Syntactic complexity in the modal μ calculus

Karoliina Lehtinen

Doctor of Philosophy
Laboratory for Foundations of Computer Science
School of Informatics
University of Edinburgh
2017

Abstract

This thesis studies how to eliminate syntactic complexity in L_μ, the modal μ calculus. L_μ is a verification logic in which a least fixpoint operator μ, and its dual ν, add recursion to a simple modal logic. The number of alternations between μ and ν is a measure of complexity called the formula's index: the lower the index, the easier a formula is to model-check. The central question of this thesis is a long standing one, the L_μ index problem: given a formula, what is the least index of any equivalent formula, that is to say, its *semantic* index?

I take a syntactic approach, focused on simplifying formulas. The core decidability results are (i) alternative, syntax-focused decidability proofs for ML and Π_1^μ, the low complexity classes of L_μ; and (ii) a proof that Σ_2^μ, the fragment of L_μ with one alternation, is decidable for formulas in the dual class Π_2^μ.

Beyond its algorithmic contributions, this thesis aims to deepen our understanding of the index problem and the tools at our disposal. I study *disjunctive form* and related syntactic restrictions, and how they affect the index problem. The main technical results are that the transformation into disjunctive form preserves Π_2^μ-indices but not Σ_2^μ-indices, and that some properties of binary trees are expressible with a lower index using disjunctive formulas than non-deterministic automata. The latter is part of a thorough account of how the L_μ index problem and the Rabin–Mostowski index problem for parity automata are related.

In the final part of the thesis, I revisit the relationship between the index problem and parity games. The *syntactic* index of a formula is an upper bound on the descriptive complexity of its model-checking parity games. I show that the *semantic* index of a formula Ψ is bounded above by the descriptive complexity of the model-checking games for Ψ. I then study whether this bound is strict: if a formula Ψ is equivalent to a formula in an alternation class C, does a formula of C suffice to describe the winning regions of the model-checking games of Ψ? I prove that this is the case for ML, Π_1^μ, Σ_2^μ, and the disjunctive fragment of any alternation class. I discuss the practical implications of these results and propose a uniform approach to the index problem, which subsumes the previously described decision procedures for low alternation classes.

In brief, this thesis can be read as a guide on how to approach a seemingly complex L_μ formula. Along the way it studies what makes this such a difficult problem and proposes novel approaches to both simplifying individual formulas and deciding further fragments of the alternation hierarchy.

Lay Summary

Verification is the field of computer science concerned with ensuring that programs behave according to the specifications describing the programmer's intentions. One verification strategy – called model-checking – is to build a model of all the possible behaviours of a program in order to then check that the specifications hold in this model. The modal μ calculus, written L_μ, is a verification logic that can be used for model-checking: formulas of L_μ describe program behaviours, such as "every possible execution of the program terminates" or "whenever a message is sent, a response is eventually received".

Model-checking can be a computationally demanding task; this becomes an issue for the verification of large systems. The computational complexity of model-checking – i.e. how much resources are needed – depends in particular on how complex the description of the specifications is: simple formulas are easy to check, even on large systems, while checking complex ones does not scale well. It is therefore judicious to use simple formulas wherever possible.

In L_μ, the complexity of a formula is measured by its *index*: the lower the index, the easier model-checking is; the higher the index, the more properties can be expressed. While it is better to use low-index formulas for model-checking, deciding whether a particular property can be expressed with a given index is a long-standing open problem. Previously, only the very simplest classes of L_μ formulas were known to be recognisable: given a formula, it is possible to find out whether it is equivalent to a formula in these classes.

This thesis extends those results. Its main goal is to understand how to analyse a seemingly complex L_μ formula to detect whether it is equivalent to a lower index formula. Its focus is on finding simplifications that yield the equivalent formulas of lower index, rather than just deciding their existence.

The main technical contributions consist of novel tools for recognising unnecessary complexity in formulas, novel procedures for identifying some classes of L_μ formulas, and an extension of the game-theoretic interpretation of L_μ model-checking to account for formulas that are equivalent to lower-index formulas.

Acknowledgements

First I thank my supervisor Julian Bradfield for his support over the past years. I am particularly grateful for his encouragements to pursue my own ideas, take risks and enjoy my time here. I thank my second supervisor Sandra Quickert for the patience with which she has scrutinised my work, and her valuable input and guidance.

I thank Richard Mayr and Damian Niwiński for agreeing to being my examiners.

Many people have given me feedback at various stages, whether by helping me direct my research, listening to my talks or reading drafts of this manuscript. For that, I thank Colin Stirling, Kousha Etessami, JC Denis, Sonja Lehtinen, Stefan Fehrenbach and the many researchers I've had the opportunity to share ideas with, both in Edinburgh and at conferences and workshops.

Others have helped me stay sane and happy during my time in Edinburgh. Thanks to my officemates Patrick, Fabian, Ricardo and Stefan for being such good company. Thanks to all those I've had the pleasure to dance with over these years, especially all my friends from New Scotland Country Dance Society. Thanks to Lydia, Jenny, Emeline and Chloë for never being as far away as geography would suggest.

I thank my family and Stefan for their moral support throughout.

I am grateful to EPSRC for funding my doctoral studies and to LFCS, SICSA and ACM for providing financial support for attending conferences and summer schools.

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.

Karoliina Lehtinen

(Karoliina Lehtinen)

Table of Contents

Chapter 1

Introduction

The field of complexity theory studies the fundamental question of what makes a problem difficult to solve. In the context of model-checking, where the problem is to decide whether a system satisfies some desirable properties, this amounts to asking what makes properties hard to recognise. A logician's approach to this question is to consider the formalisms that describe properties: properties that are algorithmically simple tend to also be expressible with simple formulas. However, a complex formula can either describe a genuinely complex property, or be equivalent to a simpler formula. This thesis studies how to identify when the apparent complexity of a formula, *i.e.* its syntactic complexity, is necessary, and a reflection of the intrinsic complexity of a property, and when is it merely incidental.

The formalism studied in this thesis is L_μ, the propositional modal μ calculus. L_μ is a powerful verification logic obtained by adding the least fixpoint operator μ and its dual, ν, to a simple modal logic. The least fixpoint μ is used to describe finite properties, such as "eventually A is true", while ν denotes infinite ones, such as "A is always true". Combining μ and ν generates properties such as "eventually A is always true". Further alternations between μ and ν allow L_μ to describe increasingly intricate mutual dependencies between finite and infinite behaviours. The number of these alternations is a measure of syntactic complexity: the fewer alternations, the easier a formula is to model-check; the more alternations, the more properties can be expressed. This is called the formula's index. To reduce the algorithmic complexity of checking a formula, it is judicious for model-checkers to use an equivalent formula with lower index, if it exists. The central question of this thesis is a long standing one, the L_μ index problem: given a formula, what is the least index of any equivalent formula?

For model-checking, recognising the least index a property can be expressed with – its semantic index – is not enough. Producing the low-index formula is just as important. I therefore take a syntactic approach to the index question, focusing on simplifying formulas. The core results are novel decidability proofs for the simplest classes of L_μ formulas, all centred around turning the input formula into a formula of the desired index. Along the way, I study the effects of syntactic restrictions on the index problem and give a thorough account of the relationship between the L_μ index problem and the related Rabin–Mostowski index problem for parity automata. Finally, I revisit the fundamental relationship between L_μ and parity games to account for the semantic index of formulas as well as their syntactic index. The goal of this research has been to develop ways to analyse and simplify high index L_μ formulas. I therefore conclude the manuscript with an inventory of methods for approaching high-complexity formulas that derive from this work.

Synopsis

The core of this manuscript is in three parts. The first one discusses how syntactic restrictions affect the index problem.

Although the syntax of L_μ is itself simple, formulas of L_μ are notoriously difficult to make sense of, for humans and algorithms alike. Restricting the syntax of L_μ forces more structure onto formulas, which then become easier to work with. Chapter 3 studies one such restriction, *disjunctive form*, a key component to most proofs in this thesis. It first briefly surveys the characteristics of disjunctive formulas that make them easy to work with, then provides a detailed rewriting of the proof that disjunctive L_μ is expressively complete. The chapter concludes with an investigation of the effects of the transformation into disjunctive form on the index of formulas. I show that it preserves the index of formulas in Π^μ_2, the alternation class of formulas of the form $\nu Y.\mu X.\phi$. The dual, however, does not hold: there are formulas in Σ^μ_2 which, in disjunctive form, have arbitrarily high alternation depth. These results are, to the best of my knowledge, novel.

In Chapter 4, I introduce the *non-deterministic* fragment of L_μ in order to take a closer look at the differences between disjunctive L_μ and non-deterministic parity automata, a related formalism, as well as their respective index problems.

The motivation for this chapter comes from the well-established idea that L_μ and alternating parity automata are closely related, if not almost identical models of computation. In the automata-theoretic world, instead of alternations, one can count the number of priorities in the acceptance condition of automata. The Rabin–Mostowski index problem is to find the least acceptance condition necessary to describe a property with a given type of automata. The parallels with the L_μ index problem are clear; however, since L_μ is defined on a more general set of input structures than traditional automata, these two index problems are distinct. This chapter aims to clarify what exactly happens to the index problem when moving between the logical and automata-theoretic frameworks.

