Formal Verification of Processes

Ulrich Schöpp

Master of Science

School of Computer Science
Division of Informatics
University of Edinburgh

2001

Abstract

We consider the problem of formally verifying of processes given in a process algebra with respect to specifications in the modal μ-calculus. For this problem, we present a Gentzen-style sequent calculus which can be used for different process algebras, and which allows compositional reasoning. The key property of this calculus, due to Dam and Gurov, is the use of explicit ordinal approximations in order to deal with properties expressed by fixed-point operators.

The model checking problem is undecidable for many process algebras allowing infinite-state processes. Therefore the sequent calculus must in general be incomplete. It is nevertheless possible to obtain completeness results for restricted classes of infinite-state processes.

As the main result of this dissertation, we establish the completeness of this sequent calculus for context-free processes. By examining a direct completeness argument, we extracted a specific tableau system for context-free processes and the full modal μ-calculus. By means of this tableau system, the proof of completeness for the sequent calculus is factorised into two manageable parts: Establishing soundness of the tableau system by reducing it to the sequent calculus, and showing completeness for the tableau system.

Acknowledgements

I would like to thank my supervisor Alex Simpson for his help, encouragement and for so much of his valuable time.
I would also like to thank my family for their constant support.

Declaration

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

(*Ulrich Schöpp*)

Table of Contents

Chapter 1

Introduction

1.1 Motivation

Concurrent systems are ubiquitous in computer science. Prominent examples are network protocols, operating systems and databases. These systems are at the heart of many critical applications, so it is very desirable to ensure that these systems work correctly. The most common approach to ensure this is testing. However, especially for concurrent system testing is likely to be insufficient, since even small concurrent systems can have a rich and complex behaviour. Consequently, the verification problem has been approached with formal methods.

One of the most promising approach of formal verification is *model checking*, where the desired properties of the system are specified in a temporal logic and a decision procedure then automatically decides whether the system satisfies these properties. Much research in the area of model checking has been focused on process algebras and the modal μ-calculus. The characteristic feature of process algebras is the use of an algebraic syntax which provides operators to build larger processes from smaller ones. Several algebraic process calculi have been considered, such as CSP [22], CCS [29] or ACP [26], their main objective is to allow the modelling of concurrent system in an abstract formal framework. The modal μ-calculus is a very powerful temporal logic which combines standard modal logic with least and greatest fixed point operators. It is of particular interest since it subsumes many other temporal logics considered for

program verification.

A very desirable property of formal verification methods is compositionality. Concurrent systems are often built in a modular fashion, that is a concurrent system is composed of smaller components. It is therefore desirable to verify the components of a system separately, and to use correctness results of the components to make conclusions about the correctness of the composed system. This approach reduces the complexity of both the specification and the verification, it helps the verification task to be integrated with the the design process.

Several techniques have been proposed for verifying processes against μ-calculus properties. Most of these algorithms, tableau systems and proof systems are based on global state-space exploration ([17, 39, 5, 11] and many more). Compositional systems have been developed for example in [27] or [2]. However, most of these techniques are only applicable to finite state system.

When modelling a concurrent system in a process algebra like CCS one easily encounters infinite state processes like unbounded stacks or idealised communication channels. One is therefore interested in verifying properties of infinite state processes. However, methods based on global state-space exploration are not applicable to infinite state systems. In fact, due to the expressive power of algebras like CCS, the verification problem for infinite state processes is in general undecidable. This is even the case for very simple infinite subclasses of CCS like BPP [18]. Nevertheless, the model checking problem has been solved for certain classes of infinite state processes. One such result, due to Muller and Schupp [30], is that model checking is decidable for pushdown transition graphs. Again, several algorithms and proof systems have been considered for model checking in different settings of process representation and program logic ([24, 23, 9, 10, 41]). An overview of results for infinite-state systems can be found in [32, 8, 18].

In the light of undecidability, we will consider a more general verification problem than model checking. We will consider the problem of formally, but not necessarily automatically, verifying processes against properties given in a temporal logic.

In consideration of this multitude of specialised systems, one would like general purpose systems which are not specifically tailored to one particular problem. In the

face of undecidability, one cannot hope to find complete general purpose systems. Nevertheless, one would like systems that allow intuitive methods of reasoning that are sufficient in practice.

In this report we investigate such a general purpose system, strongly based on a system proposed by Dam et al. in [15, 14]. The system is based on proving the validity of sequents $\Gamma \vdash \Delta$, where Γ and Δ contain correctness assertions $p : \varphi$, meaning that the process p has the property φ. The system can be used easily for a wide class of process algebras. Since it necessarily incomplete, we have to ask to what extent this system suffices in practice. We investigate how suitable the system is for the problems that are addressed by the different systems mentioned above.

Consider, for example, the problem of validity of a formula in the modal μ-calculus. The question here is whether a certain formula is satisfied by all processes. It has been shown that this problem is decidable by a special finitary axiomatisation [28, 40]. Validity assertions can be expressed in the sequent calculus by considering sequents of the form $\vdash x : \varphi$ where x is a process variable. Indeed, Dam and Gurov [15] have shown that in the proof system we will consider every valid sequent of this form is derivable.

Another interesting question is how well the system supports compositional reasoning. The sequent calculus supports compositionality by allowing processes with process variables. The sequent

$$x_1 : \varphi_1, \ldots, x_n : \varphi_n \vdash p(x_1, \ldots, x_n) : \varphi \qquad (A)$$

expresses that the processes p has the property φ under the assumption that its unspecified components $x_1, \ldots, x_n$ have the properties $\varphi_1, \ldots, \varphi_n$. Using sequents of this form allows reasoning about partly instantiated system, thus allowing the early discovery of errors in the system specification. Sequents of this form occur naturally in an application of the following, simplified, sub-term cut rule.

$$\frac{\vdash\ p(q) : \varphi}{\vdash\ q : \psi \qquad x : \psi \vdash\ p(x) : \varphi}$$

The goal of this rule, written above the line, expresses that a process p with a component q has a property φ. By virtue of this rule, this goal can be replaced with two subgoals. Namely, it is enough to find a property ψ which holds for the component

q, and to show that, under the assumption that the unspecified process x satisfies ψ, the process $p(x)$ has the property φ. This models a natural form of reasoning, and it is therefore interesting to which extent sequents of the form (A) are provable in the sequent calculus.

In this context, it is also interesting to investigate, the expressiveness of sequents of the form (A). This question depends on the process calculus used for p as well as the expressivity of the logic used for the formulae. The expressivity of the process algebras already imposes some restrictions on this question. For example, due to internal communication, one cannot expect the problem of proving a property φ for a general CCS process of the form $p|q$ to be decomposable in a problem for p and q.

Another issue is to investigate the power of the sequent calculus for certain classes of process algebras. The question here is whether the system is sufficient to prove all valid sequents of the form $\vdash p : \varphi$, where p is a closed process in the class that is under consideration. Since such a completeness result implies that the model checking problem for this class is decidable, it will therefore only be possible to obtain completeness for very restricted classes of processes. The sequent calculus has been known to be complete for finite state systems [12]. However it is an interesting question for which classes of infinite state processes the sequent calculus is complete.

In this report we consider the class of context-free processes, one of the first classes beyond finite state processes. The model checking problem for this class has been solved by different algorithms [9, 10] and proof-systems [24]. However, all of these methods where specifically designed for context-free processes. In this report, we will show that the general-purpose sequent calculus is complete for context-free processes. In order to obtain this result we make some crucial extensions to the sequent calculus by Dam et al. As a byproduct of the completeness proof, we obtain a tableau system for context-free processes and the full modal μ-calculus.

1.2 Synopsis of the Report

In chapter 3 we will present the sequent calculus and we shall prove it to be sound. We illustrate how the sequent calculus can be extended with proof rules for process

calculus operators by presenting appropriate proof rules for context-free processes.

In chapter 4 we give a tableau for context-free processes. This tableau addresses the verification problem for the full modal μ-calculus for context-free processes. We provide the soundness of the tableau by a reduction to the sequent calculus from chapter 3.

In chapter 5 we consider the completeness of the tableau system. We will use model-checking games [38] to prove the completeness. By means of the reduction of the tableau to the sequent calculus in chapter 4, we thus obtain that the sequent calculus is complete for context-free processes.

Chapter 2

Definitions

2.1 Processes

A common approach in concurrency theory is to describe processes as terms in a process algebra. Several algebraic frameworks have been considered in this aim, such as the Calculus of Communicating Systems (CCS) introduced by Milner [29] or the Algebra of Communicating Processes (ACP) due to Bergstra and Klop [26]. Both systems allow the specification of very powerful computational systems, which has the drawback that many interesting problems are infeasible. Consequently, several sub-calculi have been considered.

In this report, we will consider the Basic Process Algebra (BPA) [26], a subclass of ACP. This process algebra admits the specification of processes with an infinite state space. The signature of BPA consists of a set of constants Act, the distinguished constant ε and two binary operators $\cdot$ and $+$. The constant ε is interpreted as the process that cannot do any action, $\cdot$ represents sequential composition, and $+$ represents non-deterministic choice. The operational semantics is given in the style of structural operational semantics, introduced by Plotkin [35]. It defines, for any $a \in Act$, an action relation $\xrightarrow{a}$ between process terms:

$$\frac{}{a \xrightarrow{a} \varepsilon} \qquad \frac{p \xrightarrow{a} \varepsilon}{p \cdot q \xrightarrow{a} q} \qquad \frac{p \xrightarrow{a} p'}{p \cdot q \xrightarrow{a} p' \cdot q} \text{ if } p' \neq \varepsilon$$

$$\frac{p \xrightarrow{a} p'}{p+q \xrightarrow{a} p'} \qquad \frac{q \xrightarrow{a} q'}{p+q \xrightarrow{a} q'}$$

Process terms in the BPA-signature are not very expressive, for example such a process can only engage in a finite sequence of actions. The definition of a BPA process is therefore extended to allow (guarded) recursion.

For the ease of presentation we will use *context-free processes* in this report. The class of context-free process coincides with the class of processes defined by guarded recursion in BPA. Context-free processes are described by a context-free grammar, and since every context-free grammar can be translated effectively into Greibach Normal Form (GNF), we can restrict our attention to processes described by a context-free grammar in GNF.

2.1.1 Context-Free Processes

Definition 1 (Context-Free Process). A *context-free process* (V, Act, P, p_1) in Greibach Normal Form consists of a

- a finite set of process variables or nonterminals $V = \{p_1, \ldots, p_n\}$,
- a finite set of actions or terminals Act,
- a finite set of productions $P \subseteq V \times Act \times V^*$, written $p \xrightarrow{a} \overline{p}$ for $(p, a, \overline{p}) \in P$. We assume that some rule $p \xrightarrow{a} \overline{p}$ exists for each $p \in V$.
- a start-variable $p_1 \in V$

2.1.2 Labelled Transition System

Processes are modelled using labelled transition systems. A *labelled transition system* $\mathcal{T}$ is a triple $(T, Act, \rightarrow)$ consisting of a possibly infinite set T of states, a set Act of actions and a transition relation $\rightarrow \subseteq T \times Act \times T$. Instead of $(p, a, q) \in \rightarrow$ we will write

$p \xrightarrow{a} q$, and we will also refer to a labelled transition system as a *model*. Throughout this report, we will use the letters s and t to range over states in a transition system.

Every context-free process defines a labelled transition system. To the context-free process (V, Act, P, p_1), we associate the state p_1 in the labelled transition system $(V^*, Act, \rightarrow)$ where there are no transitions leading from ε, and $pq \xrightarrow{a} \overline{p}q$ if $p \xrightarrow{a} \overline{p}$ is a production in P.

For an example consider the process $(\{p\}, \{a,b\}, P, p)$ where P is given by $p \xrightarrow{a} pp$ and $p \xrightarrow{b} \varepsilon$. This process has the following transition system

$$\varepsilon \xleftarrow{b} p \underset{b}{\overset{a}{\rightleftarrows}} pp \underset{b}{\overset{a}{\rightleftarrows}} ppp \underset{b}{\overset{a}{\rightleftarrows}} pppp \underset{b}{\overset{a}{\rightleftarrows}} \cdots$$

Note also that this simple process has infinitely many non-bisimilar states.

2.2 Process Terms

In the sequent calculus we will represent processes by process terms.

Definition 2 (Context-Free Process Term). Let (V, Act, P, p_1) be a context-free process. The set of *process terms* for this process is the smallest set such that

- A process variable x is a process term, and
- The constant ε is a process term, and
- If $p \in V$ is a nonterminal and q is a process terms then pq is a process term.

The terms p, $p\varepsilon$ and εp will be identified.

We will use the letters p, q, r to range over process terms.

The interpretation of a process term containing is defined using a process environment. A *process environment* P is a function mapping process variables to states of the labelled transition system belonging to (V, Act, P, p_1). The function P is extended to process terms in the evident way, i.e. $\mathsf{P}(\varepsilon) = \varepsilon$, $\mathsf{P}(p) = p$ if $p \in V$, and $\mathsf{P}(pq) = \mathsf{P}(p)\mathsf{P}(q)$.

2.3 Modal μ-calculus

We present the modal μ-calculus, a modal logic enriched with fixpoint operators. The form we use here is due to Kozen [28], however, fixed point operators in logics have been used earlier, for example by Park in [33]. An overview of modal logics and μ-calculi can be found in [6].

2.3.1 Syntax

Definition 3 (μ-calculus Formula). The set of *formulas* of the modal μ-calculus is the smallest set such that:

- A variable X is a formula.
- If a is an action and φ, φ_1 and φ_2 are formulas then $\varphi_1 \wedge \varphi_2$, $\varphi_1 \vee \varphi_2$, $[a]\varphi$, $\langle a \rangle \varphi$, $\mu X.\varphi$ and $\nu X.\varphi$ are formulae.

Throughout this report we will use σ to stand for either μ or ν.
The set of free variables is defined inductively on the structure of the formulae:

$$\begin{aligned} FV(X) &= \{X\} \\ FV(\varphi_1 \wedge \varphi_2) = FV(\varphi_1 \vee \varphi_2) &= FV(\varphi_1) \cup FV(\varphi_2) \\ FV([a].\varphi) = FV(\langle a \rangle.\varphi) &= FV(\varphi) \\ FV(\mu X.\varphi) = FV(\nu X.\varphi) &= FV(\varphi) \setminus \{X\} \end{aligned}$$

A formula is *closed* is it has no free variables.

We write $\varphi[\psi/X]$ to refer to the formula φ where all free occurences of X are replaced by ψ. We will use the symbol $\equiv$ to mean syntactic identity.

We inductively define the set of subterms of a formula

$$\begin{aligned} ST(X) &= \{X\} \\ ST(\varphi_1 \wedge \varphi_2) &= \{\varphi_1 \wedge \varphi_2\} \cup ST(\varphi_1) \cup ST(\varphi_2) \\ ST(\varphi_1 \vee \varphi_2) &= \{\varphi_1 \vee \varphi_2\} \cup ST(\varphi_1) \cup ST(\varphi_2) \\ ST([a].\varphi) &= \{[a].\varphi\} \cup ST(\varphi) \\ ST(\langle a \rangle.\varphi) &= \{\langle a \rangle.\varphi\} \cup ST(\varphi) \\ ST(\mu X.\varphi) &= \{\mu X.\varphi\} \cup ST(\varphi) \\ ST(\nu X.\varphi) &= \{\nu X.\varphi\} \cup ST(\varphi) \end{aligned}$$

The subterm-relation $\leq$ is defined, such that $\varphi \leq \psi$ if, and only if, $\varphi \in ST(\psi)$. This relation is a partial order [38]. As usual, we use the strict order $<$, such that $\varphi < \psi$ if, and only if, $\varphi \leq \psi$ and $\varphi \not\equiv \psi$.

2.3.2 Semantics

The semantics of a formula is defined relative to a model $(T, Act, \rightarrow)$ and a propositional environment. A *propositional environment* V is a function mapping logical variables to subsets of T. It is used to assign meaning to free propositional variables.

The interpretation of a formula is defined inductively on the structure of the formulae.

$$\begin{aligned} \|X\|^T_{\mathsf{V}} &= \mathsf{V}(X) \\ \|\varphi_1 \vee \varphi_2\|^T_{\mathsf{V}} &= \|\varphi_1\|^T_{\mathsf{V}} \cup \|\varphi_2\|^T_{\mathsf{V}} \\ \|\varphi_1 \wedge \varphi_2\|^T_{\mathsf{V}} &= \|\varphi_1\|^T_{\mathsf{V}} \cap \|\varphi_2\|^T_{\mathsf{V}} \\ \|[a]\varphi\|^T_{\mathsf{V}} &= \left\{ s \in T \mid \forall t \in T. s \xrightarrow{a} t \text{ implies } t \in \|\varphi\|^T_{\mathsf{V}} \right\} \\ \|\langle a \rangle \varphi\|^T_{\mathsf{V}} &= \left\{ s \in T \mid \exists t \in T. s \xrightarrow{a} t \text{ and } t \in \|\varphi\|^T_{\mathsf{V}} \right\} \\ \|\mu X.\varphi\|^T_{\mathsf{V}} &= \bigcap \left\{ S \subseteq T \mid \|\varphi\|^T_{\mathsf{V}[S/X]} \subseteq S \right\} \\ \|\nu X.\varphi\|^T_{\mathsf{V}} &= \bigcup \left\{ S \subseteq T \mid S \subseteq \|\varphi\|^T_{\mathsf{V}[S/X]} \right\} \end{aligned}$$

The Knaster-Tarski-Theorem tells us that this definition of $\|\mu X.\varphi\|$ and $\|\nu X.\varphi\|$ gives us the least and the greatest fixed point, which indeed exist, of the function $f(S) = \|\varphi\|_{V[S/X]}$ respectively.

2.4 Modal μ-calculus with explicit Approximations

Standard theory tells us that if f is a monotonic function on a complete lattice then we can construct the least fixed point of f by iterating it on the bottom element on the lattice. This iteration has, in general, to be transfinite.

As the interpretation of $\mu X.\varphi$ is defined to be the least fixed point of the monotonic function $f(S) = \|\varphi\|_{V[S/X]}$ on the subset lattice on T, we can compute $\|\mu X.\varphi\|$ by repeated iteration of f on $\emptyset$. By monotonicity we get

$$\emptyset \subseteq f(\emptyset) \subseteq f^2(\emptyset) \subseteq f^3(\emptyset) \subseteq \ldots \subseteq f^\omega(\emptyset) \subseteq \ldots \subseteq f^\kappa(\emptyset) = f^{\kappa+1}(\emptyset),$$

where $\|\mu X.\varphi\| = f^\kappa(\emptyset)$ and κ is an ordinal. In this sense we can see the iterations $f^\lambda(\emptyset)$ to be ordinal approximations of $\mu X.\varphi$, and, motivated by this, the notation $\mu^\lambda X.\varphi$ is used for $f^\lambda(\emptyset)$. It is standard to write these approximations as

$$\mu^\kappa X.\varphi = \bigcup_{\lambda<\kappa} \varphi[\mu^\lambda X.\varphi / X], \qquad \mu X.\varphi = \bigcup_{\lambda} \varphi[\mu^\lambda X.\varphi / X].$$

The definition of an empty union gives us the initial case $\mu^0 X.\varphi = \emptyset$.

Dually, we have approximated greatest fixed points $\nu^\kappa X.\varphi$ which represent iterations of f on T. We have

$$\nu^\kappa X.\varphi = \bigcap_{\lambda<\kappa} \varphi[\nu^\lambda X.\varphi / X], \qquad \nu X.\varphi = \bigcap_{\lambda} \varphi[\nu^\lambda X.\varphi / X].$$

Here, the inital case $\nu^0 X.\varphi = T$ is given by the definition of an empty intersection.

The sequent calculus we are going to consider in the next chapter uses approximations explicitly in the formulae. We will now define the extension of the modal μ-calculus using explicit approximations with ordinal variables, that the sequent calculus will use.

2.4.1 Syntax

We extend the μ-calculus by ordinal approximation, we use the symbols α, β and γ to stand for ordinal variables. Ordinals will be written using the symbols κ and λ.

Definition 4. The set of μ^α-formulae of the modal μ-calculus with explicit approximations is the smallest set such that

- A μ-calculus formula is a μ^α-formula.
- If φ is a μ^α-formula and α and β are ordinal variables then $\mu^\alpha X.\varphi$, $\nu^\alpha X.\varphi$, $\forall\alpha.\varphi$, $\exists\alpha.\varphi$, $\forall(\alpha<\beta).\varphi$ and $\exists(\alpha<\beta).\varphi$ are μ^α-formulae.

