\Gamma_0, \varphi}}{\Diamond\Gamma_0, \Box\varphi}\,\Box \qquad \dfrac{\overset{\pi_1}{\overline{\varphi}, \gamma, \Gamma_1}}{\Diamond\overline{\varphi}, \Box\gamma, \Diamond\Gamma_1}\,\Box}{\Box\gamma, \Diamond\Gamma_0, \Diamond\Gamma_1}\,\mathsf{cut} \quad\longrightarrow\quad \dfrac{\dfrac{\overset{\pi_0}{\Gamma_0, \varphi} \qquad \overset{\pi_1}{\overline{\varphi}, \gamma, \Gamma_1}}{\gamma, \Gamma_0, \Gamma_1}\,\mathsf{cut}}{\Box\gamma, \Diamond\Gamma_0, \Diamond\Gamma_1}\,\Box$$

Trivial principal cut reductions

$$\dfrac{\overset{\pi_0}{\Gamma, p} \qquad \dfrac{}{\overline{p}, p}\,\mathsf{Ax1}}{\Gamma, p}\,\mathsf{cut} \quad\longrightarrow\quad \overset{\pi_0}{\Gamma, p}$$

$$\dfrac{\overset{\pi_0}{\Gamma, \overline{p}} \qquad \dfrac{}{p, \overline{p}}\,\mathsf{Ax1}}{\Gamma, \overline{p}}\,\mathsf{cut} \quad\longrightarrow\quad \overset{\pi_0}{\Gamma, \overline{p}}$$

$$\dfrac{\dfrac{\overset{\pi_0}{\Gamma_0}}{\Gamma_0, \varphi}\,\mathsf{weak} \qquad \overset{\pi_1}{\overline{\varphi}, \Gamma_1}}{\Gamma_0, \Gamma_1}\,\mathsf{cut} \quad\longrightarrow\quad \dfrac{\overset{\pi_0}{\Gamma_0}}{\Gamma_0, \Gamma_1}\,\mathsf{weak}$$

Non-principal cut reductions

We push rules, where the cut formula is not principal, upwards away from the root and unfold D rules. The presented reductions are analogous, if the right premise of the cut is labeled with a non-principal rule. Recall that we assume that all proofs are minimally focused and that π_v denotes the generated proof from π rooted at v, as defined in Definition 7.2.9.

Case R Let R be a rule different from $\Box$, f, u, D and cut. Then we transform the proof as follows:

$$\dfrac{\dfrac{\overset{\pi_1}{\Gamma_1, \varphi^u} \quad \cdots \quad \overset{\pi_n}{\Gamma_n, \varphi^u}}{\Gamma, \varphi^u}\,\mathsf{R} \qquad \overset{\pi_0}{\overline{\varphi}^u, \Gamma_0}}{\Gamma, \Gamma_0}\,\mathsf{cut}$$

$$\longrightarrow\quad \dfrac{\dfrac{\overset{\pi_1}{\Gamma_1, \varphi^u} \qquad \overset{\pi_0}{\overline{\varphi}^u, \Gamma_0}}{\Gamma_1, \Gamma_0}\,\mathsf{cut} \quad \cdots \quad \dfrac{\overset{\pi_n}{\Gamma_n, \varphi^u} \qquad \overset{\pi_0}{\overline{\varphi}^u, \Gamma_0}}{\Gamma_n, \Gamma_0}\,\mathsf{cut}}{\Gamma, \Gamma_0}\,\mathsf{R}$$

Case D

$$v: \quad \dfrac{\dfrac{\begin{array}{c}\pi_0\\ \Gamma_0, \varphi\end{array}}{\Gamma_0, \varphi}\,\mathsf{D}_\dagger \qquad \begin{array}{c}\pi_1\\ \overline{\varphi}, \Gamma_1\end{array}}{\Gamma_0, \Gamma_1}\,\mathsf{cut} \qquad \longrightarrow \qquad \dfrac{\begin{array}{c}\pi_0'\\ \Gamma_0, \varphi\end{array} \qquad \begin{array}{c}\pi_1\\ \overline{\varphi}, \Gamma_1\end{array}}{\Gamma_0, \Gamma_1}\,\mathsf{cut}$$

where π_0' is obtained from π_0 by replacing every discharged leaf labeled with $\dagger$ with π_v, where v is the left premise of the cut rule.[7]

Case f Because we assume that π is minimally focused, the premise of an f rule is labeled with D. We transform those proofs as follows:

$$v: \quad \dfrac{\dfrac{\dfrac{\begin{array}{c}\pi_0\\ \Gamma_0', \varphi^a\end{array}}{\Gamma_0', \varphi^a}\,\mathsf{D}_\dagger}{\Gamma_0, \varphi^u}\,\mathsf{f} \qquad \begin{array}{c}\pi_1\\ \overline{\varphi}^u, \Gamma_1\end{array}}{\Gamma_0, \Gamma_1}\,\mathsf{cut} \qquad \longrightarrow \qquad \dfrac{\begin{array}{c}\pi_0'\\ \Gamma_0, \varphi^u\end{array} \qquad \begin{array}{c}\pi_1\\ \overline{\varphi}^u, \Gamma_1\end{array}}{\Gamma_0, \Gamma_1}\,\mathsf{cut}$$

where π_0' is obtained from π_0 by (i) unfocusing sequents up to D rules and leaves labeled with $\dagger$ and (ii) replacing every discharged leaf labeled with $\dagger$ with the generated proof π_v from π rooted at v, where v is the left premise of the cut rule.

Case u Minimally focused proofs only contain two types of u rules. First, u rules in trivial clusters, where all formulas in its premise are out of focus. Second, u rules in proper clusters, where a single formula ψ is put out of focus and the rank of ψ is lower than the maximal rank of formulas in focus in the sequent. These rules are applied as soon as possible. In cut rules, the formulas in focus in the premises are the same as the formulas in focus in its conclusion. Therefore, there are no u rules of the second type occurring in the premises of cut rules (as they already could have been applied at its conclusion). Therefore, in minimally focused proofs, if a u rule is labeling the premise of a cut, then all the formulas in its premise are out of focus. We proceed with an inner case distinction on the rule applied at the premise of u.

Subcase f If the premise of u on is labeled with f, we do the following:

[7] Here and in the following cut reductions we replace discharge tokens $\ddagger$ with fresh discharge tokens, whenever a $\mathsf{D}_\ddagger$ rule is duplicated.

$$v:\ \dfrac{\dfrac{\dfrac{\dfrac{\overset{\pi_0}{\Gamma_0',\varphi^a}}{\Gamma_0',\varphi^a}\,\mathsf{D}_\dagger}{\Gamma_0^u,\varphi^u}\,\mathsf{f}}{\Gamma_0,\varphi^u}\,\mathsf{u}\qquad \overset{\pi_1}{\overline{\varphi}^u,\Gamma_1}}{\Gamma_0,\Gamma_1}\,\mathsf{cut} \qquad\longrightarrow\qquad \dfrac{\dfrac{\overset{\pi_0'}{\Gamma_0^u,\varphi^u}}{\Gamma_0,\varphi^u}\,\mathsf{u}\qquad \overset{\pi_1}{\overline{\varphi}^u,\Gamma_1}}{\Gamma_0,\Gamma_1}\,\mathsf{cut}$$

where π_0' is defined as in the case for f. That is, π_0' is obtained from π_0 by (i) unfocusing sequents up to D rules and leaves labeled with $\dagger$ and (ii) replacing every discharged leaf labeled with $\dagger$ with the generated proof π_v from π rooted at v, where v is the premise of the u rule.

Subcase different rule Otherwise the premise of u is labeled with a rule R different from f and u. We proceed as follows:

$$\dfrac{\dfrac{\dfrac{\overset{\pi_1}{\Gamma_1}\quad\cdots\quad\overset{\pi_n}{\Gamma_n}}{\Gamma^u,\varphi^u}\,\mathsf{R}}{\Gamma,\varphi^u}\,\mathsf{u}\qquad \overset{\pi_0}{\overline{\varphi}^u,\Gamma_0}}{\Gamma,\Gamma_0}\,\mathsf{cut}$$

$$\longrightarrow\qquad \dfrac{\dfrac{\dfrac{\overset{\pi_1}{\Gamma_1}}{\Gamma_1'}\,\mathsf{u}\quad\cdots\quad\dfrac{\overset{\pi_n}{\Gamma_n}}{\Gamma_n'}\,\mathsf{u}}{\Gamma,\varphi^u}\,\mathsf{R}\qquad \overset{\pi_0}{\overline{\varphi}^u,\Gamma_0}}{\Gamma,\Gamma_0}\,\mathsf{cut}$$

7.4 Elimination of important cuts

In this section we develop the required technical machinery to eliminate important cuts. In particular, we will prove the following key lemma.

7.4.1. Lemma (Main Lemma). *Let π be a Focus-proof of the form*

$$\dfrac{\overset{\hat{\pi}}{\Sigma_0,\varphi^u}\qquad\overset{\hat{\tau}}{\overline{\varphi}^u,\Sigma_1}}{\Sigma_0,\Sigma_1}\,\mathsf{cut}$$

where $\hat{\pi}$ and $\hat{\tau}$ are cut-free and contraction-free and φ is a μ-formula. Then we can construct a Focus-proof π' of Σ_0,Σ_1 with cut rank $<\mathsf{rank}(\varphi)$.

We will obtain the proof of Lemma 7.4.1 by the following approach:

1. In Subsection 7.4.1 we introduce *traversed proofs*; these will be the intermediate objects in the elimination of important cuts.
2. We proceed with defining a traversed proof ρ_I from π in Definition 7.4.9.
3. Then we define a construction transforming traversed proofs that stops if a proof of lower cut rank is obtained. [Definition 7.4.10]
4. Finally, in Subsection 7.4.3 we prove that this construction applied to ρ_I terminates, meaning that it produces a Focus-proof π' of cut rank $< \mathsf{rank}(\varphi)$.

7.4.1 Traversed proofs

We will utilize a *multicut rule* – a derivable generalization of the ordinary cut rule – to avoid the nuisance of cut reductions with a cut rule, which might lead to commuting cut rules without any progress. This is a common way to deal with this technicality, see for instance [FS13]. The multicut compresses several cut rules to one rule with multiple premises. For example, the following proof would be expressed by a multicut as follows:

$$\dfrac{\dfrac{\overset{\pi_0}{\Sigma_0, \psi, \varphi} \quad \overset{\pi_1}{\Sigma_1, \overline{\varphi}}}{\Sigma_0, \Sigma_1, \psi}\,\mathsf{cut} \qquad \overset{\pi_2}{\Sigma_2, \overline{\psi}}}{\Sigma}\,\mathsf{cut} \quad \longrightarrow \quad \dfrac{\overset{\pi_0}{\Sigma_0, \psi, \varphi} \quad \overset{\pi_1}{\Sigma_1, \overline{\varphi}} \quad \overset{\pi_2}{\Sigma_2, \overline{\psi}}}{\Sigma}\,\mathsf{multicut}$$

One way to decide whether a multicut rule is well-formed, is to demand that it is a compression of cut rules. That is, it can be transformed into a derivation consisting of only cut rules, where the leaves of the derivation are labeled with the premises of the multicut. However, this condition is hard to work with. An easier but equivalent condition can be formulated in terms of the so-called *cut-connection graph* G. The nodes of G are the proofs of the premises of the multicut, where two nodes are connected if their roots are *cut-connected* (that is, one contains the cut formula φ, and the other contains $\overline{\varphi}$). For instance, in the example above we have that $\mathsf{G} = (G, E)$, where G consists of three nodes π_0, π_1 and π_2; with an edge between π_0 and π_1 (witnessed by the cut formula φ), and an edge between π_0 and π_2 (witnessed by the cut formula ψ).

For a well-formed multicut rule the cut-connection graph has to be *connected* and *acyclic*. Indeed, given a connected, acyclic cut-connection graph G, one can define a derivation of cut rules, where cut rules are given by the edges of G. This is possible, because G can be brought in to the shape of a tree by letting one of its nodes be its root.

7.4.2. DEFINITION. Let C be a multiset, we call elements of C *colors*. A *colored graph over* C is a graph (G, E) where every edge $e \in E$ is labeled with a color $c \in C$. We write $E_c(v, w)$ if there is an edge between v and w labeled with c.

In the cut rule the cut formula is always out of focus. In contrast, in a multicut, we allow occurrences of the cut formula in the premises to be in focus as well. We will now formally define multicuts.

7.4.3. Definition (Multicut). A *multicut* $\mathcal{M} = (\Pi, \Psi, \mathsf{T}, \mathsf{G})$ is a quadruple such that $\Pi = \pi_1, ..., \pi_m$ and $\mathsf{T} = \tau_1, ..., \tau_n$ are multisets of Focus-proofs; $\Psi = \psi_1, ..., \psi_k$ is a multiset of formulas; and $\mathsf{G} = (\Pi \cup \mathsf{T}, E)$ is an undirected colored graph over Ψ; where Ψ and $\overline{\Psi}$ have respective decompositions in multisets $\Psi = \Psi_1, ..., \Psi_m$ and $\overline{\Psi} = \Phi_1, ..., \Phi_n$ such that the following conditions are satisfied:

1. π_i is a proof of Γ_i, Ψ_i^u for $i = 1, ..., m$,
2. τ_j is a proof of $\Delta_j, \tilde{\Phi}_j$, where $(\tilde{\Phi}_j)^- = \Phi_j$ for $j = 1, ..., n$ and
3. the graph G is *connected*, *acyclic* and with each $\psi \in \Psi$ we associate a unique edge $E_\psi(\pi_i, \tau_j)$ for some $i = 1, ..., m$ and $j = 1, ..., n$ such that $\psi \in \Psi_i$ and $\overline{\psi} \in \Phi_j$.

The sequent $\Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_n$ is called the *conclusion of* $\mathcal{M}$.

We call G the *cut-connection graph of* $\mathcal{M}$ and call π and τ *cut-connected via* ψ if $E_\psi(\pi, \tau)$. An edge $E_\psi(\pi_i, \tau_j)$ corresponds to a cut with cut formula ψ and premises π and τ. If no confusion arises we will denote a multicut $\mathcal{M} = (\Pi, \Psi, \mathsf{T}, \mathsf{G})$ by $[\Pi]\Psi[\mathsf{T}]$ and treat the cut-connection graph G implicitly. If m, n and k denote the sizes of Π, T and Ψ, respectively, then the cut-connection graph $\mathcal{G}$ consists of $m+n$ nodes and k edges. As $\mathcal{G}$ is connected and acyclic it holds that $m+n = k+1$.

Note that it is possible that $\Psi = \varnothing$. In this case the multicut $\mathcal{M}$ consists of just one proof in $\Pi \cup \mathsf{T}$ and can thus simply be seen as a Focus-proof. If Ψ is nonempty, then it follows that $\Psi_i \neq \varnothing$ and $\Phi_j \neq \varnothing$ for all indices i, j.

We define a proof-like object built around the multicut's structure, a *formula-traversed proof*. Fix a formula φ. Intuitively a φ-traversed proof is a proof that is traversed by multicuts with cut formulas in $\mathsf{Clos}(\varphi)$, meaning that on every branch of the proof there is at most one such multicut. These will be our central technical objects in the elimination of important cuts. Note that in the multicut rules in a traversed proof, annotations might vary between premises of the rule and its conclusion.

7.4.4. Definition (Traversed proof). A *φ-traversed proof* ρ of a sequent Σ is a finite derivation of Σ, where all leaves v are either closed or *traversed leaves*, meaning that they are labeled with a sequent S_v together with a multicut $\mathcal{M}_v = (\Pi, \Psi, \mathsf{T}, \mathsf{G})$ where $\Psi \subseteq \mathsf{Clos}(\varphi)$; and, if Σ is the conclusion of $\mathcal{M}_v$, then $\mathsf{S}_v^- = \Sigma^-$.

If φ is clear from the context we will just write *traversed proof*.

A traversed leaf v is called *tidy* if

1. $\Psi \neq \varnothing$ and

2. $\varphi \equiv_C \psi$ for all $\psi \in \Psi$.

A traversed proof is called *tidy* if all its traversed leaves are tidy.

Ignoring the annotations for a moment, a traversed leaf of a traversed proof can be seen as a multicut of the form

$$\dfrac{\overset{\pi_1}{\Gamma_1, \Psi_1} \quad \cdots \quad \overset{\pi_m}{\Gamma_m, \Psi_m} \quad \overset{\tau_1}{\Delta_1, \Phi_1} \quad \cdots \quad \overset{\tau_n}{\Delta_n, \Phi_n}}{\Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_n} \ \mathsf{multicut}$$

Additionally, formulas in $\Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_n$ may have different annotations in the conclusion of the rule than in its premises. In this sense, every tidy φ-traversed proof ρ corresponds to a Focus-proof π, where on every branch of the proof there is at most one multicut of rank $\mathsf{rank}(\varphi)$. Hence, transforming a φ-traversed proof to a traversed proof without traversed leaves corresponds to eliminating multicuts of rank $\mathsf{rank}(\varphi)$. Due to this correspondence with Focus-proofs with multicuts, we choose the name traversed *proof* instead of traversed derivation.

Formally, a traversed proof is a Focus-derivation with special kind of leaves. Therefore, the *cut rank* of traversed proofs is defined as for Focus-derivations. In other words, the cut rank of a traversed proof is the maximal rank of a cut below all multicuts.

We will denote a traversed leaf v labeled with a sequent Σ and a multicut $\mathcal{M}_v = (\Pi, \Psi, \mathsf{T}, \mathsf{G})$ by

$$\overset{\mathcal{M}_v}{\Sigma}$$

or, if we do not want to deal with the cut-connection graph explicitly, by

$$\overset{[\Pi]\Psi[\mathsf{T}]}{\Sigma}$$

Given a multicut $\mathcal{M}$, we need an operation that removes an edge labeled with ψ from the cut-connection graph. This might be necessary because a cut of lower rank in the proof is applied or one of the cut formulas is weakened. The multicut $\mathcal{M}(\pi, \psi)$ then consists of the remaining nodes connected to π.

7.4.5. Definition. Let $\mathcal{M} = (\Pi, \Psi, \mathsf{T}, \mathsf{G})$ be a multicut, $\pi \in \Pi$, $\psi \in \Psi$, $\tau \in \mathsf{T}$ and let $E_\psi(\pi, \tau)$ be an edge in G. We define $\mathcal{M}(\pi, \psi)$ to be the multicut $(\Pi', \Psi', \mathsf{T}', \mathsf{G}')$ obtained as follows: Remove $E_\psi(\pi, \tau)$ from G and let G' be the subgraph of G of nodes connected to π. Let $\Pi' \cup \mathsf{T}'$ be the multiset of nodes of G' such that $\Pi' \subseteq \Pi$ and $\mathsf{T}' \subseteq \mathsf{T}$ and let $\Psi' \subseteq \Psi$ be the multiset of colors of edges occurring in G'. The multicut $\mathcal{M}(\tau, \psi)$ is defined analogously replacing π with τ.

7.4.6. Lemma. *Let ρ be a φ-traversed proof with cut rank $< \mathsf{rank}(\varphi)$. Then ρ can be transformed to a tidy φ-traversed proof ρ' with cut rank $< \mathsf{rank}(\varphi)$ without introducing extra* f *rules.*

Proof:
Let v be a traversed leaf in ρ labeled with a sequent Σ and a multicut $\mathcal{M}_v = (\Pi, \Psi, \mathsf{T}, \mathsf{G})$ that is not tidy. If Ψ is empty, then we replace v with π_1 if $m = 1$, or with τ_1 if $n = 1$.

If $\Psi = \Psi', \psi$ and $\psi \not\equiv_C \varphi$, then $\mathsf{rank}(\psi) < \mathsf{rank}(\varphi)$ since $\psi \in \mathsf{Clos}(\varphi)$. Let π and τ be proofs that are cut-connected via ψ. Let Σ^l, ψ be the conclusion of $\mathcal{M}(\pi, \psi)$ and $\Sigma^r, \overline{\psi}$ be the conclusion of $\mathcal{M}(\tau, \psi)$. Then we replace v with

$$\dfrac{\begin{matrix}\mathcal{M}(\pi,\psi) \\ \Sigma^l, \psi\end{matrix} \qquad \begin{matrix}\mathcal{M}(\tau,\psi) \\ \Sigma^r, \overline{\psi}^u\end{matrix}}{\Sigma}\ \mathsf{cut}$$

This cut has rank lower than $\mathsf{rank}(\varphi)$ and we obtain a φ-traversed proof with cut rank $< \mathsf{rank}(\varphi)$. Iterating this procedure we arrive at a tidy φ-traversed proof. □

In the next subsection we will define a construction transforming traversed proofs to traversed proofs without traversed leaves. In this construction it is necessary to keep track of the dynamics of the proofs occurring in the multicuts. When transforming a traversed proof with only one traversed leaf v labeled with a multicut $[\hat{\pi}]\varphi[\hat{\tau}]$, we obtain multicuts of the form $[\Pi]\Psi[\mathsf{T}]$ where we can relate any proof π in Π to some node in $\hat{\pi}$, called the *origin* of π; and we can combine these nodes to a path in $\hat{\pi}$, called the *history* of the proof π. Analogously for proofs in T.