The starting point is disjunctive form, which was originally designed to mimic non-determinism in parity automata. However, I show that disjunctive formulas are in fact more powerful than non-deterministic automata, in the sense that they can express some properties of binary trees with a lower index than an equivalent non-deterministic automaton. In contrast, non-deterministic L_μ, as its name indicates, is a L_μ fragment that corresponds exactly to non-deterministic automata: any such formula can be turned into a non-deterministic automaton of the same index that agrees with it on all binary trees, and vice-versa. With this fragment, I highlight the differences between the index problem for disjunctive L_μ and non-deterministic automata, and give a L_μ-focused proof that non-deterministic automata are as expressive as alternating parity automata. I then study the index problem for this fragment itself.

The second part of the thesis presents three novel decision procedures. The first two, in Chapter 5, provide alternative, syntax-focused proofs that formulas expressible without any fixpoints, and those expressible with only one type of fixpoint, are recognisable. These correspond to proofs of decidability for ML, Σ^μ_1 and Π^μ_1. They give a clear account of how syntactic complexity can be eliminated to get a formula in the target class from the original formula. Chapter 6 shows that the next alternation class, Σ^μ_2, is decidable for input formulas in the dual class Π^μ_2. For general L_μ formulas, I present a sequence of formulas with eventually coincides with the input formula exactly when it is semantically in Σ^μ_2. Finding a bound on the convergence of this sequence, which would decide Σ^μ_2, remains open.

In the final part of the thesis, I revisit some of the fundamentals of L_μ-theory in light of the index problem. I show that at least for the formulas semantically in alternation classes up to Σ^μ_2, the descriptive complexity of the model-checking games is exactly the semantic complexity of the formula, rather than its syntactic complexity as thought so far.

Chapter 7 relates the index problem to the descriptive complexity of the model-checking parity games generated by a formula. It is a well established fact that the winning regions of the model-checking parity games of a formula Ψ can be described by a formula of the same index as Ψ. Say that Ψ is *interpreted* by a formula of the same index. The syntactic index of a formula is therefore an upper bound on the descriptive complexity of its model-checking parity games. I extend this relationship to semantic complexity by showing that the *semantic* index of a formula Ψ is bounded above by the descriptive complexity of the model-checking games for Ψ: if Ψ is interpreted by Φ, a formula with a smaller index than Ψ, then Ψ can be simplified into a formula of the same index as Φ. I then study how this can be used in practice to simplify formulas.

Chapter 8 asks whether the converse holds: if Ψ is semantically in an alternation class C, then is Ψ interpreted by some formula in the class C? I show that such an interpretation theorem holds for ML, and for Π^{μ}_{1} and Σ^{μ}_{2} as long as the input formula is in disjunctive form. Finally, I show a general interpretation theorem for all disjunctive alternation classes, as long as the input is in co-disjunctive form, the dual of disjunctive form. These results show how far the techniques of the previous chapters can go, and highlight the challenges that need to be addressed to take the next step up the alternation hierarchy.

I conclude the thesis with a discussion of how the results presented throughout improve our understanding of this long standing open problem and what I believe to be some promising directions for future work. A separate document, in the Appendix, compiles all the technical results of this thesis into an inventory of tools for analysing complex L_μ formulas.

The next chapter presents the prerequisite concepts, notations and results as well as the wider context of this work.

Some of the results in Part II stem from joint work with Sandra Quickert; this will be specified throughout. The contents of Section 3.4 on disjunctive form and the alternation hierarchy, and Chapter 5 on decision procedures for ML and Π^{μ}_{1} have been published [Leh15, LQ15]. The contents of Chapter 6 have been accepted for publication [LQ17]. I hope to also publish some of the content of Part III.

Chapter 2

Preliminaries

This chapter presents the prerequisite background material. In particular L_μ, its relation to parity games and automata, and the central index problem are defined and set within their historical context. Along the way, notations and conventions are fixed. A more thorough introduction to L_μ can be found in Bradfield and Stirling 2007 [BS07].

2.1 The modal μ calculus

2.1.1 A brief history

Although modal logics have roots in philosophy, their study in the context of verification is firmly grounded within the realm of computer science. The idea of formalising program specifications and proving these was pioneered in the 1960s, mainly by Floyd [Flo67] and Hoare [Hoa69]. Hoare Logic expresses assertions as precondition/postcondition pairs, that lend themselves to mechanical proofs. We can date the introduction of modal logics into verification research to the 70s: temporal and modal logics started to appear [Pra76, Pnu77] as an alternative formalism that could model non-terminating processes. Dynamic logics [HMP77] received a lot of attention, in particular Propositional Dynamic Logic (PDL) [FL79]. PDL has its semantics defined on Kripke structures [Kri63], which have since become a standard way to describe computational systems: the execution of a program is a path from node to node, each node describing the global state of the system at some stage of the execution. Although the symbols $\Diamond$ and $\Box$ were already in use in PDL, the L_μ-semantics of these modalities come directly from Hennessy–Milner logic [HM80, HM85], developed within the context of process calculi: An expression $\Diamond\phi$ denotes the existence of a next state satisfying ϕ while $\Box\phi$ denotes that all next states satisfy ϕ.

Around the same time Linear Temporal Logic (LTL) [GPSS80] introduced temporal connectives that could express global properties of paths such as "until" or "finally".

Computational Tree Logic (CTL) [EC82] generalised these ideas to branching time, again using Kripke structures.

Meanwhile, in the field of database query languages, Aho and Ullman [AU79] noted that closure properties could be expressed by introducing fixpoints. The line of research that followed on fixpoint extensions of First Order Logic (FOL) [GS86] includes the celebrated Immerman–Vardi result [Imm82, Var82] which states that in the presence of a linear order, properties expressible in FOL with a least fixpoint correspond exactly to those decidable in polynomial time. In the context of modal logics, Pratt abandoned the more intuitive temporal connectives of CTL in favour of the more agile fixpoint operator μ when he described the precursor [Pra81] of L_μ. A few years later, Kozen presented L_μ [Koz83] which has been extensively studied ever since.

The appeal of L_μ comes from its simple yet remarkably expressive syntax, its agreeable decidability properties, and its deep connections with other mathematical formalisms. Indeed, the satisfiability problem is EXPTIME-complete [EJ88] while the model-checking problem is equivalent to solving parity games [Wil01], a problem conjectured to be in P. L_μ subsumes many other verification logics and constitutes the bisimulation-invariant fragment of monadic second-order logic [JW96], another well-studied formalism. The semantics of L_μ can be described in terms of amorphous alternating parity automata, linking L_μ to the rich theory of automata. Hence L_μ is a prime example of the fertile intersection of logic, automata and games.

2.1.2 Syntax of L_μ

Fix once and for all countably infinite sets $Prop = \{P, Q, ...\}$ of propositional variables, $Var = \{X, Y, ...\}$ of fixpoint variables and a finite set $Act = \{a, b, ...\}$ of actions.

Definition 1. *(L_μ)* The syntax of L_μ is given by:

$$\phi := P \mid X \mid \neg P \mid \phi \wedge \phi \mid \phi \vee \phi \mid \langle a \rangle \phi \mid [a]\phi \mid \mu X.\phi \mid \nu X.\phi \mid \bot \mid \top$$

Conjunctions take precedence over disjunctions. The scope of fixpoint bindings extends as far as possible to the right while the scope of modalities extends as little as possible to the right. For example, $\mu X.\langle a \rangle X \wedge C \vee B$ is parsed as $\mu X.(((\langle a \rangle X) \wedge C) \vee B)$.

Unimodal L_μ is the original version of L_μ which ignores action labels on transitions. In its syntax, $\Diamond\phi$ and $\Box\phi$ replace $\langle a \rangle \phi$ and $[a]\phi$. Often, when action labels are not relevant, examples will use unimodal L_μ for clarity.

This thesis will also make extensive use of an alternative notation for modalities, introduced by Janin and Walukiewicz [JW95] to define disjunctive form: $\xrightarrow{a}\mathcal{F}$, where $\mathcal{F}$ is a set of formulas. The modality $\xrightarrow{a}\mathcal{F}$ is short for $(\bigwedge_{\psi \in \mathcal{F}} \langle a \rangle \psi) \wedge [a] \bigvee_{\psi \in \mathcal{F}} \psi$ meaning that every formula in $\mathcal{F}$ is realised by at least one a-successor and every a-successor realises at least one of the formulas in $\mathcal{F}$. The unimodal operator $\rightarrow\mathcal{F}$ is defined similarly with respect to $\Box$ and $\Diamond$. Although initially less intuitive than $\Diamond$ and $\Box$ modalities, the use of $\rightarrow\mathcal{F}$ will make formulas much clearer throughout.

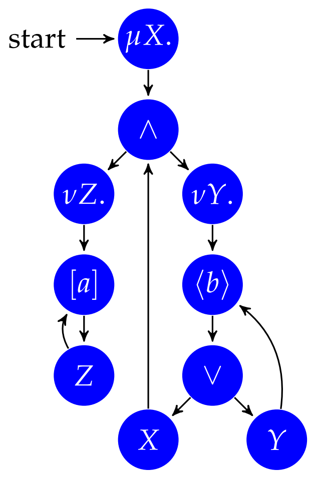
Notation and terminology

The syntax of L_μ presented here only allows formulas in positive form: negation is only applied to propositional variables. In a slight abuse of notation, $\neg\phi$ should be read as the negation of ϕ, which can be obtained syntactically by inverting μ and ν, $\wedge$ and $\vee$, $[a]$ and $\langle a\rangle$, and negating literals, *i.e.* propositional variables and their negations.