The definition of free variables is extended to ordinal variables:

$$\begin{aligned}
FV(\mu^\alpha X.\varphi) &= \{\alpha\}\cup(FV(\varphi)\setminus\{X\})\\
FV(\nu^\alpha X.\varphi) &= \{\alpha\}\cup(FV(\varphi)\setminus\{X\})\\
FV(\forall\alpha.\varphi) &= FV(\varphi)\setminus\{\alpha\}\\
FV(\exists\alpha.\varphi) &= FV(\varphi)\setminus\{\alpha\}\\
FV(\forall(\alpha<\beta).\varphi) &= \{\beta\}\cup(FV(\varphi)\setminus\{\alpha\})\\
FV(\exists(\alpha<\beta).\varphi) &= \{\beta\}\cup(FV(\varphi)\setminus\{\alpha\})
\end{aligned}$$

2.4.2 Semantics

The semantics of a μ^α-formula is defined, as before, relative to a model $(T,Act,\rightarrow)$ and a propositional environment V. However, the formulae may contain free ordinal variables. An *ordinal environment* O , which is a function mapping ordinal variables to ordinals, is used to assign a meaning to these free ordinal variables.

The semantics of the μ-calculus formulae is now extended to μ^α-formulae.

$$\|\mu^\alpha X.\varphi\|^T_{\mathsf{V},\mathsf{O}} = \bigcup_{\kappa<\mathsf{O}(\alpha)} \|\varphi\|^T_{\mathsf{M}(\kappa),\mathsf{O}}$$

where $\mathsf{M}(\kappa) = \mathsf{V}\left[\left\|\mu^{\beta}X.\varphi\right\|^{T}_{\mathsf{V},\mathsf{O}[\kappa/\beta]} \Big/ X\right]$ for some β which does not occur in φ.

$$\|\exists\alpha.\varphi\|^{T}_{\mathsf{V},\mathsf{O}} = \bigcup_{\kappa} \|\varphi\|^{T}_{\mathsf{V},\mathsf{O}[\kappa/\alpha]}$$
$$\|\exists(\alpha<\beta).\varphi\|^{T}_{\mathsf{V},\mathsf{O}} = \bigcup_{\kappa<\mathsf{O}(\beta)} \|\varphi\|^{T}_{\mathsf{V},\mathsf{O}[\kappa/\alpha]}$$

The interpretation of $\mu^{\alpha}X.\varphi$ is defined by mutual recursion with the function M. Transfinite induction shows that this is well-defined.

Dually we have

$$\|\nu^{\alpha}X.\varphi\|^{T}_{\mathsf{V},\mathsf{O}} = \bigcap_{\kappa<\mathsf{O}(\alpha)} \|\varphi\|^{T}_{\mathsf{N}(\kappa),\mathsf{O}}$$

where $\mathsf{N}(\kappa) = \mathsf{V}\left[\left\|\nu^{\beta}X.\varphi\right\|^{T}_{\mathsf{V},\mathsf{O}[\kappa/\beta]} \Big/ X\right]$ for some β which does not occur in φ.

$$\|\forall\alpha.\varphi\|^{T}_{\mathsf{V},\mathsf{O}} = \bigcap_{\kappa} \|\varphi\|^{T}_{\mathsf{V},\mathsf{O}[\kappa/\alpha]}$$
$$\|\forall(\alpha<\beta).\varphi\|^{T}_{\mathsf{V},\mathsf{O}} = \bigcap_{\kappa<\mathsf{O}(\beta)} \|\varphi\|^{T}_{\mathsf{V},\mathsf{O}[\kappa/\alpha]}$$

The following Propostion follows from the discussion at the beginning of this section.

Proposition 1.

$$\|\mu^{\alpha}X.\varphi\|_{\mathsf{V},\mathsf{O}} \Longleftrightarrow \left\|\exists(\beta<\alpha).\,\varphi\left[\mu^{\beta}X.\varphi/X\right]\right\|_{\mathsf{V},\mathsf{O}}$$
$$\|\nu^{\alpha}X.\varphi\|_{\mathsf{V},\mathsf{O}} \Longleftrightarrow \left\|\forall(\beta<\alpha).\,\varphi\left[\nu^{\beta}X.\varphi/X\right]\right\|_{\mathsf{V},\mathsf{O}}$$
$$\|\mu X.\varphi\|_{\mathsf{V},\mathsf{O}} \Longleftrightarrow \bigcup_{\kappa} \|\mu^{\alpha}X.\varphi\|_{\mathsf{V},\mathsf{O}[\kappa/\alpha]} \Longleftrightarrow \|\exists\alpha.\mu^{\alpha}X.\varphi\|_{\mathsf{V},\mathsf{O}}$$
$$\|\nu X.\varphi\|_{\mathsf{V},\mathsf{O}} \Longleftrightarrow \bigcap_{\kappa} \|\nu^{\alpha}X.\varphi\|_{\mathsf{V},\mathsf{O}[\kappa/\alpha]} \Longleftrightarrow \|\forall\alpha.\nu^{\alpha}X.\varphi\|_{\mathsf{V},\mathsf{O}}$$

2.5 Ordinal Constraints

The intention of using ordinal variables to approximate fixed points is to keep track of fixed point unfoldings. In order to relate unfoldings, we must be able to put different ordinal variables in relation. To this end ordinal constraints are introduced.

We use a strict partial order $O = (|O|, <_O)$ of ordinal variables, i.e. where set $|O|$ is a set of ordinal variables and $<_O$ is a irreflexive and transitive relation on $|O|$. Such a partial order is referred to as a set of *ordinal constraints*. We will sometimes use $\leq_O$, meaning that $\alpha \leq_O \beta$ if, and only if, $\alpha <_O \beta$ or $\alpha \equiv \beta$.

For two partial orders O and $\mathcal{P}$, we use the notation $O \subseteq \mathcal{P}$ to mean $|O| \subseteq |\mathcal{P}|$ and $<_O \subseteq <_{\mathcal{P}}$. We will also write $<_O \cup R$ to mean the transitive closure of this union.

An ordinal environment O satisfies the ordinal constraints O, if $\mathsf{O}(\alpha) < \mathsf{O}(\beta)$ whenever $\alpha <_O \beta$.

2.6 Judgements

2.6.1 Syntax

Our proof system will deal with two kinds of judgements:

1. *Process judgements* express that a process-term p has the property φ. Such judgements are written as $p : \varphi$.

2. *Transition judgements* express that a process p can do action a to become process q. The syntax is $p \xrightarrow{a} q$.

2.6.2 Semantics

The semantics of judgements is defined with respect to a model $\mathcal{T} = (T, Act, \rightarrow)$ and an environment E. An *environment* E is a triple $(\mathsf{P}, \mathsf{V}, \mathsf{O})$ consisting of a process environment, a propositional environment and an ordinal environment.

We define the satisfaction relation, $\mathcal{T} \Vdash_{\mathsf{E}} \varphi$, between a model $\mathcal{T}$, an environment E and judgement $p : \varphi$. It expresses that $p : \varphi$ is validated in the model $\mathcal{T}$ where all free variables in $p : \varphi$ are assigned by E.

$$\begin{aligned} \mathcal{T} \Vdash_{\mathsf{E}} p : \varphi \quad &\text{iff} \quad \mathsf{P}(p) \in \|\varphi\|^{T}_{\mathsf{V},\mathsf{O}} \\ \mathcal{T} \Vdash_{\mathsf{E}} p \xrightarrow{a} q \quad &\text{iff} \quad \mathsf{P}(p) \xrightarrow{a} \mathsf{P}(q) \end{aligned}$$

In a slight abuse of notation we also write

$$\mathcal{T} \Vdash_{\mathsf{E}} O \quad \text{iff} \quad \mathsf{O} \text{ satisfies } O$$

For lists of judgements Δ we write $\mathcal{T} \Vdash_{\mathsf{E}} \Delta$ to mean that, for all φ in Δ, $\mathcal{T} \Vdash_{\mathsf{E}} \varphi$.

Chapter 3

Sequent Calculus

3.1 Synopsis

In this chapter we introduce a Gentzen-style [20] sequent calculus for proving properties of processes with respect to the modal μ-calculus. Sequent calculi for this purpose have been considered in [4, 13, 14, 15, 19]. The calculus introduced by Beattie [4] uses explicit induction and coinduction arguments to prove properties that are given by fixed-point operators.

The calculus we present here is strongly based on the system by Dam, Gurov and Fredlund [13, 14, 15, 19]. The crucial idea of this calculus is to use explicit approximations of fixed points. Instead of using direct (co)induction arguments, the system is based on a global condition on derivations, which uses the ordinal approximations to apply a well-founded induction on ordinals. The global condition that we will give is a generalisation of the condition given by Dam and Gurov [15]. It is similar to the condition introduced by Fredlund [19].

Syntactically, our system differs from the system by Dam et al. in that we separate assertions about ordinal variables. Instead of allowing assertions of the $\alpha < \beta$ in the sequents, we use a partial order to record relations between ordinal variables separately. Furthermore, we extend the logic with universal quantifiers for ordinal variables, allowing assertions of the form $\forall\alpha.\,\varphi$ and $\forall(\alpha < \beta).\,\varphi$.

The extensions of formulae and global proof-condition will play a crucial role in

the proof of completeness for context-free processes.

3.2 Preliminaries

3.2.1 Sequents

In the proof system we will deal with sequents of the form $\Gamma \vdash_O \Delta$, where Γ and Δ are sets of judgements and O is a partial order of ordinal variables. The sequent $\Gamma \vdash_O \Delta$ is well-formed, if all free ordinal variables in Γ or Δ are elements of $|O|$.

The sequent $\Gamma \vdash_O \Delta$ is *satisfied* by a model $\mathcal{T}$ and environment E, if $\mathcal{T} \Vdash_{\mathsf{E}} \Gamma$ and $\mathcal{T} \Vdash_{\mathsf{E}} O$ implies that there exists a $\psi \in \Delta$ such that $\mathcal{T} \Vdash_{\mathsf{E}} \psi$. A sequent is *falsified* by $(\mathcal{T}, \mathsf{E})$, if it is not satisfied by it. A sequent is *valid*, if it is satisfied by all models and all environments.

3.2.2 Proof Rules

A *proof rule* consists of a *conclusion* sequent $\Gamma \vdash_O \Delta$ and a possibly empty list of *premises* $\Gamma_1 \vdash_{O_1} \Delta_1 \ \ldots \ \Gamma_n \vdash_{O_1} \Delta_n$. Inference rules are be written in the following form:

$$\text{(Name)}\ \frac{\Gamma \vdash_O \Delta}{\Gamma_1 \vdash_{O_1} \Delta_1 \quad \ldots \quad \Gamma_n \vdash_{O_1} \Delta_n}$$

The intuition of a proof rule is that the conclusion sequent is valid whenever all the premises are valid.

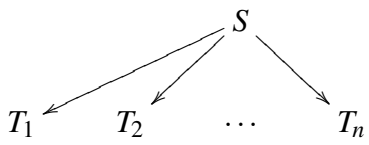
The proof rules for sequent calculi are often written with the conclusion below the line and the premises above the line. We have chosen to write the rules upside down in order to conform with the tableau-style rules in the next chapter, and also to facilitate goal-directed proofs.

3.2.3 Derivation Tree

A sequent S is a *derivation tree* in which S is both leaf and root. Given instance of a proof rule

$$\frac{S}{S_1 \quad S_2 \quad \ldots \quad S_n}$$

and *derivation trees* $T_1, \ldots, T_n$ having the roots $S_1, \ldots, S_n$ respectively, then



is an derivation tree with root S. The leaves of this tree are all the leaves of $T_1, \ldots, T_n$.

We will refer to the directed paths in this tree as *paths* in the derivation.

3.3 Proof Rules

Structural rules

$$\text{(Axiom)}\ \frac{}{\Gamma,\ A\ \vdash_O\ \Delta,\ A}$$

$$\text{(WeakL)}\ \frac{\Gamma,\ A\ \vdash_O\ \Delta}{\Gamma\ \vdash_O\ \Delta} \qquad \text{(WeakR)}\ \frac{\Gamma\ \vdash_O\ \Delta,\ A}{\Gamma\ \vdash_O\ \Delta}$$

$$\text{(Cut)}\ \frac{\Gamma\ \vdash_O\ \Delta}{\Gamma\ \vdash_O\ \Delta,\ p:\varphi \qquad p:\varphi,\ \Gamma\ \vdash_O\ \Delta}$$

$$\text{(TermCut)}^*\ \frac{\Gamma\ \vdash_O\ p[q/x]:\varphi,\ \Delta}{\Gamma\ \vdash_O\ \Delta,\ q:\psi \qquad x:\psi,\ \Gamma\ \vdash_O\ p:\varphi,\ \Delta}$$

*Restriction on (TermCut): The variable x must not occur in the conclusion.

Logical rules

$$(\vee\mathrm{L})\ \frac{\Gamma,\ p:\varphi_1\vee\varphi_2\ \vdash_O\ \Delta}{\Gamma,\ p:\varphi_1\ \vdash_O\ \Delta \qquad \Gamma,\ p:\varphi_2\ \vdash_O\ \Delta} \qquad (\vee\mathrm{R})\ \frac{\Gamma\ \vdash_O\ \Delta,\ p:\varphi_1\vee\varphi_2}{\Gamma\ \vdash_O\ \Delta,\ p:\varphi_1,\ p:\varphi_2}$$

$$(\wedge\mathrm{L})\ \frac{\Gamma,\ p:\varphi_1\wedge\varphi_2\ \vdash_O\ \Delta}{\Gamma,\ p:\varphi_1,\ p:\varphi_2\ \vdash_O\ \Delta} \qquad (\wedge\mathrm{R})\ \frac{\Gamma\ \vdash_O\ \Delta,\ p:\varphi_1\wedge\varphi_2}{\Gamma\ \vdash_O\ \Delta,\ p:\varphi_1 \qquad \Gamma\ \vdash_O\ \Delta,\ p:\varphi_2}$$

Modal rules

$$(\Diamond\mathrm{L})^*\ \frac{\Gamma,\ p:\langle a\rangle\varphi\ \vdash_O\ \Delta}{\Gamma,\ p\xrightarrow{a}x,\ x:\varphi\ \vdash_O\ \Delta} \qquad (\Diamond\mathrm{R})\ \frac{\Gamma\ \vdash_O\ \Delta,\ p:\langle a\rangle\varphi}{\Gamma\ \vdash_O\ \Delta,\ p\xrightarrow{a}q \qquad \Gamma\ \vdash_O\ \Delta,\ q:\varphi}$$

$$(\Box\mathrm{L})\ \frac{\Gamma,\ p:[a]\varphi\ \vdash_O\ \Delta}{\Gamma\ \vdash_O\ \Delta,\ p\xrightarrow{a}q \qquad \Gamma,\ q:\varphi\ \vdash_O\ \Delta} \qquad (\Box\mathrm{R})^*\ \frac{\Gamma\ \vdash_O\ \Delta,\ p:[a]\varphi}{\Gamma,\ p\xrightarrow{a}x\ \vdash_O\ \Delta,\ x:\varphi}$$

*Restriction on ($\Diamond$L) and ($\Box$R): The variable x must not occur in the conclusion.

Fixed point rules

$$(\mu\mathrm{L})\ \frac{\Gamma,\ p:\mu X.\varphi\ \vdash_O\ \Delta}{\Gamma,\ p:\exists\alpha.\mu^\alpha X.\varphi\ \vdash_O\ \Delta} \qquad (\mu\mathrm{R})\ \frac{\Gamma\ \vdash_O\ \Delta,\ p:\mu X.\varphi}{\Gamma\ \vdash_O\ \Delta,\ p:\varphi[\mu X.\varphi/X]}$$

$$(\mu^\alpha\mathrm{L})\ \frac{\Gamma,\ p:\mu^\alpha X.\varphi\ \vdash_O\ \Delta}{\Gamma,\ p:\exists(\beta<\alpha).\varphi\left[\mu^\beta X.\varphi/X\right]\ \vdash_O\ \Delta} \qquad (\mu^\alpha\mathrm{R})\ \frac{\Gamma\ \vdash_O\ \Delta,\ p:\mu^\alpha X.\varphi}{\Gamma\ \vdash_O\ \Delta,\ p:\exists(\beta<\alpha).\varphi\left[\mu^\beta X.\varphi/X\right]}$$

$$(\nu\mathrm{L})\ \frac{\Gamma,\ p:\nu X.\varphi\ \vdash_O\ \Delta}{\Gamma,p:\varphi[\nu X.\varphi/X]\ \vdash_O\ \Delta} \qquad (\nu\mathrm{R})\ \frac{\Gamma\ \vdash_O\ \Delta,\ p:\nu X.\varphi}{\Gamma\ \vdash_O\ \Delta,\ p:\forall\alpha.\nu^\alpha X.\varphi}$$

$$(\nu^\alpha\mathrm{L})\ \frac{\Gamma,\ p:\nu^\alpha X.\varphi\ \vdash_O\ \Delta}{\Gamma,\ p:\forall(\beta<\alpha).\varphi\left[\nu^\beta X.\varphi/X\right]\ \vdash_O\ \Delta} \qquad (\nu^\alpha\mathrm{R})\ \frac{\Gamma\ \vdash_O\ \Delta,\ p:\nu^\alpha X.\varphi}{\Gamma\ \vdash_O\ \Delta,\ p:\forall(\beta<\alpha).\varphi\left[\nu^\beta X.\varphi/X\right]}$$

The fixed point rules are motivated by Proposition 1. Note that the symmetric counterparts of the rules (μL), (μR), (νL) and (νR) are missing. However, for example in case

of (μL), we have $\|\mu X.\varphi\|$ if, and only if, $\|\exists\alpha.\mu^{\alpha}X.\varphi\|$, hence the symmetric counterpart is also admissible:

$$(\mu\text{R1})\ \frac{\Gamma \vdash_O \Delta,\ p : \mu X.\varphi}{\Gamma \vdash_O \Delta,\ p : \exists\alpha.\mu^{\alpha}X.\varphi}$$

However, since we will not need these symmetric rules, we have chosen not to include it in the definition.

Quantifier rules

$$(\exists_{<}\text{L})\ \frac{\Gamma,\ p : \exists(\beta < \alpha).\varphi \vdash_O \Delta}{\Gamma,\ p : \varphi[\gamma/\alpha] \vdash_{\mathcal{P}} \Delta}\ \mathcal{P} = (|O| \cup \{\gamma\},\ <_O \cup \{(\gamma,\alpha)\}) \text{ and } \gamma \notin |O|$$

$$(\exists_{<}\text{R})\ \frac{\Gamma \vdash_O \Delta,\ p : \exists(\beta < \alpha).\varphi}{\Gamma \vdash_O \Delta,\ p : \varphi[\gamma/\beta]}\ \gamma <_O \alpha$$

$$(\exists\text{L})\ \ \frac{\Gamma,\ p : \exists\alpha.\varphi \vdash_O \Delta}{\Gamma,\ p : \varphi[\beta/\alpha] \vdash_{\mathcal{P}} \Delta}\ \mathcal{P} = (|O| \cup \{\beta\}, <_O) \text{ and } \beta \notin |O|$$

$$(\exists\text{R})\ \ \frac{\Gamma \vdash_O \Delta,\ p : \exists\alpha.\varphi}{\Gamma \vdash_O \Delta,\ p : \varphi[\beta/\alpha]}\ \beta \in |O|$$

$$(\forall_{<}\text{L})\ \frac{\Gamma,\ p : \forall(\beta < \alpha).\varphi \vdash_O \Delta}{\Gamma,\ p : \varphi[\gamma/\alpha] \vdash_O \Delta}\ \gamma <_O \alpha$$

$$(\forall_{<}\text{R})\ \frac{\Gamma \vdash_O \Delta,\ p : \forall(\beta < \alpha).\varphi}{\Gamma \vdash_{\mathcal{P}} \Delta,\ p : \varphi[\gamma/\beta]}\ \mathcal{P} = (|O| \cup \{\gamma\},\ <_O \cup \{(\gamma,\alpha)\}) \text{ and } \gamma \notin |O|$$

$$(\forall\text{L})\ \ \frac{\Gamma,\ p : \forall\alpha.\varphi \vdash_O \Delta}{\Gamma,\ p : \varphi[\beta/\alpha] \vdash_O \Delta}\ \beta \in |O|$$

$$(\forall\text{R})\ \ \frac{\Gamma \vdash_O \Delta,\ p : \forall\alpha.\varphi}{\Gamma \vdash_{\mathcal{P}} \Delta,\ p : \varphi[\beta/\alpha]}\ \mathcal{P} = (|O| \cup \{\beta\}, <_O) \text{ and } \beta \notin |O|$$

Ordinal rules

$$(\text{OrdStrength})\ \frac{\Gamma \vdash_O \Delta}{\Gamma \vdash_{\mathcal{P}} \Delta}\ O \subseteq \mathcal{P} \text{ and } \alpha \not<_O \beta \text{ for all } \alpha, \beta \text{ such that } \alpha \in (|\mathcal{P}| \setminus |O|) \text{ and } \beta \in |O|.$$

It is immediate that in each rule the partial order of a premise is an extension of the partial order of the conclusion.