7.4.7. Definition. A *multicut $\mathcal{M}$ with origin* $(\hat{\pi}, \hat{\tau})$ is a triple $(\mathcal{M}, (\hat{\pi}, \hat{\tau}), \mathsf{hist})$ such that $\mathcal{M} = (\Pi, \Psi, \mathsf{T}, \mathsf{G})$ is a multicut; $(\hat{\pi}, \hat{\tau})$ is a pair of Focus-proofs; and the *history map* hist is a map with domain $\Pi \cup \mathsf{T}$ that maps proofs in Π to paths in $\hat{\pi}$ and proofs in T to paths in $\hat{\tau}$. The *origin map* of such a multicut with origin is defined as

$$\begin{aligned}\mathsf{origin} : \Pi \cup \mathsf{T} &\to \hat{\pi} \cup \hat{\tau}, \\ \pi &\mapsto \mathsf{last}(\mathsf{hist}(\pi)).\end{aligned}$$

For such a triple to qualify as a multicut with origin, the origin map origin has to satisfy the following conditions for all $i = 1, ..., m$ and $j = 1, ..., n$, where we use notations as in the definition of a multicut, Definition 7.4.3:

1. $\pi_i = \hat{\pi}_{\mathsf{origin}(\pi_i)}$ and $\tau_j = \hat{\tau}_{\mathsf{origin}(\tau_j)}$,
2. $\mathsf{S}(\mathsf{origin}(\pi_i)) = \Gamma'_i, \Psi_i^u$, where $\Gamma'_i = \Gamma_i$ or $(\Gamma'_i)^u = \Gamma_i$ and
3. $\mathsf{S}(\mathsf{origin}(\tau_j)) = \Delta'_j, \Phi'_j$, where $(\Delta'_j)^u = \Delta_j$ and $(\Phi'_j)^u = \Phi_j$.

We define the *origin depth* of a multicut $\mathcal{M} = (\Pi, \Psi, \mathsf{T}, \mathsf{G})$ with a $(\hat{\pi}, \hat{\tau})$-history as $\mathsf{oriDepth}(\mathcal{M}) := \max\{\mathsf{depth}(\mathsf{origin}(\pi)) \mid \pi \in \Pi\}$. Note that we only consider proofs in Π in this definition and not the ones in T.

Given a node v in a Focus-proof π, the generated proof π_v from π rooted at v may have a bigger depth than π; it is only guaranteed that $\mathsf{depth}(\pi) \leq \mathsf{depth}(\pi_v)$. Therefore, $\mathsf{depth}(\mathsf{origin}(\pi))$ may differ from $\mathsf{depth}(\pi)$.

7.4.8. DEFINITION. A *φ-traversed proof ρ with origin* $(\hat{\pi}, \hat{\tau})$ is defined analogously to a φ-traversed proof where traversed leaves are labeled with multicuts with origin $(\hat{\pi}, \hat{\tau})$.

We define the *origin depth* of a traversed leaf v labeled with $\mathcal{M}_v = (\Pi, \Psi, \mathsf{T}, \mathsf{G})$ as $\mathsf{oriDepth}(v) := \max\{\mathsf{depth}(\mathsf{origin}(\pi)) \mid \pi \in \Pi\}$. Note that we only consider proofs in Π in this definition and not the ones in T.

7.4.2 Proof transformations

Let π be a Focus-proof as given in Lemma 7.4.1 of the form

$$\dfrac{\begin{matrix}\hat{\pi} \\ \Sigma_0, \varphi^u\end{matrix} \qquad \begin{matrix}\hat{\tau} \\ \overline{\varphi}^u, \Sigma_1\end{matrix}}{\Sigma_0, \Sigma_1}\ \mathsf{cut}$$

where $\hat{\pi}$ and $\hat{\tau}$ are cut-free and contraction-free and φ is a μ-formula. We want to transform π to a Focus-proof π' of Σ_0, Σ_1 with cut rank $< \mathsf{rank}(\varphi)$.

Fix $\pi, \hat{\pi}, \hat{\tau}, \Sigma_0, \Sigma_1$ and φ for the remainder of this section.

7.4.9. DEFINITION. We define the *initial traversed proof* ρ_I with origin $(\hat{\pi}, \hat{\tau})$ to be the φ-traversed proof of Σ_0, Σ_1 consisting of a traversed leaf labeled with Σ_0, Σ_1 together with $[\hat{\pi}]\varphi[\hat{\tau}]$, and where $\mathsf{hist}(\hat{\pi})$ is the path consisting of the root of $\hat{\pi}$ and $\mathsf{hist}(\hat{\tau})$ is the path consisting of the root of $\hat{\tau}$. We denote the traversed proof ρ_I by

$$\begin{matrix}[\hat{\pi}]\varphi[\hat{\tau}] \\ \Sigma_0, \Sigma_1\end{matrix}$$

The high level strategy to transform ρ_I to a traversed proof without traversed leaves is as follows: We start by pushing up traversed leaves, and unfolding proofs whenever a companion node is reached. This is done similarly as one would push up multicuts. We continue pushing up the traversed leaves in the traversed proof until we find successful repeats below traversed leaves. This check will be done whenever a modal rule gets introduced.

In order to guarantee that we find such a successful repeat we have to be very careful about which formulas we put in focus. Let v be a traversed leaf labeled with

$$\begin{matrix}[\Pi]\Psi[\mathsf{T}] \\ \Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_n\end{matrix}$$

We have to decide on which formulas in $\Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_n$ we keep the annotations as in the proofs in Π and T. Our strategy is as follows: formulas in $\Delta_1, ..., \Delta_n$ will always be out of focus; and formulas in Γ_i keep the same annotation as in π_i if $\mathsf{depth}(\mathsf{origin}(\pi_i)) = \mathsf{oriDepth}(v)$ and will be unfocused otherwise for $i = 1, ..., m$. Recall that $\mathsf{oriDepth}(v)$ is the maximal depth of the nodes $\mathsf{origin}(\pi_1), ..., \mathsf{origin}(\pi_m)$.

The reason for this asymmetry stems from the following observation: The formula φ is a μ-formula, therefore all formulas in Ψ are magenta and all formulas in $\overline{\Psi}$ are navy. As π is minimally focused, only navy formulas are in focus. This means that in the proofs $\pi_1, ..., \pi_m \in \Pi$ formulas from Ψ are out of focus, whereas in the proofs $\tau_1, ..., \tau_n \in \mathsf{T}$ formulas from $\overline{\Psi}$ might be in focus. By deleting the formulas from $\overline{\Psi}$ in the proofs $\tau_1, .., \tau_n$ we cannot ensure that successful paths are still successful. Deleting formulas from Ψ in the proofs $\pi_1, ..., \pi_m$ on the other hand never removes formulas in focus.

We only keep annotations on formulas originating from the proofs $\pi_1, ..., \pi_m$. If we keep the annotations from all those proofs this could also lead to trouble – we also add applications of f potentially destroying the success condition on paths. We therefore opt to only keep annotations coming from those proofs in $\pi_1, ..., \pi_m$, where $\mathsf{origin}(\pi_1), ..., \mathsf{origin}(\pi_m)$ has maximal depth. This guarantees that at some point no f rules are applied anymore. In the case that all formulas become out of focus, this also ensures that $\mathsf{oriDepth}(v)$ got reduced and hence we can employ induction on $\mathsf{oriDepth}(v)$ in our termination argument.

In the next definition we will give a formal description of these intuitions.

7.4.10. Definition. We define the *traversed leaf reduction algorithm*; it transforms a traversed proof with origin $(\hat{\pi}, \hat{\tau})$ to a traversed proof with origin $(\hat{\pi}, \hat{\tau})$ without traversed leaves while preserving the cut rank.

Let ρ be a φ-traversed proof with origin $(\hat{\pi}, \hat{\tau})$. We may always assume that ρ is tidy (see Lemma 7.4.6). If all leaves are closed we are done. Otherwise consider the leftmost traversed leaf v labeled with

$$\begin{array}{c}[\Pi]\Psi[\mathsf{T}]\\ \Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_n\end{array}$$

We transform ρ by a case distinction on the last applied rules in the proofs in Π and T.

- $\Box$ **rule.** If the last applied rule is $\Box$ in π_i for all $i = 1, ..., m$ and in τ_j for all $j = 1, ..., n$, we make the following case distinction.
  - If there is a node c in ρ that is an ancestor of v such that $\mathsf{S}_c =_{Set} \Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_m$ and such that the path from c to v is successful, then insert a $\mathsf{D}_\dagger$ rule at c and replace v with

$$\dfrac{[\mathsf{S}_c]^\dagger}{\Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_n}\ \mathsf{weak}, \mathsf{contr}$$

with fresh discharge token †. If there is such an ancestor that is already labeled with $\mathsf{D}_\ddagger$, then let the new leaf be discharged by ‡ and do not insert an extra $\mathsf{D}_\dagger$ rule.

– Else let π_i' be the maximal subproof of π_i rooted at the child of the root of π_i for $i = 1, ..., m$ and define τ_j' analogously from τ_j for $j = 1, ..., n$. We apply a $\Box$ rule in ρ and replace the proofs π_i with π_i' for $i = 1, ..., m$ and τ_j with τ_j' for $j = 1, ..., n$.
That is, we transform a multicut of the form[8]

$$\dfrac{\dfrac{\gamma, \Theta_1, \chi_1^1, ..., \chi_1^{k_1}}{\Box\gamma, \Diamond\Theta_1, \Diamond\chi_1^1, ..., \Diamond\chi_1^{k_1}}\,\Box \quad \cdots \quad \dfrac{\Theta_l, \chi_l^1, ..., \chi_l^{k_l}, \chi_l}{\Diamond\Theta_l, \Diamond\chi_l^1, ..., \Diamond\chi_l^{k_l}, \Box\chi_l}\,\Box}{\Box\gamma, \Diamond\Theta_1, ..., \Diamond\Theta_l}\,\mathsf{multicut}$$

to the following multicut:

$$\dfrac{\dfrac{\gamma, \Theta_1, \chi_1^1, ..., \chi_1^{k_1} \quad \cdots \quad \Theta_l, \chi_l^1, ..., \chi_l^{k_l}, \chi_l}{\gamma, \Theta_1, ..., \Theta_l}\,\mathsf{multicut}}{\Box\gamma, \Diamond\Theta_1, ..., \Diamond\Theta_l}\,\Box$$

In order to claim that this is possible, we need to show that there is exactly one $\Box$-formula in the conclusion of the multicut $[\Pi]\Psi[\mathsf{T}]$. Let m, n and k denote the sizes of Π, T and Ψ, respectively. Because the cut-connection graph $\mathcal{G}$ is a connected and acyclic graph with $m + n$ many nodes and k many edges, we have that $m + n = k + 1$. Every formula ψ in Ψ is modal, thus either ψ or $\overline{\psi}$ is a $\Box$-formula. In the roots of the proofs π_i and τ_j exactly one formula is a $\Box$-formula. Of these formulas, all but one are cut formulas. Therefore, there is exactly one formula of the form $\Box\chi$ in the conclusion of $[\Pi]\Psi[\mathsf{T}]$ and the rule $\Box$ is applicable.
We define $\mathsf{hist}(\pi_i')$ as the path $\mathsf{hist}(\pi_i)$ extended with the child of $\mathsf{origin}(\pi_i)$ for $i = 1, ..., m$ and define $\mathsf{hist}(\tau_j')$ as the path $\mathsf{hist}(\tau_j)$ extended with the child of $\mathsf{origin}(\tau_j)$ for $j = 1, ..., n$.

Else we pick $i \in \{1, ..., m\}$ or $j \in \{1, ...n\}$ and reduce π_i or τ_j. We let $\Pi = \Pi', \pi_i$ and $\mathsf{T} = \mathsf{T}', \tau_j$.

- **f rule in Π.** If there is an i such that the last applied rule in π_i is f, then π_i has the form

$$\dfrac{\begin{array}{c}\pi_i' \\ \Gamma_i', \Psi_i^u\end{array}}{\Gamma_i, \Psi_i^u}\,\mathsf{f}$$

[8]For simplicity we omit annotations. Here l is the number of proofs in $\Pi \cup \mathsf{T}$, that is, $l = m + n$.

Note that all formulas in Ψ_i are magenta, thus due to Proposition 2.4.9 no formula in Ψ_i is navy. As π is minimally focused, it follows that no formula in Ψ_i is put in focus in f. We make a case distinction:

- If $\mathsf{depth}(\mathsf{origin}(\pi_i)) = \mathsf{oriDepth}(v)$, then replace v with

$$\dfrac{\begin{array}{c}[\Pi', \pi_i']\Psi[\mathsf{T}] \\ \Gamma_1, ..., \Gamma_i', ..., \Gamma_m, \Delta_1, ..., \Delta_n\end{array}}{\Gamma_1, ..., \Gamma_i, ..., \Gamma_m, \Delta_1, ..., \Delta_n}\ \mathsf{f}$$

- Otherwise replace π_i with π_i' without applying an f rule.

In both cases we define $\mathsf{hist}(\pi_i')$ as the path $\mathsf{hist}(\pi_i)$ extended with the child of $\mathsf{origin}(\pi_i)$.

- u **rule in** Π**.** If there is an i such that the last applied rule in π_i is u, then π_i has the form

$$\dfrac{\begin{array}{c}\pi_i' \\ \Gamma_i^u, \Psi_i^u\end{array}}{\Gamma_i', \Psi_i^u}\ \mathsf{u}$$

We make a case distinction:

- If there are formulas in focus in Γ_i, then we replace v with

$$\dfrac{\begin{array}{c}[\Pi', \pi_i']\Psi[\mathsf{T}] \\ \Gamma_1, ..., \Gamma_i^u, ..., \Gamma_m, \Delta_1, ..., \Delta_n\end{array}}{\Gamma_1, ..., \Gamma_i, ..., \Gamma_m, \Delta_1, ..., \Delta_n}\ \mathsf{u}$$

- Otherwise we replace π_i with π_i' without applying a u rule.

In both cases we define $\mathsf{hist}(\pi_i')$ as the path $\mathsf{hist}(\pi_i)$ extended with the child of $\mathsf{origin}(\pi_i)$.

- f **rule or** u **rule in** T**.** If there is a j such that the last applied rule R in τ_j is f or u, then τ_j has the form

$$\dfrac{\begin{array}{c}\tau_j' \\ \Delta_j', {\Phi_j}^b\end{array}}{\Delta_j, {\Phi_j}^a}\ \mathsf{R}$$

We replace τ_i with τ_i', and define $\mathsf{hist}(\tau_j')$ as the path $\mathsf{hist}(\tau_j)$ extended with the child of $\mathsf{origin}(\tau_j)$.

- D **rule.** If there is an i such that the last applied rule in π_i is D, then π_i has the form

$$\dfrac{\begin{array}{c}\pi_i'\\ \Gamma_i', \Psi_i^u\end{array}}{\Gamma_i', \Psi_i^u}\ \mathsf{D}$$

  We unfold π_i, meaning that we let $\tilde{\pi}_i$ be the proof obtained from π_i' by replacing every discharged leaf labeled with † with π_i.[9]

  We perform the following check and focus the sequent Γ_i if the origin depth of π_i is maximal:

  - if $\mathsf{depth}(\mathsf{origin}(\pi_i)) = \mathsf{oriDepth}(v)$ and $\Gamma_i' \neq \Gamma_i$, then Γ_i is unfocused and we replace v with

$$\dfrac{\begin{array}{c}[\Pi', \tilde{\pi}_i]\Psi[\mathsf{T}]\\ \Gamma_1, ..., \Gamma_i', ..., \Gamma_m, \Delta_1, ..., \Delta_n\end{array}}{\Gamma_1, ..., \Gamma_i, ..., \Gamma_m, \Delta_1, ..., \Delta_n}\ \mathsf{f}$$

  - else we replace v with

$$\begin{array}{c}[\Pi', \tilde{\pi}_i]\Psi[\mathsf{T}]\\ \Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_n\end{array}$$

  We define $\mathsf{hist}(\tilde{\pi}_i)$ as the path $\mathsf{hist}(\pi_i)$ extended with the child of $\mathsf{origin}(\pi_i)$. Note that in this case we have $\tilde{\pi}_i = \hat{\pi}_{\mathsf{origin}(\tilde{\pi}_i)}$ because the generated subproof $\hat{\pi}_{\mathsf{origin}(\tilde{\pi}_i)}$ of $\hat{\pi}$ rooted at $\mathsf{origin}(\tilde{\pi}_i)$ is defined as the unfolding of the generated subproof $\hat{\pi}_{\mathsf{origin}(\pi_i)}$. Analogously if there is a j such that the last applied rule in τ_j is D where we never apply a f rule.

In the rest of the cases we push the traversed leaf upwards. These transformations resemble the expected cut reductions for the multicut rule. Note that we assumed that π does not contain contractions. The origin map identifies proofs in Π or T with generated subproofs of π – hence these proof do not contain contractions either.

- **Non-principal rule.** If there is an i such that the last applied rule in π_i is a rule with principal formula in Γ_i, then we "push the cut upwards". Assume that π_i has the form[10]

$$\dfrac{\begin{array}{c}\pi_i^1\\ \Gamma_i^1, \Psi_i\end{array} \quad \dots \quad \begin{array}{c}\pi_i^n\\ \Gamma_i^n, \Psi_i\end{array}}{\Gamma_i, \Psi_i}\ \mathsf{R}$$

[9] Discharge tokens ‡ are replaced with fresh discharge tokens, whenever a $\mathsf{D}_‡$ rule is duplicated.

[10] Note that formally the root of π_i is labelled with Γ_i', Ψ_i' where Γ_i, Ψ_i coincides with Γ_i', Ψ_i' up to annotations. Because annotations do not change in the following reductions, we will be slightly imprecise and just write Γ_i, Ψ_i to improve readability. The same applies to the following reductions.

We let $\Gamma_1, ..., \Gamma_n = \Sigma', \Gamma_i$. Recall that $\Pi = \Pi', \pi_i$. We replace v with

$$\dfrac{\begin{matrix}[\Pi', \pi_i^1]\Psi[\mathsf{T}] \\ \Sigma', \Gamma_i^1, \Delta_1, ..., \Delta_n\end{matrix} \quad \dots \quad \begin{matrix}[\Pi', \pi_i^n]\Psi[\mathsf{T}] \\ \Sigma', \Gamma_i^n, \Delta_1, ..., \Delta_n\end{matrix}}{\Sigma', \Gamma_i, \Delta_1, ..., \Delta_n}\ \mathsf{R}$$

In this case $\mathsf{origin}(\pi_i)$ is also labeled with R and we define $\mathsf{hist}(\pi_i^k)$ as the path $\mathsf{hist}(\pi_i)$ extended with the child of $\mathsf{origin}(\pi_i)$ containing the formulas in Γ_i^k for $k = 1, ..., n$. Analogously if there is a j such that the last applied rule in τ_j is a rule with principal formula in Δ_j.

- weak **rule.** If there is an i such that the last applied rule in π_i is weak, where $\Psi_i = \Psi_i', \psi$, then π_i is of the form

$$\dfrac{\begin{matrix}\pi_i' \\ \Gamma_i', \Psi_i'\end{matrix}}{\Gamma_i', \Psi_i', \psi}\ \mathsf{weak}$$

Let $\mathcal{M}$ be the multicut at v and let $\mathcal{M}(\pi_i, \psi)$ be the multicut obtained from $\mathcal{M}$ by first removing an edge $E_\psi(\pi_i, \tau)$ for some τ, and then taking all proofs connected to π_i, as formally defined in Definition 7.4.5. Let $\mathcal{M}(\pi_i', \psi)$ be the multicut obtained from $\mathcal{M}(\pi_i, \psi)$ by replacing π_i with π_i' and let Σ be its conclusion. Then we replace v with

$$\dfrac{\begin{matrix}\mathcal{M}(\pi_i', \psi) \\ \Sigma\end{matrix}}{\Gamma_1, ..., , \Gamma_m, \Delta_1, ..., \Delta_n}\ \mathsf{weak}$$

We define $\mathsf{hist}(\pi_i')$ as the path $\mathsf{hist}(\pi_i)$ extended with the child of $\mathsf{origin}(\pi_i)$. Analogously if there is a j such that the last applied rule in τ_j is weak, where the principal formula is $\psi \in \Phi_j$.