A fixpoint variable X which does not appear in the scope of $\mu X.\phi$ or $\nu X.\phi$ is said to be *free*, and the set of free fixpoint variables of a formula ψ is written $Free_\psi$. A formula without free fixpoint variables is a sentence. A fixpoint μ or ν *binds* a variable $X \in Free_\phi$ in $\mu X.\phi$ or $\nu X.\phi$ respectively. The formula ϕ is called the binding formula of X, and will be written ϕ_X.

A formula is *guarded* if every fixpoint variable is in the scope of a modality within its binding. As is well documented in the literature [Mat02, KVW00], every L_μ formula is equivalent to a formula in guarded form. Without loss of expressivity, all L_μ formulas are assumed to be in guarded form.

The set of subformulas of a sentence Ψ is written $sf(\Psi)$. For notational purposes, call the binding formula of X, written ϕ_X, the immediate subformula of X. We call this a fixpoint regeneration. The *parse tree* of a sentence is the tree which has for nodes the subformulas of Ψ, which is rooted at the formula itself and where the child of a node consists of the parse-trees of its immediate subformulas. In other words it consists of the formula, written as a tree, with back-edges from fixpoint variables to their bindings.



Example 2. The parse-tree of the formula $\mu X.(\nu Z.[a]Z) \wedge \nu Y.\langle b\rangle(X \vee Y)$ is illustrated here. For clarity, nodes are labelled only with the outmost operator of the subformula they correspond to. Here, the subformulas X and Y are reachable from each other, via regeneration, and Z is reachable from both, but neither is reachable from Z.

A subformula ϕ of Ψ is *reachable* from another subformula ϕ' if it is reachable in the parse-tree – that is to say, if there is a sequence of subformulas $f_0 f_1 ... f_n$ such that f_0 is ϕ, f_n is ϕ' and f_{i+1} is an immediate subformula of f_i for all i, $0 \leq i < n$. $\Psi(\phi)$ means that Ψ is a formula with a subformula ϕ. Then, writing $\Psi(\phi')$ indicates Ψ where all subformulas ϕ are substituted with ϕ', also written $\Psi[\phi'/\phi]$.

2.1.3 Semantics of L_μ

L_μ formulas operate on regular trees, representing labelled transition systems.

Definition 3. *(Trees)* A labelled transition system, or simply *structure*, represented as a tree $\mathcal{M} = (S, E, P, L)$, rooted at $r \in S$ consists of:

- a set of states S,
- a successor relation $E \subseteq S \times S$,
- a labelling $P : S \to Prop_{\mathcal{M}}$ of states from a finite set $Prop_{\mathcal{M}} \subset Prop$,
- a labelling $L : E \to Act_{\mathcal{M}}$ of edges with action labels from a finite set $Act_{\mathcal{M}}$

The edge relation must be such that for every state $s \in S$, the set of its ancestors, $\{w \in S \mid \exists w_1, \ldots, w_k. \, (w, w_1) \in E, (w_1, w_2) \in E, \ldots (w_k, s) \in E\}$, is finite and well-ordered with respect to the transitive closure of E. Write $s \xrightarrow{a} s'$ for s' is an a-successor of s, that is $(s, s') \in E$ and $L(s, s') = a$. Write $s \to s'$ for $(s, s') \in E$.

The scope of this thesis is restricted to regular trees with finite but unbounded branching, which can be finitely represented as trees with back edges. Due to space concerns, this thesis uses such finite representations throughout. It will be practical to sometimes also add back edges to infinite trees to build new trees. In such cases, the L_μ formula is evaluated on the bisimilar infinite tree.

Definition 4. *(Ranked labelled transition systems)* A labelled transition system is said to be *ranked* if each node has at most one a-successor for each $a \in Act$; otherwise it is *unranked*. Unless specified, a structure should be assumed to be unranked. See Figure 2.1.1 for an example.

Binary trees where nodes have a left successor and a right successor are an example of ranked structures. This distinction enables the comparison of L_μ and automata-theoretic literature. The latter typically operates on ranked structures only, while L_μ is defined in the more general framework of unranked structures.

Definition 5. *(Semantics of L_μ)* Given a structure $\mathcal{M} = (S, E, P, L)$ rooted at $r \in S$, and an interpretation $\mathcal{I} : Var \to \mathcal{P}(S)$ of fixpoint variables, the set $\parallel \phi \parallel_{\mathcal{I}}^{\mathcal{M}} \subseteq S$ of states of $\mathcal{M}$ satisfying the formula ϕ is defined by:

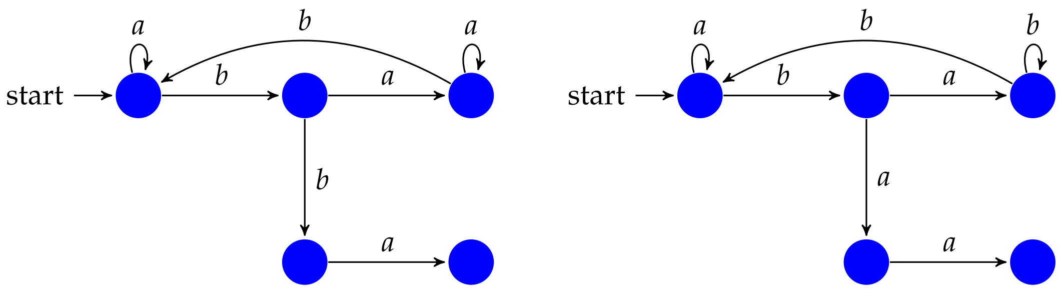


Figure 2.1.1: An example of a ranked (left) and unranked (right) structure, represented as a tree with back edges.

$$
\begin{aligned}
\| A \|_{\mathcal{I}}^{\mathcal{M}} &= \{s | A \in P(s)\} \text{ for } A \in Prop \\
\| \neg A \|_{\mathcal{I}}^{\mathcal{M}} &= \{s | A \notin P(s)\} \\
\| X \|_{\mathcal{I}}^{\mathcal{M}} &= \mathcal{I}(X) \\
\| \phi_0 \wedge \phi_1 \|_{\mathcal{I}}^{\mathcal{M}} &= \| \phi_0 \|_{\mathcal{I}}^{\mathcal{M}} \cap \| \phi_1 \|_{\mathcal{I}}^{\mathcal{M}} \\
\| \phi_0 \vee \phi_1 \|_{\mathcal{I}}^{\mathcal{M}} &= \| \phi_0 \|_{\mathcal{I}}^{\mathcal{M}} \cup \| \phi_1 \|_{\mathcal{I}}^{\mathcal{M}} \\
\| \langle a \rangle \phi \|_{\mathcal{I}}^{\mathcal{M}} &= \{s | \exists s' \in \| \phi \|_{\mathcal{I}}^{\mathcal{M}} . s \xrightarrow{a} s'\} \\
\| [a] \phi \|_{\mathcal{I}}^{\mathcal{M}} &= \{s | \forall s' . s \xrightarrow{a} s' \implies s' \in \| \phi \|_{\mathcal{I}}^{\mathcal{M}}\} \\
\| \nu X . \phi \|_{\mathcal{I}}^{\mathcal{M}} &= \bigcup \{C \subseteq S | C \subseteq \| \phi \|_{\mathcal{I}[X:=C]}^{\mathcal{M}}\} \\
\| \mu X . \phi \|_{\mathcal{I}}^{\mathcal{M}} &= \bigcap \{C \subseteq S | \| \phi \|_{\mathcal{I}[X:=C]}^{\mathcal{M}} \subseteq C\}
\end{aligned}
$$

where $\mathcal{I}[X := C]$ interprets X as the set C but otherwise agrees with $\mathcal{I}$.

For a state of $\mathcal{M}$ and a subformula ϕ of a sentence Ψ of L_μ, we write $s \models \phi$, read s models ϕ or ϕ holds in s, for $s \in \| \phi \|_{\mathcal{I}}^{\mathcal{M}}$, where $\mathcal{I}$ maps each fixpoint variable X to $\| \mu X . \phi_X \|_{\mathcal{I}}^{\mathcal{M}}$. Finally, write $\mathcal{M} \models \Psi$ if $r \models \Psi$.

Two formulas are equivalent if they agree on all structures $\mathcal{M}$.

2.1.4 Bisimulation invariance

Bisimulation is a notion of semantic equivalence for processes. It is of particular importance in the context of L_μ since this logic is bisimulation invariant: if a formula holds in a structure, it holds in all bisimilar structures.