3.4 Definition of Proof

The proof rules by themself are insufficient to establish properties that are given by fixed point formulas, as there is in general no bound on the number of unfoldings of a fixed point. Using only the rules, recursive formulae will in general result in an infinite derivation tree. In order to prove recursive properties effectively, we require a condition which allows us to consider a finite part of an infinite derivation tree to be a proof. We will use a condition which employs well-founded induction on ordinals. This condition is applicable to derivation trees of a special form, each of which represents - in a certain sense - an infinite derivation. This special form derivation trees is characterised by the property that each leaf node is a repeat of an inner node.

In order to make the notion of a repeat precise, we need some notation. Let σ be a triple of functions $\sigma = (\sigma_{\mathsf{P}}, \sigma_{\mathsf{V}}, \sigma_{\mathsf{O}})$, where σ_{P} is a function mapping process variables to process terms, σ_{V} is a function mapping logical variables to formulas, and σ_{O} is a function mapping ordinal variables to ordinal variables. The function σ can be extended to a substitution acting on judgements. Similarly, we can extend P, V and O to terms of the respective kind. Given an environment $\mathsf{E} = (\mathsf{P}, \mathsf{V}, \mathsf{O})$, $\mathsf{E} \circ \sigma$ will be used as an abbreviation for $(\mathsf{P} \circ \sigma_{\mathsf{P}}, \mathsf{V} \circ \sigma_{\mathsf{V}}, \mathsf{O} \circ \sigma_{\mathsf{O}})$.

Definition 5 (Repeat). Let M and N be different nodes in a derivation tree such that there exists a path from M to N. Let M and N be labelled with the sequents $\Gamma_M \vdash_{\mathcal{P}} \Delta_M$ and $\Gamma_N \vdash_O \Delta_N$ respectively, and let σ be a triple of functions as above. The node N is said to be a *repeat node* of M up to σ, if the following conditions hold

- $\varphi \in \Gamma_M$ implies $\sigma(\varphi) \in \Gamma_N$, and

- $\varphi \in \Delta_M$ implies $\sigma(\varphi) \in \Delta_N$, and
- $\alpha <_{\mathcal{P}} \beta$ implies $\sigma(\alpha) <_O \sigma(\beta)$.

We refer to (M,N,σ) as a *repeat*.

Definition 6 (Pre-Proof). A *pre-proof* is a derivation tree $\mathcal{D}$ together with a set $\mathcal{R}$ of repeats such that $\mathcal{R}$ contains exactly one repeat (M,N,σ) for each leaf N of $\mathcal{D}$.

Definition 7 (Pre-Proof Graph). Let $\mathcal{P} = (\mathcal{D},\mathcal{R})$ be a pre-proof. The graph $\mathcal{D}$ extended with the edges $\{(N,M) \mid (M,N,\sigma) \in \mathcal{R}\}$ is called the *pre-proof graph* for $\mathcal{P}$.

We say that a path π in the pre-proof graph contains a repeat (M,N,σ) if π contains the edge (N,M).

We go on to define a condition which establishes a pre-proof to be a proof.

Definition 8 (Discharge Order). A *discharge order* on a pre-proof $(\mathcal{D},\mathcal{R})$ is a partial ordering $\prec$ on the repeats such that if $R = (M,N,\sigma)$ and $R' = (M',N',\sigma')$ are any two repeats for which $R \not\preceq R'$ and $R' \not\preceq R$ then every path $M \ldots N'M' \ldots N$ through the pre-proof graph contains a repeat R'' with $R \prec R''$ and $R' \prec R''$.

Note that any linear order of repeats is a discharge order.

Proposition 2. *It is decidable if a partial order on repeats is a discharge order.*

Proof. It suffices to consider only acyclic paths $M \ldots N'M' \ldots N$ in the pre-proof graph, of which there are finitely many. □

Proposition 3. *Let $\mathcal{P}$ be a pre-proof and let $\prec$ be a partial order. Then the following are equivalent*

1. *$\prec$ is a discharge order on $\mathcal{P}$.*
2. *In every infinite path through the pre-proof graph of $\mathcal{P}$ there is repeat R occurring infinitely often such that $R' \preceq R$ holds for any repeat R' occurring infinitely often.*

Proof.

($1. \Rightarrow 2.$) Consider any infinite path. Any such path has tail π containing only repeats which occur infinitely often. Let $R_1 = (N_1, M_1, \sigma_1), \ldots, R_k = (N_k, M_k, \sigma_k)$ be all repeats that occur infinitely often in π.

We will now show that there is a an i such that $R_j \preceq R_i$ for all j. We assume the contrary, i.e. that for all i there is a j such that $R_j \not\preceq R_i$. Note that $j \neq i$, by reflexivity of $\preceq$. This assumption is now used to construct an infinite ascending chain with respect to $\prec$. Starting with arbitrary R_i, the assumption is that there exists a $j \neq i$ for which $R_j \not\preceq R_i$. If $R_i \prec R_j$, we choose R_j to be the next element of the chain. Suppose $R_i \not\prec R_j$. Because we have assumed $R_j \not\preceq R_i$, we know that then also $R_i \not\preceq R_j$ holds. Since R_i and R_j occur infinitely often in π, we know that π has infinitely many sections the form $N_i \ldots N_j \ldots N_i \ldots N_j$. By the definition of a discharge order, such a path contains a repeat R for which $R_i \prec R$ and $R_j \prec R$. Since there are only finitely many repeats, one such R must occur infinitely often in π, hence R is R_k for some k. We have therefore found an R_k with $R_i \prec R_k$; we choose this R_k to be the next element of the chain.

The construction starts with an arbitrary R_i, so it can be applied repeatedly to give an infinite ascending chain $R_{i_1} \prec R_{i_2} \prec R_{i_3} \prec \ldots$, and since $\prec$ is a partial order on a finite set, this is a contradiction.

($2. \Rightarrow 1.$) Suppose $\prec$ is not a discharge order, i.e. there are two repeats $R = (M, N, \sigma)$ and $R' = (M', N', \sigma')$ such that $R \not\preceq R'$ and $R' \not\preceq R$ and there exists a path $M \ldots N' M' \ldots N$ which does not contain a repeat R'' with $R \prec R''$ and $R' \prec R''$. We then have a cycle $M \ldots N' M' \ldots N M$, and thus an infinite path in which the repeats R and R' occur infinitely often. By assumption, there is no repeat occurring in this path which subsumes R and R' under $\preceq$. This gives a contradiction to the assumption 2.

□

Definition 9 (Progress, Preservation). Let $R = (M, N, \sigma)$ be a repeat, let M and N be labelled with the sequents $\Gamma_M \vdash_{\mathcal{P}} \Delta_M$ and $\Gamma_N \vdash_O \Delta_N$ respectively, and let $\alpha \in |\mathcal{P}|$ be an ordinal variable. We say that R *progresses* on α if $\sigma(\alpha) <_O \alpha$, and R *preserves* α if $\sigma(\alpha) \leq_O \alpha$.

Definition 10 (Discharge). A pre-proof $(\mathcal{D}, \mathcal{R})$ is *dischargeable* with respect to a discharge order $\prec$, if for every repeat R there is an ordinal variable α_R such that R progresses on α_R and R preserves $\alpha_{R'}$ whenever $R \prec R'$.

Definition 11 (Proof). A pre-proof $(\mathcal{D}, \mathcal{R})$ is a *proof* of the root sequent of $\mathcal{D}$ if it is dischargeable with respect to some discharge order $\prec$.

3.4.1 Finding a Repeat

Suppose we want to show that

$$\begin{array}{c} \Gamma_M \vdash_{\mathcal{P}} \Delta_M \\ \vdots \\ \Gamma_N \vdash_{O} \Delta_N \end{array}$$

is a repeat. To do so, we have to find a substitution σ which satisfies, among others, the condition that whenever $\alpha <_{\mathcal{P}} \beta$ then $\sigma(\alpha) <_O \sigma(\beta)$.

In presence of the rule (OrdStrength) it is enough just to demand this condition for the ordinal variables in

$$\mathcal{P}_d = \{\alpha \mid \alpha \leq_{\mathcal{P}} \beta \text{ and } \beta \in FV(\Gamma_M, \Delta_M)\}.$$

Suppose we have a function σ, mapping from $|\mathcal{P}_d|$ to $|O|$, for which $\sigma(\alpha) <_O \sigma(\beta)$ holds whenever $\alpha <_{\mathcal{P}_d} \beta$. We extend this function to $\mathcal{P}$ by mapping the elements of $\mathcal{P} \setminus |\mathcal{P}_d|$ to distinct ordinal variables that are not in O. Using the rule (OrdStrength) we can now extend the partial order O to contain these fresh variables in $\sigma(\mathcal{P} \setminus |\mathcal{P}_d|)$:

$$\text{(OrdStrength)}\ \frac{\Gamma_N \vdash_O \Delta_N}{\Gamma_N \vdash_{O'} \Delta_N}$$

Furthermore, we can do this such that $\sigma(\alpha) <_{O'} \sigma(\beta)$ whenever $\alpha <_{\mathcal{P}} \beta$. The side-condition of (OrdStrength) is satisfied because of the choice of $\mathcal{P}_d$. We have thus constructed a function σ of the required form to give us a repeat.

3.5 Soundness

All interpretations in this section are taken with respect to a fixed model $\mathcal{T}$.

Lemma 1 (Local Soundness). *For each proof rule*

$$\frac{\Gamma \vdash_O \Delta,}{\Gamma_1 \vdash_{O_1} \Delta_1 \quad \ldots \quad \Gamma_n \vdash_{O_n} \Delta_n}$$

and any falsifying interpretation E *of its conclusion sequent* $\Gamma \vdash_O \Delta$*, there exists a falsifying interpretation* E' *of one of its premises* $\Gamma_i \vdash_{O_i} \Delta_i$ *such that* E *and* E' *agree on all common free variables of* $\Gamma \vdash_O \Delta$ *and* $\Gamma_i \vdash_{O_i} \Delta_i$*.*

Proof. Since E is a falsifying interpretation for $\Gamma \vdash_O \Delta$, we know that $\mathsf{E} \Vdash \Gamma$, $\mathsf{E} \Vdash O$, and $\mathsf{E} \nVdash \psi$ for all $\psi \in \Delta$. We consider representative cases:

- (TermCut). By assumption, E invalidates $p[q/x] : \varphi$. If E invalidates $q : \psi$, then E is a falsifying interpretation for the left premise of the rule. Otherwise, suppose $\mathsf{E} \Vdash q : \psi$. Using $\mathsf{E}' = \mathsf{E}[q/x]$, this means that E' validates $x : \psi$. Since x does not occur in the conclusion, and therfore not in q, and by the construction of the process interpretations, we have $\mathsf{E}(p[q/x]) = \mathsf{E}'(p)$. Hence, $\mathsf{E}' \nVdash p : \varphi$. Since x does not occur in Γ or Δ, the judgements in these sets have the same interpretation under E as under E'. We have therefore constructed a falsifying interpretation for the right-hand premise of the rule.

- ($\Diamond$L). Since E validates $p : \langle a \rangle \varphi$, we find some $t \in T$ such that $\mathsf{E}(p) \xrightarrow{a}_T t$ and $t \in \|\varphi\|^T_{\mathsf{E}}$. By letting $\mathsf{E}' = \mathsf{E}[t/x]$, we have that $\mathsf{E}' \Vdash p \xrightarrow{a} x$ and $\mathsf{E}' \Vdash x : \varphi$. As x does not occur in the conclusion, we therefore have a falsifying interpretation for $\Gamma, p \xrightarrow{a} x,\ x : \varphi \vdash_O \Delta$.

- ($\Diamond$R). In this case, E invalidates $p : \langle a \rangle \varphi$. By the definition of the semantics, this means that $\mathsf{E}(p) \not\xrightarrow{a}_T t$ or $t \notin \|\varphi\|^T_{\mathsf{E}}$ for all $t \in T$. In the first case, we have $\mathsf{E} \nVdash p \xrightarrow{a} q$ for any process term q, hence the left premise is falsified by E. In the second case, we have $\mathsf{E} \nVdash q : \varphi$, thus the right premise is falsified by E.

- ($\Box$L), ($\Box$R). These cases follow by a dual argument.

- (ν^αL). The environment E validates $p : \nu^\alpha X.\varphi$. By Proposition 1, E also validates $\forall(\beta < \alpha).\, \varphi\left[\nu^\beta.\varphi/X\right]$. Thus E is a falsifying interpretation for the premise of this rule.

- (ν^αR). This case is similar to (ν^αL), by using the other direction of the implication in Proposition 1.

- (νL), (νR) follow similarly to the previous two cases.

- ($\forall_<$L). By the form of the conclusion sequent, we know that $\mathsf{E} \Vdash p : \forall(\beta < \alpha).\varphi$. The side condition of the rule, $\gamma <_O \alpha$, shows $\mathsf{E}(\gamma) < \mathsf{E}(\alpha)$. Hence, $\mathsf{E} \Vdash p : \varphi[\gamma/\alpha]$. The environment E is thus a falsifying interpretation of the premise of this rule.

- ($\forall_<$R). In this case $\Delta = \Delta',\ p : \forall(\beta < \alpha).\varphi$. We have $\mathsf{E} \nVdash p : \forall(\beta < \alpha).\varphi$. This shows that there exists an ordinal κ with $\kappa < \mathsf{E}(\alpha)$ such that $\mathsf{E}[\kappa/\gamma] \nVdash \varphi[\gamma/\beta]$ for a fresh γ. For the environment $\mathsf{E}' = \mathsf{E}[\kappa/\gamma]$ we therefore have $\mathsf{E}'(\gamma) < \mathsf{E}'(\alpha)$ as well as $\mathsf{E}' \nVdash \varphi[\gamma/\beta]$. Hence, E' falsifies the premise of the rule.

- ($\forall$L). We have $\mathsf{E} \Vdash p : \forall\alpha.\varphi$, and then all the more $\mathsf{E} \Vdash p : \varphi[\beta/\alpha]$ for some $\beta \in O$. We use E as the falsifying interpretation.

- ($\forall$R). We have $\mathsf{E} \nVdash p : \forall\alpha.\varphi$. The partial order of ordinal constraints is extended with a fresh β, we choose $\mathsf{E}' = \mathsf{E}[\kappa/\beta]$ where κ is a witness for $\mathsf{E}' \nVdash p : \forall\alpha.\varphi$. Since $\mathsf{E}' \nVdash p : \forall\alpha.\varphi[\beta/\alpha]$, this E' is a falsifying interpretation of the premise sequent.

- ($\exists_<$L),($\exists_<$R),($\exists$L),($\exists$R). These cases follow by a dual argument to the $\forall$-cases.

- (OrdStrength). Let $\mathsf{E} = (\mathsf{P}, \mathsf{O})$. We have to find an environment which satisfies $\mathcal{P}$. We can use the following function

$$\mathsf{O}'(\beta) = \begin{cases} \mathrm{succ}\,(\max\,(\{\mathsf{O}'(\alpha) \mid \alpha <_{\mathcal{P}} \beta\})) & \text{if } \beta \in (|\mathcal{P}| \setminus |O|), \\ \mathsf{O}(\beta) & \text{otherwise.} \end{cases}$$

  where $\max(\emptyset) = 0$. This is well-defined since $\mathcal{P}$ is finite.

  We now show that we have $\mathsf{O}'(\alpha) < \mathsf{O}'(\beta)$ whenever $\alpha <_{\mathcal{P}} \beta$. Thus, suppose $\alpha <_{\mathcal{P}} \beta$. If $\beta \in (|\mathcal{P}| \setminus |O|)$ then we know that α is in the set $\{\alpha \mid \alpha <_{\mathcal{P}} \beta\}$. Hence the construction gives $\mathsf{O}'(\alpha) < \mathsf{O}'(\beta)$, as required.
  Suppose $\beta \notin (|\mathcal{P}| \setminus |O|)$. From the side-condition of the rule follows that α cannot

be an element of $(|\mathcal{P}| \setminus |O|)$. We therefore have that both α and β are elements of $|O|$. We then have $\mathsf{O}'(\alpha) = \mathsf{O}(\alpha) < \mathsf{O}(\beta) = \mathsf{O}'(\beta)$, because O satisfies the ordinal constraints of the conclusion sequent.

The environment $\mathsf{E}' = (\mathsf{P}, \mathsf{O}')$ therefore has the required property.

□

Lemma 2. *Let $R = (M, N, \sigma)$ be a repeat in a derivation tree $\mathcal{D}$ and let E be an environment falsifying the sequent at N. Then the sequent at M is falsified by $\mathsf{E} \circ \sigma$. Moreover $(\mathsf{E} \circ \sigma)(\alpha) \leq \mathsf{E}(\alpha)$ if R preserves α, and $(\mathsf{E} \circ \sigma)(\alpha) < \mathsf{E}(\alpha)$ if R progresses on α.*

Proof. Let N and M be labelled with the sequents $\Gamma_N \vdash_O \Delta_N$ and $\Gamma_M \vdash_{\mathcal{P}} \Delta_M$ respectively. By assumption, we have $\mathsf{E} \Vdash \Gamma_N$, $\mathsf{E} \Vdash O$ and $\mathsf{E} \nVdash \psi$ for all $\psi \in \Delta_N$. By definition of a repeat, we then have $(\mathsf{E} \circ \sigma) \Vdash \Gamma_M$, $(\mathsf{E} \circ \sigma) \Vdash \mathcal{P}$ and $(\mathsf{E} \circ \sigma) \nVdash \psi$ for all $\psi \in \Delta_M$. Hence $(\mathsf{E} \circ \sigma)$ is a falsifying interpretation for $\Gamma_M \vdash_{\mathcal{P}} \Delta_M$. The assertion about progress and preservation follows immediately from the definition. □

Lemma 3. *Let $\mathcal{P} = (\mathcal{D}, \mathcal{R})$ be a pre-proof and let π be an infinite path in the pre-proof graph of $\mathcal{P}$. Let α be an ordinal variable which is preserved by all repeats that occur infinitely often in π. Then π has a tail π' in which every node is labelled with a sequent $\Gamma \vdash_O \Delta$ such that $\alpha \in |O|$.*

Proof. As π is an infinite path, there must be at least one repeat, say (M, N, σ), occurring infinitely often in π. We consider a tail π' of π which starts with M and which contains only repeats occurring infinitely often.

Let $R_1 = (M_1, N_1, \sigma_1), \ldots, R_n = (M_n, N_n, \sigma_n)$ be all repeats that occur infinitely often in π'. Every node K occurring in π' must be below some M_i, i.e. there is a path from M_i to K in the derivation tree. This is true for the first node M of π', and it remains true after each step along π.

We now note that, due to preservation, α must be in the partial order of the sequent at each M_i, for $i = 1, \ldots, n$. Because of the form of the proof rules, the partial order of ordinal variables can only be extended along a path of the derivation tree. Since any node K in π' is below some M_i, we can therefore conclude that α occurs in the partial order of any K. Hence, π' is a tail with the required property. □

Theorem 1 (Soundness). *Let* $(\mathcal{D}, \mathcal{R})$ *be a proof of* $\Gamma \vdash_O \Delta$. *Then* $\Gamma \vdash_O \Delta$ *is valid.*

Proof. We show the contraposition. Assuming that $\Gamma \vdash_O \Delta$ is not valid, we show that the pre-proof $(\mathcal{D}, \mathcal{R})$ cannot be a proof. Suppose $\Gamma \vdash_O \Delta$, the sequent at the root node N_0 of $\mathcal{D}$, is not valid, i.e. there exists a falsifying environment E_0.

Starting from (N_0, E_0), we now construct an infinite sequence π of pairs (N_i, E_i) where N_i is a node in the pre-proof graph of $(\mathcal{D}, \mathcal{R})$ and E_i is an interpretation falsifying the sequent at N_i. The crucial property of this sequence will be that there is an ordinal variable having its value decreased infinitely often. The sequence is constructed inductively. Let $(N_0, \mathsf{E}_0) \dots (N_i, \mathsf{E}_i)$ be the sequence constructed so far. The next pair is chosen according to the type of N_i.

1. N_i is an inner node of $\mathcal{D}$. Lemma 1 shows that there exists a node N_{i+1} in $\mathcal{D}$ and an environment E_{i+1} falsifying N_{i+1}. Furthermore, the environments E_i and E_{i+1} agree on all variables that occur in the partial order of the sequent at N_i. Choose the pair $(N_{i+1}, \mathsf{E}_{i+1})$.

2. N_i is a leaf node of $\mathcal{D}$. The set $\mathcal{R}$ contains a unique repeat (M_i, N_i, σ_i), and, by Lemma 2, the environment $\mathsf{E}_i \circ \sigma_i$ falsifies M_i. Choose the pair $(M_i, \mathsf{E}_i \circ \sigma_i)$.