In the remaining cases a non-modal formula $\psi \in \Psi$ is principal in the last applied rules in π_i and τ_j. Let π_i and τ_j be cut-connected proofs via ψ in respectively Π and T. We let $\Psi = \Psi', \psi$; $\Psi_i = \Psi_i', \psi$ and $\Phi_j = \Phi_j', \psi$ as well as $\Pi = \Pi', \pi_i$ and $\mathsf{T} = \mathsf{T}', \tau_j$.

- $\vee$ **rule.** If $\psi = \psi_0 \vee \psi_1$, then π_i has the form

$$\dfrac{\begin{matrix}\pi_i' \\ \Gamma_i, \Psi_i', \psi_0^u, \psi_1^u\end{matrix}}{\Gamma_i, \Psi_i', \psi_0 \vee \psi_1^u}\ \vee$$

and τ_j has the form

$$\dfrac{\begin{matrix}\tau_j^0 & & \tau_j^1 \\ \Delta'_j, \Phi'_j, \overline{\psi_0}^a & & \Delta'_j, \Phi'_j, \overline{\psi_1}^a\end{matrix}}{\Delta'_j, \Phi'_j, \overline{\psi_0} \wedge \overline{\psi_1}^a} \wedge$$

Then we replace v with

$$\dfrac{\begin{matrix}[\Pi', \pi'_i]\Psi', \psi_0, \psi_1[\mathsf{T}', \tau_j^0, \tau_j^1] \\ \Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_j, \Delta_j, ..., \Delta_n\end{matrix}}{\Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_j, ..., \Delta_n} \ \mathsf{contr}$$

where ψ_0 is cut-connected to π'_i and τ_j^0; and ψ_1 is cut-connected to π'_i and τ_j^1. We define $\mathsf{hist}(\pi'_i)$ as the path $\mathsf{hist}(\pi_i)$ extended with the child of $\mathsf{origin}(\pi_i)$. The path $\mathsf{hist}(\tau_j^k)$ is defined as $\mathsf{hist}(\tau_j)$ extended with the child of $\mathsf{origin}(\tau_j)$ containing the auxiliary formula $\overline{\psi_k}^a$ for $k = 0, 1$.

- $\wedge$ **rule.** The case $\wedge$ is dual to $\vee$.
- μ **rule.** If $\psi = \mu x.\chi$ then π_i has the form

$$\dfrac{\begin{matrix}\pi'_i \\ \Gamma_i, \Psi'_i, \chi[x/\mu x.\chi]^u\end{matrix}}{\Gamma_i, \Psi'_i, \mu x.\chi^u} \ \mu$$

and τ_j has the form

$$\dfrac{\begin{matrix}\tau'_j \\ \Delta'_j, \Phi'_j, \overline{\chi}[x/\nu x.\overline{\chi}]^a\end{matrix}}{\Delta'_j, \Phi'_j, \nu x.\overline{\chi}^a} \ \nu$$

Then we replace v with

$$\begin{matrix}[\Pi', \pi'_i]\Psi', \chi[x/\mu x.\chi][\mathsf{T}', \tau'_j] \\ \Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_n\end{matrix}$$

We define $\mathsf{hist}(\pi'_i)$ as the path $\mathsf{hist}(\pi_i)$ extended with the child of $\mathsf{origin}(\pi_i)$, and define $\mathsf{hist}(\tau'_j)$ as the path $\mathsf{hist}(\tau_j)$ extended with the child of $\mathsf{origin}(\tau_j)$.

- ν **rule.** As ρ is tidy, $\psi \equiv_C \varphi$ for every formula $\psi \in \Psi$. Therefore ψ is magenta and due to Proposition 2.4.9 this means that ψ is never a ν-formula.
- **Axioms.** As ψ is magenta it is never of the form p or $\overline{p}$. Hence the last applied rule in π_i or τ_j is not an axiom.

7.4.11. Remark. It may seem that the construction is formulated in a non-deterministic way, yet this is only superficially so. All choices can be made canonical, depending on an arbitrary but fixed order on proof rules and proofs in $\Pi \cup \mathsf{T}$. For example, we could give priority to cases where a formula $\psi \in \Psi$ is principal on both sides and take an arbitrary order on $\Pi \cup \mathsf{T}$, where proofs in Π are of higher priority than proofs in T. Importantly, the particular choice of orders does not matter in the termination proof.

7.4.3 Proof of termination

We prove that the traversed leaf reduction algorithm given in Subsection 7.4.2 yields the desired proof. First we show that the transformation only terminates if a traversed proof without traversed leaves is reached. In Lemma 7.4.13 we then show that the algorithm terminates when applied to ρ_I.

7.4.12. Lemma. *If v is a tidy traversed leaf in a traversed proof ρ, then one of the cases in the case distinction in Definition 7.4.10 is applicable.*

Proof:
Let v be labeled with $[\Pi]\Psi[\mathsf{T}]$. If there is a proof in $\Pi \cup \mathsf{T}$, where the last applied rule is different from a rule with principal formula in Ψ and different from $\Box$, then we can transform that proof. Otherwise, for all $i = 1, ..., m$ the last applied rule in π_i is either $\Box$ or a rule with principal formula in Ψ_i and analogously for all $j = 1, ..., n$ the last applied rule in τ_j is either $\Box$ or a rule with principal formula in $\overline{\Phi_j}$. If the last applied rule in all those proofs is $\Box$ we are in the first case of Definition 7.4.10. Else let Ψ' be the non-empty submultiset of Ψ consisting of all non-modal formulas in Ψ. Let $\Pi' \subseteq \Pi$ and $\mathsf{T}' \subseteq \mathsf{T}$ be the respective subset of proofs of Π and T, where the last applied rule is different than $\Box$. Let G' be the subgraph of the cut-connection graph G with nodes $\Pi' \cup \mathsf{T}'$ and edges labeled with formulas in Ψ'. Then G' is non-empty and acyclic. Moreover, we may assume that G' is connected, otherwise continue with a maximally connected subgraph of G'. Let $k' = |\Psi'|$, $m' = |\Pi'|$ and $n' = |\mathsf{T}'|$, then $m' + n' = k' + 1$. At every node in $\Pi' \cup \mathsf{T}'$ the principal formula of the last applied rule in the proof is in Ψ' or in $\overline{\Psi'}$. As $k' < m' + n'$ there is $\psi \in \Psi'$ and an edge $E_\psi(\pi_i, \tau_j)$ in $\mathcal{G}'_v$ such that ψ is principal in the last applied rule in π_i and $\overline{\psi}$ is principal in the last applied rule in τ. ☐

7.4.13. Lemma. *The traversed leaf reduction algorithm given in Definition 7.4.10 applied to the initial traversed proof ρ_I with origin $(\hat{\pi}, \hat{\tau})$ terminates and yields a* Focus*-proof ρ_T.*

Proof:
In this proof we will simply write traversed proofs for traversed proofs with origin $(\hat{\pi}, \hat{\tau})$ and treat the hist and origin maps implicitly.

Let ρ_k and ρ_l be traversed proofs. We write $\rho_k \leq_t \rho_l$ if ρ_l can be obtained from ρ_k by the traversed leaf reduction algorithm. It holds that $\leq_t$ is a partial order. Moreover, if $\rho_k \leq_t \rho_l$, then ρ_k is a sub-traversed proof of ρ_l, in the sense that ρ_l can be obtained from ρ_k by replacing some traversed leaves in ρ_k by traversed proofs and inserting nodes labeled with D. Thus, ρ_l consists of at least the nodes in ρ_k and we can identify nodes in ρ_k with nodes in ρ_l.

Fix an arbitrary traversed proof ρ with $\rho_I \leq_t \rho$.

For every node v in ρ we can find $\rho' \leq_t \rho$, where v is the leftmost open leaf. If there are multiple ones, then choose the minimal. Note that we intentionally overuse v to denote the node in ρ and the traversed leaf in ρ'. Using the fact that we can identify the node v in ρ with the traversed leaf v in ρ', we define $\mathsf{oriDepth}(v)$ for a node v in ρ to be the depth $\mathsf{oriDepth}(v)$ of the traversed leaf v in ρ'.

For the termination argument we need some measures on the proof π. Let $n_l := |\mathsf{Clos}(\Sigma_0^-, \varphi)|$ and $n_r := |\mathsf{Clos}(\Sigma_1^-, \overline{\varphi})|$, that is, the sizes of the closures of the roots of $\hat{\pi}$ and $\hat{\tau}$, respectively. Let m_l be the number of nodes in $\hat{\pi}$.

<u>`Claim 1`</u>: Nodes in ρ are labeled with at most $2^{2 \cdot n_l + n_r}$ many distinct sequents up to $=_{\mathsf{Set}}$.

`Proof of Claim 1`: For every node v in ρ there is $\rho' \leq_t \rho$ such that v is an open leaf in ρ'. Let v be labeled with

$$\begin{array}{c}[\Pi]\Psi[\mathsf{T}] \\ \Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_n\end{array}$$

Because of the definition of the origin map, for each $i = 1, ..., m$ there is a node w in $\hat{\pi}$ such that $\mathsf{S}_w = \Gamma_i', \Psi_i^u$ where $\Gamma_i' = \Gamma_i$ or $(\Gamma_i')^u = \Gamma_i$. Similarly, for each $j = 1, ..., n$ there is a node w in $\hat{\tau}$ such that $\mathsf{S}_w = \Delta_j', \Phi_j'$ where $(\Delta_j')^u = \Delta_j$.

Therefore the sequent $\Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_n$ consists of the union of sequents in $\{\Gamma \subseteq \mathsf{S}_w \mid w \in \hat{\pi}\}$, $\{\Gamma^u \mid \Gamma \subseteq \mathsf{S}_w \text{ and } w \in \hat{\pi}\}$ and $\{\Delta^u \mid \Delta \subseteq \mathsf{S}_w \text{ and } w \in \hat{\tau}\}$. Because $\hat{\pi}$ and $\hat{\tau}$ are cut-free, all formulas at nodes $w \in \hat{\pi}$ are in $\mathsf{Clos}(\Sigma_0^-, \varphi)$ and formulas at nodes $w \in \hat{\tau}$ are in $\mathsf{Clos}(\Sigma_1^-, \overline{\varphi})$. This implies that only $2 \cdot n_l + n_r$ many annotated formulas occur in S_v. Hence, v is labeled with at most $2^{2 \cdot n_l + n_r}$ many distinct sequents up to $=_{\mathsf{Set}}$. ⊣

Let $\alpha = a_0 ... a_g$ be a path in ρ from the root of ρ to a traversed leaf. For every node a_k on α we let $\rho_k \leq_t \rho$ be such that a_k is the leftmost open leaf in ρ_k. If there are multiple ones, then choose the minimal. To fix notation we let a_k in ρ' be labeled with

$$\begin{array}{c}[\Pi^k]\Psi^k[\mathsf{T}^k] \\ \Gamma_1^k, ..., \Gamma_{m'}^k, \Delta_1^k, ..., \Delta_{n'}^k\end{array}$$

Let α be the path from the root of ρ to an open leaf v labeled with

$$\frac{[\Pi]\Psi[\mathsf{T}]}{\Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_n}$$

Let $\Pi = \pi_1, ..., \pi_m$ and $\mathsf{T} = \tau_1, ..., \tau_n$. We define corresponding paths $\alpha_i := \mathsf{hist}(\pi_i)$ in $\hat{\pi}$ for $i = 1, ..., m$ and $\beta_j := \mathsf{hist}(\tau_j)$ in $\hat{\tau}$ for $j = 1, ..., n$. We call α_i the *i-th projection* of α to $\hat{\pi}$.

Next we want to show that for every n: If a path α in ρ has certain length (depending on n) then there are n modal nodes on α. For that aim we define M to be the maximal length of a path in π without a modal node. Notably, M is smaller than the number of nodes in π. We write $l(\alpha)$ for the length of a path α.

`Claim 2`: Let α be a path in ρ starting from a node a_0. Let $s = |\Pi^0| + |\mathsf{T}^0|$. If $l(\alpha) \geq s \cdot 2^{M+1}$, then there is a modal node a_k on α. Moreover $|\Pi^k| + |\mathsf{T}^k| \leq s \cdot 2^{M+1}$.

`Proof of Claim 2`: Let $\alpha = a_0 a_1 ... a_k$ be a path without a modal node and let $s_j = |\Pi^j| + |\mathsf{T}^j|$ for $j = 0, ..., k$. Let α_i be the i-th projection of α in $\hat{\pi}$ for $i = 1, ..., |\Pi^k|$. Then the first node of α_i is $\mathsf{origin}(\pi)$ for some $\pi \in \Pi^0$. Thus the paths α_i form a forest F_l consisting of $|\Pi^0|$ many trees with roots $\mathsf{origin}(\pi)$ for $\pi \in \Pi^0$. Analogously, the paths β_j in $\hat{\tau}$ form a forest F_r. Let $F = F_l \cup F_r$, then F consists of s trees. All rules in the Focus system have at most two premises. Therefore in all reduction steps of the traversed leaf reduction algorithm, at most two new proofs are added; hence every node in F has at most two children.[11] If the modal rule is never applied in α, the lengths of all branches in F are bound by M. Thus, every tree in F consists of at most 2^M nodes and therefore $|F| \leq s \cdot 2^M$.

If a traversed leaf is transformed in the construction, meaning that a child is added, then also one proof of Π or T is transformed. After that, we may add D rules. But as we reuse D rules for all leaves labeled with the same sequent up to $=_{\mathsf{Set}}$, there are at most as many nodes labeled with D as other nodes. Let k be the length of α, then $s + k/2 \leq |F|$. Hence $k \leq 2|F| - 2s \leq s \cdot 2^{M+1}$, meaning that after at most $s \cdot 2^{M+1}$ transformations all proofs in Π and T must have a modal node at the root. In every step of the construction the size of $\Pi \cup \mathsf{T}$ is increased by at most one, hence $s_{j+1} \leq s_j + 1$ and therefore $s_k \leq s + k \leq 2|F| \leq s \cdot 2^{M+1}$. $\dashv$

`Claim 3`: Let α be a path in ρ starting from the root. For every $n \in \mathbb{N}$, if $l(\alpha) \geq 2^{(M+1)\cdot(n+2)+1}$, then there are at least n modal rules on α.

`Proof of Claim 3`: For the root r of ρ it holds that $s = |\{\hat{\pi}\}| + |\{\hat{\tau}\}| = 2$. We can find modal nodes $b_1, ..., b_n$ on α using Claim 2. Doing so the length of the path from r to b_n can be bound by $\sum_{j=1}^{n} s_j \cdot 2^{M+1} = \sum_{j=1}^{n} 2 \cdot 2^{(M+1)\cdot j} \cdot 2^{M+1} =$

[11] Note that this would not be possible if we would allow contraction rules in π, as a reduction with a contraction would potentially double the size of the multicut.

$2 \cdot \sum_{j=2}^{n+1} 2^{(M+1)\cdot j} \leq 2 \cdot 2^{(M+1)\cdot(n+2)}$, where $s_j = |\Pi^j| + |\mathsf{T}^j|$ corresponds to the number of proofs in the traversed leaf at b_j for $j = 1, ..., n$. ⊣

Note that in the construction of ρ, a modal node is added only if the root of every proof in Π is a modal node. Hence there are also n modal nodes on every projection α_i for $i = 1, ..., m$.

For later use we define the function $f_M(n) := 2^{(M+1)\cdot(n+2)+1}$

Recall that we defined the depth $\mathsf{oriDepth}(v)$ of a traversed leaf v as $\mathsf{oriDepth}(v) = \max\{\mathsf{depth}(\mathsf{origin}(\pi)) \mid \pi \in \Pi\}$, where v is labeled with

$$\frac{[\Pi]\Psi[\mathsf{T}]}{\Gamma_1, ..., \Gamma_m, \Delta_1, ..., \Delta_n}$$

If $\alpha = a_0 \dots a_g$ is a path in ρ, then $\mathsf{oriDepth}$ is not increasing on α. That is, for $i < j < g$ we have $\mathsf{oriDepth}(a_i) \geq \mathsf{oriDepth}(a_j)$. Next we want to show that if α is of a certain length then $\mathsf{oriDepth}$ is at some point strictly decreasing.

Recall that m_l is the number of nodes in $\hat{\pi}$.

`Claim 4`: Let a be a node in ρ with $\mathsf{oriDepth}(a) = d$. Then between a and every traversed leaf v with $\mathsf{oriDepth}(v) = d$ there are at most $m_l + 2^{2\cdot n_l + n_r}$ many modal nodes.

`Proof of Claim 4`: Suppose that v is a traversed leaf and $\alpha = a_0 a_1 \dots$ is the path from $a = a_0$ to v with more than $m_l + 2^{2\cdot n_l + n_r}$ many modal nodes on α. Let b be the lowest node on α, such that there are m_l modal rules between a and b and let β be the subpath of α from b to v.

Let $w_1, ..., w_k$ be a path in $\hat{\pi}$, where $\mathsf{depth}(w_j) = d$ for all $j = 1, ..., k$. If $k \geq m_l$, then w_k is in a proper cluster. Hence, for all $a_j \in \beta$ and $\pi \in \Pi^j$ we have that if $\mathsf{depth}(\mathsf{origin}(\pi)) = d$ then $\mathsf{origin}(\pi)$ is in a proper cluster. In proper clusters, no f rules are applied. In the construction an f rule is only added if for some i the root of π_i is labeled with f and it holds $\mathsf{depth}(\mathsf{origin}(\pi_i)) = \mathsf{oriDepth}(a_j)$. For nodes in β this is not possible, as long the depth of a_j is d. Moreover, for every a_j in β, there is a formula in focus, as there is a $\pi \in \Pi^j$ such that $\mathsf{origin}(\pi)$ is in a proper cluster of depth d and the same formulas in focus are added to ρ in the reductions for f and D. This is the case as all formulas $\psi \in \Psi$ are out of focus in the proofs $\pi \in \Pi_j$.

There are more than $2^{2\cdot n_l + n_r}$ many modal nodes on β. By Claim 1 these modal nodes are labeled with at most $2^{2\cdot n_l + n_r}$ many sequents up to $=_{Set}$. Hence, there are modal nodes c and w in β such that $\mathsf{S}(c) =_{\mathsf{Set}} \mathsf{S}(w)$. On the path from c to w there is a modal rule applied, no f rules are applied and all sequents have a formula in focus. Hence, the path from c to w is successful and the node w would get discharged in the construction. This contradicts the fact that the path α has more than $m_l + 2^{2\cdot n_l + n_r}$ many modal nodes. ⊣

Let $d = \mathsf{depth}(\hat{\pi})$. Iterating Claim 4 we obtain that for every traversed leaf v, the path α from the root of ρ to v has at most $(d+1)\cdot(m_l + 2^{2\cdot n_l + n_r})$ many modal nodes.