Definition 6. *(Bisimulation)* Two structures are bisimilar if there is a bisimulation between them. Given two structures $\mathcal{T} = (T, E_T, P_T, L_T)$ and $\mathcal{M} = (M, E_M, P_M, L_M)$, a bisimulation $B \subseteq T \times M$ is a binary relation such that for all $(t, m) \in B$ and all $a \in Act$ it is the case that $P_T(t) = P_M(m)$, and for all $t' \in T$ if $t \xrightarrow{a} t'$ then there is a $m' \in M$ such that $(t',m') \in B$ and $m \xrightarrow{a} m'$; and symmetrically, for all $m' \in M$ if $m \xrightarrow{a} m'$ then there is a $t' \in T$ such that $(t',m') \in B$ and $t \xrightarrow{a} t'$.

Theorem 7. *[JW96] L_μ is bisimulation invariant.*

2.2 The alternation hierarchy

Complexity in L_μ can be measured in terms of alternations between the two fixpoint operators. Each additional alternation increases the expressive power of the logic, as was shown independently by Bradfield [Bra98] and Lenzi [Len96]. However, additional alternations also increase the cost of model-checking, at least for any of the currently known algorithms. The properties expressible in L_μ can therefore be stratified into a strict hierarchy of alternation classes, and for each class, we can ask whether membership is decidable. This is the problem of deciding the alternation hierarchy, or the L_μ index problem.

Syntactically, what exactly should count as an alternation is not entirely trivial. For example, there is a distinction between the alternations in the following formulas:

$$\mu X.\nu Y.\mu W.\Box Y \wedge (C \vee \Diamond W) \vee B \wedge \Diamond X$$

$$\mu X.(\nu Y.\mu W.\Box Y \wedge (C \vee \Diamond W)) \vee B \wedge \Diamond X$$

In the latter, the subformula $(\nu Y.\mu W.\Box Y \wedge (C \vee \Diamond W))$ is properly embedded in the binding of X so that there is no mutual dependency between the definitions of ϕ_X and ϕ_Y. This means that the apparent alternation between X and Y is not in fact reflected in the algorithmic complexity of model-checking. The definition of alternation depth, here also referred to as index to match the automata-theoretic terminology, is meant to only capture alternations which generate algorithmic complexity, matching the definition given in Niwiński 1986 [Niw86]. The presentation here emphasizes the relationship of a formula's alternation depth to the priorities in parity games and automata acceptance conditions (see the next sections). A thorough discussion on how to define alternation depth, and comparison of definitions used in the literature can be found in Bradfield and Stirling 2007 [BS07].

Definition 8. *(Priority assignment and index)* A priority assignment for a sentence Ψ is a mapping $\Omega : Var_\Psi \to \{m,...,q-1,q\}$, where $m \in \{0,1\}$ and q is a positive integer, of priorities to the fixpoint variables Var_Ψ of Ψ such that μ-bound variables receive odd priorities while ν-bound variables receive even priorities. A priority assignment is order preserving if whenever X is free in the formula binding Y, $\Omega(X) \geq \Omega(Y)$.

A formula has index I if it admits an order preserving priority assignment with co-domain I.

Definition 9. *(Alternation hierarchy)* The base of the alternation hierarchy is ML, the modal fragment of L_μ, consisting of formulas without any fixpoints. Formulas with only ν-bound fixpoints, or only μ-bound fixpoints respectively, have index $\{0\}$, or $\{1\}$, corresponding to the alternation classes Π^μ_1, or Σ^μ_1.

The classes Π^μ_i and Σ^μ_i for positive even i correspond to formulas with indices $\{1,...,i\}$ and $\{0,...,i-1\}$ respectively, while for odd i they correspond to formulas with indices $\{0,...,i-1\}$ and $\{1,...,i\}$ respectively.

The semantic alternation class of a formula is the the least alternation class of any equivalent formula.

Example 10. The formula $\mu X.\nu Y.\Box Y \wedge \mu Z.\Box(X \vee Z)$ accepts the priority assignment $\Omega(X) = 1, \Omega(Y) = 0$ and $\Omega(Z) = 1$, so it has index $\{1,0\}$ and is in the class Σ^μ_2. However, it is equivalent to $\mu X.\Box X$ which holds in structures without infinite paths, and is therefore semantically in Σ^μ_1.

Theorem 11. *[Bra98, Len96] The alternation hierarchy is strict: for each index $I = \{m,...,q\}$, where $m \in \{0,1\}$, there are formulas that are not equivalent to a formula with smaller index.*

Definition 12. *(Ambiguous classes)* Some formulas have both index $\{0,...,q\}$ and $\{1,...,q+1\}$ and therefore belong to both an alternation class Π^μ_i and its dual Σ^μ_i. The intersection $\Pi^\mu_i \cap \Sigma^\mu_i$ is called an ambiguous class. The first ambiguous class collapses to the modal level: $\Pi^\mu_1 \cap \Sigma^\mu_1 = ML$. This is also true semantically: if a formula is equivalent to both a Π^μ_1 and a Σ^μ_1 formula, then it is semantically in ML. Formulas in $\Pi^\mu_2 \cap \Sigma^\mu_2$ are called *alternation free*: they don't have pairs of mutually reachable μ- and ν-bound fixpoint variables. If a formula is semantically in both Π^μ_2 and Σ^μ_2, then it is equivalent to an alternation-free formula [SA05].

Definition 13. *(Weak alternation hierarchy)* The weak alternation hierarchy is defined within the alternation free class $\Pi^\mu_2 \cap \Sigma^\mu_2$. If a formula in $\Pi^\mu_2 \cap \Sigma^\mu_2$ admits an order-preserving priority assignment Ω with co-domain I such that $\Omega(X) \leq \Omega(Y)$ if the binding formula ϕ_Y of Y is a subformula of the binding formula ϕ_X of X, then the formula has *weak index I*. The weak alternation classes, weak Π^μ_i and weak Σ^μ_i, are then defined similarly to Π^μ_i and Σ^μ_i but with respect to the weak index of a formula.

Deciding an alternation class C means deciding, given an arbitrary L_μ formula, whether it is equivalent to any formula in C.

So far only the first levels ML, Π^μ_1 and Σ^μ_1 are known to be decidable [Ott99, KW02]. Some results in automata theory [CKLV13] can be interpreted within the L_μ context to yield another decidability result: given a formula in the Π^μ_2 alternation class, it is decidable whether it is equivalent on ranked trees to a formula in the Σ^μ_2 alternation class.

2.3 Parity games

The semantics of logics are often described in anthropomorphic terms with games between a Verifier and Falsifier. For first-order logic, for example, the two players (often named Adam and Eve after their respective quantifiers $\forall$ and $\exists$) engage in a reachability game to decide whether a first-order formula holds. For L_μ, the corresponding games are infinite, and use a parity winning condition, first explicitly described by Emerson and Jutla [EJ91]. The antagonism between the players represents the duality between conjunctions and disjunctions, existential and universal quantifiers, and in the case of L_μ between greatest and least fixpoints. For an introduction to some of the games that logicians play, see Grädel 2011 [Grä11].

Informally, a parity game – the model-checking game for L_μ – is a potentially infinite two-player game between the verifier called Even, and her opponent Odd. The arena is a rooted finite graph (equivalently, a regular tree) in which each node belongs to one of the two players and has a priority. Starting at an initial position, at each turn the player to whom the current node belongs chooses the next position amongst the current node's successors. The winner of the game is decided by the parity of the highest priority seen infinitely often along a play.

To check whether a sentence Ψ holds in a structure $\mathcal{M}$, such a game can be played on an arena built from Ψ and $\mathcal{M}$. The verifying player, Even, and can win this game if and only if $\mathcal{M} \models \Psi$. This means that model-checking L_μ reduces to deciding the winner of parity games. The exact complexity of this task is a long standing open problem: while it is conjectured to be solvable in polynomial time, the best current algorithms are exponential in a function of the alternation depth of the formula. A survey of algorithms pre-dating 2011 for solving parity games can be found in [AG11]. However, this is an active area, with several new algorithms [DMPV16, BDM16, MRR16], one of which is quasi-polynomial [CJK$^+$16], published within the last year.

Definition 14. *(Parity game)* A parity game arena is $G = (V, E, v_\iota, \Omega)$ consisting of: a set V of states partitioned into those belonging to Even, V_e and those belonging to Odd, V_o; an edge relation $E \subseteq V \times V$; an initial node v_ι; and a priority assignment Ω which assigns a priority to each position. The co-domain I of Ω is said to be the index of G. Ω can always be a prefix of the natural numbers, starting at either 0 or 1.

A play in a parity game is a potentially infinite sequence of positions starting with v_ι. The winner of a finite play depends on the final position v: if $\Omega(v)$ is even, Even wins, otherwise Odd wins. For infinite plays, the winner depends on the highest priority seen infinitely often, called the dominant priority. The winner is the player of the parity of the dominant priority.

A positional (or memoryless) strategy σ for Even (and similarly for Odd) consists of a choice $\sigma(v)$ of successor at the nodes v in V_e (or V_o). A play $\pi = v_0, v_1, ...,$ where $v_0 = v_\iota$, agrees with a strategy σ if, at every position $v_i \in V_e$ (or V_o) along the play, $v_{i+1} = \sigma(v_i)$. A strategy can be seen as a tree of which each branch is a play which agrees with the strategy.

A pair of strategies, one for each player, induces a unique play. A strategy for one of the players is winning if every play that agrees with it is winning for the player. A parity game G is said to be winning for a player if that player has a winning strategy.