We identify the sequence of pairs π with the path through $\mathcal{D}$ that it induces. Proposition 3 tells us that there exists a repeat R which occurs infinitely often such that $R' \preceq R$ for any repeat R' occurring infinitely often. This also holds for any tail of π. We consider the tail π' which contains only repeats that occur infinitely often.

Since $(\mathcal{D}, \mathcal{R})$ is dischargeable, there exists an ordinal variable α_R such that R progresses on α_R and any other repeat R' in π' preserves α_R. We can now apply Lemma 3 to get a tail π'' of π' such that α_R occurs in the partial order of each sequent in π''. Let π'' be the sequence $(N_n, \mathsf{E}_n)(N_{n+1}, \mathsf{E}_{n+1}) \dots$

We consider the values of α_R in the interpretations E_i for $i \geq n$. If the step $(N_i, \mathsf{E}_i)(N_{i+1}, \mathsf{E}_{i+1})$ was chosen according to case 1 above then $\mathsf{E}_i(\alpha_R) = \mathsf{E}_{i+1}(\alpha_R)$. In case 2 was used, we have, by the fact that every repeat in π'' preserves α_R and Lemma 2, that $\mathsf{E}_i(\alpha_R) \geq \mathsf{E}_{i+1}(\alpha_R)$. If the repeat used in case 2 was R then $\mathsf{E}_i(\alpha_R) > \mathsf{E}_{i+1}(\alpha_R)$, due to progress. Since R occurs infinitely often, we get an infinite descending chain

$$\mathsf{E}_{i_0}(\alpha_R) > \mathsf{E}_{i_1}(\alpha_R) > \mathsf{E}_{i_2}(\alpha_R) > \dots$$

for appropriate indices i_k. This gives a contradiction because the ordinals are well-founded, thus the assumption that $\Gamma \vdash_O \Delta$ is invalid must have been false. □

3.6 Rules for Context-Free Processes

In this section we provide rules dealing with transition judgements $p \xrightarrow{a} q$. It has been shown by Simpson [37] how such rules can be found for process algebras with an operational semantics specified in the GSOS format. The GSOS format, SOS stands for structural operational semantics [35], is very general, allowing to express most process algebras. In this dissertation we instantiate the rules of [37] to the special case of context-free processes.

Let (V, Act, P, p_1) be a context-free process. The process-rules are given by:

$$(\varepsilon\mathrm{L})\ \frac{\Gamma, \varepsilon \xrightarrow{a} x \vdash_O \Delta}{}$$

$$(p\mathrm{L})\ \frac{\Gamma, pq \xrightarrow{a} x \vdash_O \Delta}{\{\Gamma[\overline{p}q/x] \vdash_O \Delta[\overline{p}q/x]\}_{\{\overline{p} \mid (p,a,\overline{p}) \in P\}}}$$

$$(p\mathrm{R})\ \frac{\Gamma \vdash_O pq \xrightarrow{a} \overline{p}q, \Delta}{} \text{ if } (p,a,\overline{p}) \in P$$

In this definition we are slightly sloppy with the use of p and $\overline{p}$. Note that p and $\overline{p}$ are process terms, whereas $(p,a,\overline{p}) \in P$ requires them to be elements of V and V^* respectively. However, the translation from process term to context-free processes and vice versa should be clear from the definition.

Note that it is very simple to extend the process algebra. One could, for example, define a parallel operator $|$ on processes. The rules for this operator reduce the transition judgements for terms of the form $p|q$ to judgements that can be dealt with using the rules above:

$$\frac{\Gamma \vdash_O p|q \xrightarrow{a} p'|q, \Delta}{\Gamma \vdash_O p \xrightarrow{a} p', \Delta} \qquad \frac{\Gamma \vdash_O p|q \xrightarrow{a} p|q', \Delta}{\Gamma \vdash_O q \xrightarrow{a} q', \Delta}$$

$$\frac{\Gamma,\ p|q \xrightarrow{a} r \vdash_O \Delta}{\Gamma[p'|q\ /\ r],\ p \xrightarrow{a} p' \vdash_O \Delta[p'|q\ /\ r] \qquad \Gamma[p|q'\ /\ r],\ q \xrightarrow{a} q' \vdash_O \Delta[p|q'\ /\ r]}$$

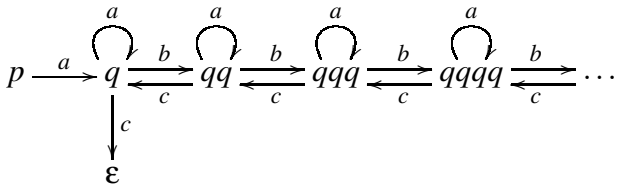
These rules are only given to illustrate the generality of the approach, we will not use the parallel operator in this report.

3.7 An Example Proof

This section provides an example for a derivation in the sequent calculus. The example is chosen to illustrate the use of ordinal quantification. We consider the process p given by the productions

$$p \xrightarrow{a} q, \quad q \xrightarrow{a} q, \quad q \xrightarrow{b} qq, \quad q \xrightarrow{c} \varepsilon.$$

This process has the following transition graph.



The property we will prove is $\nu X.[a]\nu Y.\langle b\rangle Y \wedge \langle c\rangle X$.

For the ease of presentation, we use the following two derivations as atomic rules:

$$(\nu\mathrm{R1}) \qquad \frac{\dfrac{\Gamma \vdash_O \Delta,\ p : \nu X.\varphi}{\Gamma \vdash_O \Delta,\ p : \forall\alpha.\nu^{\alpha}X.\varphi}\ (\nu\mathrm{R})}{\Gamma \vdash_{\mathcal{P}} \Delta,\ p : \nu^{\alpha}X.\varphi}\ (\forall\mathrm{R})$$

where $\mathcal{P} = (|O| \cup \{\alpha\}, <_O)$ and $\alpha \notin |O|$, and

$$(\nu^{\alpha}\mathrm{R1}) \qquad \frac{\dfrac{\Gamma \vdash_O \Delta,\ p : \nu^{\alpha}X.\varphi}{\Gamma \vdash_O \Delta,\ p : \forall(\beta < \alpha).\varphi\left[\nu^{\beta}X.\varphi/X\right]}\ (\nu^{\alpha}\mathrm{R})}{\Gamma \vdash_{\mathcal{P}} \Delta,\ p : \varphi\left[\nu^{\beta}X.\varphi/X\right]}\ (\forall_{<}\mathrm{R})$$

where $\mathcal{P} = (|O| \cup \{\beta\},\ <_O \cup \{(\beta,\alpha)\})$ and $\beta \notin |O|$.

The following abbreviations are be used in the derivation.

$$\begin{aligned}\varphi_\alpha &= \nu^\alpha X.[a]\nu Y.\langle b\rangle Y \wedge \langle c\rangle X\\ \psi_\alpha &= \nu Y.\langle b\rangle Y \wedge \langle c\rangle \varphi_\alpha\\ \psi_\alpha^\beta &= \nu^\beta Y.\langle b\rangle Y \wedge \langle c\rangle \varphi_\alpha\end{aligned}$$

The partial orders $O_1, \ldots, O_8$ used in the derivation are the smallest partial orders such that:

$$\begin{aligned}
&|O_1| = \{\alpha\},\\
&|O_2| = \{\alpha, \alpha'\}, && \alpha' <_{O_2} \alpha\\
&|O_3| = \{\alpha, \alpha', \beta\}, && \alpha' <_{O_3} \alpha\\
&|O_4| = \{\alpha, \alpha', \beta, \beta'\}, && \alpha' <_{O_4} \alpha,\ \beta' <_{O_4} \beta\\
&|O_5| = \{\alpha, \alpha', \alpha'', \beta, \beta'\}, && \alpha'' <_{O_5} \alpha',\ \alpha' <_{O_5} \alpha,\ \beta' <_{O_5} \beta\\
&|O_6| = \{\alpha, \alpha', \beta, \beta', \beta''\}, && \alpha' <_{O_6} \alpha,\ \beta'' <_{O_6} \beta',\ \beta' <_{O_6} \beta\\
&|O_7| = \{\alpha, \alpha', \alpha'', \beta, \beta', \beta''\}, && \alpha'' <_{O_7} \alpha',\ \alpha' <_{O_7} \alpha,\ \beta'' <_{O_7} \beta',\ \beta' <_{O_7} \beta\\
&|O_8| = \{\alpha, \alpha', \alpha'', \beta, \beta', \beta'', \gamma\}, && \alpha'' <_{O_8} \alpha',\ \alpha' <_{O_8} \alpha,\ \beta'' <_{O_8} \beta',\ \beta' <_{O_7} \beta
\end{aligned}$$

The derivation starts thus:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\vdash_\emptyset\ p : \nu X.[a]\nu Y.\langle b\rangle Y \wedge \langle c\rangle X}{\vdash_{O_1}\ p : \nu^\alpha X.[a]\nu Y.\langle b\rangle Y \wedge \langle c\rangle X}(\nu\mathrm{R1})}{\vdash_{O_2}\ p : [a]\psi_{\alpha'}}(\nu^\alpha\mathrm{R1})}(\Box\mathrm{R})}{p \xrightarrow{a} x \vdash_{O_2}\ x : \psi_{\alpha'}}(p\mathrm{L})}{\vdash_{O_2}\ q : \psi_{\alpha'}}(\nu\mathrm{R1})\quad\bullet}{\vdash_{O_3}\ q : \psi_{\alpha'}^\beta}(\nu^\alpha\mathrm{R1})}{\vdash_{O_4}\ q : \langle b\rangle\psi_{\alpha'}^{\beta'} \wedge \langle c\rangle\varphi_{\alpha'}}(\wedge\mathrm{R})$$

$$\dfrac{\dfrac{\vdash_{O_4}\ q : \langle b\rangle\psi_{\alpha'}^{\beta'}}{\vdash_{O_4}\ q \xrightarrow{b} qq \qquad \vdash_{O_4}\ qq : \psi_{\alpha'}^{\beta'}}(\Diamond\mathrm{R})}{(A)}$$

$$\dfrac{\dfrac{\vdash_{O_4}\ q : \langle c\rangle\varphi_{\alpha'}}{\vdash_{O_4}\ q \xrightarrow{c} \varepsilon \qquad \dfrac{\dfrac{\dfrac{\vdash_{O_4}\ \varepsilon : \varphi_{\alpha'}}{\vdash_{O_5}\ \varepsilon : [a]\psi_{\alpha''}}(\nu^\alpha\mathrm{R1})}{\varepsilon \xrightarrow{a} x \vdash_{O_5}\ x : \psi_{\alpha''}}(\Box\mathrm{R})}{}(\varepsilon\mathrm{L})}(\Diamond\mathrm{R})$$

For the sake of clarity, we will from now on contract the steps for ($\Box$) and ($\Diamond$). The derivations of the respective cases are as in the derivation above.
We continue the derivation at (A) with an application of the (TermCut) rule.

$$\dfrac{\dfrac{\dfrac{\vdash_{\mathcal{O}_5}\ q : [a]\psi_{\alpha''}}{\vdash_{\mathcal{O}_5}\ q : \psi_{\alpha''}}}{\vdash_{\mathcal{O}_4}\ q : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''}} \qquad \dfrac{\dfrac{x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_6}\ qx : \langle b\rangle\psi^{\beta''}_{\alpha'} \wedge \langle c\rangle\varphi^{\alpha'}}{x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_4}\ qx : \psi^{\beta'}_{\alpha'}}}{\vdash_{\mathcal{O}_4}\ qq : \psi^{\beta'}_{\alpha'}}\ \star}{(B)}$$

The derivation at (B) is

$$\dfrac{\dfrac{\dfrac{x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_6}\ qx : \langle b\rangle\psi^{\beta''}_{\alpha'}}{x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_6}\ qqx : \psi^{\beta''}_{\alpha'}}}{(C)} \qquad \dfrac{\dfrac{\dfrac{x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_6}\ qx : \langle c\rangle\varphi_{\alpha'}}{x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_6}\ \varepsilon x : \varphi_{\alpha'}}}{x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_7}\ \varepsilon x : [a]\psi_{\alpha''}}}{x : [a]\psi_{\alpha''} \vdash_{\mathcal{O}_7}\ \varepsilon x : [a]\psi_{\alpha''}}\ (\forall_<\mathrm{L})}{x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_6}\ qx : \langle b\rangle\psi^{\beta''}_{\alpha'} \wedge \langle c\rangle\varphi^{\alpha'}}$$

We conclude with the derivation (C), again by applying (TermCut) first.

$$\dfrac{\dfrac{\dfrac{\dfrac{x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_6}\ qx : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''}}{x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_7}\ qx : [a]\psi_{\alpha''}}}{x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_7}\ qx : \psi_{\alpha''}}}{x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_8}\ qx : \psi^{\gamma}_{\alpha''}} \qquad x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_6}\ qx : \psi^{\beta''}_{\alpha'}}{x : \forall(\alpha'' < \alpha').\,[a]\psi_{\alpha''} \vdash_{\mathcal{O}_6}\ qqx : \psi^{\beta''}_{\alpha'}}$$

The leaf in derivation (A) is a repeat of the node marked with $\bullet$. This repeat progesses on α'. Both leafs in derivation (C) are repeats of the node $\star$. The left leaf progresses on α', the right leaf progresses on β' and preserves α'. We now use a linear order of the repeats in which the repeat which progresses on β' occurs last. This must be a discharge order, since any linear order is a discharge order. Moreover it is dischargable. We have therefore established the property of p.

Note that in order to get progress on α' for the leaf in derivation (A), we have crucially used the quantifier $\forall(\alpha'' < \alpha')$ in the application of (TermCut). Had we introduced the formula $[a]\psi_{\alpha'}$, the repeat would not have progressed on any ordinal variable. The same situation applies in the left leaf of derivation (C).

Chapter 4

A Tableau for Context-Free Processes

4.1 Synopsis

In this chapter we present a tableau for context-free processes and the full modal μ-calculus.

The main obstacle in giving such a tableau is that context-free processes are in general infinite state processes. We therefore need to devise a strategy for decomposing process terms. We use the idea of a property transformer of [7] to systematically decompose process terms. We will decompose a process of the form pq into the two processes px and q, where x is a process variable. In order to show that pq has the property φ it is enough to find a property ψ which holds for q, and to show that whenever a process variable x satisfies ψ that then px has the property φ. Since a process pqx can thus be decomposed into py and qx, we can therefore restrict our attention to processes of the form rx for some nonterminal r.

The tableau is based on sequents of the form $\mathsf{Z} \;;\; \Phi \Vdash p : \varphi$, which express that the process px enjoys the property φ under the assumption that x satisfies all formulae in the set Φ. In $\Phi \cup \{\varphi\}$ we allow formulas with free variables, which are bound by definitions in the list Z. The decomposition is modelled by a cut rule of the form

$$(\text{Cut})\ \frac{\mathsf{Z} \;;\; \Phi \Vdash pq : \varphi}{\{\mathsf{Z} \;;\; \Phi \Vdash q : \psi\}_{\psi \in \Psi} \qquad \mathsf{Z} \;;\; \Psi \Vdash p : \varphi}$$

Note the similarity to the rule of sequential composition in Hoare logic.

In order to deal with the full modal μ-calculus, we extend the formulae by modifiers $(X = \sigma X.\varphi X.)$ and $(+X)$. Whereas the first modifier is merely a syntactic extension to the μ-calculus , similar to a let expression, the latter is a real extension. The main reason for introducing these modifiers is to deal with effects caused by the cut rule.

The tableau is inspired by a tableau system by Hungar and Steffen [24], which decides the alternation-free μ-calculus for context-free processes. The sequent $\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi$, would be written $p \vdash \langle \varphi, \Phi \rangle$ in their system. Since only addressing the alternation-free μ-calculus, they do not need to use open formulae or modifiers in the formulae.

4.2 The Tableau

4.2.1 The Sequents

The sequents in the tableau have the form

$$\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi,$$

where Z is a definition-list of the free variables of the formulae in $\Phi \cup \{\varphi\}$, Φ is a set of formulae, p is a context-free process, and φ is a formula. The list Z consists of definitions of the form $X = \sigma X.\varphi_X$, which are used to record the definition of the free variables of φ and Φ.

Definition 12 (Tableau formula). The set of *tableau formulae* is the smallest set such that any μ-calculus formula is a tableau formula, and if φ is a tableau formula then $(+X).\varphi$ and $(X = \sigma X.\varphi_X).\varphi$ are tableau formulae.

We extend the notion of free variables from formulae in the μ-calculus to tableau-formulae:

$$\begin{aligned} FV((+X).\varphi) &= \{X\} \cup FV(\varphi) \\ FV((X = \sigma X.\varphi_X).\varphi) &= FV(\sigma X.\varphi_X) \cup (FV(\varphi) \setminus \{X\}) \end{aligned}$$

We will refer to both $(+X)$ and $(X = \sigma X.\varphi_X)$ as *modifiers*. The latter will be abbreviated (X) when its definition is clear from the context. The modifier $(X = \sigma X.\varphi_X)$

is used to assign a definition to the free occurrences of X. An analogous syntax has been used in logical proof systems, e.g. by Hasegawa [21] and Hurkens et al. [25]. Hurkens et al. write t' where $\{x = t\}$, and Hasegawa uses expressions of the form letrec $X = t$ in t'.

Note that any tableau formula is a μ-calculus formula with a prefix of modifiers. We will say that X is *prefixed* in φ if φ contains a modifier $(+X)$ or (X) in its prefix.

In order to start a derivation for $p : \varphi$, we will represent the process p by $p\varepsilon$ and decompose the problem of showing $p\varepsilon : \varphi$ into the problem of finding a Φ which holds for ε and which is a sufficient assumption on x to establish that px has the property φ. It is therefore necessary to prove the formulae $\psi \in \Phi$ for the deadlocked process ε. This will be done by a second kind of sequent

$$\mathsf{Z} \Vdash_\varepsilon \psi.$$

From now on, we will only consider well-formed sequents.

Definition 13 (Well-Formed Sequent). Consider a sequent $\mathsf{Z} \Vdash_\varepsilon \varphi$ or $\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi$, where

$$\mathsf{Z}\ \text{ is }\ X_1 = \sigma_1 X_1.\varphi_1,\ \ldots,\ X_n = \sigma_n X_n.\varphi_n.$$

The sequent is *well-formed*, if for any $\psi \in \Phi \cup \{\varphi\}$ the formula

$$(X_1 = \sigma_1 X_1.\varphi_1) \ldots (X_n = \sigma_n X_n.\varphi_n).\, \psi$$

is closed.

4.2.2 Tableau Rules

All the rules assume that premise and conclusion are well-formed sequents. With the exception of the rule (Cut), this assumption is automatically satisfied for the premises if the conclusion is well-formed.