Combining this with Claim 3, we obtain that the height of traversed leaves is bound by $f_M((d+1)\cdot(m_l + 2^{2\cdot n_l+n_r}))$. In conclusion, as every constructed tree is finitely branching, after finitely many steps a traversed proof ρ_T without traversed leaves – a Focus-proof – is constructed. □

7.4.4 Example

Let φ, ψ, χ and δ be the following formulas, with their intuitive meaning written on the right:

$$\begin{aligned} \varphi :=& \nu x.\Box x \wedge \mu y.\Diamond y \vee \overline{p}, && \text{"everywhere } \overline{p} \text{ is reachable"} \\ \psi :=& \mu x.\Diamond x \vee p, && \text{"} p \text{ is reachable"} \\ \chi :=& \mu x.\Diamond x \vee q, && \text{"} q \text{ is reachable"} \\ \delta :=& \mu x.\Diamond x \vee (p \wedge \overline{q}), && \text{"} p \wedge \overline{q} \text{ is reachable"} \end{aligned}$$

Note that "p is reachable" means that there is a finite path to a state where p holds. The negation $\overline{\delta}$ of δ translates to $\nu x.\Box x \wedge (\overline{p} \vee q)$ which intuitively means "everywhere p implies q". The negation $\overline{\varphi}$ of φ is $\mu x.\Diamond x \vee \nu y.\Box y \wedge p$ and reads as "there is a reachable state, where everywhere it holds p". It thus holds that $\overline{\varphi}$ and $\overline{\delta}$ imply χ, in other words the sequent φ, δ, χ is valid. We give a Focus-proof using an important cut with ψ:

$$\dfrac{\begin{matrix}\hat{\pi} \\ \varphi, \psi\end{matrix} \qquad \begin{matrix}\hat{\tau} \\ \overline{\psi}, \delta, \chi\end{matrix}}{\varphi, \delta, \chi}\ \mathsf{cut}$$

where the proofs $\hat{\pi}$ and $\hat{\tau}$ are given as follows. We let $\gamma := \mu y.\Diamond y \vee \overline{p}$ and mention that $\overline{\psi} = \nu x.\Box x \wedge \overline{p}$. Note that in $\hat{\tau}$ the cut formula $\overline{\psi}$ is the only formula containing a ν-operator; it is therefore essential in the successful repeat. In this example we omit annotations of u for readability.

$$
\hat{\pi}:\quad
\begin{prooftree}
\AxiomC{$[\varphi^f, \psi]^\dagger$}
\RightLabel{$\Box$}
\UnaryInfC{$\Box\varphi^f, \Diamond\psi$}
\RightLabel{weak}
\UnaryInfC{$\Box\varphi^f, \Diamond\psi, p$}
\AxiomC{}
\RightLabel{Ax1}
\UnaryInfC{$\overline{p}, p$}
\RightLabel{weak}
\UnaryInfC{$\Diamond\gamma, \overline{p}, \Diamond\psi, p$}
\RightLabel{$\vee$}
\UnaryInfC{$\Diamond\gamma \vee \overline{p}^u, \Diamond\psi, p$}
\RightLabel{μ}
\UnaryInfC{$\gamma^f, \Diamond\psi, p$}
\RightLabel{$\wedge$}
\BinaryInfC{$\Box\varphi \wedge \gamma^f, \Diamond\psi, p$}
\RightLabel{ν}
\UnaryInfC{$b:\ \varphi^f, \Diamond\psi, p$}
\RightLabel{$\vee$}
\UnaryInfC{$\varphi^f, \Diamond\psi \vee p$}
\RightLabel{μ}
\UnaryInfC{$a:\ \varphi^f, \psi$}
\RightLabel{$\mathsf{D}_\dagger$}
\UnaryInfC{φ^f, ψ}
\RightLabel{f}
\UnaryInfC{φ^u, ψ}
\end{prooftree}
$$

$$
\hat{\tau}:\quad
\begin{prooftree}
\AxiomC{$[\overline{\psi}^f, \delta, \chi]^\ddagger$}
\RightLabel{$\Box$}
\UnaryInfC{$\Box\overline{\psi}^f, \Diamond\delta, \Diamond\chi$}
\RightLabel{weak}
\UnaryInfC{$s:\ \Box\overline{\psi}^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q$}
\AxiomC{}
\RightLabel{Ax1}
\UnaryInfC{$\overline{p}^f, p, q$}
\AxiomC{}
\RightLabel{Ax1}
\UnaryInfC{$\overline{p}^f, \overline{q}, q$}
\RightLabel{$\wedge$}
\BinaryInfC{$\overline{p}^f, p \wedge \overline{q}, q$}
\RightLabel{weak}
\UnaryInfC{$t:\ \overline{p}^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q$}
\RightLabel{$\wedge$}
\BinaryInfC{$\Box\overline{\psi} \wedge \overline{p}^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q$}
\RightLabel{ν}
\UnaryInfC{$r:\ \overline{\psi}^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q$}
\RightLabel{$\vee$}
\UnaryInfC{$\overline{\psi}^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi \vee q$}
\RightLabel{μ}
\UnaryInfC{$\overline{\psi}^f, \Diamond\delta, p \wedge \overline{q}, \chi$}
\RightLabel{$\vee$}
\UnaryInfC{$\overline{\psi}^f, \Diamond\delta \vee (p \wedge \overline{q}), \chi$}
\RightLabel{μ}
\UnaryInfC{$w:\ \overline{\psi}^f, \delta, \chi$}
\RightLabel{$\mathsf{D}_\ddagger$}
\UnaryInfC{$\overline{\psi}^f, \delta, \chi$}
\RightLabel{f}
\UnaryInfC{$\overline{\psi}^u, \delta, \chi$}
\end{prooftree}
$$

We want to eliminate the important cut as in the construction given in Subsection 7.4.2. We start by defining the traversed proof ρ_I as above by

$$
\begin{array}{c}
[\hat{\pi}]\psi[\hat{\tau}] \\
\varphi, \delta, \chi
\end{array}
$$

We proceed by reducing $\hat{\pi}$. The last applied rule in $\hat{\pi}$ is f and $\mathsf{depth}(\mathsf{origin}(\hat{\pi}))$ is maximal (there is only one proof). We therefore add f to ρ_I. Afterwards the proof is unfolded and then ψ is principal. On the right hand side in $\hat{\tau}$ the f rule is ignored and then the proof is unfolded. The following rules μ and $\vee$ are non-principal and the cut will be pushed upwards. This yields the following traversed proof. Note that $\hat{\pi}_a$ denotes the generated proof from $\hat{\pi}$ rooted at the node a.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}[\hat{\pi}_a]\psi[\hat{\tau}_r]\\ \varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q\end{array}}{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi \vee q}\,\vee}{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \chi}\,\mu}{\varphi^f, \Diamond\delta \vee (p \wedge \overline{q}), \chi}\,\vee}{\varphi^f, \delta, \chi}\,\mu}{\varphi^u, \delta, \chi}\,\mathsf{f}$$

Now ψ is principal on both sides and gets reduced. First the reduction for μ is applied and then for $\vee$, giving the following traversed proof

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}[\hat{\pi}_b]\Diamond\psi, p[\hat{\tau}_s, \hat{\tau}_t]\\ \varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q\end{array}}{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q}\,\mathsf{contr}}{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi \vee q}\,\vee}{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \chi}\,\mu}{\varphi^f, \Diamond\delta \vee (p \wedge \overline{q}), \chi}\,\vee}{\varphi^f, \delta, \chi}\,\mu}{\varphi^u, \delta, \chi}\,\mathsf{f}$$

This traversed proof is not tidy, as $p \not\equiv_C \psi$. We transform it into a tidy traversed proof by adding a cut of lower rank.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}[\hat{\pi}_b]\Diamond\psi[\hat{\tau}_s]\\ \varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q, p\end{array} \qquad \begin{array}{c}\hat{\tau}_t\\ \overline{p}, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q\end{array}}{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q}\,\mathsf{cut}}{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q}\,\mathsf{contr}}{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi \vee q}\,\vee}{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \chi}\,\mu}{\varphi^f, \Diamond\delta \vee (p \wedge \overline{q}), \chi}\,\vee}{\varphi^f, \delta, \chi}\,\mu}{\varphi^u, \delta, \chi}\,\mathsf{f}$$

We continue reducing non-principal rules, until a $\Box$ rule is applied on the left branch and the cut formula gets weakened on the right branch.

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{\begin{matrix}[\hat{\pi}_a]\psi[\hat{\tau}_w]\\ v:\ \varphi^f, \delta, \chi\end{matrix}}{\Box\varphi^f, \Diamond\delta, \Diamond\chi}\ \Box
        }{\Box\varphi^f, \Diamond\delta, \Diamond\chi, p}\ \mathsf{weak}
        \qquad
        \dfrac{
          \dfrac{
            \dfrac{
              \dfrac{\dfrac{}{\overline{p}, p}\ \mathsf{Ax1}}{\Diamond\gamma, \overline{p}, \Diamond\delta, \Diamond\chi, p}\ \mathsf{weak}
            }{\Diamond\gamma \vee \overline{p}^u, \Diamond\delta, \Diamond\chi, p}\ \vee
          }{\gamma^f, \Diamond\delta, \Diamond\chi, p}\ \mu
        }{}
      }{\Box\varphi \wedge \gamma^f, \Diamond\delta, \Diamond\chi, p}\ \wedge
    }{\varphi^f, \Diamond\delta, \Diamond\chi, p}\ \nu
  }{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q, p}\ \mathsf{weak}
  \qquad
  \begin{matrix}\hat{\tau}_t\\ \overline{p}, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q\end{matrix}
}{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q}\ \mathsf{cut}
$$

$$
\dfrac{
  \dfrac{
    \dfrac{
      \dfrac{
        \dfrac{
          \dfrac{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q}{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi, q}\ \mathsf{contr}
        }{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \Diamond\chi \vee q}\ \vee
      }{\varphi^f, \Diamond\delta, p \wedge \overline{q}, \chi}\ \mu
    }{\varphi^f, \Diamond\delta \vee (p \wedge \overline{q}), \chi}\ \vee
  }{c:\ \varphi^f, \delta, \chi}\ \mu
}{\varphi^u, \delta, \chi}\ \mathsf{f}
$$

Now the traversed leaf v is labeled with the same sequent as its ancestor c and the path from c to v is successful. We can therefore insert a $\mathsf{D}_{\ddagger}$ rule at c and discharge v by $\ddagger$. This yields a Focus-proof of φ, δ, χ, where the only cut is of lower rank. Note that in the construction of Definition 7.4.10, the check whether a successful repeat is reached is only carried out when a $\Box$ rule is applied. Thus, the proof would get transformed further until we reach a node labeled with $\Box\varphi^f, \Diamond\delta, \Diamond\chi$ again. Only then we would discharge the leaf.

7.5 Elimination of unimportant cuts

In this section we reduce unimportant cuts to important cuts of the same rank. The strategy is as follows. Given a proper cluster S with unimportant cuts, we inductively unfold leaves in S and push cuts in S upwards, until we find repeats below all cuts. In S there are no occurrences of f rules and cut reductions for all other rules do not affect formulas in focus. Therefore, if we can find a repeat path below all cuts, this will constitute a successful path – assuming it contains a modal rule.

In this process, all cuts that are pushed outside of S become important cuts. Due to the presence of contractions we have to work with a generalization of the cut rule, the mix rule, which allows to introduce the cut formula multiple times and can therefore be seen as a combination of cut and contractions.

7.5.1. Definition. The mix rule is the following rule:

$$\textsf{mix}: \frac{\varphi^u, ..., \varphi^u, \Gamma_0 \qquad \overline{\varphi}^u, ..., \overline{\varphi}^u, \Gamma_1}{\Gamma_0, \Gamma_1}$$

where φ^u does not occur in Γ_0 and $\overline{\varphi}^u$ does not occur in Γ_1. Note that there are finitely many occurrences of φ^u the left premise of mix and there are finitely many occurrences of $\overline{\varphi}^u$ in the right premise and that the number of occurrences of φ^u in the left premise might differ from the number of occurrences of $\overline{\varphi}^u$ in the right premise.

We use the same terminology for mix as we do for cut. For instance, we say that φ is the mix formula of the mix rule depicted above and we define the rank of a mix rule as the rank of its mix formula.

7.5.2. Definition. The cyclic proof system $\textsf{Focus}^{\textsf{m}}$ is defined as the variant of Focus in which the cut rule is replaced by the mix rule.

As mix is a generalization of cut, every Focus-proof may be seen as a $\textsf{Focus}^{\textsf{m}}$-proof by simply replacing cut rules with mix rules. Conversely, every $\textsf{Focus}^{\textsf{m}}$-proof can be translated to a Focus-proof by replacing mix rules with cut rules and contractions. Importantly, the rank of cut/mix formulas is not affected in this translation

We call a sequent Γ *modal*, if all formulas in Γ are modal formulas. We call a $\textsf{Focus}^{\textsf{m}}$-derivation π *local*, if π does not contain the rules $\Box$, f and D.

The following lemma deals with the finitary part of the mix elimination: We can push mixes upwards, until all premises of a mix are modal sequents. First we need to define focused proofs with assumptions. Recall that a closed leaf is either discharged or labeled with an axiom.

7.5.3. Definition. Let $\mathcal{A}$ be a set of sequents. Recall that a $\textsf{Focus}^{\textsf{m}}$-*proof with assumptions* $\mathcal{A}$ is a finite $\textsf{Focus}^{\textsf{m}}$-derivation π, where every leaf of π is either closed or labeled with a sequent in $\mathcal{A}$.

A proof π with assumptions $\mathcal{A}$ is called *focused*, if for every assumption Γ in $\mathcal{A}$ that contains a formula in focus, every node on the path from the root of π to any occurrence of Γ in π contains a formula in focus.

In the $\textsf{Focus}^{\textsf{m}}$ system we replace the cut rule with a mix rule. To eliminate mix rules we employ mix reduction; These reductions can be defined as the cut reductions in Subsection 7.3.4, where cut rules are replaced by mix rules. If we employ a mix reduction to a rule different from f, then formulas do not loose focus. This is the content of the following lemma and can be observed by looking at the mix reductions.

7.5.4. Proposition. *Let π be a focused* $\mathsf{Focus}^{\mathsf{m}}$*-proof with assumptions $\mathcal{A}$. Let π' be obtained from π by applying a mix reduction to π. Then π' is focused.*

7.5.5. Lemma. *Let $\mathcal{A}$ be a set of modal sequents. Let π be a local* $\mathsf{Focus}^{\mathsf{m}}$*-proof with assumptions $\mathcal{A}$ and only one* mix *rule of rank n at the root of π. Then π can be transformed to a local* $\mathsf{Focus}^{\mathsf{m}}$*-proof π' with assumptions $\mathcal{A}$ of the same sequent, where the premises of all* mix *rules are open assumptions in $\mathcal{A}$ and all* mix *rules have rank $\leq n$. Additionally, if π is focused, then π' is focused as well.*

Proof (Sketch):
Note that π does not contain D rules. Therefore π is a finitary proof without cycles and we may employ cut elimination (more precisely: mix elimination) for finitary proofs, see for example [Tak87]. The mix reductions that are used resemble the cut reductions in Subsection 7.3.4, but then for the more general mix rule. As the focus of this chapter is not on cut elimination for finitary proofs, we omit the details. The overall strategy is to inductively "push the mix upwards" in π until one of its premises is an axiom and the mix can be omitted. In our situation we have to consider the additional case where one of the premises of the mix is an assumption in $\mathcal{A}$. In this situation, as $\mathcal{A}$ consists of modal sequents, the mix formula is a modal formula. Then the mix formula is never principal in π: the latter does not contain modal rules. Therefore, we can push the mix upwards even further until both premises of the mix are open assumptions. Proposition 7.5.4 implies that π' is focused if π is focused. ☐

We will now use Lemma 7.5.5 to inductively push mixes upwards until there are enough modal rules below every mix, which guarantees that we find successful repeats below every mix. First, we need a variant of the infinite unfolding of a derivation. Recall that we call a leaf l *outermost* if $c(l)$ is the root of a proper cluster in π. The *unfolding* of an outermost leaf l in π is the derivation obtained from π by replacing l with the maximal subderivation $\pi_{c(l)}$ of π rooted at $c(l)$.[12]

7.5.6. Definition. Let π be a $\mathsf{Focus}^{\mathsf{m}}$-derivation and v be a node in π. The *infinite unfolding of* $\mathsf{comp}(v)$ in π, written π^{*v}, is the $\mathsf{Focus}^{\mathsf{m}}$-derivation obtained from π by recursively unfolding outermost leaves l that are component descendants of v, and removing nodes labeled with $\mathsf{D}_{\dagger}$ whenever no discharged leaf is labeled with †.

7.5.7. Lemma. *Let π be a* Focus*-proof of cut rank n such that all cuts of rank n are unimportant and in the root cluster. Then we can transform π into a* Focus*-proof π' of the same sequent with cut rank $\leq n$, where all cuts are important.*

[12] In order to guarantee that D rules are labeled with unique discharge tokens, in $\pi_{c(l)}$ discharge tokens are replaced by fresh discharge tokens not occurring in π.

Proof:
First we need to introduce some notions. Let ρ be a (possibly infinite) Focus-derivation. The *focused trunk* $\mathsf{FT}(\rho)$ of ρ is the subderivation of ρ with the same root up to the first occurrences of unfocused sequents. Note that the focused trunk of ρ may be infinite. Without loss of generality we may assume that between any node in the focused trunk of a derivation ρ and any D rule there is a modal node, otherwise we can unfold D rules. We let the *extended focused trunk* $\mathsf{eFT}(\rho)$ be the subderivation of ρ with the same root up to the first occurrences of modal nodes outside of the focused trunk of ρ. We define the *k-fragment of ρ* to be the subderivation of $\mathsf{eFT}(\rho)$ with the same root up to the k-th application of a modal rule.

By replacing cuts with mixes we let π be a $\mathsf{Focus}^{\mathsf{m}}$-proof without renaming it. Let Γ be the sequent at the root r of π and let π^{*r} be the infinite unfolding of $\mathsf{comp}(r)$ in π.

We want to push the cuts (also the ones with cut rank $< n$) occurring in $\mathsf{eFT}(\pi^{*r})$ upwards until the mix-free subproof of $\mathsf{eFT}(\pi^{*r})$ is big enough. This is formalized in the following claim.

`Claim 1`: For every k we can construct a $\mathsf{Focus}^{\mathsf{m}}$-derivation π_k of Γ without open assumptions, where all mix rules have mix rank $\leq n$ and are outside of the k-fragment of π_k. Additionally, all mix rules outside of the focused trunk of π_k are important.

`Proof of Claim 1`: We prove the claim by induction on k. For $k = 0$ the derivation π^{*r} satisfies the requirements. Let π_k be a derivation satisfying the requirements of the claim for $k \geq 0$. We construct the desired derivation π_{k+1} by an inner induction on the number l of mix rules in the $(k+1)$-fragment of π_k. If $l = 0$, then π_k already satisfies the requirements for $k+1$ and we are done. If $l > 0$ let C be an occurrence of a mix rule in the $k+1$-fragment of π_k such that there is no mix rule above C in the $(k+1)$-fragment of π_k. Let ρ be the subderivation of the $(k+1)$-fragment of π_k rooted at the conclusion of C and let $\mathcal{A}$ be the set of assumptions of ρ. Then ρ satisfies the assumption of Lemma 7.5.5 and applying the lemma yields a focused proof ρ', where the premises of all mix rules are open assumptions in $\mathcal{A}$. We can replace ρ with ρ' in π_k and apply the following mix reduction to all mix rules in ρ':

$$\dfrac{\dfrac{\varphi,\Gamma}{\Box\varphi,\Diamond\Gamma}\,\Box \qquad \dfrac{\overline{\varphi},...,\overline{\varphi},\gamma,\Gamma}{\Diamond\overline{\varphi},...,\Diamond\overline{\varphi},\Box\gamma,\Diamond\Gamma}\,\Box}{\Box\gamma,\Diamond\Gamma}\,\mathsf{mix} \quad\longrightarrow\quad \dfrac{\dfrac{\varphi,\Gamma \qquad \overline{\varphi},...,\overline{\varphi},\gamma,\Gamma}{\gamma,\Gamma}\,\mathsf{mix}}{\Box\gamma,\Diamond\Gamma}\,\Box$$

This results in a derivation as desired with $l-1$ many occurrences of mix rules in its $(k+1)$-fragment. We can thus apply the inner induction hypothesis to obtain a proof π_{k+1} of Γ without open assumptions, where all mix rules have mix rank $\leq n$ and all mix rules are outside of the k-fragment of π_{k+1}.

All mix rules that where pushed outside of the focused trunk of π_{k+1} are important: no cut reduction puts formulas in focus and therefore the conclusion of those mix rules are unfocused, implying that they are important. Because all mix reductions preserve the cut rank, all mix rules have mix rank $\leq n$. $\dashv$

Let m be the number of modal formulas in $\mathsf{Clos}(\Gamma)$ and let $k := 4^m + 1$. Let π_k be given as in Claim 1. The k-fragment of π_k is mix-free, therefore all conclusions of modal rules in the k-fragment of π_k consist of modal formulas in $\mathsf{Clos}(\Gamma)$, where every formula could occur in focus or out of focus. Thus, conclusions of such modal rules are labeled with at most 4^m many distinct sequents up to $=_{\mathsf{Set}}$. Each branch $\beta = v_0 ... v_n$ in the k-fragment of π_k, where at least one of the children of v_n is in $\mathsf{eFT}(\pi_k)$, contains $4^m + 1$ many modal nodes. Hence, on such a branch β there are nodes v and l such that v and l are labeled with the same sequent up to $=_{\mathsf{Set}}$ and such that on the path from v to l a modal rule is applied. As all nodes in the k-fragment of π_k contain a formula in focus, this implies that the path from v to l is successful.