Parity games are positionally determined [EJ91, Mar75]: positional strategies suffice and exactly one of the players has a winning strategy from every position.

For every structure $\mathcal{M}$ and formula Ψ, there is a model-checking parity game $\mathcal{M} \times \Psi$ such that Even has a winning strategy in $\mathcal{M} \times \Psi$ if and only if $\mathcal{M} \models \Psi$. Furthermore, Even has a winning strategy from each position $s \times \phi$ of $\mathcal{M} \times \Psi$, where s is a state of $\mathcal{M}$ and ϕ is a subformula of Ψ, if and only if $s \models \phi$.

Definition 15. *(Model-checking parity game)* Let Ψ be a sentence of L_μ and Ω_Ψ a priority assignment with co-domain I on the fixpoint variables of Ψ. Then, for any labelled transition system $\mathcal{M}$, define the model-checking parity game $\mathcal{M} \times \Psi$ as follows: $\mathcal{M} \times \Psi = (V, E, v_\iota, \Omega)$ where:

- V is the set of states $s \times \phi$, where s is a state of $\mathcal{M}$ and $\phi \in sf(\Psi)$;
- V_o consists of positions $s \times \phi \wedge \psi$ and $s \times [a]\phi$ while $V_e = V \setminus V_o$;
- If $s \not\models C$, position $s \times C \wedge \phi$, where C is a conjunction of literals, is terminal and of odd priority; else it has a unique successor $s \times \phi$;
- Positions $s \times C$, where C is a conjunction of literals, are terminal and if $s \models C$, of even priority; else it is of odd priority;
- There is an edge from $s \times \phi \vee \psi$ to $s \times \phi$ and $s \times \psi$;
  an edge from $s \times \phi \wedge \psi$ to $s \times \phi$ and $s \times \psi$;
  an edge from $s \times X$ and $s \times \mu X.\phi_X$ or $\nu X.\phi_X$ to $s \times \phi_X$;
  an edge from $s \times \langle a \rangle \phi$ and $s \times [a]\phi$ to $s' \times \phi$ for every a-successor s' of s;
- v_ι the initial position is $r \times \Psi$, where r is the root of $\mathcal{M}$;
- Ω assigns $\Omega_\Psi(X)$ to positions $s \times X$; it assigns the minimal odd priority from I (or 1 if all elements of I are even) to $s \times C$ and $s \times C \wedge \phi$ when the conjunction C of literals does not hold in s; it assigns the minimal even priority of I (or 0 if all elements of I are odd) to $s \times C$ when C holds in s and to $s \times \neg P$ if P does not hold in S. Ω assigns the minimal priority of I to all other positions.

Theorem 16. *[AG11] If Ω is an order-preserving parity assignment for Ψ, then $\mathcal{M} \times \Psi$ is winning for Even if and only if $\mathcal{M} \models \Psi$.*

2.4 Alternating parity automata

There is a close connection between L_μ and alternating parity automata. The connection between the L_μ alternation depth and the indices in Rabin automata was first noted by Niwiński [Niw97]. Janin and Walukiewicz gave the first automata-theoretic semantics for L_μ in 1995 [JW95], and Wilke revisited these some years later [Wil01]. In this paper, the automata model for L_μ is stated to be alternating parity automata. To be slightly more precise, like automata for CTL [BG93], these automata are *amorphous*: they operate on unranked structures, which allow several successors via the same label, like L_μ. This distinguishes these L_μ-automata from the variety of alternating parity automata encountered in other branches of automata-theoretic literature, which operate on ranked structures: every node has exactly one successor per action label, and the branching degree is bounded by the finite number of action labels. In the literature this distinction tends to be implicit and apparent in the definitions, but rarely explicitly stated. Its significance becomes clear when we consider the index problem, both further down in this section and in detail in Chapter 4.

Here I review the definition of amorphous alternating parity automata using L_μ notation in the transition condition. This definition subsumes both the original L_μ automata [Wil01], which it generalises to multimodal L_μ, and non-amorphous alternating parity automata, as found in much of the automata theoretic literature [KV98, FMS13, AN07], which it generalises onto unranked structures. The definition of non-amorphous alternating parity automata can be retrieved by restricting the input trees to ranked trees; ignoring action labels yields the original unimodal automata [Wil01].

Throughout this thesis, L_μ-automata refer to amorphous alternating parity automata while alternating parity automata refer to the non-amorphous variation. Similarly, non-deterministic automata, discussed in Chapter 4, are non-amorphous.

Definition 17. *(L_μ-automata)* A L_μ-automaton is a tuple $A = (S, s_i, \delta, \Omega)$ where

- S is a finite set of states with an initial state $s_i \in S$;
- δ is a transition function, from states to *transition conditions*, defined below;
- $\Omega : S \to I$ is a priority function.

A transition condition is:

- $\top, \bot, P$ or $\neg P$ for $P \in Prop$;
- $s, [a]s, \langle a \rangle s$ for $s \in S$ and $a \in Act$;
- $s \vee s', s \wedge s'$ for $s \in S$.

The codomain I of Ω is the index of A. The acceptance conditions are standard [Wil01]. As with formulas, the evaluation of A boils down to a parity game of index I.

Theorem 18. *[Wil01] For every L_μ formula Ψ of index I there is an amorphous alternating parity automaton A of index I such that A accepts a structure $\mathcal{M}$ if and only if $\mathcal{M} \models \Psi$. Conversely, for any amorphous alternating parity automaton of index I, there is a L_μ formula of index I such that A accepts a structure $\mathcal{M}$ if and only if $\mathcal{M} \models \Psi$.*

The original proof of this theorem [Wil01] only considers the unimodal case, but the extension to multiple action-labels is trivial. The intuition of this transformation between automata and formulas is simple: the states of the automaton correspond to subformulas of Ψ and the transition relation is derived from the parse-tree of Ψ. The priorities in the automaton are inherited from the priority assignment of Ψ.

The index problem can be defined for any class of parity automata in the same way it is defined for L_μ: given a property, what is the least index required to express it as an automaton of the specified type? For the L_μ-automata defined above, this is of course exactly the same problem as the L_μ index problem. For non-amorphous parity automata, it is the Rabin–Mostowski index problem, which has been studied on non-deterministic [CL08], deterministic [NW05], and game automata [FMS13] for example. While the model-checking problems for L_μ and alternating parity automata reduce to each other, it is to the best of my knowledge open whether there is a reduction between the respective index problems. The distinction between the Rabin–Mostowski index problem and the L_μ index problem stems from the fact that an automaton can have a different index according to whether it is amorphous or not.

Example 19. If $[a]\psi \vee [a]\neg\psi$ is considered as a L_μ formula, or an (amorphous) L_μ-automaton, its complexity depends on ψ and can be arbitrarily high. When considered as a non-amorphous automaton – restricted to ranked trees – this property becomes trivial: there is only ever one a-successor, which will of course satisfy either ψ or $\neg\psi$, so the property is equivalent to $\top$.

The index of a property on unranked structures is higher than its index on ranked structures. Deciding its index on unranked structures only gives an upper bound to its index on ranked structures; deciding its index on ranked structures gives no guarantees at all about its index on unranked structures. The differences between the index problems for non-deterministic (non-amorphous) automata and fragments of L_μ are studied in Chapter 4. Despite these differences, some techniques can be generalised from ranked structures into the L_μ framework – for example, the decidability proof of Chapter 6 relied in part on generalising work on the Rabin–Mostowski index.

While further definitions will be introduced as required, this concludes the presentation of the foundations for discussing the L_μ index problem.

Part I

On the effects that syntactic restrictions have on the index problem.

The syntax of L_μ is itself simple, but it allows us to build formulas that quickly become difficult to make sense of, for humans and algorithms alike. Syntactic restrictions can impose some semblance of structure onto formulas and make them both more manageable for the human reader and easier to manipulate within proofs. In this first part, I study two syntactic fragments of L_μ and how they contribute to understanding complexity in L_μ: in particular, how does turning formulas into these fragments affect the alternation depth of the formula and how does the index problem for these fragments differ from the L_μ index problem?

One syntactic restriction – *disjunctive form* – is of particular importance throughout this thesis: most of the decision procedures in Part II as well as theorems in Part III, rely on the properties of disjunctive formulas. The next chapter is therefore dedicated to studying this fragment. Its main contributions are:

- A brief survey of the various properties of disjunctive formulas;
- A detailed rewriting of the transformation of an arbitrary L_μ formula into disjunctive form [JW95, Wal95a];
- A proof that any Π_2^μ formula is equivalent to a disjunctive Π_2^μ formula;
- A proof that the transformation into disjunctive form can blow up the alternation depth of a formula an arbitrary amount.

These last two results are, to the best of my knowledge, novel. They do however find parallels in automata theory: non-deterministic Büchi automata are as expressive as alternating Büchi automata [MS95] while the same is not known to be true of co-Büchi automata. This is one of many parallels between disjunctive formulas and non-deterministic automata, and in fact disjunctive form was initially conceived as a logical counterpart to non-deterministic automata. However, the exact relationship between disjunctive L_μ and non-deterministic automata is more subtle, and Chapter 4 clarifies its details by introducing the *non-deterministic* fragment of L_μ.