Start rule:

$$(\text{Start})\ \frac{p : \varphi}{\{ \Vdash_\varepsilon \psi \}_{\psi \in \Psi} \qquad ;\ \Psi \Vdash p : \varphi}$$

Epsilon rules:

$$(\varepsilon\text{-Axiom})\ \frac{\mathsf{Z} \Vdash_\varepsilon [a]\varphi}{}$$

$$(\varepsilon\text{-}\wedge)\ \frac{\mathsf{Z} \Vdash_\varepsilon \varphi_1 \wedge \varphi_2}{\mathsf{Z} \Vdash_\varepsilon \varphi_1 \qquad \mathsf{Z} \Vdash_\varepsilon \varphi_2} \qquad (\varepsilon\text{-}\vee 1)\ \frac{\mathsf{Z} \Vdash_\varepsilon \varphi_1 \vee \varphi_2}{\mathsf{Z} \Vdash_\varepsilon \varphi_1} \qquad (\varepsilon\text{-}\vee 2)\ \frac{\mathsf{Z} \Vdash_\varepsilon \varphi_1 \vee \varphi_2}{\mathsf{Z} \Vdash_\varepsilon \varphi_2}$$

$$(\varepsilon\text{-}\sigma)\ \frac{\mathsf{Z} \Vdash_\varepsilon \sigma X.\varphi_X}{\mathsf{Z} \Vdash_\varepsilon (X = \sigma X.\varphi_X).X} \qquad (\varepsilon\text{-Un})\ \frac{\mathsf{Z} \Vdash_\varepsilon X}{\mathsf{Z} \Vdash_\varepsilon \varphi_X}\ X = \sigma X.\varphi_X \text{ is in } \mathsf{Z}$$

$$(\varepsilon\text{-}(X))\ \frac{\mathsf{Z} \Vdash_\varepsilon (X = \sigma X.\varphi_X).\varphi}{\mathsf{Z}, X = \sigma X.\varphi_X \Vdash_\varepsilon \varphi}\ X \text{ not in } \mathsf{Z}$$

$$(\varepsilon\text{-Z-L})\ \frac{\mathsf{Z}, X = \sigma X.\varphi_X \Vdash_\varepsilon \varphi}{\mathsf{Z} \Vdash_\varepsilon \varphi}\ X \text{ not free in } \varphi$$

Axiom:

$$(\text{Axiom})\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash \varepsilon : \varphi}{}\ \varphi \in \Phi$$

Structural Rules:

$$(\text{Weak})\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi}{\mathsf{Z}\ ;\ \Psi \Vdash p : \varphi}\ \Psi \subseteq \Phi$$

Logical rules:

$$(\wedge)\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi_1 \wedge \varphi_2}{\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi_1 \qquad \mathsf{Z}\ ;\ \Phi \Vdash p : \varphi_2}$$

$$(\vee 1)\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi_1 \vee \varphi_2}{\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi_1} \qquad (\vee 2)\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi_1 \vee \varphi_2}{\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi_2}$$

Modal rules:

$$(\Box)\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : [a]\varphi}{\{\mathsf{Z}\ ;\ \Phi \Vdash q : \varphi\}_{q \in \{q \mid p \xrightarrow{a} q\}}} \qquad (\Diamond)\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : \langle a\rangle\varphi}{\mathsf{Z}\ ;\ \Phi \Vdash q : \varphi}\ p \xrightarrow{a} q$$

Fixed point rules:

$$(\mu)\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : \mu X.\varphi_X}{\mathsf{Z}\ ;\ \Phi \Vdash p : (X = \mu X.\varphi_X).X} \qquad (\nu)\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : \nu X.\varphi_X}{\mathsf{Z}\ ;\ \Phi \Vdash p : (X = \nu X.\varphi_X).X}$$

$$(\text{Un-}\mu)\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : X}{\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi_X}\ X = \mu X.\varphi_X \text{ is in } \mathsf{Z}$$

$$(\text{Un-}\nu)\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : X}{\mathsf{Z}\ ;\ \Phi \Vdash p : (+X).\varphi_X}\ X = \nu X.\varphi_X \text{ is in } \mathsf{Z}$$

Quantifier rules:

$$((X))\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : (X = \sigma X.\varphi_X).\varphi}{\mathsf{Z}, X = \sigma X.\varphi_X\ ;\ \Phi \Vdash p : \varphi}\ X \text{ not in } \mathsf{Z}$$

$$((+X))\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : (+X).\varphi}{\mathsf{Z}\ ;\ \Phi \cup \{\rho.\psi \mid \rho.(+X).\psi \in \Phi\} \Vdash p : \varphi}\ X \text{ is prefixed in any } \psi \in \Phi$$

$$(\text{Z-L})\ \frac{\mathsf{Z}, X = \sigma X.\varphi_X\ ;\ \Phi \Vdash p : \varphi}{\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi}\ X \text{ not free in } \varphi \text{ or } \Phi$$

$$((X)\text{-L1})\ \frac{\mathsf{Z}\ ;\ \Phi, \rho.(X = \sigma X.\varphi_X).\psi \Vdash p : \varphi}{\mathsf{Z}\ ;\ \Phi, \rho.\psi \Vdash p : \varphi}\ X = \sigma X.\varphi_X \text{ is in } \mathsf{Z}$$

$$((X)\text{-L2})\ \frac{\mathsf{Z}\ ;\ \Phi, \rho.(X = \sigma X.\varphi_X).\psi \Vdash p : \varphi}{\mathsf{Z}\ ;\ \Phi, \rho.(+X).\psi \Vdash p : \varphi}\ X = \sigma X.\varphi_X \text{ is in } \mathsf{Z}$$

Cut:

$$(\text{Cut})\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash pq : \varphi}{\{\mathsf{Z}\ ;\ \Phi \Vdash q : \psi\}_{\psi \in \Psi} \qquad \mathsf{Z}\ ;\ \Psi \Vdash p : \varphi}\ \text{any free } X \text{ in } \psi \in \Psi \text{ is in } \mathsf{Z}$$

In the rules $((+X))$, $((X)\text{-L1})$ and $((X)\text{-L2})$, we use the notation $\rho.(X).\psi$. In this notation, ρ always stands for a sequence of modifiers and ψ is an ordinary tableau formula. Intuitively, in these rules, one can thereby 'change an inner modifier'.

The epsilon rules are more complicated than necessary - they were chosen to allow a uniform treatment of all sequents.

The meaning of the rules will become clear in the translation to the sequent calculus in the soundness section.

A tableau is a proof tree whose root is labelled with a sequent of the form $p : \varphi$ for closed φ. A leaf in the tableau is a leaf node which is not closed by an axiom.

Definition 14 (Repeat). Let N be a leaf node in a tableau and leaf M be a different node such that there is a path from M to N. The node M is called a repeat of N if

- N and M are labelled with sequents $\mathsf{Z}' \,;\, \Phi' \Vdash p : \varphi$ and $\mathsf{Z} \,;\, \Phi \Vdash p : \varphi$ respectively, and Z is a prefix of Z' and $\Phi \subseteq \Phi'$.
- N and M are labelled with sequents $\mathsf{Z}' \Vdash_\varepsilon \varphi$ and $\mathsf{Z} \Vdash_\varepsilon \varphi$ respectively, and Z is a prefix of Z'.

Definition 15 (Progress, Preservation). A path *preserves* X if X occurs in the list of free fixed point variables in every sequent along the path. A path *progresses* on X if $((+X))$ is applied and X is preserved by the path.

Definition 16 (Successful Leaf/Derivation). A leaf N in the tableau labelled with the sequent $\mathsf{Z} \,;\, \Phi \Vdash p : \varphi$ is *successful*, if N has a repeat M such that there is a fixed point variable X, on which the path from M to N progresses.

A derivation in the tableau is *successful*, if each of its leaves is successful.

Definition 17. For closed φ, the judgement $p : \varphi$ is *derivable* if there is a successful tableau with the root sequent $p : \varphi$.

4.3 An Example Derivation

Consider the process P, defined by $P \xrightarrow{a} PP$ and $P \xrightarrow{b} \varepsilon$. We prove that P has the property $\nu X.\mu Y.[a]X \wedge [b]Y$. This formula expresses that in any sequence of a, b-actions containing infinitely many b-actions there must be infinitely many a-actions.

We use (X) and (Y) to abbreviate the modifiers $(X = \nu X.\mu Y.[a]X \wedge [b]Y)$ and $(Y = \mu Y.[a]X \wedge [b]Y)$ respectively. In the definition-list of free variables, we record only the defined variable but not its definition.

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{}{\Vdash_{\varepsilon} (X).(Y).Y}}{\vdots} \qquad \dfrac{P : \nu X.\mu Y.[a]X \wedge [b]Y}{; (X).(Y).Y \Vdash P : \nu X.\mu Y.[a]X \wedge [b]Y}\,(\nu)}{; (X).(Y).Y \Vdash P : (X).X}\,((X))}{X \;;\; (X).(Y).Y \Vdash P : X}\,(\text{Un-}\nu)}{X \;;\; (X).(Y).Y \Vdash P : (+X).\mu Y.[a]X \wedge [b]Y}\,((+X))}{X \;;\; (X).(Y).Y \Vdash P : \mu Y.[a]X \wedge [b]Y}\,((X)\text{-L1/2})}{X \;;\; (+X).(Y).Y,\ (Y).Y \Vdash P : \mu Y.[a]X \wedge [b]Y}\,(\mu)\ \star}{X \;;\; (+X).(Y).Y,\ (Y).Y \Vdash P : (Y).Y}\,((Y))\ \bullet}{X,Y \;;\; (+X).(Y).Y,\ (Y).Y \Vdash P : Y}\,(\text{Un-}\mu)}{X,Y \;;\; (+X).(Y).Y,\ (Y).Y \Vdash P : [a]X \wedge [b]Y}$$

$$(A)$$

We continue with the right-hand branch (A).

$$\dfrac{\dfrac{\dfrac{X,Y \;;\; (+X).(Y).Y,\ (Y).Y \Vdash P : [a]X \wedge [b]Y}{X,Y \;;\; (+X).(Y).Y,\ (Y).Y \Vdash P : [a]X}\,(\wedge)}{X,Y \;;\; (+X).(Y).Y,\ (Y).Y \Vdash PP : X}\,(\Box) \qquad \dfrac{X,Y \;;\; (+X).(Y).Y,\ (Y).Y \Vdash P : [b]Y}{X,Y \;;\; (Y).Y \Vdash P : [b]Y}\,(\text{Weak})}{(B)}\,(\Box)$$

$$\dfrac{\dfrac{X,Y \;;\; (Y).Y \Vdash \varepsilon : Y}{X,Y \;;\; Y \Vdash \varepsilon : Y}\,((Y)\text{-L1})}{}\,(\text{Axiom})$$

Consider the left branch (B) of this derivation:

$$\dfrac{\dfrac{\dfrac{X,Y \;;\; (+X).(Y).Y,\ (Y).Y \Vdash PP : X}{X,Y \;;\; (+X).(Y).Y,\ (Y).Y \Vdash P : (+X).(Y).Y}\,((+X))}{X,Y \;;\; (+X).(Y).Y,\ (Y).Y \Vdash P : (Y).Y}\,(\text{Z-L})}{X \;;\; (+X).(Y).Y,\ (Y).Y \Vdash P : (Y).Y}\,(\text{Cut}) \qquad (C)$$

The sequent at this leaf is a repeat of the node at $\bullet$, X occurs in the list of free variable all along the path, and $(+X)$ is applied along the path. Thus this leaf is successful.

Finally, the derivation at (C) is

$$\dfrac{\dfrac{\dfrac{X,Y\ ;\ (+X).(Y).Y \Vdash P : X}{X,Y\ ;\ (+X).(Y).Y \Vdash P : (+X).\mu Y.[a]X \wedge [b]Y}\ \text{(Un-}\nu\text{)}}{X,Y\ ;\ (+X).(Y).Y,\ (Y).Y \Vdash P : \mu Y.[a]X \wedge [b]Y}\ ((+X))}{X\ ;\ (+X).(Y).Y,\ (Y).Y \Vdash P : \mu Y.[a]X \wedge [b]Y}\ \text{(Z-L)}$$

This leaf is a successful repeat of $\star$. We have therefore established the property of P.

4.4 Soundness

The soundness of the tableau system will be proved by reducing it to the sequent calculus from the previous chapter. One benefit of this reduction is that a completeness proof for the tableau will also establish completeness for the sequent calculus.

We will now encode the tableau in the sequent calculus. Tableau-sequents of the form $\mathsf{Z} \Vdash_\varepsilon \varphi$ are translated to

$$\vdash_O\ \varepsilon : e(\varphi, \mathsf{Z}, s),$$

and sequents of the form $\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi$ are translated to

$$\{x : e(\psi, \mathsf{Z}, s)\}_{\psi\in\Phi}\ \vdash_O\ px : e(\varphi, \mathsf{Z}, s).$$

where O and e are explained below. In both cases, s is a function mapping each *greatest* fixed point variable in Z to an ordinal variable in $|O|$. The function e uses this mapping to translate the open formulae that are encountered in the tableau to formulae in the sequent calculus. Note that, since the tableau sequent is well-formed, we know that all free fixed point variables of φ and Φ occur in Z and hence all occurrences of $s(X)$ in the following definition are well-defined. In the following definition, we assume $\alpha \notin |O|$ and that each α can only be used once.

$$e(\varphi, \mathsf{Z}, s) = \begin{cases} \forall\alpha.\, e(\varphi_1, (\mathsf{Z}, X = \sigma X.\varphi_X), s[\alpha / X]) & \text{if } \varphi \text{ is } (X = \sigma X.\varphi_X).\varphi_1, \\ \forall(\alpha < s(X)).\, e(\varphi_1, \mathsf{Z}, s[\alpha/X]) & \text{if } \varphi \text{ is } (+X).\varphi_1, \\ \varphi[\sigma'_n X_n.\varphi_n / X_n] \ldots [\sigma'_1 X_1.\varphi_n / X_1] & \text{otherwise.} \end{cases}$$

where Z is $X_1 = \sigma_1 X_1.\varphi_1, \ldots, \sigma_n X_n.\varphi_n$ and, for all $i = 1, \ldots, n$,

$$\sigma_i' = \begin{cases} \mu & \text{if } \sigma_i \text{ is } \mu, \\ \nu^\alpha \text{ where } \alpha \text{ is } s(X_i) & \text{if } \sigma_i \text{ is } \nu. \end{cases}$$

Note that, because of the order of the substitution and because the sequents are well-formed, the result of this translation has no free propositional variables. A function e similar to this one is described in [28].

We will now use this translation to map a tableau to a derivation in the sequent calculus. It will be shown next that each tableau rule can be translated to a derivation in the sequent calculus. We start with a technical lemma.

Lemma 4. *Let φ be a formula in the sequent calculus, let O be a partial order of ordinal variables, and let $Q_1, \ldots, Q_n$ be quantifiers such that Q_i is of the form $\forall\alpha_i$ or $\forall(\alpha_i < \beta_i)$ for some $\beta_i \in |O|$. Using Q to abbreviate $Q_1 \ldots Q_{n-1}$, the following are derivable in the sequent calculus:*

$$\frac{\Gamma,\ q : Q.\forall\alpha.\varphi \vdash_O \Delta}{\Gamma,\ q : Q.\varphi \vdash_O \Delta} \tag{4.1}$$

$$\frac{\Gamma,\ q : Q.\forall\alpha.\varphi \vdash_O \Delta}{\Gamma,\ q : Q.\forall(\alpha < \gamma).\varphi \vdash_O \Delta.}\ \gamma \in |O| \tag{4.2}$$

$$\frac{\Gamma,\ q : Q.\forall(\alpha < \beta).\varphi \vdash_O \Delta}{\Gamma,\ q : Q.\varphi \vdash_O \Delta.}\ \beta \in |O| \text{ and } \alpha <_O \beta \tag{4.3}$$

$$\frac{\Gamma,\ q : Q.\forall(\alpha < \beta).\varphi \vdash_O \Delta}{\Gamma,\ q : Q.\forall(\alpha < \gamma).\varphi \vdash_O \Delta}\ \beta \in |O|, \gamma \in |O| \text{ and } \gamma <_O \beta \tag{4.4}$$

Proof. We consider 4.4, the other cases are similar. Using (Cut),(WeakL) and (WeakR) we get the following derivation

$$\dfrac{\dfrac{\Gamma,\ q : Q.\forall(\alpha < \beta).\ \varphi \vdash_O \Delta}{q : Q.\forall(\alpha < \beta).\varphi \vdash_O q : Q.\forall(\alpha < \gamma).\varphi} \qquad \Gamma,\ q : Q.\forall(\alpha < \gamma).\varphi \vdash_O \Delta}{(A)}$$

We continue the derivation at (A) by eliminating the quantifiers in Q. According to the outmost quantifier, we can apply ($\forall$R) and ($\forall$L) or ($\forall_<$R) and ($\forall_<$L) in this order to remove the outmost quantifier on both sides of the sequent. We arrive at a sequent of the form,

$$q : \forall(\alpha < \beta).\varphi' \vdash_{O'} q : \forall(\alpha < \gamma).\varphi',$$

we continue the derivation using ($\forall_<$R) and ($\forall_<$L):

$$\dfrac{\dfrac{q : \forall(\alpha < \beta).\varphi' \vdash_{O'} q : \forall(\alpha < \gamma).\varphi'}{\dfrac{q : \forall(\alpha < \beta).\varphi' \vdash_{O''} q : \varphi'}{q : \varphi' \vdash_{O''} q : \varphi'}\ \text{if } \alpha <_{O''} \beta}\ \text{where } <_{O''} \text{ is } <_{O'} \text{ with } \alpha <_{O''} \gamma$$

In order to justify the application of the second rule ($\forall_<$L) we need to show $\alpha <_{O''} \beta$. The first rule gives us $\alpha <_{O''} \gamma$. Together with the assumption $\gamma <_O \beta$ and the fact that $O \subseteq O''$, we therefore have the required $\alpha <_{O''} \beta$. Since we arrive at an axiom, our derivation is thus finished. □

We are now in the position to show that the tableau-rules can be encoded in the sequent calculus.

Lemma 5. *For every tableau-rule of the form*

$$\dfrac{\mathsf{Z} \Vdash_\varepsilon \varphi}{\mathsf{Z}' \Vdash_\varepsilon \psi \qquad \ldots}$$

there exists a derivation in the sequent calculus

$$\dfrac{\vdash_O \varepsilon : e(\varphi, \mathsf{Z}, s)}{\vdash_{\mathcal{P}} \varepsilon : e(\psi, \mathsf{Z}', t) \qquad \ldots}$$

For every tableau-rule of the form

$$\dfrac{\mathsf{Z}\ ;\ \Phi \Vdash p : \varphi}{\mathsf{Z}'\ ;\ \Psi \Vdash p : \psi \qquad \ldots}$$

there exists a derivation in the sequent calculus

$$\dfrac{\{x : e(\psi, \mathsf{Z}, s)\}_{\psi \in \Phi} \vdash_O px : e(\varphi, \mathsf{Z}, s)}{\{x : e(\psi, \mathsf{Z}', t)\}_{\psi \in \Psi} \vdash_{\mathcal{P}} qx : e(\psi, \mathsf{Z}', t) \qquad \ldots}$$

Moreover, the following assertions hold for all X and Y on which s and t are defined.

1. *If the tableau-rule is not* (X) *then* $t(X) \leq_{\mathcal{P}} s(X)$*, and if the tableau-rule is* $(+X)$ *then* $t(X) <_{\mathcal{P}} s(X)$.

2. *If* $\alpha \not<_O s(X)$ *for all* $\alpha \in |O|$ *then* $\beta \not<_{\mathcal{P}} t(X)$ *for all* $\beta \in |\mathcal{P}|$.

3. *If* $s(X) \neq s(Y)$ *for all* $X \neq Y$ *then* $t(X) \neq t(Y)$ *for all* $X \neq Y$.

Proof. We omit the translation of epsilon rules, since these cases are very similar to the following.

- (Axiom) is translated to an axiom in the sequent calculus.
- (Weak). This rule can be derived using a repeated application of (WeakL).
- $(\wedge)$. In this case, we note that $e(\varphi_1 \wedge \varphi_2, Z, s)$ is $e(\varphi_1, Z, s) \wedge e(\varphi_2, Z, s)$. We apply $(\wedge R)$ in the sequent calculus to get the corresponding derivation.
- $(\vee 1)$, $(\vee 2)$ The translations for these rules can be obtained directly by and application of $(\vee R)$,(WeakR) and the respective process rules.
- $(\Box)$. We note that $e([a]\varphi, Z, s)$ is $[a](e(\varphi, Z, s))$. The derivation is

$$\dfrac{\dfrac{\Gamma \vdash_O px : [a](e(\varphi, Z, s))}{\Gamma, px \xrightarrow{a} y \vdash_O y : e(\varphi, Z, s)}\,(\Box R)}{\{\Gamma \vdash_O qx : e(\varphi, Z, s)\}_{\{q \mid p \xrightarrow{a} q\}}}\,(pL)$$

  where y is a fresh variable and Γ is used to stand for $\{x : e(\psi, Z, s)\}_{\psi \in \Phi}$.

- $(\Diamond)$. Let $p \xrightarrow{a} q$. We note that $e(\langle a \rangle \varphi, Z, s)$ is $\langle a \rangle (e(\varphi, Z, s))$. We have the following derivation

$$\dfrac{\dfrac{}{\Gamma \vdash_O px \xrightarrow{a} qx}\,(pR) \qquad \Gamma \vdash_O px : \langle a \rangle (e(\varphi, Z, s))}{\Gamma \vdash_O qx : e(\varphi, Z, s)}\,(\Diamond R)$$

  where Γ is used to stand for $\{x : e(\psi, Z, s)\}_{\psi \in \Phi}$.

- (ν) This case is dealt with by an application of the (νR) rule. Let Z_1 be the longest prefix on Z which does not contain a definition of X. We note that $e(\nu X.\varphi_X, \mathsf{Z}, s) \equiv \nu X.e(\varphi_X, \mathsf{Z}_1, s)$. The derivation is then

$$(\nu\mathrm{R})\ \frac{\Gamma \vdash_O px : \nu X.e(\varphi_X, \mathsf{Z}_1, s)}{\Gamma \vdash_O px : \forall\alpha.\nu^\alpha X.e(\varphi_X, \mathsf{Z}_1, s)}$$

  By definition of e, we have that

$$\begin{aligned}\forall\alpha.\nu^\alpha X.e(\varphi_X, \mathsf{Z}_1, s) &\equiv \forall\alpha.e(X, (\mathsf{Z}_1, X = \nu X.\varphi_X), s[\alpha/X]) \\ &\equiv e((X = \nu X.\varphi_X).X, \mathsf{Z}_1, s),\end{aligned}$$

  which completes this case.