For each such branch choose the root-most such nodes v and l, insert a $\mathsf{D}_\dagger$ rule at v with fresh discharge token $\dagger$ and replace l with

$$\frac{[\mathsf{S}_v]^\dagger}{\mathsf{S}_l}\ \mathsf{weak,contr}$$

By König's Lemma this results in a finite $\mathsf{Focus^m}$-proof π'. All mix rules in π' are outside of the k-fragment of π_k and thus are important. Hence, the proof π' has mix rank $\leq n$, where all mix rules are important. By replacing all mixes in π' with cuts and contractions we obtain the desired Focus-proof. $\square$

7.6 Elimination of contractions

It is well-known that contractions pose one of the major difficulties to cut elimination. In our case, in the elimination of important cuts, cut reductions of the multicut with contractions may double the size of the multicut. This ruins our termination proof as we rely on a bound on the size of the multicut. We thus first opt to eliminate contractions from cut-free proofs and aim to prove the following lemma.

7.6.1. LEMMA. *Let π be a cut-free Focus-proof. Then there is a cut-free and contraction-free Focus-proof π' of the same sequent.*

The elimination of contractions shares similarities with the elimination of cuts, in the sense that we treat contractions in trivial clusters differently from those in proper clusters. In the first step of our procedure, we push all contractions in trivial clusters upwards, until all contractions are in proper clusters. For this to work, we need to be able to swap occurrences of contractions with the rules $\vee$, $\wedge$ and η. To that end, we first show that these rules are invertible in Subsection 7.6.1. Contractions in proper cluster are eliminated in a similar way as unimportant cuts: We push the contraction upwards until we can find successful repeats below them. The proof of termination of this process is more complicated, as we need to find repeats without introducing new contractions. For this purpose we refer to the results on well-quasi-orders from Subsection 7.1.2.

Recall that the depth of a node v in a proof π is the maximal number of proper clusters on an upward path starting from v.

7.6.2. Definition. We define the *shallow depth* of a node v in a proof π as the maximal length of a path in π starting at v and not containing nodes in proper clusters, where the shallow depth of v equals 0, if v is in a proper cluster. The contr-*free shallow depth* of v is defined as the shallow depth without counting nodes labeled with contr.

Let C be an occurrence of a contr rule with conclusion v in a Focus-proof π. The *depth* and *shallow depth* of C are defined as the depth and shallow depth of v, respectively. The *contraction depth* of a proof π is defined as the maximal depth of an occurrence of a contr rule in π.

7.6.1 Strongly invertible rules

7.6.3. Definition. Let π be a Focus-proof. We call π a $\mathsf{Focus}^{\mathsf{c}}$-proof if all occurrences of contractions in π are in *proper clusters.*

This definition might seem unusual at first glance, but recall that our aim is to push contractions in trivial clusters upwards. This is only possible if the rules $\vee$, $\wedge$ and η are invertible. However, their invertibility relies on the absence of contractions in trivial clusters higher up in the proof tree. For this reason, we disallow such contractions in $\mathsf{Focus}^{\mathsf{c}}$-proofs.

We say that a Focus-proof π is *k-focused* if every node of depth $\geq k$ has a formula in focus.

7.6.4. Definition. Let $\dfrac{\Gamma_1 \quad \cdots \quad \Gamma_n}{\Gamma}\ \mathsf{R}$ be a rule in Figure 7.1. We call R *strongly invertible* in $\mathsf{Focus}^{\mathsf{c}}$, if every $\mathsf{Focus}^{\mathsf{c}}$-proof π of Γ can be transformed, for every $i = 1, ..., n$, to a $\mathsf{Focus}^{\mathsf{c}}$-proof π_i of Γ_i with the same depth, shallow depth and such that for every k, if π is k-focused then π_i is k-focused as well.

7.6.5. Lemma. *The rules $\vee$, $\wedge$ and η are strongly invertible in $\mathsf{Focus}^{\mathsf{c}}$.*

Proof:
We only prove that $\wedge$ is strongly invertible, the proofs for the other rules are

similar. Let π be a proof of $\varphi \wedge \psi^a, \Gamma$ with depth m and shallow depth l. The proof goes by induction on m with an inner induction on l.

First assume that $l = 0$, meaning that the root cluster is proper. Then π is the following proof on the left, where π_0 is the subderivation of π with the same root as π up to the first occurrences of (i) $\wedge$ rules with $\varphi \wedge \psi^b$ principal or (ii) nodes outside the root cluster. We transform π into a proof of φ^a, Γ as follows, where π_0^φ is obtained from π_0 by replacing $\varphi \wedge \psi^c$ with φ^c at every node, analogously for π_0^ψ.

$$
\begin{array}{c}
\dfrac{\begin{array}{cc}
[\varphi \wedge \psi^a, \Gamma]^\dagger & [\varphi \wedge \psi^a, \Gamma]^\dagger \\
\vdots\,\pi_l & \vdots\,\pi_r \\
\varphi^b, \Delta & \psi^b, \Delta
\end{array}}{\varphi \wedge \psi^b, \Delta}\ \wedge \\
\vdots\,\pi_0 \\
\dfrac{\varphi \wedge \psi^a, \Gamma}{\varphi \wedge \psi^a, \Gamma}\ \mathsf{D}_\dagger
\end{array}
\quad\longrightarrow\quad
\begin{array}{c}
\dfrac{[\varphi^a, \Gamma]^\dagger \qquad
\begin{array}{c}
\dfrac{[\varphi^a, \Gamma]^\dagger \quad [\psi^a, \Gamma]^\ddagger}{\varphi \wedge \psi^a, \Gamma}\ \wedge \\
\vdots\,\pi_r \\
\psi^b, \Delta \\
\vdots\,\pi_0^\psi \\
\dfrac{\psi^a, \Gamma}{\psi^a, \Gamma}\ \mathsf{D}_\ddagger
\end{array}}{\varphi \wedge \psi^a, \Gamma}\ \wedge \\
\vdots\,\pi_l \\
\varphi^b, \Delta \\
\vdots\,\pi_0^\varphi \\
\dfrac{\varphi^a, \Gamma}{\varphi^a, \Gamma}\ \mathsf{D}_\dagger
\end{array}
$$

Note that for any node outside the root cluster labeled with $\varphi \wedge \psi^a, \Gamma$, we inductively obtain proofs of φ^a, Γ and of ψ^a, Γ of the same depth. Therefore the above transformation yields a proof of φ^a, Γ of depth m and shallow depth 0. An analogous transformation gives a proof of ψ^a, Γ.

If $l > 0$ we proceed with a case distinction on the applied rule R at the root of π. Note that $\mathsf{R} \neq \mathsf{contr}$ because π is a $\mathsf{Focus}^{\mathsf{c}}$-proof. If $\mathsf{R} = \wedge$ with principal formula $\varphi \wedge \psi^a$, then the proofs rooted at the premises of R are the desired proofs. If $\mathsf{R} = \mathsf{weak}$ the transformation is obvious. If any other rule is applied, we transform the proof as follows, where $\pi_1', ..., \pi_n'$ are obtained from respectively $\pi_1, ..., \pi_n$ by applying the induction hypothesis.

$$\dfrac{\overset{\pi_1}{\varphi \wedge \psi^b, \Gamma_1} \quad \cdots \quad \overset{\pi_n}{\varphi \wedge \psi^b, \Gamma_n}}{\varphi \wedge \psi^a, \Gamma}\ \mathsf{R} \quad \longrightarrow \quad \dfrac{\overset{\pi_1'}{\varphi^b, \psi^b, \Gamma_1} \quad \cdots \quad \overset{\pi_n'}{\varphi^b, \psi^b, \Gamma_n}}{\varphi^a, \psi^a, \Gamma}\ \mathsf{R}$$

It is clear that in all cases the resulting proof is k-focused if π is k-focused. We thus have shown that $\wedge$ is strongly invertible. □

7.6.2 Reduction of contractions

In this subsection, we introduce the contraction reductions used in the rest of this section. The reader may prefer to skip this part and return to the reductions as they arise. Whenever the strong invertibility of a rule R is employed, we denote it by a doubled line and the rule name R^I. For readability, we omit annotations whenever they are not affected by the reduction.

Principal reductions

$$\dfrac{\dfrac{\overset{\pi'}{\varphi, \psi, \varphi \vee \psi, \Gamma}}{\varphi \vee \psi, \varphi \vee \psi, \Gamma}\ \vee}{\varphi \vee \psi, \Gamma}\ \mathsf{contr} \quad \longrightarrow \quad \dfrac{\dfrac{\dfrac{\dfrac{\overset{\pi'}{\varphi, \psi, \varphi \vee \psi, \Gamma}}{\varphi, \varphi, \psi, \psi, \Gamma}\ \vee^I}{\varphi, \varphi, \psi, \Gamma}\ \mathsf{contr}}{\varphi, \psi, \Gamma}\ \mathsf{contr}}{\varphi \vee \psi, \Gamma}\ \vee$$

$$\dfrac{\dfrac{\overset{\pi_0}{\varphi, \varphi \wedge \psi, \Gamma} \quad \overset{\pi_1}{\psi, \varphi \wedge \psi, \Gamma}}{\varphi \wedge \psi, \varphi \wedge \psi, \Gamma}\ \wedge}{\varphi \wedge \psi, \Gamma}\ \mathsf{contr} \quad \longrightarrow \quad \dfrac{\dfrac{\dfrac{\overset{\pi_0}{\varphi, \varphi \wedge \psi, \Gamma}}{\varphi, \varphi, \Gamma}\ \wedge^I}{\varphi, \Gamma}\ \mathsf{contr} \quad \dfrac{\dfrac{\overset{\pi_1}{\psi, \varphi \wedge \psi, \Gamma}}{\psi, \psi, \Gamma}\ \wedge^I}{\psi, \Gamma}\ \mathsf{contr}}{\varphi \wedge \psi, \Gamma}\ \wedge$$

$$\dfrac{\dfrac{\overset{\pi'}{\varphi[\eta x.\varphi/x], \eta x.\varphi, \Gamma}}{\eta x.\varphi, \eta x.\varphi, \Gamma}\ \eta}{\eta x.\varphi, \Gamma}\ \mathsf{contr} \quad \longrightarrow \quad \dfrac{\dfrac{\dfrac{\overset{\pi'}{\varphi[\eta x.\varphi/x], \eta x.\varphi, \Gamma}}{\varphi[\eta x.\varphi/x], \varphi[\eta x.\varphi/x], \Gamma}\ \eta^I}{\varphi[\eta x.\varphi/x], \Gamma}\ \mathsf{contr}}{\eta x.\varphi, \Gamma}\ \eta$$

$$\dfrac{\dfrac{\begin{matrix}\pi' \\ \varphi, \psi, \psi, \Gamma\end{matrix}}{\Box\varphi, \Diamond\psi, \Diamond\psi, \Diamond\Gamma}\ \Box}{\Box\varphi, \Diamond\psi, \Diamond\Gamma}\ \mathsf{contr} \quad\longrightarrow\quad \dfrac{\dfrac{\begin{matrix}\pi' \\ \varphi, \psi, \psi, \Gamma\end{matrix}}{\varphi, \psi, \Gamma}\ \mathsf{contr}}{\Box\varphi, \Diamond\psi, \Diamond\Gamma}\ \Box$$

$$\dfrac{\dfrac{\begin{matrix}\pi' \\ \varphi, \Gamma\end{matrix}}{\varphi, \varphi, \Gamma}\ \mathsf{weak}}{\varphi, \Gamma}\ \mathsf{contr} \quad\longrightarrow\quad \begin{matrix}\pi' \\ \varphi, \Gamma\end{matrix}$$

Non-principal reductions

Case R Let R be a rule different from $\Box$, u, f and D. Then we reduce R as follows.

$$\dfrac{\dfrac{\begin{matrix}\pi_1 \\ \varphi^a, \varphi^a, \Gamma_1\end{matrix} \quad \cdots \quad \begin{matrix}\pi_n \\ \varphi^a, \varphi^a, \Gamma_n\end{matrix}}{\varphi^a, \varphi^a, \Gamma}\ \mathsf{R}}{\varphi^a, \Gamma}\ \mathsf{contr}$$

$$\longrightarrow\quad \dfrac{\dfrac{\begin{matrix}\pi_1 \\ \varphi^a, \varphi^a, \Gamma_1\end{matrix}}{\varphi^a, \Gamma_1}\ \mathsf{contr} \quad \cdots \quad \dfrac{\begin{matrix}\pi_n \\ \varphi^a, \varphi^a, \Gamma_n\end{matrix}}{\varphi^a, \Gamma_n}\ \mathsf{contr}}{\varphi^a, \Gamma}\ \mathsf{R}$$

Case D We unfold D rules in the same way that we did in the cut reductions.

$$v : \quad \dfrac{\dfrac{\begin{matrix}\pi_0 \\ \varphi^a, \varphi^a, \Gamma\end{matrix}}{\varphi^a, \varphi^a, \Gamma}\ \mathsf{D}_\dagger}{\varphi^a, \Gamma}\ \mathsf{contr} \quad\longrightarrow\quad \dfrac{\begin{matrix}\pi_0' \\ \varphi^a, \varphi^a, \Gamma\end{matrix}}{\varphi^a, \Gamma}\ \mathsf{contr}$$

where π_0' is obtained from π_0 by replacing every discharged leaf labeled with $\dagger$ with π_v, where v is the left premise of the contr rule.[13]

Case f Because proofs are minimally focused, the premise of an f rule is labeled

[13] As in the cut reductions, we replace discharge tokens $\ddagger$ with fresh discharge tokens, whenever a $\mathsf{D}_\ddagger$ rule is duplicated.

with D. We reduce those contractions as follows.

$$
\dfrac{\dfrac{\dfrac{\begin{matrix}[\varphi^a,\varphi^a,\Gamma']^\dagger\\ \vdots\\ \pi'\\ \vdots\\ \varphi^a,\varphi^a,\Gamma'\end{matrix}}{\varphi^a,\varphi^a,\Gamma'}\,\mathsf{D}_\dagger}{\varphi^u,\varphi^u,\Gamma}\,\mathsf{f}}{\varphi^u,\Gamma}\,\mathsf{contr}
\quad\longrightarrow\quad
\dfrac{\dfrac{\dfrac{\begin{matrix}\dfrac{[\varphi^a,\Gamma']^\dagger}{\varphi^a,\varphi^a,\Gamma'}\,\mathsf{weak}\\ \vdots\\ \pi'\\ \vdots\\ \varphi^a,\varphi^a,\Gamma'\end{matrix}}{\varphi^a,\Gamma'}\,\mathsf{contr}}{\varphi^a,\Gamma'}\,\mathsf{D}_\dagger}{\varphi^u,\Gamma}\,\mathsf{f}
$$

Note that we treat D rules that are premises of f rules differently than those D rules that are not. The reason for that is, that the former occur at the root of proper clusters and the latter occur inside proper clusters. We deal with D rules at the root of proper clusters when reducing contractions in trivial clusters, and with D rules inside proper clusters when reducing contractions in proper clusters. The treatment of those different occurrences of contractions differs as will see in the next subsections.

Case u We only consider minimally focused proofs. This implies that premises of u rules in trivial clusters are out of focus. In proper clusters, a formula φ is put out of focus iff φ is of a non-maximal rank. Let v be a node in a proper cluster labeled with a contraction rule with principal formula φ. We may assume that the formula φ is not put out of focus at the premise of the contraction rule – if φ is of non-maximal rank it would already be put out of focus at v. Therefore the annotations of both occurrences of φ in the premise of a u rule are the same. We reduce those u rules as follows.

$$
\dfrac{\dfrac{\begin{matrix}\pi'\\ \varphi^b,\varphi^b,\Gamma'\end{matrix}}{\varphi^a,\varphi^a,\Gamma}\,\mathsf{u}}{\varphi^a,\Gamma}\,\mathsf{contr}
\quad\longrightarrow\quad
\dfrac{\dfrac{\begin{matrix}\pi'\\ \varphi^b,\varphi^b,\Gamma'\end{matrix}}{\varphi^b,\Gamma'}\,\mathsf{contr}}{\varphi^a,\Gamma}\,\mathsf{u}
$$

7.6.3 Contractions in trivial clusters

7.6.6. LEMMA. *Let π be a cut-free* Focus*-proof of contraction depth m. Then π can be transformed to a cut-free* Focus*-proof π' of contraction depth $\leq m$ of the same sequent, where all contractions are in proper clusters.*

Proof:
Let $\mathsf{C}_1, ..., \mathsf{C}_n$ be the occurrences of contraction rules in trivial clusters in π with

respective contr-free shallow depths $d_1, ..., d_n$. We prove the lemma by induction on the Dershowitz–Manna ordering on the multiset $\{d_1, ..., d_n\}$ induced by the natural order on $\mathbb{N}$.

Let C be an occurrence of a contraction rule $\dfrac{\varphi^a, \varphi^a, \Gamma}{\varphi^a, \Gamma}\,\mathsf{contr}$ in a trivial cluster with contr-free shallow depth d, such that there is no contraction rule in a trivial cluster in π above C. Note that the subproof of π rooted at the premise of C is a $\mathsf{Focus^c}$-proof. Let the premise of C be labeled with R. We proceed with a case distinction based on the shape of R and apply contraction reductions to R from Subsection 7.6.2.

If $\mathsf{R} = \mathsf{f}$, then we perform the following transformation:

$$
\begin{array}{c}
[\varphi^a, \varphi^a, \Gamma']^\dagger \\
\vdots \\
\pi' \\
\vdots \\
\dfrac{\dfrac{\dfrac{\varphi^a, \varphi^a, \Gamma'}{\varphi^a, \varphi^a, \Gamma'}\,\mathsf{D}_\dagger}{\varphi^u, \varphi^u, \Gamma}\,\mathsf{f}}{\varphi^u, \Gamma}\,\mathsf{contr}
\end{array}
\quad\longrightarrow\quad
\begin{array}{c}
\dfrac{[\varphi^a, \Gamma']^\dagger}{\varphi^a, \varphi^a, \Gamma'}\,\mathsf{weak} \\
\vdots \\
\pi' \\
\vdots \\
\dfrac{\dfrac{\dfrac{\varphi^a, \varphi^a, \Gamma'}{\varphi^a, \Gamma'}\,\mathsf{contr}}{\varphi^a, \Gamma'}\,\mathsf{D}_\dagger}{\varphi^u, \Gamma}\,\mathsf{f}
\end{array}
$$

This results in a Focus-proof with one less contraction rule in a trivial cluster and we can therefore apply the induction hypothesis.

Because π is minimally focused and C is in a trivial cluster, the premise of C cannot be labeled with D. If $\mathsf{R} \neq \mathsf{f}$ and φ^a is not principal in R, then we can exchange the order in which the rules R and contr are applied and thus reduce d.

Otherwise, assume that φ^a is principal in the rule $\mathsf{R} = \vee$. We transform the proof π as follows:

$$
\dfrac{\dfrac{\begin{array}{c}\pi' \\ \varphi, \psi, \varphi \vee \psi, \Gamma\end{array}}{\varphi \vee \psi, \varphi \vee \psi, \Gamma}\,\vee}{\varphi \vee \psi, \Gamma}\,\mathsf{contr}
\quad\longrightarrow\quad
\dfrac{\dfrac{\dfrac{\dfrac{\begin{array}{c}\pi' \\ \varphi, \psi, \varphi \vee \psi, \Gamma\end{array}}{\varphi, \varphi, \psi, \psi, \Gamma}\,\vee^I}{\varphi, \varphi, \psi, \Gamma}\,\mathsf{contr}}{\varphi, \psi, \Gamma}\,\mathsf{contr}}{\varphi \vee \psi, \Gamma}\,\vee
$$

where $\vee^I$ describes an application of the invertibility of $\vee$ (Lemma 7.6.5) and thus does not increase the contr-free shallow depth. Both introduced contraction rules have contr-free shallow depth $d - 1$, thus we may apply the induction hypothesis.

If φ^a is principal in a different rule, we can perform similar transformations using the invertibility results shown in Lemma 7.6.5. Note that in all those transformations the depth of π remained the same. □

7.6.4 Contractions in proper clusters

The idea to reduce the depth of contractions in proper clusters is to push contractions upwards until we find successful repeats below all contractions. This resembles the elimination of unimportant cuts in Lemma 7.5.7. Here, we have to be a bit more careful, as the reductions of the contraction rule rely on the invertibility of $\vee$, $\wedge$ and $\eta-$ which only holds for $\mathsf{Focus}^{\mathsf{c}}$-proofs. We therefore have to make sure that we apply reductions only at those nodes v, where no contraction rules appear in trivial clusters above v. We therefore opt to only unfold leaves in the root component when needed, compared to the proof of Lemma 7.5.7, where we already started the process with the infinite unfolding of the root component.