Indeed, disjunctive formulas are defined on unranked structures, while non-deterministic automata operate on ranked structures only. When restricted to ranked structures, disjunctive formulas are more powerful: they can express properties using a smaller index than a non-deterministic automaton. Non-deterministic automata can be generalised onto unranked structures to match exactly a sub-fragment of disjunctive L_μ: non-deterministic L_μ. Any non-deterministic automaton is equivalent to a non-deterministic formula of the same index and vice-versa. This fragment allows the detailed study of the relationship between non-deterministic automata and L_μ formulas, a relation often alluded to, but rarely dwelled on, in the literature. Non-deterministic formulas also boast a curious property which neither disjunctive L_μ nor non-deterministic automata have: semantic equivalence implies similarity of tableaus. As a result, I propose this fragment as an interesting fragment for which solving the index problem may be within reach. The contributions of Chapter 4 can be summarised as:

- The introduction of the *non-deterministic* fragment of L_μ as a generalisation of non-deterministic automata onto unranked structures;
- A transformation of any disjunctive formula into a formula in this fragment which agrees with the original formula *on ranked structures* – this is an alternative, logic-centric proof that alternating parity automata are simulated by non-deterministic automata [MS95];
- A proof that the non-deterministic L_μ is less powerful than disjunctive L_μ;
- A reduction of the index problem for non-deterministic L_μ to a question of finding the least parity assignment in any sufficiently similar tableau.

The aim is to clarify how L_μ and automata-theoretic literature compare with respect to their respective index problems, highlighting some of the difficulties of moving between the two frameworks. In particular, while the objects under scrutiny may be practically identical, as is the case with non-deterministic automata and formulas, the differences in the frameworks they are defined in mean that they have distinct index problems which can behave very differently indeed.

Chapter 3

Disjunctive L_μ

In this chapter I introduce and study the disjunctive fragment of L_μ. This fragment is as expressive as the whole L_μ, but tends to be better behaved. Specifically, disjunctive form restricts the use of conjunctions, so that universal branching – Odd's choices – only stems from modalities. As a result, in the model-checking game of a disjunctive formula, Even can use strategies which only agree with one play per branch.

This fragment was first introduced by Janin and Walukiewicz in the context of describing L_μ semantics in an automata-theoretic setting [JW95]. Its restriction on conjunctions was devised as a way to mimic the non-deterministic subset of alternating parity automata. It is however more powerful than non-deterministic automata, in the sense that it can describe properties of ranked structures using a smaller index than an equivalent non-deterministic automaton would require. Disjunctive form was most notably used in Walukiewicz' proof of completeness of Kozen's axiomatisation [Wal95a]. Since then however, disjunctive form has received relatively little attention compared to its cousin, the non-deterministic automaton.

In this thesis, the structural properties of disjunctive formulas are repeatedly used for studying optimizations for L_μ formulas. This chapter lays out the groundwork by discussing these properties and studying how turning a formula into disjunctive form affects its index.

Here is an intuition for how a restriction on conjunctions can help tackle the index problem. Consider the following formula:

$$(\Box\Box A) \wedge (\Diamond\Diamond(\neg A \wedge \psi) \vee B)$$

In this example, ψ could be any formula, perhaps one of high complexity. It is easy to see that ψ is unnecessary: a strategy for Even that reaches $\neg A \wedge \psi$ at a state v also reaches A at the same state. Then Odd always wins by choosing whichever one of A or $\neg A$ does not hold at v. The formula therefore simplifies to $\Box\Box A \wedge B$. Again, supposing ψ is of arbitrary complexity, consider this formula:

$$(\mu X.\Box X) \wedge \psi$$

The first clause, $\mu X.\Box X$, states that the structure does not have infinite branches. On such structures, μ and ν are interchangeable (for guarded formulas), so the above formula is semantically in Σ^μ_1.

These are just some examples of the ways in which conjunctions can obfuscate redundancies. Given how difficult L_μ formulas can be to parse, these situations are not always easy to identify. Turning a formula into disjunctive form can eliminate some complexity, and failing that, it makes the formula easier to manipulate and reason about. This will be further studied in later chapters; the present chapter focuses on the properties of disjunctive form and the transformation into it.

After the formal introduction of the disjunctive fragment in Section 3.1, Section 3.2 provides a brief survey of some of the pleasing properties of disjunctive form. In particular it establishes some of the fundamental properties of disjunctive form which will be used throughout this thesis.

The transformation into disjunctive form is described in 3.3. It is a detailed rewriting of the original proof [JW95] that all L_μ formulas are equivalent to disjunctive ones, spelling out some details originally omitted.

Finally, Section 3.4 considers the effect of this transformation on the alternation depth of formulas. It establishes that while Π^μ_2 formulas remain Π^μ_2 formulas when turned into disjunctive form, in general the transformation can cause an arbitrary increase or decrease in the number of alternations. The latter result was published in [Leh15].

Discussion of the exact relationship between disjunctive L_μ and non-deterministic automata is left to the next chapter.

3.1 Definitions

Throughout this chapter, and wherever this thesis uses disjunctive form, the modal operator $\xrightarrow{a}\mathcal{F}$ (or $\rightarrow\mathcal{F}$), introduced in Section 2.1.2, replaces the traditional modalities $\langle a\rangle$ and $[a]$ (or $\Diamond$ and $\Box$). Recall that given a set of formulas $\mathcal{F}$, the formula $\xrightarrow{a}\mathcal{F}$ is short for $(\bigwedge_{\psi\in\mathcal{F}}\langle a\rangle\psi) \wedge [a]\bigvee_{\psi\in\mathcal{F}}\psi$ – that is to say, every formula in $\mathcal{F}$ is realised by at least one a-successor and every a-successor realises at least one of the formulas in $\mathcal{F}$. Note that $\langle a\rangle\psi$ can be expressed with $\xrightarrow{a}\{\psi,\top\}$ and $[a]\psi$ can be expressed with

$\xrightarrow{a}\psi \vee \xrightarrow{a}\{\}$: this notation does not affect the expressiveness of the logic. $\xrightarrow{a}\{\}$ is true for states without a-successors while $\xrightarrow{a}\{\bot\}$ is false in all states.

Informally, disjunctive formulas are built from subformulas, called *modal* subformulas, of the form $P \wedge \bigwedge_{a\in A} \xrightarrow{a}\mathcal{F}_a$, or in the unimodal case $P \wedge \rightarrow\mathcal{F}$, where P is a conjunction over a set of literals, $A \subseteq Act$ is a finite set of actions and $\mathcal{F}$ is a set of formulas. These subformulas are joined by disjunctions and fixpoint operators bind fixpoint variables as usual.

Definition 20. *(Disjunctive L_μ)* The set of disjunctive formulas of (unimodal) L_μ is the smallest set $\mathcal{D}$ satisfying:

- Literals, $\bot$ and $\top$ and fixpoint variables are in $\mathcal{D}$;
- If $\psi \in \mathcal{D}$ and $\phi \in \mathcal{D}$ then $\psi \vee \phi \in \mathcal{D}$;
- If P is the conjunction of a finite set of literals, and $\mathcal{F} \subseteq \mathcal{D}$ is a finite set of disjunctive formulas, then $P \wedge \rightarrow\mathcal{F}$ is in $\mathcal{D}$, and so is P.
- $\mu X.\psi \in \mathcal{D}$ and $\nu X.\psi \in \mathcal{D}$ as long as $\psi \in \mathcal{D}$ and X is guarded in ψ.

For multimodal disjunctive form, the rule for modal subformulas is:

- $P \wedge \bigwedge_{a\in A} \xrightarrow{a}\mathcal{F}_a$ where each $\mathcal{F}_a$ is a finite set of disjunctive formulas, P is a finite conjunction of literals, and $A \subseteq Act$ is a finite set of actions.

Note that a fixpoint variable X can only appear positively and does not appear in a context $X \wedge \alpha$ for any formula α. As a result, if $\mu X.\phi(X)$ is in disjunctive form then $\phi(\mu X.\phi(X))$ is also in disjunctive form.

Definition 21. A disjunctive formula is said to be S-complete with respect to some set of propositions $S \subset Prop$ if for every modal subformula $P \wedge \rightarrow B$, the conjunction P contains for each proposition L in S either L or its negation. A formula is said to be complete if it is complete with respect to the set of propositions that appear in it.

In the multimodal case, we can also talk about A-completeness with respect to some finite set of actions $A \subseteq Act$: every modality must be of the form $\bigwedge_{a\in A} \xrightarrow{a}B_a$ for the fixed set A.

Example 22.

$$(\mu X.P \vee \rightarrow\{X\}) \wedge (\mu Y.S \vee \rightarrow\{Y\})$$

This formula expresses the reachability of both P and S, and is not in disjunctive form. It is equivalent to the following disjunctive formula:

$$\mu X.(P \wedge S) \vee (P \wedge \rightarrow\{\mu Y.S \vee \rightarrow\{Y\}\}) \vee (S \wedge \rightarrow\{\mu Z.P \vee \rightarrow\{Z\}\}) \vee \rightarrow\{X\}$$

While the original formula just states two distinct requirements, the disjunctive formula spells out the local behaviour generated by the interaction between the conjuncts: one must reach a point where either P and S are seen simultaneously, or one of P or S is seen first, after which the other must still be reached.