- (μ) This case is similar to (ν).

- (Un-μ). Let Z be $\mathsf{Z}_1, X = \mu X.\varphi_X, \mathsf{Z}_2$. By the side-condition to the rule, Z must contain an occurrence of X. We first note that $e(X, \mathsf{Z}, s) = e(\mu X.\varphi_X, \mathsf{Z}_1, s) = \mu X.e(\varphi_X, \mathsf{Z}_1, s)$. Using ψ as an abbreviation for $e(\varphi_X, \mathsf{Z}_1, s)$, we can apply the ($\mu$R) rule in the sequent calculus to derive

$$\frac{\{x : e(\psi, \mathsf{Z}, s)\}_{\psi\in\Phi} \vdash_O px : \mu X.\psi}{\{x : e(\psi, \mathsf{Z}, s)\}_{\psi\in\Phi} \vdash_O px : \psi[\mu X.\psi/X]}\ (\mu\mathrm{R})$$

  Now, we have

$$\begin{aligned}\psi[\mu X.\psi/X] &\equiv e(\varphi_X, \mathsf{Z}_1, s)[\mu X.e(\varphi_X, \mathsf{Z}_1, s)/X] \\ &\equiv e(\varphi_X, (\mathsf{Z}_1, X = \mu X.\varphi_X), s) \\ &\equiv e(\varphi_X, \mathsf{Z}, s)\end{aligned}$$

- (Un-ν). Let Z be $\mathsf{Z}_1, X = \nu X.\varphi_X, \mathsf{Z}_2$. As in the previous case, we have $e(X, \mathsf{Z}, s) \equiv e(\nu^{s(X)} X.\varphi_X, \mathsf{Z}_1, s) \equiv \nu^{s(X)} X.e(\varphi_X, \mathsf{Z}_1, s)$. Using ψ as an abbreviation for $e(\varphi_X, \mathsf{Z}_1, s)$, the ($\nu^\kappa$R) rule can be applied to derive

$$\frac{\{x : e(\psi, \mathsf{Z}, s)\}_{\psi\in\Phi} \vdash_O px : \nu^{s(X)} X.\psi}{\{x : e(\psi, \mathsf{Z}, s)\}_{\psi\in\Phi} \vdash_O px : \forall(\alpha < s(X)).\psi[\nu^\alpha X.\psi/X]}\ (\nu^\kappa\mathrm{R})$$

  Expanding ψ gives $\forall(\alpha < s(X)).e(\varphi_X, \mathsf{Z}_1, s)[\nu^\alpha X.e(\varphi_X, \mathsf{Z}_1, s)/X]$, which is identical to $\forall(\alpha < s(X)).e(\varphi_X, (\mathsf{Z}_1, X = \nu X.\varphi_X), s[\alpha/X])$, and this is $e((+X).\varphi_X, \mathsf{Z}, s)$.

- $((X))$. By definition, $e((X).\varphi, \mathsf{Z}, s) = \forall\alpha.e(\varphi, \mathsf{Z}, s[\alpha/X])$ for some $\alpha \notin |O|$. We have
$$\frac{\{x : e(\psi, \mathsf{Z}, s)\}_{\psi\in\Phi} \vdash_O px : \forall\alpha.e(\varphi, \mathsf{Z}, s[\alpha/X])}{\{x : e(\psi, \mathsf{Z}, s[\alpha/X])\}_{\psi\in\Phi} \vdash_{\mathcal{P}} px : e(\varphi, \mathsf{Z}, s[\alpha/X])}\ (\forall\text{R}),$$
where $\mathcal{P} = (|O| \cup \{\alpha\}, <_O)$. Note that X does not occur in Z, and therefore $e(\varphi, \mathsf{Z}, s) = e(\varphi, \mathsf{Z}, s[\alpha/X])$. This allows us to replace s by $s[\alpha/X]$ on the left hand side.

- $((+X))$. By definition, $e((+X).\varphi, \mathsf{Z}, s) = \forall(\alpha < s(X)).e(\varphi, \mathsf{Z}, s[\alpha/X])$ for some $\alpha \notin |O|$. We therefore apply the $(\forall <\text{R})$ rule first:
$$\frac{\{x : e(\psi, \mathsf{Z}, s)\}_{\psi\in\Phi} \vdash_O px : \forall(\alpha < s(X)).e(\varphi, \mathsf{Z}, s[\alpha/X])}{\{x : e(\psi, \mathsf{Z}, s)\}_{\psi\in\Phi} \vdash_{\mathcal{P}} px : e(\varphi, \mathsf{Z}, s[\alpha/X])}\ (\forall <\text{R}),$$
where $\mathcal{P} = (|O| \cup \{\alpha\}, <_O \cup\{(\alpha, s(X)\})$. This gives us the appropriate right-hand side.

  We now consider the the formulae on the left-hand side. Note that, by assumption on the tableau-rule, X is prefixed in all formulae on the left side. For formulae prefixed with (X), we can safely replace s by $s[\alpha/X]$. Formulae prefixed with $(+X)$, are dealt with by the third and fourth case of Lemma 4.

- (Z-L) Note that the translation of the upper and the lower sequent are identical, so we do not need to apply any rule in the sequent calculus.

- $((X)$-L1). Apply the first derivation of Lemma 4.

- $((X)$-L2). Apply the second derivation of Lemma 4.

- (Cut). We use the sub-term-cut rule to split the process pqx into py and qx. Using Γ and φ' as abbreviations for $\{x : e(\psi, \mathsf{Z}, s)\}_{\psi\in\Phi}$ and $e(\varphi, \mathsf{Z}, s)$ respectively, the derivation is:
$$\frac{\Gamma \vdash_O pqx : \varphi'}{\dfrac{\Gamma \vdash_O qx : \bigwedge_{\psi\in\Psi} e(\psi, \mathsf{Z}, s)}{\{\Gamma \vdash_O qx : e(\psi, \mathsf{Z}, s)\}_{\psi\in\Psi}}\ (\wedge\text{R}) \qquad \dfrac{y : \bigwedge_{\psi\in\Psi} e(\psi, \mathsf{Z}, s) \vdash_O py : \varphi'}{\{y : e(\psi, \mathsf{Z}, s)\}_{\psi\in\Psi} \vdash_O py : \varphi'}\ (\vee\text{L})}\ (\text{TermCut})$$

- (Start). This is a similar (Cut) above, using the fact that any process p is identified with $p\varepsilon$. We use φ' to abbreviate $e(\varphi, Z_0, s)$, where Z_0 is the empty list and s is the everywhere undefined function.

$$\dfrac{\vdash_\emptyset\ p\varepsilon : \varphi'}{\dfrac{\vdash_\emptyset\ \varepsilon : \bigwedge_{\psi\in\Psi} e(\psi, Z_0, s)}{\{\vdash_\emptyset\ \varepsilon : e(\psi, Z_0, s)\}_{\psi\in\Psi}}\ (\wedge\mathrm{R}) \qquad \dfrac{x : \bigwedge_{\psi\in\Psi} e(\psi, Z_0, s)\ \vdash_\emptyset\ px : \varphi'}{\{x : e(\psi, Z_0, s)\}_{\psi\in\Psi}\ \vdash_\emptyset\ px : \varphi'}\ (\vee\mathrm{L})}\ (\mathrm{TermCut})$$

The cases above show that the only cases, where the partial order of ordinal variables changes are $((X))$ and $((+X))$. Let (O, s) and $(\mathcal{P}, t)$ the corresponding partial orders. We see that $t(X) <_\mathcal{P} s(X)$ if the rule is $((+X))$, and $t(X) \leq_\mathcal{P} s(X)$ if the rule is not $((X))$, hence property 1 follows. Property 2 and 3 follow from the fact that the ordinal variables introduced in these rules are fresh. □

We now use this Lemma to translate a successful tableau with root $p_0 : \varphi_0$ for closed φ_0 to a derivation in the sequent calculus. Using Lemma 5, we can build a derivation in the sequent calculus 'along the tableau'. The translation of the (Start) rule implies that root of the translated derivation has an empty partial order of ordinal variables. The function s used in this translation is undefined on all its arguments. For the root sequent of the translation we thus have $\alpha \not<_O s(X)$ for all X for which s is defined. By Lemma 5, we may therefore assume this property for any sequent in the translated derivation. Furthermore, we can assume $s(X) \neq s(Y)$ whenever $X \neq Y$.

Using this property we can now show that each repeat in the tableau is translated to a repeat in the sequent calculus. Consider a repeat of the form

$$\begin{array}{c} Z\ ;\ \Phi \Vdash p : \varphi \\ \vdots \\ Z'\ ;\ \Phi' \Vdash p : \varphi \end{array}$$

in the tableau derivation, i.e. Z is a prefix of Z' and $\Phi \subseteq \Phi'$, and consider the translation of this repeat:

$$\begin{array}{c} \{x : e(\psi, Z, s)\}_{\psi\in\Phi}\ \vdash_O\ px : e(\varphi, Z, s) \\ \vdots \\ \{x : e(\psi, Z', t)\}_{\psi\in\Phi'}\ \vdash_\mathcal{P}\ px : e(\varphi, Z', t) \end{array}$$

Since the tableau-sequents are well-formed, all free variables of Φ and φ occur in Z. Furthermore, $\mathrm{dom}(s) \subseteq \mathrm{dom}(t)$, as Z is a prefix of Z'. Hence, for any $\psi \in \Phi \cup \{\varphi\}$, $e(\psi, \mathsf{Z}, s)$ and $e(\psi, \mathsf{Z}', t)$ differ only in the ordinal variables.

In section 3.4.1 we have seen that in order to find a repeat it is enough just to consider the ordinal variables α for which $\alpha \leq_O \beta$ holds for some β occurring in the inner sequent. In the repeat we are considering here, the variables occurring in the sequent are of the form $s(X)$ and we have $\alpha \not<_O s(X)$ for all X where s is defined. We therefore know that the following functions induce a mapping σ which satisfies the conditions for a repeat: $\sigma_{\mathsf{V}} = id$, $\sigma_{\mathsf{P}} = id$ and

$$\sigma_{\mathsf{O}}(\alpha) = \begin{cases} t(X) & \text{if } \alpha \text{ is } s(X) \\ \text{undefined} & \text{otherwise.} \end{cases}$$

Note that σ_{O} is well-defined because $s(X) \neq s(Y)$ whenever $X \neq Y$. The case of a repeat of epsilon-sequents is omitted here, since it is very similar to this case.

It is now immediate from Lemma 5 that the translated repeat preserves (progresses on) $s(X)$ if the tableau-repeat preserves (progresses on) X.

As the tableau is successful, we can choose a repeat node for each leaf in the translated derivation, giving us a pre-proof. As in the previous chapter, we associate a tree with the tableau derivation and add edges from each repeat leaf to its companion node. By the form of the translation, each path in the pre-proof graph of the translated derivation has a corresponding path in the so constructed tableau-graph, and we can therefore restrict our attention to paths in the tableau.

We now prove that this construction gives indeed a proof in the sequent calculus.

Theorem 2 (Soundness). *If $p_0 : \varphi_0$ is tableau-derivable then there is a sequent-proof of $\vdash\ p_0 : \varphi_0$.*

Proof. We need to find a discharge order $\prec$ such that the pre-proof is dischargeable using this order. We will say that Z_R is the list of free variables for a repeat R if Z_R is a prefix of every node in the corresponding tableau-repeat.

Let $R = (M_R, N_R, \sigma_R)$ and $S = (M_S, N_S, \sigma_S)$ be repeats in the translated derivation and let Z_R and Z_S be the respective lists of free variables. Define $S \prec R$ if, and only if, Z_R is a prefix of Z_S.

We now show that $\prec$ is indeed a discharge order. We have to show that whenever $S \not\prec R$ and $R \not\prec S$ then any path of the form $M_R \ldots N_S M_S \ldots N_R$ contains a repeat T such that $R \prec T$ and $S \prec T$. Thus, suppose $S \not\prec R$ and $R \not\prec S$. If there is no path of the form $M_R \ldots N_S M_R \ldots N_R$ then we are done. Otherwise, we consider the paths $M_R \ldots N_S$ and $N_S \ldots N_R M_R$. We note that all nodes on the unique shortest path connecting M_R and N_S in the spanning tree of the pre-proof graph must occur in both paths. By the form of the rules, in each step along this unique shortest these path the list of free variables of the corresponding sequent can only change by adding or deleting a node at the end. Therefore, there the shortest path must contain a node K labelled with a sequent with a list Z of free variables which is a prefix of both Z_R and Z_S. Hence, the paths under consideration have the form $M_R \ldots K \ldots N_S$ and $N_S \ldots K \ldots N_R M_R$. Now, because N_S is a leaf node, the path $K \ldots N_S \ldots K$ must contain a repeat $T = (M_T, N_T, \sigma_T)$ such that the node K is between M_T and N_T in the tree, i.e. there is a directed path $M_T \ldots K \ldots N_T$ in the derivation tree. This means that the list of variables of the repeat T is a prefix of Z, and this in turn implies $R \prec T$ and $S \prec T$. We have thus shown that any path of the form $M_R \ldots N_S \ldots N_R$ contains a repeat T which is a larger than R and S with respect do $\prec$. Hence, $\prec$ is a discharge order.

Since the tableau is successful, a repeat R preserves all ordinal variables $s(X)$ for which X occurs in Z_R. Because $\prec$ is the prefix-relation, it follows that if $S \prec R$ then S preserves all $s(X)$ for which X occurs in Z_R. Furthermore, each repeat R progresses on one of the ordinal variables in its list Z_R variables. This argument shows that the pre-proof in the translated derivation is dischargeable, hence the derivation is indeed a proof. □

Chapter 5

Completeness

In this chapter we will prove the completeness of the tableau system with respect to the full modal μ-calculus. The completeness proof uses the property checking games introduced by Stirling [38] to construct a tableau-derivation. In the following we will briefly present these games.

5.1 Property Checking Games

Definition 18 (Normal Formula). A closed μ-calculus formula φ is *normal* if every occurrence of a binder μX or νX in φ binds a distinct variable.

Every μ-calculus formula can be converted into a normal formula by renaming of bound variables. In a formula φ in normal form we can use the variable X to uniquely identify the sub-formula $\sigma X.\varphi_X$ of φ. When referring to fixed point variables, we will therefore write $X > Y$ to abbreviate $\sigma X.\varphi_X > \sigma Y.\varphi_Y$.

The property checking *game* $\mathsf{G}(p_0, \varphi_0)$, when p_0 is a process and φ_0 is a normal μ-calculus formula, is a game played by two players, the Refuter and the Verifier. The Refuter attempts to refute $\Vdash p_0 : \varphi_0$, whereas the Verifier tries to show that it is true.

A *play* of the game $\mathsf{G}(p_0, \varphi_0)$ is a finite or infinite sequence of the form

$$(p_0, \varphi_0)(p_1, \varphi_1)\ldots(p_n, \varphi_n)\ldots$$

where each formula φ_i is a sub-formula of φ_0 and each p_i is a process which is reachable from p_0. If (p_n, φ_n) is a position in a play then then the next position is chosen in

the following way:

- If φ_n is $\psi_1 \wedge \psi_2$ then the Refuter chooses a conjunct ψ_i where $i \in \{1,2\}$, the next position is (p_n, ψ_i).
- If φ_n is $\psi_1 \vee \psi_2$ then the Verifier chooses a disjunct ψ_i where $i \in \{1,2\}$, the next position is (p_n, ψ_i).
- If φ_n is $[a]\psi$ then the Refuter chooses a transition $p_n \xrightarrow{a} p_{n+1}$, the next position is (p_{n+1}, ψ).
- If φ_n is $\langle a \rangle \psi$ then the Verifier chooses a transition $p_n \xrightarrow{a} p_{n+1}$, the next position is (p_{n+1}, ψ).
- If φ_n is $\sigma X.\psi$ then the next position is (p_n, X).
- If $\varphi_n = X$ and X corresponds to $\sigma X.\varphi_X$ in φ_0 then the next position is (p_n, φ_X).

Proposition 4. *If $(p_0, \varphi_0)\ldots(p_n, \varphi_n)\ldots$ is an infinite play of the game $\mathsf{G}(p_0, \varphi_0)$ then there is a unique variable X corresponding to the fixed-point $\sigma X.\varphi_X$ which*

1. *occurs infinitely often, that is for infinitely many i, $\varphi_i = X$, and*
2. *$\sigma X.\varphi_X \geq \varphi_i$ for almost all φ_i, i.e. the property holds for all but finitely many φ_i.*

A proof can be found in [38].

The winning conditions for the respective players are the following.

The Refuter wins a play if

1. The play is $(p_0, \varphi_0)\ldots(p_n, \varphi_n)$ and φ_n is $\langle a \rangle \psi$ and $\{q \mid p_n \xrightarrow{a} q\} = \emptyset$; or
2. The play is infinite and the unique variable X which occurs infinitely often and which subsumes all other variables that occur infinitely often identifies a least fixed point in φ_0.

The Verifier wins a play if

1. The play is $(p_0, \varphi_0)\ldots(p_n, \varphi_n)$ and φ_n is $[a]\psi$ and $\{q \mid p_n \xrightarrow{a} q\} = \emptyset$; or

2. The play is infinite and the unique variable X which occurs infinitely often and which subsumes all other variables that occur infinitely often identifies a greatest fixed point in φ_0.

A strategy is a family of rules which tell a player how to move, a history-free strategy is a strategy which does not depend upon previous positions in the play. A more precise definition can be found in [38].

The following theorem is due to Stirling [38], a proof can be found there.

Theorem 3.

1. $\Vdash p_0 : \varphi_0$ *if, and only if, the Verifier has a history-free winning strategy for* $\mathsf{G}(p_0, \varphi_0)$.

2. $\nVdash p_0 : \varphi_0$ *if, and only if, the Refuter has a history-free winning strategy for* $\mathsf{G}(p_0, \varphi_0)$.

Let the Verifier have a history-free winning strategy for $\mathsf{G}(p_0, \varphi_0)$, and let π be a play of this game. We will call π a *V-play* if the Verifier uses his winning strategy in π. We say that a play π_2 *extends* a play π_1, if π_2 starts in the same position in which π_1 ends. Given two plays $\pi_1 = (p, \varphi) \dots (q, \psi)$ and $\pi_2 = (q, \psi) \dots (r, \delta)$, we write $\pi_1\pi_2$ to mean the play $(p, \varphi) \dots (q, \psi) \dots (r, \delta)$. If both plays π_1 and π_2 are V-plays, then $\pi_1\pi_2$ is a V-play, since the strategy for the Verifier is history-free.

Definition 19. Let π be a play of the form $(p_1, \varphi_1)(p_1, \varphi_2) \dots (p_n, \varphi_n)$, and let X be a fixed point variable corresponding to $\nu X.\varphi_X$. The play π *preserves* X if $\nu X.\varphi_X > \varphi_i$ for all $i = 1, \dots, n$. The play π *progresses* on X if it preserves X and it contains a move of the form $(p, X)(p, \varphi_X)$.

Note that a play $\pi_1\pi_2$ preserves X if, and only if, both plays π_1 and π_2 preserve X, a play $\pi_1\pi_2$ progresses on X if, and only if, both plays π_1 and π_2 preserve X and at least one of π_1 and π_2 progresses on X.

We now give a characterisation of the progress and preservation properties of a play.

Proposition 5. *Let π be a play in the game $\mathsf{G}(p_0, \varphi_0)$ ending in the position (p, φ). Let $X_1, \ldots, X_n$ be all variables such that $\sigma X_i.\varphi_i > \varphi$ which are preserved by π. Let $Y_1, \ldots, Y_n$ be all variables such that $\sigma Y_i.\varphi_i > \varphi$ which are not preserved by π. Then there exist permutations i and j such that*

$$X_{i(1)} > \ldots > X_{i(n)} > Y_{j(1)} > \ldots > Y_{j(m)}.$$

Moreover, π progresses on at most one fixed point variable, which is $X_{i(n)}$.

Proof. The formula φ must contain at least one variable X, since the only atomic formulae are variables. Since φ_0 is in normal form, X occurs only once in φ_0, and therefore φ can occur only once as a sub-formula of φ_0. Therefore, the sub-formulae of φ_0 that are identified by the variables $X_1, \ldots, X_n$ and $Y_1, \ldots, Y_m$ must all contain this occurrence of φ. Hence these sub-formulae can be ordered linearly.

We have to show that the order has the required form. We therefore note that if π does not preserve X, then it cannot preserve any Y with $Y < X$. This shows that the Y_i occur at the end of the ordering.