In Lemma 7.5.7 the algorithm stops when for every path β we found a pair of nodes v, l such that v is an ancestor of l, the path from v to l is successful and $S_v =_{\mathsf{Set}} S_l$. Then we could apply weakenings and contractions at l to obtain a successful repeat. Now we do not want to introduce contr rules and we therefore only demand that $S_v \subseteq S_l$: In this case we only need to apply weakenings to obtain a successful repeat.

In the proof of termination, finding such nodes v, l becomes more tricky. Our solution is to use results on well-quasi-orders: Let $\mathcal{M}_X$ be the set of sequents occurring in a cut-free proof and let $[\![\cdot]\!]_\infty$ be the infinity norm defined as $[\![A]\!]_\infty := \max\{\sigma_A(\varphi) \mid \varphi \in X\}$. In Lemma 7.1.8 we saw that $(\mathcal{M}_X, \subseteq, [\![\cdot]\!]_\infty)$ is a normed well-quasi-order and so we can find a bound N, such that on all paths longer than N we can find such nodes v, l as desired.

To guarantee that on every repeat path there is a modal node we need the following technical lemma. It states that in a cut-free and contraction-free proof all repeat paths contain a modal node.

7.6.7. LEMMA. *Let β be a repeat path in a* Focus*-derivation that does not contain nodes labeled with* cut, contr *and* f*. Then β contains a node labeled with* $\Box$*.*

Proof:
Let φ and ψ be formulas. We let $\varphi \to_C^- \psi$ if $\varphi \to_C \psi$ and φ is not a modal formula. The relation $\twoheadrightarrow_C^-$ is defined as the reflexive and transitive closure of $\to_C^-$. Note that all formulas are assumed to be guarded. Therefore for no formulas φ and ψ with $\varphi \neq \psi$ it holds that $\varphi \twoheadrightarrow_C^- \psi$ and $\psi \twoheadrightarrow_C^- \varphi$.

We let $\mathsf{Clos}^-(\varphi)$ be the least superset of $\{\varphi\}$ that is closed under $\twoheadrightarrow_C^-$. We define $\mathsf{nmf}(\varphi) := |\mathsf{Clos}^-(\varphi)|$ to be the number of non-modal formulas in $\mathsf{Clos}^-(\varphi)$. For a sequent Γ we define $\mathsf{nmf}(\Gamma)$ to be the *multiset* $\{\mathsf{nmf}(\varphi) \mid \varphi \in \Gamma^-\}$. We let $<_{\mathrm{DM}}$ be the Dershowitz-Manna ordering on multisets of natural numbers induced by the natural order on $\mathbb{N}$. Let Γ be a premise and Γ' be the conclusion of a rule R. Then,

1. if $\mathsf{R} = \vee, \wedge, \eta$ or weak then $\mathsf{nmf}(\Gamma) <_{\mathrm{DM}} \mathsf{nmf}(\Gamma')$,
2. if $\mathsf{R} = \Box$ then $\mathsf{nmf}(\Gamma) \geq_{\mathrm{DM}} \mathsf{nmf}(\Gamma')$, and
3. if $\mathsf{R} = \mathsf{u}$ then $\mathsf{nmf}(\Gamma) = \mathsf{nmf}(\Gamma')$.

This can easily be verified. For instance, for the rule $\wedge$ this holds as $\mathsf{nmf}(\varphi) < \mathsf{nmf}(\varphi \wedge \psi)$ because of $\varphi \not\twoheadrightarrow_C^- \varphi \wedge \psi$. Now let β be a repeat path where all nodes on β are labeled with the rules $\vee$, $\wedge$, $\Box$, η, u or weak. First note that β cannot only consist of nodes labeled with u. All other rules apart from $\Box$ increase $\mathsf{nmf}(\Gamma)$ and the only rule that reduces $\mathsf{nmf}(\Gamma)$ is $\Box$. Hence, there has to be a node labeled with $\Box$ on β. □

Let the root r of a Focus-proof π be labeled with $\mathsf{D}_\dagger$. Recall that the *unfolding of* π is obtained from π by replacing every discharged leaf labeled with $\dagger$ with π, and removing the node r.

Note that the unfolding π' of a proof π may have a bigger depth than π. However, the depth of nodes without formulas in focus does not increase. Recall that a Focus-proof π is *k-focused* if every node of depth $\geq k$ has a formula in focus.

7.6.8. LEMMA. *Let $k \in \mathbb{N}$ and let π be a k-focused* Focus*-proof. Let the root of π be labeled with $\mathsf{D}_\dagger$ and let π' be the unfolding of π. Then π' is k-focused.*

Proof:
Every node v in π' of depth $\geq k$ is a copy of a node u in π of depth $\geq k$. □

7.6.9. LEMMA. *Let $k \in \mathbb{N}$ and let π be a k-focused* Focus*-proof. Let π' be obtained from π by applying a contraction reduction from Subsection 7.6.2 to the root of π. Then π' is k-focused.*

Proof:
The case of a D rule follows from Lemma 7.6.8. In the other cases we use the fact that whenever the strong invertibility of a rule is applied, k-focused proofs are transformed to k-focused proofs, see Lemma 7.6.5. The lemma then follows straightforwardly. □

7.6.10. LEMMA. *Let π be a cut-free* Focus*-proof of depth m where contractions only occur in proper clusters and such that the root cluster of π is proper. Then π can be transformed to a cut-free* Focus*-proof π' of the same sequent where all contractions have depth $< m$.*

Proof:
We start by defining some notions. Given a Focus-proof ρ, let $\rho_{\geq m}$ be the set of nodes in ρ of depth $\geq m$. If the depth of ρ is $\geq m$, then $\rho_{\geq m}$ is non-empty and forms a subtree of ρ containing the root of ρ. Let $\rho^-_{\geq m}$ be the maximal connected subset of $\rho_{\geq m}$ with the same root up to (and including) the first occurrences of contraction or discharge rules.

We call a maximal path $\beta = v_0 ... v_n$ in $\rho^-_{\geq m}$ *critical*, if at least one of the children of v_n is in $\rho_{\geq m}$. We call a critical path β *tamed* if there are nodes v and l on β such that v is a proper ancestor of l with $\mathsf{S}_v \subseteq \mathsf{S}_l$, and *untamed* otherwise.

Now consider the Focus-proof π. Let Γ be the sequent at the root r of π. Then $\pi_{\geq m}$ is simply the root cluster of π and $\pi^-_{\geq m}$ consists of the single node r.

We transform π using the following algorithm:[14]

1. If all critical paths in $\pi^-_{\geq m}$ are tamed, then stop.

2. Else if there is a node v in a trivial cluster in π labeled with an occurrence C of a contr rule such that no contr rule is applied in a trivial cluster above v, then apply a reduction from Subsection 7.6.2 to C.

3. Else take a node v in $\pi^-_{\geq m}$ labeled with $\mathsf{D}_\dagger$ and unfold it, meaning that every discharged leaf l labeled with $\dagger$ is replaced by π_v and the node v is removed.

As π is a proof, at some point a principal reduction to a contraction rule is applied and therefore at some point the length of all critical paths in $\pi^-_{\geq m}$ increases. To prove termination, it therefore suffices to show that every critical path of a certain length is tamed.

Every node in $\pi^-_{\geq m}$ is labeled with a sequent consisting of formulas in $\mathsf{Clos}(\Gamma)$, hence by a multiset over the finite set $X := \mathsf{Clos}(\Gamma)$. As shown in Lemma 7.1.8 we have that $\mathsf{M}_X = (\mathcal{M}_X, \subseteq, [\![\cdot]\!]_\infty)$ is a normed well-quasi-order. Any untamed critical path in $\pi^-_{\geq m}$ corresponds to a bad sequence over M_X. We can therefore use the bounds on controlled bad sequences over M_X given by Lemma 7.1.5 to obtain a bound on the length of critical paths in $\pi^-_{\geq m}$. It remains to find a control function and a starting value.

Given a premise Δ and the conclusion Δ' of a rule R it holds that $[\![\Delta]\!]_\infty \leq [\![\Delta']\!]_\infty + 2$. Thus we can choose the control function $f : n \mapsto n + 2$, let $t := [\![\Gamma]\!]_\infty$ be the starting value and let $N := L[\mathsf{M}_X, f](t)$. Any untamed critical path in $B\pi^-_{\geq m}$ corresponds to an (f, t)-controlled bad sequence over M_X. But the length of (f, t)-controlled bad sequences over M_X is bound by N and therefore the length of untamed critical paths in $\pi^-_{\geq m}$ is bound by N as well. This suffices to show termination.

[14] Note that π may change in the process and consequently $\pi_{\geq m}$ and $\pi^-_{\geq m}$ may change as well. In particular, the root cluster may become trivial, and thus $\pi_{\geq m}$ may consist of multiple clusters.

Let $\tilde{\pi}$ be the proof obtained by this algorithm. For any critical path β in $\tilde{\pi}^{-}_{\geq m}$ let v and l be the root-most nodes such that v is a proper ancestor of l and $\mathsf{S}_v \subseteq \mathsf{S}_l$. We add a node labeled with $\mathsf{D}_\dagger$ at v and replace l with

$$\frac{[\mathsf{S}_v]^\dagger}{\mathsf{S}_l}\ \mathsf{weak}$$

This results in a Focus-derivation ρ. All remaining nodes labeled with contractions were pushed out of $\rho_{\geq m}$ and therefore have depth $< m$. It remains to show that all repeat leaves are discharged. Because of Lemma 7.6.8 and Lemma 7.6.9 all sequents in $\rho_{\geq m}$ have a formula in focus. Clearly no f rules were introduced, hence no node in $\rho^{-}_{\geq m}$ is labeled with f. All newly introduced repeat paths β_l do not contain nodes labeled with cut or contr, therefore Lemma 7.6.7 implies that there is a modal node on β_l. Hence, all repeat paths are successful and we obtain a cut-free proof of the same sequent, where all contractions have depth $< m$. □

We can now combine the Lemmas 7.6.6 and 7.6.10 and prove the elimination of contractions.

Proof of Lemma 7.6.1:
We prove the Lemma by induction on the contraction depth m of π. By Lemma 7.6.6 we can transform π to a proof π_0 with contraction depth m, where all contractions are in proper clusters. We can apply Lemma 7.6.10 to every subproof of π_0 rooted at a proper cluster containing contractions of depth m. This yields a cut-free Focus-proof π' of the same sequent with contraction depth $< m$. The statement then follows by the induction hypothesis. □

7.7 Cut-elimination theorem

We put together the elimination of important and unimportant cuts and obtain cut elimination for the Focus system. There is one extra step that we have to carry out, namely to push important cuts upwards until the cut formula is a fixpoint formula.

7.7.1. DEFINITION. Let π be a Focus-proof and C be an important cut in π. We call C *essential* if its cut formula is a fixpoint-formula.

7.7.2. LEMMA. *Let π be a contraction-free* Focus*-proof of cut rank n, where the only cut of rank n is important and at the root. Then there is a* Focus*-proof π' of the same sequent with cut rank $\leq n$, where all cuts are essential.*

Proof:
Using the cut reductions from Subsection 7.3.4 we can push the cuts of rank n

upwards. All cut reductions apart from η do not increase the syntactic size of the cut formula and in the cut reduction for $\Box$ the syntactic size of the cut formula decreases. As on every repeat path there is an application of $\Box$, the syntactic size of cut formulas decreases until all cut formulas of rank n are fixpoint-formulas. $\Box$

7.7.3. THEOREM (Cut elimination). *We can transform every* Focus*-proof* π *into a cut-free* Focus*-proof* π' *of the same sequent.*

Proof:
Let $P_1, ..., P_k$ be the proper clusters in π that do contain cut rules, where n_j^u is the maximal rank of a cut in P_j for $j = 1, .., k$. Let $S_1, ..., S_m$ be the trivial clusters in π that do contain an essential cut rule, where S_i contains a cut of rank n_j^e for $j = 1, ..., m$. Let $T_1, ..., T_l$ be the trivial clusters in π that do contain an important, but not essential cut rule, where T_i contains a cut of rank n_j^i for $j = 1, ..., l$.

We define the *cut order* $o(\pi)$ of π as the multiset

$$\{3 \cdot n_1^u + 2, ..., 3 \cdot n_k^u + 2, 3 \cdot n_1^i + 1, ..., 3 \cdot n_l^i + 1, 3 \cdot n_1^e, ..., 3 \cdot n_m^e\}.$$

Let $<_{\mathrm{DM}}$ be the Dershowitz-Manna ordering on multisets of natural numbers induced by the natural order on $\mathbb{N}$. We prove the lemma by $<_{\mathrm{DM}}$-induction on $o(\pi)$. The definition of $o(\pi)$ guarantees that $o(\pi)$ becomes $<_{\mathrm{DM}}$-smaller if either

(i) one proper cluster with unimportant cuts of rank n is replaced by multiple important cuts in trivial clusters with rank $\leq n$, or

(ii) one non-essential, important cut of rank n in a trivial cluster is replaced by multiple essential cuts of rank n, or

(iii) one essential cut of rank n in a trivial cluster is replaced by multiple cuts of rank $< n$.

Let π_0 be a subproof of π, where all cuts are in the root cluster of π_0 and let n be the cut rank of π_0. If the root cluster is proper then all cuts in the root cluster of π_0 are unimportant. Otherwise there is one important cut at the root of π_0.

In the first case Lemma 7.5.7 yields a proof π_1 with cut rank n, where all cuts of rank n are important. In the second case, Lemma 7.6.1 transforms π_0 to π_0', where π_0' does not contain contractions and has one important cut with rank n at the root. If the cut is not essential, then Lemma 7.7.2 yields a proof π_1 with cut rank n, where all cuts are essential. Otherwise the cut is essential and Lemma 7.4.1 yields a proof π_1 with cut rank $< n$.

In all cases, we substitute π_0 with π_1 in π and obtain a proof π', where $o(\pi') <_{\mathrm{DM}} o(\pi)$. We can apply the induction hypothesis in order to obtain a cut-free proof. □

7.7.4. Corollary. *We can transform every* Focus*-proof* π *into a cut-free and contraction-free* Focus*-proof* π' *of the same sequent.*

Proof:
Combine Theorem 7.7.3 and Lemma 7.6.1. □

7.8 Conclusion

We presented a syntactic cut-elimination procedure for a cyclic proof system for the alternation-free modal μ-calculus. Several possible extensions and adaptations of the presented approach are worth mentioning.

First, the result can be readily extended to the *polymodal case*, where a set of modalities is considered.

Perhaps most interesting is the applicability to temporal and dynamic logics – such as PDL, LTL, and CTL – since these can be viewed as fragments of the alternation-free μ-calculus. Although our cut-elimination result does not apply to them directly, a similar method can be used. To illustrate this, consider PDL. As discussed in Section 2.5, PDL is equivalent to the completely additive μ-calculus $\mathcal{L}_\mu^{ca}$, with translations provided between PDL and $\mathcal{L}_\mu^{ca}$. Since $\mathcal{L}_\mu^{ca}$ is a fragment of the alternation-free μ-calculus, our cut-elimination result transfers directly to the Focus system when restricted to sequents of $\mathcal{L}_\mu^{ca}$-formulas.

Let us now consider the proof system CPDL_f introduced in Chapter 6. As mentioned in the conclusion to this chapter, an analogous proof system PDL_f for PDL can be defined. Using the translations between PDL and $\mathcal{L}_\mu^{ca}$, we can define translations between Focus-proofs of $\mathcal{L}_\mu^{ca}$-sequents and PDL_f-proofs. However, these translations may introduce cuts, preventing a direct transfer of our cut-elimination result. Nonetheless, since annotations and the soundness condition in PDL_f are simpler than in Focus, it should be possible to adapt our cut-elimination method and apply it directly to PDL_f without difficulty. This aligns with our conclusion in Chapter 6 that the analytic cut rule in CPDL_f is only necessary for handling converse modalities and can be omitted in PDL_f.

Regarding the extension to more expressive logics, it is worth investigating whether our technique can be generalized to the full modal μ-calculus. Candidate proof systems include BT and JS introduced in Chapter 4. Our construction relies on a key property of $\mathcal{L}_\mu^{af}$-formulas φ: either φ or its negation $\overline{\varphi}$ is not contained in the closure of a ν-formula of the same rank. Since such formulas can never be in focus, descendants of such a formula are not essential for the success-condition of repeat paths. For general $\mathcal{L}_\mu$ formulas, this property need not hold, and a more sophisticated method would be required to handle the resulting complexity of annotations.

Also of interest is to determine the precise *complexity* of our cut-elimination procedure. As we currently rely on results concerning well-quasi-orders, we can only establish an Ackermannian upper bound. Whether the termination argument can be simplified to yield a tighter bound remains an open question.

The cut-elimination result also extends to NW-proofs of $\mathcal{L}_\mu^{af}$-sequents. Given an NW-proof with cuts of an $\mathcal{L}_\mu^{af}$-sequent Γ, we can first translate π to a Focus-proof ρ of Γ^u. This translation is given in [MV21a] to show completeness of the Focus system, and resembles the translations in Chapter 4 for showing completeness of BT. Our cut-elimination result then yields a cut-free Focus proof ρ'. By unfolding ρ' and omitting annotations, we obtain a cut-free NW proof π' of Γ. We can therefore use the annotated proof system Focus to obtain results on the trace-based proof system NW.

Bibliography

[ACG24] M. Acclavio, G. Curzi, and G. Guerrieri. "Infinitary Cut-Elimination via Finite Approximations". In: *32nd EACSL Annual Conference on Computer Science Logic, CSL*. Vol. 288. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl, 2024, 8:1–8:19. DOI: 10.4230/LIPICS.CSL.2024.8.

[AEL22] B. Afshari, S. Enqvist, and G. E. Leigh. "Cyclic proofs for the first-order μ-calculus". In: *Logic Journal of the IGPL* 32.1 (2022), pp. 1–34. DOI: 10.1093/jigpal/jzac053.

[AELMV23] B. Afshari, S. Enqvist, G. Leigh, J. Marti, and Y. Venema. "Proof Systems for Two-way Modal mu-Calculus". In: *Journal of Symbolic Logic* 90.3 (2023), pp. 1211–1260. DOI: DOI:10.1017/jsl.2023.60.

[AF09] L. Alberucci and A. Facchini. "The Modal mu-Calculus Hierarchy over Restricted Classes of Transition Systems". In: *Journal of Symbolic Logic* 74.4 (2009), pp. 1367–1400.

[AJL19] B. Afshari, G. Jäger, and G. E. Leigh. "An Infinitary Treatment of Full Mu-Calculus". In: *International Workshop on Logic, Language, Information, and Computation, WoLLIC*. Vol. 11541. Lecture Notes in Computer Science. Springer, 2019. DOI: 10.1007/978-3-662-59533-6_2.

[AK24] B. Afshari and J. Kloibhofer. "Cut elimination for Cyclic Proofs: A Case Study in Temporal Logic". In: *Twelfth Workshop on Fixed Points in Computer Science, FICS*. Vol. 435. Electronic Proceedings in Theoretical Computer Science. 2024, pp. 21–40. DOI: 10.4204/EPTCS.435.3.

[AK25] B. Afshari and J. Kloibhofer. *Cut-elimination for the alternation-free modal mu-calculus*. preprint. 2025. DOI: 10.48550/arXiv.2510.11293.

[AL16] B. Afshari and G. E. Leigh. "Circular proofs for the modal mu-calculus". In: *Proceedings in Applied Mathematics and Mechanics* 16.1 (2016), pp. 893–894. DOI: 10.1002/pamm.201610435.

[AL17] B. Afshari and G. E. Leigh. "Cut-Free Completeness for Modal Mu-Calculus". In: *Proceedings of the 32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*. IEEE, 2017.

[AL19] B. Afshari and G. E. Leigh. "Lyndon Interpolation for Modal μ-Calculus". In: *Language, Logic, and Computation: 13th International Tbilisi Symposium, TbiLLC 2019, Revised Selected Papers.* 2019. DOI: 10.1007/978-3-030-98479-3_10.

[ALM21] B. Afshari, G. E. Leigh, and G. Menéndez Turata. "Uniform Interpolation from Cyclic Proofs: The Case of Modal Mu-Calculus". In: *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX*. Lecture Notes in Computer Science. Springer, 2021, pp. 335–353. DOI: 10.1007/978-3-030-86059-2_20.

[ALM25] B. Afshari, G. E. Leigh, and G. Menèndez Turata. "Demystifying µ". In: *Fundamenta Informaticae* Volume 194, Issue 2: Fixed Points in Computer Science (2025). DOI: 10.46298/fi.12773.

[Bae12] D. Baelde. "Least and Greatest Fixed Points in Linear Logic". In: *ACM Trans. Comput. Logic* 13.1 (2012), 2:1–2:44. DOI: 10.1145/2071368.2071370.