A disjunctive formula can be seen as a tree with back edges (see Example 2) with two types of inner nodes: modal nodes $P \wedge \rightarrow\mathcal{F}$, and disjunctive nodes $\bigvee\mathcal{F}$ – the set of successors for both types of nodes is $\mathcal{F}$, the immediate subformulas. The leaves are either just P, a conjunction of literals, or a fixpoint variable X with a back edge pointing to the formula binding X.

Write $P{\rightarrow}\mathcal{F}$ for $P \wedge \rightarrow\mathcal{F}$, where P is a conjunction of literals and $\mathcal{F}$ is a set of formulas. Recall that conjunctions bind more closely than disjunctions. This notation reduces the length of formulas, and potential confusion about operator precedence. It is also meant to distinguish the conjunctions that are allowed in disjunctive formulas from arbitrary conjunctions by emphasising the notion that in disjunctive form, sets of literals "guard" modalities: as long as the conjunction of literals P is true at a state, Even can play to the modality $\rightarrow\mathcal{F}$. Similarly, when multiple modalities are at play, write $P \bigwedge_{a\in A} \xrightarrow{a}\mathcal{F}_a$ for $P \wedge \bigwedge_{a\in A} \xrightarrow{a}\mathcal{F}_a$. Assume that in such formulas $A \subseteq Act$ is a finite set of actions – this will be the case throughout, and will not be explicitly stated each time.

3.2 Disjunctive L_μ: a well-behaved fragment

So far, we have defined the disjunctive fragment of L_μ. This section presents a brief survey of the various ways in which this fragment is well-behaved. The first subsection presents the fundamental property of disjunctive formulas which will be used throughout the later chapters; the subsequent sections give an overview of some other properties of disjunctive formulas.

3.2.1 Well-behaved strategies

Disjunctive form affects the dynamics of the model-checking parity games. For L_μ formulas in general, a strategy σ for either player in a model checking game $\mathcal{M} \times \Psi$ will generate for every node v of the structure $\mathcal{M}$ a set of subformulas ϕ of Ψ such that σ reaches $v \times \phi$ in $\mathcal{M} \times \Psi$. Given a disjunctive formula Ψ and a strategy σ for Even in a model checking game $\mathcal{M} \times \Psi$, this set reduces to a singleton set for all nodes that σ reaches, provided $\mathcal{M}$ and σ satisfy some simple assumptions. In other words, Even's strategy fixes which subformula can be seen at which position and Odd's strategy only affects which positions the play will visit.

To obtain this property, Even has to avoid playing to the same state at two distinct formulas. Recall that at a modality $\rightarrow\{\phi, \psi\}$, Odd can either choose a successor, in which case Even chooses one of $\{\phi, \psi\}$, or he can choose one of $\{\phi, \psi\}$ himself and let Even pick the successor. Even's strategy should choose a distinct successor to pair up with each of ϕ and ψ. The next definition formalises this notion.

Definition 23. *(Well-behaved strategies and structures)* Let Ψ be a disjunctive formula, $\mathcal{M}$ a structure represented as a tree, and σ a strategy in the model-checking game $\mathcal{M} \times \Psi$. Then, $\mathcal{M}$ and σ are *well-behaved* with respect to each other if for each position $s \times \xrightarrow{a}\mathcal{B}$ reachable with σ, and each $\phi \in \mathcal{B}$, the state s has distinct a-successors s_ϕ such that σ reaches s_ϕ at $s_\phi \times \psi$ for no other $\psi \in \mathcal{B}$ than ϕ, and every a-successor of s is reached at a position $s \times \phi$ for exactly one $\phi \in \mathcal{B}$. That is to say, whenever Odd chooses $\phi \in \mathcal{B}$ at a position $s \times \xrightarrow{a}\mathcal{B}$, Even chooses s_ϕ and whenever Odd chooses s_ϕ, Even chooses ϕ . Note that s may have additional a-successors beyond s_ϕ for each $\phi \in B$.

In other words, a well-behaved strategy is one which agrees with only one play per branch.

Given a structure $\mathcal{M}$ and strategy σ in $\mathcal{M} \times \Psi$, the structure $\mathcal{M}$ is bisimilar to a structure $\mathcal{M}'$ such that $\mathcal{M}'$ and the strategy that σ induces on $\mathcal{M}' \times \Psi$ are well-behaved. This bisimilar structure is obtained by simply duplicating successor states: If σ reaches both $s' \times \phi$ and $s' \times \psi$ from $s \times \xrightarrow{a}\mathcal{B}$, then duplicate s' into two a-successor states s'_ψ and s'_ϕ, each the root of a duplicate of the subtree rooted at s', and let σ play to $s'_\phi \times \phi$ and $s'_\psi \times \psi$.

Fact 24. If a formula Ψ is in disjunctive form, without loss of generality we can assume any pair of structure $\mathcal{M}$ and Even's strategy in $\mathcal{M} \times \Psi$ to be well-behaved.

Definition 25. *(Odd position tree)* If σ is a strategy for Even, the Odd position tree for σ consists of the positions $(s \times P \bigwedge_{a\in A} \xrightarrow{a}\mathcal{B}_a)$ belonging to the Odd player which are reachable by a play respecting σ. One step in this tree corresponds to the odd player choosing an action a, and either an a-successor or a formula in $\mathcal{B}_a$, followed by as many moves dictated by σ as necessary to reach the next position belonging to the Odd player.

Lemma 26. *If Ψ is in disjunctive form and a structure $\mathcal{M}$ represented as a tree is well-behaved with respect to a strategy σ in $\mathcal{M} \times \Psi$, then each state of $\mathcal{M}$ appears in the Odd position tree of σ at most once.*

Proof. We prove that for a well-behaved strategy σ, each position $t \times \psi$ in the Odd position tree only has one successor position per successor state s of t. Each position in the Odd position tree is of the form $(t \times P \bigwedge_{a\in A} \xrightarrow{a}\mathcal{B}_a)$ and its successors correspond to the possible choices of Odd: either Odd evaluates P on t and ends the game, or he picks a formulas from some $\mathcal{B}_a$ in which case Even picks an a-successor to evaluate that formula at, or Odd chooses an a successor for some a, in which case Even picks a formula from $\mathcal{B}_a$ to evaluate at the successor state. Since σ is well-behaved, if Odd chooses either $\phi \in \mathcal{B}_a$ or the a-successor t_ϕ for some a, the game goes to $(t_\phi \times \phi)$ so each successor t_ϕ only appears in one successor position of $(t \times \psi)$ in the Odd position tree. Furthermore, if t has some a-successor t' which is distinct from t_ϕ for any $\phi \in \mathcal{B}_a$, then the play only moves to a position $t' \times \phi$ if Odd chooses t'. Since σ uniquely dictates ϕ from $\mathcal{B}$ when Odd chooses a successor, a single position $t' \times \phi'$ appears in the Odd position tree.

Since $\mathcal{M}$ is represented as a tree, and disjunctive formulas are guarded, each of its states can only appear once in the Odd position tree. □

Corollary 27. *The consistent assignment induced by Even's winning strategy σ in a parity game $\mathcal{M} \times \Psi$ that assigns to each state of the structure $\mathcal{M}$ the subformulas of Ψ such that σ reaches $(s \times \psi)$, assigns at most one Odd subformula $P \bigwedge_{a \in A} \xrightarrow{a} \mathcal{B}$ to each state.*

This is a fundamental characteristic of disjunctive formulas: to prove that $\mathcal{M}$ satisfies a disjunctive formula Ψ it is sufficient to prove that each state satisfies one formula $P \bigwedge_{a \in A} \xrightarrow{a} \mathcal{B}$.

This property is reminiscent of non-deterministic automata for which runs also visit each node of the ranked input tree at a single state. In contrast, disjunctive formulas lose this property if the input space is restricted to ranked structures. Since successors can no longer be duplicated at will, from a modal position $s \times \xrightarrow{k} \{\phi, \psi\}$, any strategy for Even has to reach the k-successor of s at both ϕ and ψ. The next chapter will discuss the relationship between disjunctive L_μ and non-deterministic automata in more detail, and will discuss a fragment of disjunctive L_μ which preserves well-behaved strategies on ranked structures.

Finally, note that while the added structure of disjunctive form simplifies the parity games by reducing the number of formulas reached at a state to one, it comes at a cost. The symmetry between Even and Odd is lost and dualisation of results is no longer self-evident. For example the negation of a disjunctive Σ^μ_n formula is not necessarily a disjunctive Π^μ_n formula.

Instead, define *co-disjunctive form*, the exact dual of disjunctive form, which has the dual properties of disjunctive form: the Odd player has well-behaved strategies which only agree with one play per branch. This co-disjunctive form will only be used in Chapter 8.