The play π can progress on at most one fixed point variable. Suppose it would progress on two fixed points. Then the play must have the form

$$(p,X)(p,\varphi_X)\ldots(p,Z)(p,\varphi_Z),$$

and, because of preservation, we have $\nu X.\varphi_X > \varphi_Z$ as well as $\nu Z.\varphi_Z > \varphi_X$. By definition of the sub-term-relation, this implies $\varphi_X \geq \varphi_Z$ and $\varphi_Z \geq \varphi_X$, hence $\varphi_X \equiv \varphi_Z$. Therefore, X and Z must identify the same sub-formula in φ_0, and, as φ_0 is normal, X and Z must be identical.

Suppose, π would progress on a variable other than $X_{i(n)}$, say $X_{i(k)}$ for $1 \leq k < n$. Then the play π would contain the move $(q, X_{i(k)})(\sigma X_{i(k)}.\varphi_{i(k)})$. But since $X_{i(k)} > X_{i(n)}$, then $X_{i(n)}$ could not be preserved. □

5.2 Completeness of the Tableau

In this section we will construct a derivation for $p_0 : \varphi_0$, for closed φ_0. All plays considered in this section are parts of plays of the game $\mathsf{G}(p_0, \varphi_0)$, hence all formulae

encountered are sub-formulae of φ_0. We will use (X) as an abbreviation $(X = \sigma X.\varphi_X)$, where $\sigma X.\varphi_X$ is the sub-formula of φ_0 identified by X.

Definition 20 (Characteristic Formula). Let π be a play ending in the position (p, φ). The *characteristic formula* of π is

$$(+X).(Y_1)\ldots(Y_n).\varphi,$$

where X is the fixed point variable, if any, on which π progresses, and $Y_1 > \cdots > Y_n$ are all fixed point variables such that, for all $i = 1, \ldots, n$, Y_i corresponds to $\sigma Y_i.\varphi_i$ where $\sigma Y_i.\varphi_i > \varphi$, and Y_i is not preserved by π.

This definition is justified by Proposition 5. Note that a characteristic formula of a play can contain at most one modifier for each fixed point variable.

Definition 21. Let π be a V-play ending in the position (pq, φ). The set $pc(p, \varphi, \pi)$ of post-conditions of p at π is the smallest set such that whenever there exists a V-play π_1 of the form $(pq, \varphi)\ldots(q, \psi)$, then $pc(p, \varphi, \pi)$ contains the characteristic formula of π_1.

We define $\mathsf{Z}(\varphi)$ to be the list or all fixed point variables $X_1 > \ldots > X_n$ such that, for any $i = 1, \ldots, n$, $\sigma X_i.\varphi_i > \varphi$ holds where $\sigma X_i.\varphi_i$ is the formula identified by X_i.

Note that the set of assumptions is consistent as a V-play can reach only valid positions. Moreover, the set of post-conditions not only depends on p and φ, but also on q, the context in which the process p occurs. The list $\mathsf{Z}(\varphi)$ depends on the formula φ_0 of the game $\mathsf{G}(p_0, \varphi_0)$, of which π is a part.

In the construction of the tableau we will now restrict our attention to, by definition well-formed, sequents of the form $\mathsf{Z}(\varphi)\ ;\ pc(p, \varphi, \pi) \Vdash p : \varphi$. We will continue to build a derivation using these sequents, and then use the winning condition for the property checking games to conclude that this derivation is successful.

The crucial property of *pc* for the completeness proof is the following.

Lemma 6. *There are only finitely many sequents of the form* $\mathsf{Z}(\varphi)\ ;\ pc(p, \varphi, \pi) \Vdash p : \varphi$, *where* p *is a process constant and* π *is a play of the game* $\mathsf{G}(p_0, \varphi_0)$.

Proof. Let π be any $\mathsf{G}(p_0, \varphi_0)$ play ending in a position of the form (pq, φ). All formulae occurring in a V-play extending π are sub-formulae of φ_0. A characteristic formula

of such a play contains at most one modifier for each fixed point variable, thus there are only finitely many possible different prefixed formulae for each sub-formula of φ_0. Together with the fact that there are only finitely many process constants, this shows that there can only be finitely many sets $pc(p,\varphi,\pi)$. Furthermore, since there are only finitely many φ, there can be only finitely many lists $\mathsf{Z}(\varphi)$. This shows that there are only finitely many sequents of this form. □

The following Lemma allows us to maintain the restricted form of the tableau sequents.

Lemma 7. *Let π be a V-play ending in the position (pr,φ), let π_1 be a V-play of the form $(pr,\varphi)\ldots(qr,\psi)$, and let χ_1 be the characteristic formula of π_1. Then the following is derivable.*

$$\frac{\mathsf{Z}(\varphi)\ ;\ pc(p,\varphi,\pi) \Vdash q:\chi_1 \qquad (A)}{\mathsf{Z}(\psi)\ ;\ pc(q,\psi,\pi\pi_1) \Vdash q:\psi \qquad (B)}$$

Moreover, the unique path from (A) to (B) progresses on X if π_1 progresses on X, and the path preserves X if π_1 preserves X.

Proof. We will start the derivation with a repeated application of the (Z-L) rule, to arrive at Z_1, which is the prefix of $\mathsf{Z}(\varphi)$ containing all the fixed point variables which are preserved by π_1. Note that π_1 can only preserve variables X such that $\sigma X.\varphi_X > \varphi$, and these are just the variables occurring in $\mathsf{Z}(\varphi)$. In order to apply (Z-L), we have to show that, for $\mathsf{Z}(\varphi) = \mathsf{Z}_1\mathsf{Z}_2$, none of variables in Z_2 occurs freely in the sequent. We consider the first element X of Z_2. This is the largest X which is bound by (X) in χ_1, and, as χ_1 is the characteristic formula of π_1, the play π_1 does not preserve X. Any other variable Y in Z_2 is smaller than X, which shows that Y is not preserved by π_1, and this in turn implies that Y does not occur free in χ_1.
Since the play π_1 does not preserve any variable in Z_2, neither does the play $\pi_1\pi_2$, therefore no fixed point variable in Z_2 can occur free in the characteristic formula χ_{12} of such a play $\pi_1\pi_2$. This justifies the following derivation:

$\Phi = \{\chi_{12} \mid \chi_{12} \in pc(p,\varphi,\pi)$ and χ_{12} is the characteristic formula of $\pi_1\pi_2$, for some $\pi_2\ \}$,

$$\dfrac{\dfrac{\mathsf{Z}(\varphi)\ ;\ pc(p,\varphi,\pi) \Vdash q:\chi_1}{\mathsf{Z}(\varphi)\ ;\ \Phi \Vdash q:\chi_1}\ \text{(Weak)}}{\mathsf{Z}_1\ ;\ \Phi \Vdash q:\chi_1}\ \text{(Z-L)}$$

We will continue the derivation with the elimination of all modifiers in χ_1 using the following two rules. These rules are derivable using $((X))$, $((+X))$ and repeated applications of $((X)$-L1) and $((X)$-L2). The rule for (X) is applicable because of the form of Z_1.

$$\text{(A)}\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : (X).\varphi}{\mathsf{Z}, X\ ;\ W_{(X)}(\Phi) \Vdash p : \varphi}\ X \notin \mathsf{Z} \qquad \text{(B)}\ \frac{\mathsf{Z}\ ;\ \Phi \Vdash p : (+X).\varphi}{\mathsf{Z}\ ;\ W_{(+X)}(\Phi), W_{(X)}(\Phi) \Vdash p : \varphi}$$

$$\begin{aligned} W_{(X)}(\Phi) &= \{\psi,\ ((+X).\psi)\ , ((X).\psi) \mid ((X).\psi) \in \Phi\} \\ W_{(+X)}(\Phi) &= \{\psi,\ ((+X).\psi) \mid ((+X).\psi) \in \Phi\} \end{aligned}$$

Note that $W_{(X)}(\Phi)$ and $W_{(+X)}(\Phi)$ both contain Φ.

We now show that we arrive at a sequent having every element of $pc(q, \psi, \pi\pi_1)$ on its left side. Each element of $pc(q, \psi, \pi\pi_1)$ is, by definition, the characteristic formula χ_2 of a play π_2 of the form $(qr, \psi)\ldots(r, \delta)$, for some formula δ. Since the play $\pi_1\pi_2$ has the form $(pr, \varphi)\ldots(r, \delta)$, the characteristic formula χ_{12} of $\pi_1\pi_2$ is then an element of Φ.

The formula χ_2 will be constructed from this corresponding χ_{12}. By construction, the formulae χ_2 and χ_{12} can differ only in their modifier prefix. We show that they can only differ in modifiers for a variable X for which there is a modifier in the prefix of χ_1. We consider the variables X which are not prefixed in χ_1. There are two possible cases in which X is not prefixed in χ_1:

- $\nu X.\varphi_X \not> \psi$. If $\nu X.\varphi_X > \delta$, then X is prefixed by (X) in χ_2 as well as χ_{12}. Otherwise, neither χ_2 nor χ_{12} have a modifier for X.

- $\nu X.\varphi_X > \psi$ and π_1 preserves X. Then $\pi_1\pi_2$ preserves (progresses on) X if, and only if, π_2 preserves (progresses on) X. Hence, χ_2 and χ_{12} have the same modifier, if any, for X.

This shows that χ_2 and χ_{12} differ only in modifiers for those fixed point variables for which there is a modifier in χ_1.

We will now show that after each application of one of the rules (A) and (B) above we have a formula on the left-hand side of the sequent which differs from χ_2 and agrees

with χ_{12} in all modifiers for variables for which there occurs a modifier in the formula on the right.

We consider the modifier in a rule application:

- (X). The rule applied is (A) to a formula $(X).\varphi$ on the right. This formula must be a sub-formula of χ_1. This means that π_1 does not preserve X. Then $\pi_1\pi_2$ does not preserve X and χ_{12} is contains the modifier (X).

  The play π_2 may progress on X, it may not preserve X or it may preserve X. The prefix of the characteristic formula χ_2 of π_2 may thus contain the modifiers $(+X)$, (X) or may be un-prefixed.

  We have a formula $\chi \in \Phi$ which differs from χ_2 only in the prefixes for variables that are prefixed in $(X).\varphi$. This formula agrees with χ_{12} on the prefixes for these variables.

  We note that the formula χ', which is obtained from χ by changing the modifier for X to the modifier for X in χ_2, is in $W_{(X)}(\Phi)$. Hence, there is a formula of the required form on the left-hand side.

- $(+X)$. We apply rule (B) to a formula $(+X).\varphi$ on the right. As $(+X)$ occurs in the prefix of χ_1, we have that π_1 progresses on X.

  As in the previous case, we have a formula $\chi \in \Phi$ which differs from χ_2 only in the prefixes for variables that are prefixed in $(+X).\varphi$, and which agrees with χ_{12} on the prefixes for these variables.

  By the form of the rule, we have to show that χ', the formula which results from χ by changing the modifier for X to the corresponding one in χ_2, is in $W_{(+X)}(\Phi)$ or in $W_{(X)}(\Phi)$.

  There are two cases two consider

  - π_2 preserves X. In this case, $\pi_1\pi_2$ progresses on X and χ_{12} is abstracted with $(+X)$. In case π_2 progresses on X we have that χ_2 is abstracted with $(+X)$, otherwise it has no abstraction for X. In both cases we can find an appropriate formula in $W_{(+X)}(\Phi)$.

– π_2 does not preserve X. Then $\pi_1\pi_2$ does not preserve X, the required formula is among the $W_{(X)}(\Phi)$.

After repeated application of the rules (A) and (B), we arrive at a sequent having the formula ψ on the right, because ψ is the modifier-free part of χ_1. The argument above shows that χ_2 occurs on the left in this sequent. Therefore, every formula in $pc(q,\psi,\pi\pi_1)$ must occur on the left, and we use weakening, if necessary, to arrive at the exact set $pc(q,\psi,\pi\pi_1)$ required on the left-hand side.

By definition of the characteristic formula, χ_1 has a prefix of the form $\rho(Y_1)\ldots(Y_m)$, where $Y_1 > \ldots > Y_m$ and $\sigma Y_i.\varphi_i > \psi$ and π_1 does not preserve Y_i, for $i = 1,\ldots,m$. In the applications of the rule (A), therefore the variables $Y_1,\ldots,Y_m$ are added, in this order, to the end of Z_1. The list Z_1 consists, by construction, of all variables X which are preserved by π_1. Note that preservation of X implies that $\sigma X.\varphi_X > \psi$ holds. The resulting list thus consists of all fixed point variables X such that $\sigma X.\varphi_X > \psi$, i.e. it is $\mathsf{Z}(\psi)$.

The derivation that we have constructed has only one path from the root to a leaf. If π_1 preserves X, then X occurs in Z_1 and therefore all along the path. Hence the path preserves X. If π_1 progresses on X, then the characteristic formula χ_1 of π_1 contains the modifier $(+X)$. Then the rule $((+X))$ is applied and the path in the derivation progresses. □

Lemma 8. *Let π be a V-play ending in the position $(p_1\ldots p_n r,\varphi)$, where all p_i are nonterminals. For any sequent of the form*

$$\mathsf{Z}(\varphi)\ ;\ pc(p_1\ldots p_n,\varphi,\pi) \Vdash p_1\ldots p_n : \varphi \qquad (A)$$

there exists a derivation D for this sequent, such that each leaf has the form

$$\mathsf{Z}(\psi_i)\ ;\ pc(p_i,\psi_i,\pi\pi_i) \Vdash p_i : \psi_i \qquad (B_i)$$

for some i, some ψ_i, and some V-play π_i extending π.

Moreover, the unique path from (A) to (B_i) progresses on X if π_i progresses on X, and preserves X if π_i preserves X.

Proof. We will prove the assertion for the special case where $n = 2$, the general case follows by repeated application of this special case.

Consider a sequent of the form $\mathsf{Z}(\varphi)\ ;\ pc(p_1p_2,\varphi,\pi) \Vdash p_1p_2 : \varphi$ By definition of pc, we know that the V-play π must end in a position (p_1p_2r,φ). Note that then $pc(p_1,\varphi,\pi)$ is also defined. We use the (Cut) rule.

$$\dfrac{\dfrac{\{\mathsf{Z}(\varphi)\ ;\ pc(p_1p_2,\varphi,\pi) \Vdash p_2 : \chi\}_{\chi\in pc(p_1,\varphi,\pi)}}{\{\mathsf{Z}(\psi)\ ;\ pc(p_2,\psi,\pi\pi_1) \Vdash p_2 : \psi\}_{\chi\in pc(p_1,\varphi,\pi)}}\ \text{(Lemma 7)} \qquad \mathsf{Z}(\varphi)\ ;\ pc(p_1,\varphi,\pi) \Vdash p_1 : \varphi}{\mathsf{Z}(\varphi)\ ;\ pc(p_1p_2,\varphi,\pi) \Vdash p_1p_2 : \varphi}$$

Since $\chi \in pc(p_1,\varphi,\pi)$, we know that χ is the characteristic formula of some play π_1 extending π. Note that, by the form of π_1, every free variables of χ must be in $\mathsf{Z}(\varphi)$. Therefore, all sequents are well-formed and Lemma 7 is applicable.

The assertion about progress and preservation follows directly from Lemma 7. □

The completeness construction is based on building a derivation using the following Lemma.

Lemma 9. *Let p be a nonterminal, and let π be a V-play ending in the position (pr,φ). The sequent*

$$\mathsf{Z}(\varphi)\ ;\ pc(p,\varphi,\pi) \Vdash p : \varphi \qquad (A)$$

has a derivation in which each leaf has the form

$$\mathsf{Z}(\psi)\ ;\ pc(q,\psi,\pi\pi_1) \Vdash q : \psi \qquad (B)$$

where π_1 is a V-play containing at least one move, and q is a nonterminal.

Moreover, the unique path from (A) to (B) progresses on X if π_1 progresses on X, and the path preserves X if π_1 preserves X.

Proof. The play π ends in the position (pr,φ).

We consider all possible cases for φ:

- $\varphi \equiv \varphi_1 \wedge \varphi_2$: The play π ends in a conjunction $(pr,\varphi_1\wedge\varphi_2)$, so the Refuter has the turn; he has to choose either of the conjuncts. This gives a V-plays, $\pi\pi_1$ and $\pi\pi_2$, where $\pi_1 = (pr,\varphi_1\wedge\varphi_2)(pr,\varphi_1)$ and $\pi_2 = (pr,\varphi_1\wedge\varphi_2)(pr,\varphi_2)$. Applying $(\wedge)$ and Lemma 7 gives the desired derivation:

$$\dfrac{\dfrac{\mathsf{Z}(\varphi_1\wedge\varphi_2)\ ;\ pc(p,\varphi_1\wedge\varphi_2,\pi) \Vdash p : \varphi_1}{\mathsf{Z}(\varphi_1)\ ;\ pc(p,\varphi_1,\pi\pi_1) \Vdash p : \varphi_1}\ \text{(Lemma 7)} \qquad \text{(symmetric case)}}{\mathsf{Z}(\varphi_1\wedge\varphi_2)\ ;\ pc(p,\varphi_1\wedge\varphi_2,\pi) \Vdash p : \varphi_1\wedge\varphi_2}\ (\wedge)$$

- $\varphi \equiv \varphi_1 \vee \varphi_2$: The play π ends in a disjunction, the Verifier chooses one of the disjuncts, say φ_1. We extend π with $\pi_1 = (pr, \varphi_1 \vee \varphi_2)(pr, \varphi_1)$. We have the following derivation:

$$\dfrac{\dfrac{\mathsf{Z}(\varphi_1 \vee \varphi_2)\ ;\ pc(p, \varphi_1 \vee \varphi_2, \pi) \Vdash p : \varphi_1 \vee \varphi_2}{\mathsf{Z}(\varphi_1 \vee \varphi_2)\ ;\ pc(p, \varphi_1 \vee \varphi_2, \pi) \Vdash p : \varphi_1}\ (\vee)}{\mathsf{Z}(\varphi_1)\ ;\ pc(p, \varphi_1, \pi\pi_1) \Vdash p : \varphi_1}\ \text{(Lemma 7)}$$

- $\varphi \equiv [a]\varphi_1$: For each q with $p \xrightarrow{a} q$ there is an extension $\pi_q = (pr, [a]\varphi_1)(qr, \varphi_1)$ of the play π. We start the derivation thus:

$$\dfrac{\mathsf{Z}([a]\varphi_1)\ ;\ pc(p, [a]\varphi_1, \pi) \Vdash p : [a]\varphi_1}{\{\mathsf{Z}([a]\varphi_1)\ ;\ pc(p, [a]\varphi_1, \pi) \Vdash q : \varphi_1\}_{q \in \{q \mid p \xrightarrow{a} q\}}}\ (\Box)$$

  We continune the derivation for each q. If q is ε then we apply the rule (Axiom). This rule is applicable since, because of the form of π_q, the characteristic formula of π_q, which is φ_1, must be an element of $pc(p, [a]\varphi_1, \pi)$.

$$\dfrac{\mathsf{Z}([a]\varphi_1)\ ;\ pc(p, [a]\varphi_1, \pi) \Vdash \varepsilon : \varphi_1}{}\ \text{(Axiom)}$$

  If q is not ε then we continue the derivation in the following way:

$$\dfrac{\mathsf{Z}([a]\varphi_1)\ ;\ pc(p, [a]\varphi_1, \pi) \Vdash q : \varphi_1}{\mathsf{Z}(\varphi_1)\ ;\ pc(q, \varphi_1, \pi\pi_q) \Vdash q : \varphi_1}\ \text{(Lemma 7)}$$

  In this derivation, q may be a sequence of more than one nonterminal. In this case, Lemma 8 is applied to complete the derivation.

- $\varphi \equiv \langle a \rangle \varphi_1$: The Verifier uses his winning strategy to extend the play π with a new position $\pi_1 = (pr, \langle a \rangle \varphi_1)(qr, \varphi_1)$ for some q such that $p \xrightarrow{a} q$. Such a q exists since otherwise π would not be a V-play. We first apply the rule ($\Diamond$).

$$\dfrac{\mathsf{Z}(\langle a \rangle \varphi_1)\ ;\ pc(p, \langle a \rangle \varphi_1, \pi) \Vdash p : \langle a \rangle \varphi_1}{\mathsf{Z}(\langle a \rangle \varphi_1)\ ;\ pc(p, \langle a \rangle \varphi_1, \pi) \Vdash q : \varphi_1}\ (\Diamond),\ p \xrightarrow{a} q$$

  If q is ε then, as in the case for $\varphi = [a]\varphi_1$, we can apply the rule (Axiom). Otherwise, we continue the derivation in the following way:

$$\dfrac{\mathsf{Z}(\langle a \rangle \varphi_1)\ ;\ pc(p, \langle a \rangle \varphi_1, \pi) \Vdash q : \varphi_1}{\mathsf{Z}(\varphi_1)\ ;\ pc(q, \varphi_1, \pi\pi_1) \Vdash q : \varphi_1}\ \text{(Lemma 7)}$$

In this derivation, q may be a sequence of more than one nonterminal. In this case, Lemma 8 is applied to complete the derivation.