[BBV19] M. Benedikt, P. Bourhis, and M. Vanden Boom. "Definability and Interpolation within Decidable Fixpoint Logics". In: *Logical Methods in Computer Science* 15.3 (2019). DOI: 10.23638/LMCS-15(3:29)2019.

[BCV15] M. Benedikt, B. ten Cate, and M. Vanden Boom. "Interpolation with Decidable Fixpoint Logics". In: *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*. IEEE, 2015, pp. 378–389. DOI: 10.1109/LICS.2015.43.

[BDKS22] D. Baelde, A. Doumane, D. Kuperberg, and A. Saurin. "Bouncing Threads for Circular and Non-Wellfounded Proofs: Towards Compositionality with Circular Proofs". In: *37th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*. ACM, 2022, 63:1–63:13. DOI: 10.1145/3531130.3533375.

[BDS16] D. Baelde, A. Doumane, and A. Saurin. "Infinitary Proof Theory: the Multiplicative Additive Case". In: *25th EACSL Annual Conference on Computer Science Logic, CSL*. Vol. 62. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl, 2016, 42:1–42:17. DOI: 10.4230/LIPICS.CSL.2016.42.

[Ben08] J. van Benthem. "The many faces of interpolation". In: *Synthese* 164.3 (2008), pp. 451–460. DOI: 10.1007/s11229-008-9351-5.

[Bet53] E. W. Beth. "On Padoa's methods in the theory of definitions". In: *Indagationes mathematicae* 15 (1953), pp. 330–339.

[BGHRDV25] M. Borzechowski, M. Gattinger, H. H. Hansen, R. Ramanayake, V. T. Dalmas, and Y. Venema. "Propositional Dynamic Logic has Craig Interpolation: a tableau-based proof". preprint. 2025. DOI: 10.48550/arXiv.2503.13276.

[BHP82] M. Ben-Ari, J. Y. Halpern, and A. Pnueli. "Deterministic propositional dynamic logic: Finite models, complexity, and completeness". In: *Journal of Computer and System Sciences* 25.3 (1982), pp. 402–417. DOI: 10.1016/0022-0000(82)90018-6.

[BL07] F. Baader and C. Lutz. "Description Logics". In: *Handbook of Modal Logic*. Ed. by P. Blackburn, J. van Benthem, and F. Wolter. Vol. 3. Elsevier, 2007, pp. 757–820. DOI: https://doi.org/10.1016/S1570-2464(07)80016-4.

[BL69] J. R. Büchi and L. H. Landweber. "Solving Sequential Conditions by Finite-State Strategies". In: *Transactions of the American Mathematical Society* 138 (1969), pp. 295–311. DOI: 10.2307/1994916.

[Bor88] M. Borzechowski. "Tableau-Kalkül für PDL und Interpolation". MA thesis. Department of Mathematics, Freie Universität Berlin, 1988.

[Bra98] J. C. Bradfield. "The modal mu-calculus alternation hierarchy is strict". In: *Theoretical Computer Science* 195.2 (1998), pp. 133–153. DOI: 10.1016/S0304-3975(97)00217-X.

[Bro06] J. Brotherston. "Sequent calculus proof systems for inductive definitions". PhD thesis. School of Informatics; The University of Edinburgh, 2006.

[BRV01] P. Blackburn, M. d. Rijke, and Y. Venema. *Modal Logic*. Cambridge Tracts in Theoretical Computer Science 53. Cambridge University Press, 2001.

[BS07] J. Bradfield and C. Stirling. "Modal mu-Calculi". In: *Handbook of Modal Logic*. Ed. by P. Blackburn, J. van Benthem, and F. Wolter. Vol. 3. Elsevier, 2007, pp. 721–756. DOI: 10.1016/S1570-2464(07)80015-2.

[BS25] E. Bauer and A. Saurin. "On the cut-elimination of the modal mu-calculus: Linear Logic to the rescue". In: *Foundations of Software Science and Computation Structures, FoSSaCS*. Lecture Notes in Computer Science. Springer, 2025, pp. 133–154. DOI: 10.1007/978-3-031-90897-2_7.

[Büc62] J. R. Büchi. "On a Decision Method in Restricted Second Order Arithmetic". In: *Logic, methodology and philosophy of science*. Ed. by A. T. E. Nagel P. Suppes. Stanford University Press, 1962, pp. 1–11.

[CFS13] B. ten Cate, E. Franconi, and I. Seylan. "Beth Definability in Expressive Description Logics". In: *Journal of Artificial Intelligence Research* 48 (2013), pp. 347–414. DOI: 10.1613/JAIR.4057.

[CJKLS17] C. Calude, S. Jain, B. Khoussainov, W. Li, and F. Stephan. "Deciding parity games in quasipolynomial time". In: *Proceedings of the 49th Annual ACM SIGACT Symposium on Theory of Computing, (STOC 2017)*. Ed. by H. Hatami, P. McKenzie, and V. King. 2017, pp. 252–263.

[Cra57] W. Craig. "Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory". In: *Journal of Symbolic Logic* 22.3 (1957), pp. 269–285. DOI: 10.2307/2963594.

[CV14] F. Carreiro and Y. Venema. "PDL Inside the μ-calculus: A Syntactic and an Automata-theoretic Characterization". In: *Advances in Modal Logic, AiML*. College Publications, 2014, pp. 74–93.

[DAg18] G. D'Agostino. "μ-Levels of Interpolation". In: *Larisa Maksimova on Implication, Interpolation, and Definability*. Springer International Publishing, 2018, pp. 155–170. DOI: 10.1007/978-3-319-69917-2_8.

[DGL16] S. Demri, V. Goranko, and M. Lange. *Temporal Logics in Computer Science: Finite-State Systems*. Cambridge Tracts in Theoretical Computer Science. Cambridge University Press, 2016.

[DH00] G. D'Agostino and M. Hollenberg. "Logical questions concerning the μ-calculus". In: *Journal of Symbolic Logic* 65 (2000), pp. 310–332.

[Dic13] L. E. Dickson. "Finiteness of the Odd Perfect and Primitive Abundant Numbers with n Distinct Prime Factors". In: *American Journal of Mathematics* 35.4 (1913), pp. 413–422. DOI: 10.2307/2370405.

[DKMV23] M. Dekker, J. Kloibhofer, J. Marti, and Y. Venema. "Proof Systems for the Modal μ-Calculus Obtained by Determinizing Automata". In: *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX*. Lecture Notes in Computer Science. Springer, 2023, pp. 242–259. DOI: 10.1007/978-3-031-43513-3_14.

[DM79] N. Dershowitz and Z. Manna. "Proving termination with multiset orderings". In: *Communications of the ACM* 22.8 (1979), pp. 465–476. DOI: 10.1145/359138.359142.

[Dou17] A. Doumane. "Constructive completeness for the linear-time μ-calculus". In: *32nd Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*. IEEE, 2017, pp. 1–12. DOI: 10.1109/LICS.2017.8005075.

[DP18] A. Das and D. Pous. "Non-Wellfounded Proof Theory For (Kleene+Action)(Algebras+Lattices)". In: *27th EACSL Annual Conference on Computer Science Logic CSL*. Vol. 119. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl, 2018, 19:1–19:18. DOI: 10.4230/LIPIcs.CSL.2018.19.

[EHKMV19] S. Enqvist, H. H. Hansen, C. Kupke, J. Marti, and Y. Venema. "Completeness for game logic". In: *34th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*. IEEE, 2019. DOI: 10.1109/LICS.2019.8785676.

[EJ99] E. Emerson and C. Jutla. "The complexity of tree automata and logics of programs". In: *SIAM Journal of Computing* 29.1 (1999), pp. 132–158.

[ESV18] S. Enqvist, F. Seifan, and Y. Venema. "Completeness for the modal μ-calculus: Separating the combinatorics from the dynamics". In: *Theoretical Computer Science* 727 (2018), pp. 37–100. DOI: 10.1016/j.tcs.2018.03.001.

[FFSS11] D. Figueira, S. Figueira, S. Schmitz, and P. Schnoebelen. "Ackermannian and Primitive-Recursive Bounds with Dickson's Lemma". In: *26th Annual Symposium on Logic in Computer Science, LICS*. IEEE, 2011. DOI: 10.1109/LICS.2011.39.

[FKVW15] S. Fogarty, O. Kupferman, M. Y. Vardi, and T. Wilke. "Profile trees for Büchi word automata, with application to determinization". In: *Information and Computation* 245 (2015), pp. 136–151.

[FKWV13] S. Fogarty, O. Kupferman, T. Wilke, and M. Vardi. "Unifying Büchi Complementation Constructions". In: *Logical Methods in Computer Science* 9.1 (2013).

[FL13] O. Friedmann and M. Lange. "Deciding the unguarded modal μ-calculus". In: *Journal of Applied Non-Classical Logics* 23.4 (2013), pp. 353–371. DOI: 10.1080/11663081.2013.861181.

[FL79] M. J. Fischer and R. E. Ladner. "Propositional dynamic logic of regular programs". In: *Journal of Computer and System Sciences* 18.2 (1979), pp. 194–211. DOI: 10.1016/0022-0000(79)90046-1.

[Fre06] T. French. "Bisimulation Quantifiers for Modal Logic". PhD thesis. School for Computer Science and Software Engineering, University of Western Australia, 2006.

[Fre07] T. French. "Idempotent Transductions for Modal Logics". In: *Proceedings of the 6th International Symposium on Frontiers of Combining Systems (FroCoS 2007)*. Ed. by B. Konev and F. Wolter. Vol. 4720. Lecture Notes in Computer Science. Springer, 2007, pp. 178–192. DOI: 10.1007/978-3-540-74621-8_12.

[FS13] J. Fortier and L. Santocanale. "Cuts for circular proofs: semantics and cut-elimination". In: *22th EACSL Annual Conference on Computer Science Logic, CSL*. Vol. 23. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl, 2013, pp. 248–262. DOI: 10.4230/LIPIcs.CSL.2013.248.

[FV18] G. Fontaine and Y. Venema. "Some model theory for the modal μ-calculus: syntactic characterisations of semantic properties". en. In: *Logical Methods in Computer Science* Volume 14, Issue 1 (2018), p. 4225. DOI: 10.23638/LMCS-14(1:14)2018.

[Gen35] G. Gentzen. "Untersuchungen über das logische Schließen. I". de. In: *Mathematische Zeitschrift* 39.1 (1935), pp. 176–210. DOI: 10.1007/BF01201353.

[GGRJ25] N. Galatos, V. Greati, R. Ramanayake, and G. S. John. *Complexities of Well-Quasi-Ordered Substructural Logics*. preprint. 2025. DOI: 10.48550/arXiv.2504.21674.

[Gir95] J.-Y. Girard. "Linear logic: its syntax and semantics". In: *Proceedings of the workshop on Advances in linear logic*. Cambridge University Press, 1995, pp. 1–42.

[GKL14] J. Gutierrez, F. Klaedtke, and M. Lange. "The mu-calculus alternation hierarchy collapses over structures with restricted connectivity". In: *Theoretical Computer Science* 560 (2014), pp. 292–306. DOI: 10.1016/j.tcs.2014.03.027.

[GL94] G. D. Giacomo and M. Lenzerini. "Description Logics with Inverse Roles, Functional Restrictions, and N-ary Relations". In: *Logics in Artificial Intelligence, European Workshop, JELIA '94, 1994, Proceedings*. Springer, 1994. DOI: 10.1007/BFB0021982.

[GM05] D. M. Gabbay and L. Maksimova. *Interpolation and Definability: modal and intuitionistic logics*. Oxford University Press, 2005.

[GTW02] E. Grädel, W. Thomas, and T. Wilke, eds. *Automata, Logic, and Infinite Games*. Vol. 2500. LNCS. Springer, 2002.

[HKQ03] T. A. Henzinger, O. Kupferman, and S. Qadeer. "From Pre-Historic to Post-Modern Symbolic Model Checking". In: *Formal Methods in System Design* 23.3 (2003), pp. 303–327. DOI: 10.1023/A:1026228213080.

[HSS25] S. Horvat, B. Sierra Miranda, and T. Studer. "Non-wellfounded Proof Theory for Interpretability Logic". In: *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX*. Lecture Notes in Computer Science. Springer, 2025, pp. 201–219. DOI: 10.1007/978-3-032-06085-3_11.

[JKS08] G. Jäger, M. Kretz, and T. Studer. "Canonical completeness of infinitary μ". In: *The Journal of Logic and Algebraic Programming* 76.2 (2008), pp. 270–292. DOI: 10.1016/j.jlap.2008.02.005.

[JLPW21] J. C. Jung, C. Lutz, H. Pulcini, and F. Wolter. "Separating Data Examples by Description Logic Concepts with Restricted Signatures". In: *Proceedings of the 18th International Conference on Principles of Knowledge Representation and Reasoning, KR*. 2021. DOI: 10.24963/KR.2021/37.

[Jun10] N. Jungteerapanich. "Tableau systems for the modal μ-calculus". PhD thesis. School of Informatics; The University of Edinburgh, 2010.

[JW96] D. Janin and I. Walukiewicz. "On the Expressive Completeness of the Propositional μ-Calculus w.r.t. Monadic Second-Order Logic". In: *Proceedings of the Seventh International Conference on Concurrency Theory, CONCUR '96*. Vol. 1119. LNCS. 1996, pp. 263–277.

[Kas94] R. Kashima. "Cut-free sequent calculi for some tense logics". In: *Studia Logica* 53 (1994), pp. 119–136.

[Klo23] J. Kloibhofer. *A note on the incompleteness of Afshari & Leigh's system Clo*. preprint. 2023. arXiv: 2307.06846 [math.LO].

[KO17] T. Kowalski and H. Ono. "Analytic cut and interpolation for bi-intuitionistic Logic". In: *Review of Symbolic Logic* 10.2 (2017), pp. 259–283. DOI: 10.1017/S175502031600040X.

[Koz06] D. Kozen. *Theory of Computation*. Springer Science & Business Media, 2006. DOI: 10.1007/1-84628-477-5.

[Koz83] D. Kozen. “Results on the propositional μ-calculus”. In: *Theoretical Computer Science* 27 (1983), pp. 333–354.

[Koz88] D. Kozen. “A finite model theorem for the propositional μ-calculus”. In: *Studia Logica* 47.3 (1988), pp. 233–241. DOI: 10.1007/BF00370554.

[KPP21] D. Kuperberg, L. Pinault, and D. Pous. “Cyclic proofs, system T, and the power of contraction”. In: *Proceedings of the ACM on Programming Languages* 5.POPL (2021), 1:1–1:28. DOI: 10.1145/3434282.

[KTV25] J. Kloibhofer, V. Trucco Dalmas, and Y. Venema. “Interpolation for Converse PDL”. In: *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX*. Lecture Notes in Computer Science. Springer, 2025, pp. 258–277. DOI: 10.1007/978-3-032-06085-3_14.

[KV25] J. Kloibhofer and Y. Venema. “Interpolation for the two-way modal μ-calculus”. In: *40th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS*. IEEE, 2025, pp. 155–168. DOI: 10.1109/LICS65433.2025.00019.

[KW08] D. Kähler and T. Wilke. “Complementation, Disambiguation, and Determinization of Büchi Automata Unified”. In: *Automata, Languages and Programming*. Ed. by L. Aceto, I. Damgård, L. A. Goldberg, M. M. Halldórsson, A. Ingólfsdóttir, and I. Walukiewicz. Springer, 2008, pp. 724–735. DOI: 10.1007/978-3-540-70575-8_59.

[LL32] C. I. Lewis and C. H. Langford. *Symbolic Logic*. The Century Philosophy Series. The Century Company, 1932.

[LP19] C. Löding and A. Pirogov. “Determinization of Büchi Automata: Unifying the Approaches of Safra and Muller-Schupp”. In: *46th International Colloquium on Automata, Languages, and Programming, ICALP*. Ed. by C. Baier, I. Chatzigiannakis, P. Flocchini, and S. Leonardi. Vol. 132. Leibniz International Proceedings in Informatics (LIPIcs). Schloss Dagstuhl, 2019, 120:1–120:13. DOI: 10.4230/LIPIcs.ICALP.2019.120.

[LPZ85] O. Lichtenstein, A. Pnueli, and L. D. Zuck. “The Glory of the Past”. In: *Logics of Programs*. Ed. by R. Parikh. Vol. 193. Lecture Notes in Computer Science. Springer, 1985, pp. 196–218. DOI: 10.1007/3-540-15648-8_16.

[Lut05] C. Lutz. “PDL with Intersection and Converse Is Decidable”. In: *14th EACSL Annual Conference on Computer Science Logic, CSL*. Vol. 3634. Lecture Notes in Computer Science. Berlin, Heidelberg: Springer, 2005, pp. 413–427. DOI: 10.1007/11538363_29.

[LW24] G. E. Leigh and D. Wehr. “From GTC to Reset: Generating reset proof systems from cyclic proof systems”. In: *Annals of Pure and Applied Logic* 175.10 (2024). DOI: 10.1016/j.apal.2024.103485.

[Mae61] S. Maehara. “Craig’s interpolation theorem”. In: *Sugaku* 12 (4 1961). Japanese, pp. 235–237. DOI: 10.11429/sugaku1947.12.235.

[McM05] K. L. McMillan. “Applications of Craig Interpolants in Model Checking”. In: *Tools and Algorithms for the Construction and Analysis of Systems, 11th International Conference, TACAS 2005, Proceedings*. 2005. DOI: 10.1007/978-3-540-31980-1_1.

[McN66] R. McNaughton. “Testing and generating infinite sequences by a finite automaton”. In: *Information and Control* 9.5 (1966), pp. 521–530. DOI: 10.1016/S0019-9958(66)80013-X.

[Mos91] A. Mostowski. “Games with forbidden positions”. Technical Report 78, Instytut Matematyki, Uniwersytet Gdański, Poland. 1991.

[MS95] D. E. Muller and P. E. Schupp. “Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of Rabin, McNaughton and Safra”. In: *Theoretical Computer Science* 141.1 (1995), pp. 69–107. DOI: 10.1016/0304-3975(94)00214-4.

[MSZ24] B. S. Miranda, T. Studer, and L. Zenger. “Coalgebraic Proof Translations of Non-Wellfounded Proofs”. In: *Advances in Modal Logic, AiML*. Vol. 15. College Publications, 2024.

[Mul63] D. Muller. “Infinite sequences and finite machines”. In: *4th Annual Symposiom on Switching Circuit Theory and Logical Design. Proceedings*. IEEE, 1963, pp. 3–16. DOI: 10.1109/SWCT.1963.8.

[MV21a] J. Marti and Y. Venema. “A Focus System for the Alternation-Free μ-Calculus”. In: *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX*. Lecture Notes in Computer Science. Springer, 2021, pp. 371–388. DOI: 10.1007/978-3-030-86059-2_22.

[MV21b] J. Marti and Y. Venema. “Focus-style proof systems and interpolation for the alternation-free μ-calculus”. preprint. 2021. arXiv: 2103.01671.

[Nis80] H. Nishimura. “A Study of Some Tense Logics by Gentzen’s Sequential Method”. In: *Publications of the Research Institute for Mathematical Sciences* 16.2 (1980), pp. 343–353.

[Nol21] R. Nollet. “Circular representations of infinite proofs for fixed-points logics: expressiveness and complexity”. PhD thesis. Université Paris Cité, 2021.

[NW96] D. Niwinski and I. Walukiewicz. “Games for the mu-Calculus”. In: *Theoretical Computer Science* 163.1&2 (1996), pp. 99–116. DOI: 10.1016/0304-3975(95)00136-0.

[Par78] R. Parikh. “The Completeness of Propositional Dynamic Logic”. In: *7th Symposium of Mathematical Foundations of Computer Science, Proceedings*. Vol. 64. Lecture Notes in Computer Science. Springer, 1978, pp. 403–415. DOI: 10.1007/3-540-08921-7_88.

[Pra80] V. Pratt. “A near-optimal method for reasoning about action”. In: *Journal of Computer and System Sciences* 20 (1980), pp. 231–254. DOI: 10.1016/0022-0000(80)90061-6.

[Rab69] M. O. Rabin. “Decidability of Second-Order Theories and Automata on Infinite Trees”. In: *Transactions of the American Mathematical Society* 141 (1969), pp. 1–35. DOI: 10.2307/1995086.

[Roo21] J. Rooduijn. “Cyclic Hypersequent Calculi for Some Modal Logics with the Master Modality”. In: *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX*. Lecture Notes in Computer Science. Springer, 2021, pp. 354–370. DOI: 10.1007/978-3-030-86059-2_21.

[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. DOI: 10.1147/rd.32.0114.

[RV23] J. Rooduijn and Y. Venema. “Focus-style proofs for the two-way alternation-free μ-calculus”. In: *International Workshop on Logic, Language, Information, and Computation, WoLLIC*. Vol. 13923. Lecture Notes in Computer Science. Springer, 2023, pp. 318–335. DOI: 10.1007/978-3-031-39784-4_20.

[Saf88] S. Safra. “On The Complexity of ω-Automata”. In: *29th Symposium on the Foundations of Computer Science*. Proceedings. IEEE Computer Society Press, 1988, pp. 319–327.

[San02] L. Santocanale. “A Calculus of Circular Proofs and Its Categorical Semantics”. In: *Foundations of Software Science and Computation Structures, FoSSaCS*. Lecture Notes in Computer Science. Springer, 2002, pp. 357–371. DOI: 10.1007/3-540-45931-6_25.

[Sau23] A. Saurin. "A Linear Perspective on Cut-Elimination for Non-wellfounded Sequent Calculi with Least and Greatest Fixed-Points". In: *Automated Reasoning with Analytic Tableaux and Related Methods, TABLEAUX*. Lecture Notes in Computer Science. Springer, 2023, pp. 203–222. DOI: 10.1007/978-3-031-43513-3_12.

[SD03] C. Sprenger and M. Dam. "On the Structure of Inductive Reasoning: Circular and Tree-Shaped Proofs in the μCalculus". In: *Foundations of Software Science and Computation Structures, FoSSaCS*. Lecture Notes in Computer Science. Springer, 2003, pp. 425–440. DOI: 10.1007/3-540-36576-1_27.

[SE89] R. S. Streett and E. A. Emerson. "An automata theoretic decision procedure for the propositional mu-calculus". In: *Information and Computation* 81.3 (1989), pp. 249–264. DOI: 10.1016/0890-5401(89)90031-X.

[Seg77] K. Segerberg. "A completeness theorem in the modal logic of programs". In: *Notices of the American Mathematical Society* 24 (1977), p. 522.

[Sha14] D. S. Shamkanov. "Circular proofs for the Gödel-Löb provability logic". In: *Mathematical Notes* 96.3-4 (2014), pp. 575–585. DOI: 10.1134/s0001434614090326.

[Sha25] D. Shamkanov. "On structural proof theory of the modal logic K+ extended with infinitary derivations". In: *Logic Journal of the IGPL* 33.3 (2025). DOI: 10.1093/jigpal/jzae121.

[Sim17] A. Simpson. "Cyclic Arithmetic Is Equivalent to Peano Arithmetic". In: *Foundations of Software Science and Computation Structures, FoSSaCS*. Lecture Notes in Computer Science. Springer, 2017, pp. 283–300. DOI: 10.1007/978-3-662-54458-7_17.

[SS20] Y. Savateev and D. Shamkanov. "Non-well-founded Proofs for the Grzegorcyk Modal Logic". In: *The Review of Symbolic Logic* 14.1 (2020), pp. 22–50. DOI: 10.1017/s1755020319000510.

[SSW20] P. M. Schuster, M. Seisenberger, and A. Weiermann, eds. *Well-Quasi Orders in Computation, Logic, Language and Reasoning*. Vol. 53. Springer Cham, 2020. DOI: https://doi.org/10.1007/978-3-030-30229-0.

[Sta15] R. P. Stanley. *Catalan Numbers*. Cambridge University Press, 2015. DOI: 10.1017/CBO9781139871495.

[Sti14] C. Stirling. "A Tableau Proof System with Names for Modal Mu-calculus". In: *HOWARD-60. A Festschrift on the Occasion of Howard Barringer's 60th Birthday.* Ed. by A. Voronkov and M. Korovina. Vol. 42. EPiC Series in Computing. EasyChair, 2014, pp. 306–318. DOI: `10.29007/lwqm`.

[Stu08] T. Studer. "On the Proof Theory of the Modal mu-Calculus". In: *Studia Logica* 89.3 (2008), pp. 343–363. DOI: `10.1007/s11225-008-9133-6`.

[Tak87] G. Takeuti. *Proof Theory.* Second edition. North-Holland, 1987.

[Tar55] A. Tarski. "A lattice-theoretical fixpoint theorem and its applications." In: *Pacific Journal of Mathematics* 5.2 (1955), pp. 285–309.

[TB23] N. Troquard and P. Balbiani. "Propositional Dynamic Logic". In: *The Stanford Encyclopedia of Philosophy.* Ed. by E. N. Zalta and U. Nodelman. Metaphysics Research Lab, Stanford University, 2023.

[Var98] M. Y. Vardi. "Reasoning about the past with two-way automata". In: *International Colloquium on Automata, Languages, and Programming, ICALP.* Vol. 1443. Lecture Notes in Computer Science. Springer, 1998, pp. 628–641. DOI: `10.1007/BFb0055090`.

[Ven20] Y. Venema. *Lectures on the modal μ-calculus.* Lecture Notes. ILLC, University of Amsterdam. 2020.

[Wal00] I. Walukiewicz. "Completeness of Kozen's axiomatisation of the propositional μ-calculus". In: *Information and Computation* 157 (2000), pp. 142–182.

[Wal93] I. Walukiewicz. "On completeness of the mu-calculus". In: *8th Annual IEEE Symposium on Logic in Computer Science, LICS.* IEEE, 1993, pp. 136–146. DOI: `10.1109/LICS.1993.287593`.

Index

Samenvatting

De titel van dit proefschrift luidt in het Nederlands: *Cykels met Annotaties. Niet-Welgefundeerde Bewijstheorie voor Modale Dekpuntlogica's*. In de niet-welgefundeerde bewijstheorie kunnen bewijzen oneindig lange takken of cykels bevatten. Om absurde redeneringen te voorkomen, wordt een zogenaamde *correctheidsvoorwaarde* geformuleerd voor die oneindige takken en cykels. De belangrijkste uitdaging in de niet-welgefundeerde bewijstheorie is het omgaan met deze correctheidsvoorwaarden. Dit proefschrift behandelt verschillende soorten correctheidsvoorwaarden en is opgebouwd rond twee hoofdthema's.

Ten eerste laten we zien hoe *annotaties* kunnen worden gebruikt om oneindige bewijssystemen te verkrijgen met eenvoudige padgebaseerde correctheidsvoorwaarden; en we transformeren deze laatste calculi in cyclische systemen met lokale correctheidsvoorwaarden. Ten tweede laten we zien hoe dergelijke geannoteerde cyclische bewijssystemen kunnen worden gebruikt om resultaten af te leiden over de onderliggende logica's.

De logica's die we beschouwen zijn modale dekpuntlogica's. De centrale logica die we bestuderen is de *modale μ-calculus*, die de basismodale logica uitbreidt met expliciete kleinste en grootste dekpunt-operatoren. We onderzoeken ook uitbreidingen daarvan, zoals de tweerichtings modale μ-calculus, die achterwaartse modaliteiten omvat, en fragmenten daarvan, zoals de alternatievrije modale μ-calculus en tweerichtings propositionele dynamische logica.

In hoofdstuk 3 ontwikkelen we methoden om ω-automaten te determinizeren, als een technische basis voor de volgende hoofdstukken. Hoofdstuk 4 richt zich op niet-welgefundeerde bewijssystemen voor de modale μ-calculus. Met behulp van de automaten-theoretische resultaten die in hoofdstuk 3 zijn vastgesteld, voegen we annotaties toe om cyclische bewijssystemen met lokale correctheidsvoorwaarden te construeren. Daarnaast stellen we vast dat het bewijssysteem Clo, geïntroduceerd door Afshari en Leigh, onvolledig is.

In hoofdstuk 5 introduceren we verschillende bewijssystemen voor de tweerichtings modale μ-calculus. Voortbouwend op de resultaten van hoofdstukken

3 en 4, introduceren we een geannoteerd cyclisch bewijssysteem, dat we vervolgens gebruiken om de *Craig-interpolatie-eigenschap* voor de tweerichtings modale μ-calculus vast te stellen. Hoofdstuk 6 is gewijd aan tweerichtings propositionele dynamische logica. We introduceren een geannoteerd cyclisch bewijssysteem en gebruiken dit om te bewijzen dat ook deze logica voldoet aan *Craig-interpolatie*.

Ten slotte behandelt hoofdstuk 7 *snede-eliminatie* voor geannoteerde cyclische bewijssystemen. Dit resultaat wordt verkregen binnen het Focus-systeem, geïntroduceerd door Marti en Venema voor de alternatievrije modale μ-calculus.

Abstract

This thesis studies *non-wellfounded proof theory*. In this setting, proofs may contain infinitely long branches or cycles. In order to disallow absurd reasoning, a so-called *soundness condition* is formulated on those infinite branches and cycles. The main challenge in non-wellfounded proof theory is to handle this soundness condition. This thesis addresses several kinds of soundness conditions and is organized around two main themes.

First, we show how to employ *annotations* to obtain infinitary proof systems with simple path-based soundness conditions; and we transform the latter into cyclic systems with local soundness conditions. Second, we demonstrate how to use such annotated cyclic proof systems to derive results about their underlying logics.

The logics we consider are modal fixpoint logics. The central logic studied is the *modal μ-calculus*, which extends basic modal logic with explicit least and greatest fixpoint operators. We also investigate extensions of it such as the two-way modal μ-calculus, which includes backwards modalities, and fragments thereof such as the alternation-free modal μ-calculus and Converse Propositional Dynamic Logic.

In Chapter 3 we develop *determinization* methods for ω-automata, which form a technical foundation for the subsequent chapters. Chapter 4 focuses on non-wellfounded proof systems for the modal μ-calculus. Using the automata-theoretic results established in Chapter 3, we add annotations to construct cyclic proof systems with local soundness conditions. Additionally, we establish that the proof system Clo, introduced by Afshari and Leigh, is incomplete.

In Chapter 5 we introduce several proof systems for the two-way modal μ-calculus. Building on the results of Chapters 3 and 4, we derive an annotated cyclic proof system, which we then use to establish the *Craig interpolation property* for the two-way modal μ-calculus. Chapter 6 is devoted to Converse Propositional Dynamic Logic. We introduce an annotated cyclic proof system and employ it to prove that the logic satisfies *Craig interpolation*.

Finally, Chapter 7 addresses *cut elimination* for annotated cyclic proof systems. This result is obtained within the Focus system, defined by Marti and Venema for the alternation-free modal μ-calculus.

Acknowledgments

Doing research and writing a thesis is not a solitary activity, and having the right people to work with and to support you is hugely important. I was fortunate to have such people around me which I would like to thank here.

First I would like to thank my supervisors. Yde, without your efforts this thesis could not have been written. Throughout the four years of working together I learned a lot from you and left each of our meetings feeling more motivated than before. I admire your passion and commitment. Thanks for being interested in all my ideas and your constant support in the last years.

Thanks Bahareh for always believing in me, from letting me teach in my first year, to writing papers, and connecting me to the research community. I warmly look back to a wonderful stay in Gothenburg. Thanks for your enthusiastic hospitality and all the fun social activities.

At this point I would also like to thank Stefan. You first got me interested in the sort of logic I am doing. Thanks for supporting me even far beyond my master's thesis and finishing a paper that took almost as long as my PhD.

I pursued my PhD at the ILLC, an institute full of kind and interesting people that I am grateful for. I would like to start with thanking the members of the ILLC office. Without you, nothing would work as nicely as we all got accustomed to.

Thanks to my office mates and everyone who had lunch and coffee breaks with me – you made me excited about going to the office. A special thanks goes to the colleagues from the MCL unit and the organizers of the LLAMA seminar enabling a lot of interesting talks and discussions. Inside this unit we also were a small group devoted to proof theory. Thanks Bahareh, Guillermo, Iris, Jan, Johannes, Lide, Marianna and Yde for making our cyclic meetings both enjoyable and enlightening.

I was fortunate to work with some great colleagues, many of whom I became friends with during this time. Jan, you share a PhD topic and a name with me. As such you have been great in paving my way. Lide, we quickly became friends when having a great time as flatmates in Sweden. I highly appreciate your openness, interest and understanding; and special thanks for revising my introduction. I am very happy to have both of you as paranymphs for my defense. Daniël, we had some great stays at conferences together. I will miss your timely arrivals at noon for going for lunch together. Thanks Rodrigo for your enthusiasm, whether it is about drinking coffee, organizing seminars and barbecues, or discussing free algebras, letics or lives. Thanks Daira for all the relaxing coffees and chats about everything else apart from work.

I am embedded in an open and welcoming research community. Each time I was going to some event I was looking forward to meeting you and I always left feeling more motivated about research.

A shout-out goes to the coauthors that I had the joy of working with. The first paper I wrote during my PhD was a bumpy ride. Johannes, I remember countless discussions on ever so slightly non-working constructions. It was even more satisfying when we finally worked it out; and what I learned then proved very important for this thesis. I also had the pleasure to write a paper together with Francisco. I recall the week you were visiting in Amsterdam as one of the most enjoyable and productive weeks of my PhD.

A special thanks goes to the members of my committee for taking their time and energy to read and evaluate this thesis.

I am grateful to everyone organizing workshops and conferences which allowed me to present my work. Thanks to the reviewers of my papers who, due to there anonymous nature, are never acknowledged enough. Not only do I want to thank them for accepting my works, but also for (mostly) giving valuable feedback that indirectly also improved this thesis.

I would like to express my gratitude to the research community by mentioning some people with whom I have had interesting discussions on various occasions: Alyssa, Amir, Anton, Anupam, Armand, Borja, Daniel, Dominik, Giacamo, Graham, Ian, Justus, Lev, Lukas, Matteo, Revantha, Robin, Sebastian, Sonja, Thomas, Timo, Tjeerd. I surely forgot many of you; sorry for my bad memory.

Last but not least, I want to thank all my non-academic friends and my family that kept me sane and provided me with some distraction from work. A special thanks to my friends and family in Austria, I really appreciate that on every visit back home you are happily welcoming me back and I love to spend time with you. Finally, thank you Iris for all your unconditional love and support throughout the years and for the beautiful cover of this thesis.

Titles in the ILLC Dissertation Series:

ILLC DS-2021-03: **Seyyed Hadi Hashemi**
Modeling Users Interacting with Smart Devices

ILLC DS-2021-04: **Sophie Arnoult**
Adjunction in Hierarchical Phrase-Based Translation

ILLC DS-2021-05: **Cian Guilfoyle Chartier**
A Pragmatic Defense of Logical Pluralism

ILLC DS-2021-06: **Zoi Terzopoulou**
Collective Decisions with Incomplete Individual Opinions

ILLC DS-2021-07: **Anthia Solaki**
Logical Models for Bounded Reasoners

ILLC DS-2021-08: **Michael Sejr Schlichtkrull**
Incorporating Structure into Neural Models for Language Processing

ILLC DS-2021-09: **Taichi Uemura**
Abstract and Concrete Type Theories

ILLC DS-2021-10: **Levin Hornischer**
Dynamical Systems via Domains: Toward a Unified Foundation of Symbolic and Non-symbolic Computation

ILLC DS-2021-11: **Sirin Botan**
Strategyproof Social Choice for Restricted Domains

ILLC DS-2021-12: **Michael Cohen**
Dynamic Introspection

ILLC DS-2021-13: **Dazhu Li**
Formal Threads in the Social Fabric: Studies in the Logical Dynamics of Multi-Agent Interaction

ILLC DS-2021-14: **Álvaro Piedrafita**
On Span Programs and Quantum Algorithms

ILLC DS-2022-01: **Anna Bellomo**
Sums, Numbers and Infinity: Collections in Bolzano's Mathematics and Philosophy

ILLC DS-2022-02: **Jan Czajkowski**
Post-Quantum Security of Hash Functions

ILLC DS-2022-03: **Sonia Ramotowska**
Quantifying quantifier representations: Experimental studies, computational modeling, and individual differences

ILLC DS-2022-04: **Ruben Brokkelkamp**
How Close Does It Get?: From Near-Optimal Network Algorithms to Suboptimal Equilibrium Outcomes

ILLC DS-2022-05: **Lwenn Bussière-Carae**
No means No! Speech Acts in Conflict

ILLC DS-2022-06: **Emma Mojet**
Observing Disciplines: Data Practices In and Between Disciplines in the 19th and Early 20th Centuries

ILLC DS-2022-07: **Freek Gerrit Witteveen**
Quantum information theory and many-body physics

ILLC DS-2023-01: **Subhasree Patro**
Quantum Fine-Grained Complexity

ILLC DS-2023-02: **Arjan Cornelissen**
Quantum multivariate estimation and span program algorithms

ILLC DS-2023-03: **Robert Paßmann**
Logical Structure of Constructive Set Theories

ILLC DS-2023-04: **Samira Abnar**
Inductive Biases for Learning Natural Language

ILLC DS-2023-05: **Dean McHugh**
Causation and Modality: Models and Meanings

ILLC DS-2023-06: **Jialiang Yan**
Monotonicity in Intensional Contexts: Weakening and: Pragmatic Effects under Modals and Attitudes

ILLC DS-2023-07: **Yiyan Wang**
Collective Agency: From Philosophical and Logical Perspectives

ILLC DS-2023-08: **Lei Li**
Games, Boards and Play: A Logical Perspective

ILLC DS-2023-09: **Simon Rey**
Variations on Participatory Budgeting

ILLC DS-2023-10: **Mario Giulianelli**
Neural Models of Language Use: Studies of Language Comprehension and Production in Context

ILLC DS-2023-11: **Guillermo Menéndez Turata**
Cyclic Proof Systems for Modal Fixpoint Logics

ILLC DS-2023-12: **Ned J.H. Wontner**
Views From a Peak: Generalisations and Descriptive Set Theory

ILLC DS-2024-01: **Jan Rooduijn**
Fragments and Frame Classes: Towards a Uniform Proof Theory for Modal Fixed Point Logics

ILLC DS-2024-02: **Bas Cornelissen**
Measuring musics: Notes on modes, motifs, and melodies

ILLC DS-2024-03: **Nicola De Cao**
Entity Centric Neural Models for Natural Language Processing

ILLC DS-2024-04: **Ece Takmaz**
Visual and Linguistic Processes in Deep Neural Networks: A Cognitive Perspective

ILLC DS-2024-05: **Fatemeh Seifan**
Coalgebraic fixpoint logic Expressivity and completeness result

ILLC DS-2024-06: **Jana Sotáková**
Isogenies and Cryptography

ILLC DS-2024-07: **Marco Degano**
Indefinites and their values

ILLC DS-2024-08: **Philip Verduyn Lunel**
Quantum Position Verification: Loss-tolerant Protocols and Fundamental Limits

ILLC DS-2024-09: **Rene Allerstorfer**
Position-based Quantum Cryptography: From Theory towards Practice

ILLC DS-2024-10: **Willem Feijen**
Fast, Right, or Best? Algorithms for Practical Optimization Problems

ILLC DS-2024-11: **Daira Pinto Prieto**
Combining Uncertain Evidence: Logic and Complexity

ILLC DS-2024-12: **Yanlin Chen**
On Quantum Algorithms and Limitations for Convex Optimization and Lattice Problems

ILLC DS-2024-13: **Jaap Jumelet**
Finding Structure in Language Models

ILLC DS-2025-01: **Julian Chingoma**
On Proportionality in Complex Domains

ILLC DS-2025-02: **Dmitry Grinko**
Mixed Schur-Weyl duality in quantum information

ILLC DS-2025-03: **Rochelle Choenni**
Multilinguality and Multiculturalism: Towards more Effective and Inclusive Neural Language Models

ILLC DS-2025-04: **Aleksi Anttila**
Not Nothing: Nonemptiness in Team Semantics

ILLC DS-2025-05: **Niels M. P. Neumann**
Adaptive Quantum Computers: decoding and state preparation

ILLC DS-2025-06: **Alina Leidinger**
Towards Language Models that benefit us all: Studies on stereotypes, robustness, and values

ILLC DS-2025-07: **Zhi Zhang**
Advancing Vision and Language Models through Commonsense Knowledge, Efficient Adaptation and Transparency

ILLC DS-2025-08: **Sophie Klumper**
The Gap and the Gain: Improving the Approximate Mechanism Design Frontier in Constrained Environments

ILLC DS-2026-01: **Bryan Eikema**
A Sampling-Based Exploration of Neural Text Generation Models

ILLC DS-2026-02: **Marten Folkertsma**
Empowering Quantum Computation with: Measurements, Catalysts, and Guiding States

ILLC DS-2026-03: **Valentin Richard**
Presuppositional and Dynamic Aspects of Questions

ILLC DS-2026-04: **Puyu Yang**
Bringing Science to the Public: The Role of Wikipedia in Scientific Communication

Institute for Logic, Language and Computation