- $\varphi \equiv \mu X.\varphi_X$: Extend π with $\pi_1 = (pr, \mu X.\varphi_X)(pr, X)$. The characteristic formula of π_1 is $(X).X$, and the derivation is:

$$\dfrac{\dfrac{\mathsf{Z}(\mu X.\varphi_X)\ ;\ pc(p, \mu X.\varphi_X, \pi) \Vdash p : \mu X.\varphi_X}{\mathsf{Z}(\mu X.\varphi_X)\ ;\ pc(p, \mu X.\varphi_X, \pi) \Vdash p : (X).X}\ (\mu)}{\mathsf{Z}(X)\ ;\ pc(p, X, \pi\pi_1) \Vdash p : X}\ \text{(Lemma 7)}$$

- $\varphi \equiv \nu X.\varphi_X$: In this case, let $\pi_1 = (pr, \nu X.\varphi_X)(pr, X)$. The play π_1 progresses on no fixed point variable and preserves all fixed point variables larger than X. The characteristic formula of π_1 is thus $(X).X$, and the following is derivable:

$$\dfrac{\dfrac{\mathsf{Z}(\nu X.\varphi_X)\ ;\ pc(p, \nu X.\varphi_X, \pi) \Vdash p : \nu X.\varphi_X}{\mathsf{Z}(\nu X.\varphi_X)\ ;\ pc(p, \nu X.\varphi_X, \pi) \Vdash p : (X).X}\ (\nu)}{\mathsf{Z}(X)\ ;\ pc(p, X, \pi\pi_1) \Vdash p : X}\ \text{(Lemma 7)}$$

- $\varphi \equiv X$ and X corresponds to $\mu X.\varphi_X$. We use the play $\pi_1 = (pr, X)(pr, \varphi_X)$ and have:

$$\dfrac{\dfrac{\mathsf{Z}(X)\ ;\ pc(p, X, \pi) \Vdash p : X}{\mathsf{Z}(X)\ ;\ pc(p, X, \pi) \Vdash p : \varphi_X}\ (X)}{\mathsf{Z}(\varphi_X)\ ;\ pc(p, \varphi_1, \pi\pi_1) \Vdash p : \varphi_X}\ \text{(Lemma 7)}$$

- $\varphi = X$ and X corresponds to $\nu X.\varphi_X$. The play π_1 is $(pr, X)(pr, \varphi_X)$, having the characteristic formula $(+X).\varphi_1$. We have:

$$\dfrac{\dfrac{\mathsf{Z}(X)\ ;\ pc(p, X, \pi) \Vdash p : X}{\mathsf{Z}(X)\ ;\ pc(p, X, \pi) \Vdash p : (+X).\varphi_X}\ (X)}{\mathsf{Z}(\varphi_X)\ ;\ pc(p, \varphi_X, \pi\pi_1) \Vdash p : \varphi_X}\ \text{(Lemma 7)}$$

In all cases except ($\Diamond$) and ($\Box$), the assertion about progress and preservation follows directly from Lemma 7. In the cases for ($\Diamond$) and ($\Box$), the required property is obtained by Lemma 7 and 8 together with the facts that a play $\pi_1\pi_2$ preserves X if, and only if, both plays π_1 and π_2 preserve X, and that a play $\pi_1\pi_2$ progresses on X, if, and only if, it preserves X and at least one of the plays π_1 or π_2 progresses on X. $\square$

Lemma 10. *For any context-free process p, any closed formula* φ *such that* $\Vdash p : \varphi$*, the sequent* $\mathsf{Z}(\varphi)\ ;\ pc(p,\varphi,(p,\varphi)) \Vdash p : \varphi$ *is derivable.*

Proof. Note that (p,φ) is a V-play, since $\Vdash p : \varphi$.

We now construct a derivation for

$$\mathsf{Z}(\varphi)\ ;\ pc(p,\varphi,(p,\varphi)) \Vdash p : \varphi.$$

We use Lemma 9 repeatedly to extend each unsuccessful leaf of the derivation. Suppose this would not give a successful derivation. Then there would be an infinite path of the form

$$\begin{array}{c} \mathsf{Z}(\varphi)\ ;\ pc(p,\varphi,(p,\varphi)) \Vdash p : \varphi \\ \vdots \\ \mathsf{Z}(\varphi_1)\ ;\ pc(p_1,\varphi_1,(p,\varphi)\pi_1) \Vdash p_1 : \varphi_1 \\ \vdots \\ \mathsf{Z}(\varphi_2)\ ;\ pc(p_2,\varphi_2,(p,\varphi)\pi_1\pi_2) \Vdash p_2 : \varphi_2 \\ \vdots \end{array}$$

where each sequent is unsuccessful and each π_i contains at least one move. Then $\pi = (p,\varphi)\pi_1\pi_2\ldots$ would be an infinite play in which the Verifier uses his winning strategy. Since $\Vdash p : \varphi$, the play must be won by the Verifier. Hence, there exists a fixed point variable X which occures infinitely often in π and which subsumes all other fixed point variables occuring infinitely often in π.

We consider a suffix π' of π, where each formula occuring in π' is subsumed by $\nu X.\varphi_X$. Such a suffix exists by Proposition 4.2.

We consider the infinite path in the derivation along π'. Since by Lemma 6 there are only finitely many tableau sequents, there exists a repeat of the form

$$\begin{array}{c} \mathsf{Z}(\psi)\ ;\ pc(q,\psi,\pi) \Vdash q : \psi \\ \vdots \\ \mathsf{Z}(\psi)\ ;\ pc(q,\psi,\pi\pi_R) \Vdash q : \psi, \end{array}$$

Since X occurs infinitely often in π', we may assume that π_R is a part of π' which contains a move of the form $(r,X)(r,\varphi_X)$. In order to show that π_R progresses on X it remains to show that π_R preserves X. Since $\nu X.\varphi_X$ subsumes any formula occurring in π', and thus also in π_R, the play π_R must preserve X. Therefore, π_R progresses on X.

The derivation was constructed using Lemma 9, hence the repeat progresses on X, in other words, it is successful. This is a contradiction to the assumption that each sequent in the infinite path is unsuccessful, hence the construction must result in a successful derivation. □

Lemma 11. *Let π be a V-play ending in the position (ε, φ). The sequent*

$$\mathsf{Z}(\varphi) \Vdash_{\varepsilon} \varphi \qquad (A)$$

has a derivation in which each leaf has the form

$$\mathsf{Z}(\psi) \Vdash_{\varepsilon} \psi \qquad (B)$$

where there exists a V-play π_1 extending π which ends in the position (ε, ψ).

Moreover, the unique path from (A) to (B) progresses on X if π_1 progresses on X.

Proof. The proof is very similar to the proof of Lemma 9, but much simpler. □

Lemma 12. *For any context-free process p, any closed formula φ such that $\Vdash p : \varphi$, the sequent $\Vdash_{\varepsilon} \chi$ is derivable for any $\chi \in pc(p, \varphi, (p, \varphi))$.*

Proof. We notice that every $\chi \in pc(p, \varphi, (p, \varphi))$ is the characteristic formula of a play of the form $(p, \varphi) \dots (\varepsilon, \psi)$. As φ is closed, such a play cannot progress on any variable, which implies that χ is prefixed only by modifiers of the form (X). Hence, we start the derivation of $\Vdash_{\varepsilon} \chi$ with a repeated application of the $(\varepsilon\text{-}(X))$ rule to eliminate the modifiers. We then proceed using Lemma 11; the rest of the proof is almost identical to the proof of Lemma 10, but again much simpler. □

Theorem 4. *For any context-free process p_0 and any closed formula φ_0 such that $\Vdash p_0 : \varphi_0$, the judgement $p_0 : \varphi_0$ is derivable in the tableau.*

Proof. In order to give a derivation for $p_0 : \varphi_0$, we first apply the start rule.

$$(\text{Start}) \; \frac{p_0 : \varphi_0}{\{\varepsilon : \chi\}_{\chi \in pc(p_0, \varphi_0, (p_0, \varphi_0))} \qquad \mathsf{Z}(\varphi_0) \; ; \; pc(p_0, \varphi_0, (p_0, \varphi_0)) \Vdash p_0 : \varphi_0}$$

Note that $\mathsf{Z}(\varphi_0)$ is the empty list. All sequents in this derivation are well-formed, since any formula in $pc(p_0, \varphi_0, (p_0, \varphi_0))$ must be closed. This is the case because a play starting with a closed formula φ_0 does not preserve any fixed point variable.

Now, Lemma 10 and 12 provide the required successful derivation. □

Chapter 6

Conclusions and Further Work

In this report we have considered a sequent calculus for verification of processes. We have introduced this calculus as a general purpose system which is capable of dealing with different classes of processes, in particular classes of infinite-state processes. In order to substantiate this claim we have instantiated the sequent calculus to context-free processes and shown that it is complete for this class of infinite-state processes. As a byproduct of the completeness proof, we have extracted a sound and complete tableau system for context-free processes and the full modal μ-calculus.

An immediate point for further work is the whether the completeness result in this report can be extended to super-classes of context-free processes. The natural next step in this direction would be to consider pushdown processes, as this is the next-larger class beyond context-free processes. Another class, context-free processes with parallel composition and communication, was considered by Hungar [23]. He has given a sound and complete tableau system for these processes and the alternation-free μ-calculus. Although the tableau is in general undecidable, Hungar has shown completeness for a restricted subclass class of these processes. As Hungar's tableau uses a similar idea to the tableau that we have presented in this report, it is conceivable that our tableau can be adapted to this class of processes.

Furthermore, in consideration of compositionality, it would be interesting to investigate whether the sequent calculus is complete with respect to the following, more

general, form of sequents

$$x_1 : \varphi_1, \ldots, x_n : \varphi_n \vdash p(x_1, \ldots, x_n) : \varphi. \qquad (A)$$

It would be of interest, for which classes of process terms p such completeness results can be obtained. In this context, not only the completeness of the sequent calculus is of interest, but also how expressive such sequents are. Answers to this questions could help assessing the ability of the system to support compositional reasoning.

But for all that, the obtainable completeness results are limited by the undecidability of model checking for many interesting classes of infinite state systems. Nevertheless, it is still desirable to have a system supporting the formal verification of processes in such classes. With an automatic system impossible, one wants a system which transfers as much work as possible to an automatic procedure. It is therefore of interest to develop proof tactics and and heuristics to aid the proof-process.

With the goal of a partial automatisation of the verification process, one is interested in implementing the proof system. General-purpose theorem provers such as Isabelle [34], Coq [16] or PVS [36] are specifically made for formulating different logics. They provide a framework for dealing with common problems in the implementation of proof systems, and they facilitate tasks such as automation or user-interaction. Unfortunately, none of these theorem provers provides support for storing the derivation tree. Instead, only the open goals, i.e. the leaves of a derivation tree, are stored. However, for the application of the global proof condition, access to whole derivation structure is crucial. One possible solution to this problem is to encode the derivation tree in the logic. It is possible to tag each sequent in a derivation tree with the path from the root of the tree to this sequent. This allows the whole derivation tree to be reconstructed from its leaves. An efficient way of storing this information has been investigated in [31].

A significant part of my MSc-project was devoted to developing an implementation of the sequent calculus for experimental purposes. This was helpful for understanding the proof system and experimenting with different changes to the sequent calculus. However, due to the time and space limits for the MSc projects, it was infeasible to give a detailed description of both, the completeness proof and the implementation. I will therefore only give a brief sketch of the implementation. I have implemented

the calculus in Isabelle, using the solution sketched above: Instead of working with sequents of the form $\Gamma \vdash \Delta$, this implementation works with sequents $(\Pi)\ \Gamma \vdash \Delta$, where Π is the path from the root to the sequent in question. This allows to reconstruct the whole derivation from the open goals of an Isabelle proof-state. The global proof condition itself is implemented in Standard ML, being invoked from Isabelle as an oracle. A convenient user interface is provided by Proof General [3], allowing easy proof editing and pretty printing. The only drawback of the implementation is the inefficient storage of the proof tree, which makes it only suitable for smaller proofs. However, due to the high-level implementation in a theorem prover , it is well suited for experimenting with various changes to the proof system.

In contrast, Fredlund et al. [1, 19] have implemented a similar proof system in Standard ML rather than in one of the generic theorem provers mentioned above. This allows a more efficient implementation, as their proof checker, named the 'ERLANG Verification Tool', is aimed for larger verification tasks. However, such an implementation requires significantly more time and is less suitable for experimenting than an implementation in a theorem prover.

Nevertheless, it should be possible to extend theorem provers such as Isabelle to allow an efficient representation of proof systems which require access to the derivation tree.

Apart from completeness questions, another interesting goal is simplifying the sequent calculus. The ordinal variables, and in particular the set of ordinal constraints, are notationally heavy and complicate the proof construction. It would be worthwhile to have a simpler notation than ordinal variables. To simplify the calculus, one could replace ordinal variables with modifiers like (X) and $(+X)$ that are used in our tableau for context-free processes. We have shown with our tableau system, that such an approach is possible for a restricted form of sequents. The difficultly to overcome in the sequent calculus is to handle interdependencies between fixed points in the sequents, e.g. as in sequents of the form (A).

The global proof condition of the sequent calculus is a way of showing that an infinite derivation represents a proof. In this condition a finite representation, based on repeats, is used represent infinite derivations. Some global condition is then used

to establish this infinite derivation as a proof. However, these definitions are to some extent arbitrary, there is, for example, no real reason to demand that a repeat and its companion node are connected by a directed path in the derivation. It would therefore be worth investigating more a general, abstract formulation of the idea on which the proof system is based. This would serve for a better understanding of the approach and may help to discover other proof conditions.

Bibliography

[1] Lars åke Fredlund, Dilian Gurov, Thomas Noll, Mads Dam, Thomas Arts, and Gennady Chugunov. A verification tool for Erlang. Software Tools for Technology Transfer. submitted, conditionally accepted for publication after reviewing.

[2] Andersen, Stirling, and Winskel. A Compositional Proof System for the Modal mu-calculus. In *LICS: IEEE Symposium on Logic in Computer Science*, 1994.

[3] David Aspinall. Proof general — organize your proofs! http://www.proofgeneral.org.

[4] John Beatti. A Sequent Calculus for Proving Properties of Processes. MSc Thesis, University of Edinburgh, 1997.

[5] Julian Bradfield and Colin Stirling. Local model checking for infinite state spaces. *Theoretical Computer Science*, 96(1):157–174, April 1992.

[6] Julian Bradfield and Colin Stirling. Modal logics and mu-calculi: an introduction. In *Handbook of Process Algebra*. Elsevier, 2001.

[7] Olaf Burkart. Automatic verification of sequential infinite-state processes. In *Lecture Notes in Computer Science*, volume 1354. Springer, 1997.

[8] Olaf Burkart and Javier Esparza. More infinite results. *Electronic Notes in Theoretical Computer Science*, 6, 1996.

[9] Olaf Burkart and Bernhard Steffen. Model checking for context-free processes. In *International Conference on Concurrency Theory*, pages 123–137, 1992.

[10] Olaf Burkart and Bernhard Steffen. Model checking the full modal mu-calculus for infinite sequential processes. *Theoretical Computer Science*, 221(1–2):251–270, 1999.

[11] R. Cleaveland, M. Klein, and B. Steffen. Faster model checking for the modal mu-calculus. In G. von Bochmann and D. K. Probst, editors, *Computer Aided Verification: Proc. of the Fourth International Workshop CAV'92*, pages 410–422. Springer, Berlin, Heidelberg, 1993.

[12] Mads Dam. Compositional proof systems for model checking infinite state processes. In *International Conference on Concurrency Theory*, pages 12–26, 1995.

[13] Mads Dam. Proving properties of dynamic process networks. *Information and Computation*, 140(2):95–114, 1998.

[14] Mads Dam, Lars-åke Fredlund, and Dilian Gurov. Toward parametric verification of open distributed systems. In A. Pnueli H. Langmaack and W.-P. de Roever, editors, *Compositionality: the Significant Difference*. Springer, 1998.

[15] Mads Dam and Dilian Gurov. Mu-calculus with explicit points and approximations(abstract). In *Fixed Points in Computer Science*, 2000.

[16] Gilles Dowek, Amy Felty, Hugo Herbelin, Gérard Huet, Chet Murthy, Catherine Parent, Christine Paulin-Mohring, and Benjamin Werner. The Coq proof assistant user's guide. Technical Report 154, INRIA, Rocquencourt, France, 1993.

[17] E. Allen Emerson and Chin-Laung Lei. Efficient model-checking in fragments of the propositional mu-calculus. In *Proceedings 1st Annual IEEE Symp. on Logic in Computer Science, LICS'86, Cambridge, MA, USA, 16–18 June 1986*, pages 267–278. IEEE Computer Society Press, Los Alamitos, CA, 1986.

[18] Javier Esparza. Decidability of model checking for infinite-state concurrent systems. *Acta Informatica*, 34(2):85–107, 1997.

[19] Lars-åke Fredlund. A framework for reasoning about erlang code. PhD Thesis, Swedish Institute of Computer Science, 2001.

[20] Gerhard Gentzen. Untersuchungen über das logische Schliessen. *Mathematische Zeitschrift*, 39:176–210 and 405–431, 1935.

[21] Masahito Hasegawa. Models of sharing graphs: A categorical semantics of let and letrec. PhD Thesis, University of Edinburgh, 1997.

[22] C. A. R. Hoare. *Communicating Sequential Processes*. Prentice Hall, 1985.

[23] Hardi Hungar. Local model checking for parallel compositions of context-free processes. In *International Conference on Concurrency Theory*, pages 114–128, 1994.

[24] Hardi Hungar and Bernhard Steffen. Local model checking for context-free processes. *Nordic Journal of Computing*, 1(3):364–385, Fall 1994.

[25] Antonius J. C. Hurkens, Monica McArthur, Yiannis N. Moschovakis, Lawrence S. Moss, and Glen T. Whitney. The logic of recursive equations. *The Journal of Symbolic Logic*, 63(2):451–478, 1998.

[26] J.A. Bergstra and J.W. Klop. Algebra of Communicating Processes with Abstraction. *Theoretical Computer Science*, 37(1):77–121, 1985.

[27] Liu Xinxin Kim G. Larsen . Compositionality through an operational semantics of contexts. *Journal of Logic and Computation*, 1:526–539, 1991.

[28] Dexter Kozen. Results on the propositional μ-calculus. In *Automata, Languages and Programming*, pages 348–359, 1982.

[29] Robin A. Milner. *Communicating and Concurrency*. Prentice Hall, New York, 1989.

[30] David E. Muller and Paul E. Schupp. The theory of ends, pushdown automata, and second-order logic. *Theoretical Computer Science*, 37:51–75, 1985.

[31] George C. Necula and Peter Lee. Efficient representation and validation of logical proofs. In *Proceedings of the 13th Annual Symposium on Logic in Computer Science (LICS'98)*, pages 93–104, Indianapolis, Indiana, 1998. IEEE Computer Society Press.

[32] Faron Moller Olaf Burkart, Didier Caucal and Bernhard Steffen. Verification over infinite states. In *Handbook of Process Algebra*, pages 545–623. Elsevier, 2001.

[33] David Park. Fixpoint Induction and Proofs of Program Properties. In B. Meltzer and D. Michie, editors, *Machine Intelligence*, volume 5, pages 59–78. Edinburgh University Press, 1969.

[34] Lawrence C Paulson. The foundation of a generic theorem prover. *Journal of Automated Reasoning*, 5:363–397, 1989.

[35] Gordon D. Plotkin. A Structural Approach to Operational Semantics. Technical Report DAIMI FN-19, University of Aarhus, 1981.

[36] S. Owre, S. Rajan, J. M. Rushby, N. Shankar, and M. K. Srivas. PVS: Combining specification, proof checking, and model checking. In Rajeev Alur and Thomas A. Henzinger, editors, *Proceedings of the Eighth International Conference on Computer Aided Verification CAV*, volume 1102, pages 411–414, New Brunswick, NJ, USA, 1996. Springer Verlag.

[37] Alex K. Simpson. Compositionality via cut-elimination: Hennessy-milner logic for an arbitrary GSOS. In *Logic in Computer Science*, pages 420–430, 1995.

[38] Colin Stirling. *Modal and Temporal Properties of Processes*. Texts in Computer Science. Springer, 2001.

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

[40] Igor Walukiewicz. Notes on the propositional μ-calculus: Completeness and related results. Technical Report NS-95-1, University of Aarhus, 1995.

[41] Igor Walukiewicz. Pushdown Processes: Games and Model-Checking. *Information and Computation*, 164(2):234–263, January 2001.