3.2.2 Characteristic formulas

The characteristic formula of a structure $\mathcal{M}$ is a formula which is true in exactly the structures bisimilar to $\mathcal{M}$. In other words, it is a formula describing a bisimulation class. Although the modal fragment of L_μ is strong enough to separate any pair of structures that are not bisimilar [HM85], it is not expressive enough to formulate characteristic formulas. Such formulas can be derived in L_μ [Ste89, MO98]. Since L_μ is bisimulation invariant, this is the most restrictive type of formula there is, excluding $\bot$, in the sense that the property expressed corresponds to exactly one bisimulation class of structures. It turns out that the natural way of building characteristic formulas for a regular tree yields a disjunctive Π_1^μ formula without disjunctions. Furthermore, any complete disjunctive Π_1^μ formula without disjunctions (except $\bot$) is a characteristic formula.

In this section, take *complete* to mean completeness with respect to both sets of literals and actions, *i.e.* modal conjunctions are of the form $\bigwedge_{a\in Act} \mathcal{F}_a$. The bisimulation relation in question is of course the one which preserves the set of literals appearing in the formula or structure at hand.

Definition 28. *(Characteristic formulas)* A sentence ψ of L_μ is the characteristic formula of a structure $\mathcal{M}$ if for any structure $\mathcal{N}$, it is the case that $\mathcal{N} \models \psi$ if and only if $\mathcal{N}$ and $\mathcal{M}$ are bisimilar.

It is easy to see that any complete disjunctive formula without disjunctions or least fixpoints is a characteristic formula. Its model is essentially the parse-tree of the formula: every state corresponds to a modal subformula $P \bigwedge_{a\in\mathcal{A}} \xrightarrow{a} \mathcal{B}_a$; P dictates the propositions which hold at the state; there is an a-edge from the state corresponding to $P \bigwedge_{a\in\mathcal{A}} \xrightarrow{a} \mathcal{B}_a$ to the ones corresponding to the next modal subformula in each element of every $\mathcal{B}_a$. A greatest fixpoint variable induces a back-edge to the state corresponding to the next modal position after its binding.

It is equally easy to see that any structure that satisfies such a formula is bisimilar to the above structure. Conversely, given a structure, its characteristic formula is built in the obvious way, as detailed in the next definition.

Definition 29. *(The characteristic formula of $\mathcal{M}$)* Any finitely representable structure is also finitely representable by a tree with back edges so let $\mathcal{M}$ be a structure, represented finitely by a tree with back edges. Let $\mathcal{P}$ be the set of propositional variables under consideration – usually, but not necessarily, this is simply the set of propositions assigned to the labels of $\mathcal{M}$. Given a node n, write $\mathcal{A}(n)$ for the conjunction of propositional variables from $\mathcal{P}$ true at n together with the negations of propositional variables from $\mathcal{P}$ which do not hold at n.

To build a characteristic formula, first label the leaves n without back edges with the formula $\mathcal{A}(n) \bigwedge_{a\in Act} \xrightarrow{a} \{\}$. If a leaf m is the source of a-back-edges $e \in E_a$ leading to some node n_e, then label it with $\mathcal{A}(m) \bigwedge_{a\in Act} \xrightarrow{a} \{X_{n_e} | e \in E_a\}$. If an inner node n has successors n_i, themselves labelled with ψ_i, and is not the target of a back edge, label n with the formula $\mathcal{A}(n) \bigwedge_{a \in Act} \xrightarrow{a} \bigcup_i \{\psi_i\}$. If n is the target of a back edge, then the formula at n must also bind the fixpoint variable X_n: the label of n is then $\nu X_n.\mathcal{A}(n) \bigwedge_{a \in Act} \xrightarrow{a} \bigcup_i \{\psi_i\}$. We only use greatest fixpoint bindings since the aim is to express the infinite behaviour of the structure rather than impose finiteness. Note that νX_n may bind several instances of X_n, if there are several back-edges to n. Since there are no μ variables involved, the order of binding is irrelevant.

The label of the root of $\mathcal{M}$, which is a sentence of L_μ is then the characteristic formula $\psi_\mathcal{M}$ of $\mathcal{M}$.

By construction, this formula is in $\mathcal{P}$-complete disjunctive form and does not contain any μ-operations nor disjunctions. To see that this is indeed a characteristic formula of $\mathcal{M}$, it suffices to observe that $\mathcal{M}$ is exactly the structure generated by the construction of a model from a characteristic formula.

Let us now consider briefly how the addition of μ and $\vee$ to the syntax of characteristic formulas increases the complexity of the class of structures described. First note that adding μ-operators without adding disjunctions is nonsensical: any μ-bound formula in disjunctive form without disjunctions induces a parity game where the Odd player can always force the game from $\mu X.\phi$ to X, yielding a play in which the highest priority seen infinitely often is odd. Therefore such subformulas are equivalent to $\bot$.

On the other hand, we can add disjunctions to formulas without μ-operators to expand the class of accepted structures. Taking the disjunction of two characteristic formulas $\psi_\mathcal{M}$ and $\psi_\mathcal{N}$ simply produces the union of the two bisimulation classes of structures. Introducing disjunctions within formulas has more subtle effects. In a modal formula, that is to say a formula without fixpoints, introducing a disjunction within a characteristic formula simply allows two different completions of the model. However, as soon as the disjunction is within the scope of a fixpoint operator, be it μ or ν, we get a potentially infinite union of bisimulation classes satisfying the formula. In the simplest case, some subformula $\phi(X)$ of a characteristic formula $\psi_\mathcal{M}$ containing a fixpoint variable X is replaced with $\phi(X) \vee \psi_\mathcal{N}$ where ψ_N is a characteristic formula. Then the class of structures satisfying the new formula is the infinite union which contains not only the bisimulation class of structures characterised by $\psi_\mathcal{M}[\psi_\mathcal{N}/\phi(X)]$ and $\psi_\mathcal{M}$, but also those characterised by each finite approximation of the fixpoints. Finally, with the introduction of alternations between μ and ν we get a full hierarchy of L_μ expressible properties.

Describing structures as formulas also allows us to express relations between structures and formulas as formulas. For instance $\mathcal{M} \models \phi$ if and only if $\psi_\mathcal{M} \wedge \neg\phi$ is unsatisfiable, *i.e.* if and only if $\psi_\mathcal{M}$ implies ϕ.

3.2.3 Satisfiability and synthesis

While the satisfiability of $\psi \vee \phi$ stems directly from the satisfiability of ϕ and ψ, deciding the satisfiability of $\psi \wedge \phi$ is more involved. Pushing conjunctions as far as possible, as the transformation into disjunctive form does, makes any interactions between conjuncts explicit. This makes deciding whether a disjunctive formula is satisfiable easier – of linear complexity, to be precise.

Indeed, a disjunctive formula Ψ is satisfiable if and only if the formula in which fixpoints are approximated to their initial level – that is to say ν-bound and μ-bound fixpoint variables are replaced with $\top$ and $\bot$ respectively – is satisfiable. Formally, disjunctive Ψ is equi-satisfiable with $\Psi[\bot/X;\top/Y]_{X\in\mu Var;Y\in\nu Var}$, where μVar and νVar are the least and greatest fixpoint variables of Ψ respectively. Since this is a modal formula, its satisfiability is linear-time decidable and consists of finding a strategy σ for Even – that is to say choices at disjunctions – which reaches neither $\bot$ nor an inconsistent conjunction of literals.

For synthesis, the strategy σ induces a characteristic formula on Ψ: at each disjunction eliminate all disjuncts except the one σ chooses; complete modalities $\bigwedge_{a\in A} \xrightarrow{a} \mathcal{F}_a$ to $\bigwedge_{a\in A} \xrightarrow{a} \mathcal{F}_a$ where $\mathcal{F}_a = \{\}$ for $a \notin A$. Since σ only reaches ν-bound variables, we can build a structure from this formula as detailed in the previous section.

3.2.4 Unsatisfiability in disjunctive formulas

Having discussed satisfiable disjunctive formulas in the previous section, this section briefly considers unsatisfiable ones.

Turning a formula into disjunctive form can be considered as an exponential complexity pre-processing step which introduces some order into a formula whilst weeding out some redundancies. For example, a L_μ formula may have some propositions A and $\neg A$ buried deep inside the formula so that in every structure, the Even player must prove both A and $\neg A$ at the same state. Such a formula is of course unsatisfiable, but the formula may be complex enough that it is not entirely obvious that this is the case. However, when such a formula is turned into disjunctive form, this will produce a subformula with a conjunct $A \wedge \neg A$. It is reasonable to require that any conjunct of propositional variables appearing in a formula must be consistent or rewritten as $\bot$, that the subformula $\rightarrow\{\bot\}$ should just be written $\bot$ and that any subformula $\phi \vee \bot$ should be simplified to ϕ while $\bot \wedge \phi$ should be simplified to $\bot$. As a result, after these trivial simplifications, the only disjunctive formula with $\bot$ in it is $\bot$ itself.

The following lemma is a curious consequence of the way $\bot$ gets simplified away in disjunctive form.

Lemma 30. *Any disjunctive formula Ψ in Π_1^μ without inconsistent conjunctions of literals, or $\bot$, is satisfiable.*

Proof. In the previous section, satisfiability of a disjunctive formula was reduced to finding a strategy for Even which does not reach $\bot$, a least fixpoint or any inconsistent conjunction of literals. A disjunctive Π_1^μ formula in which the previously mentioned syntactic simplifications have eliminated $\bot$ and inconsistent conjunctions, is free of all three, so any strategy for Even will do. □

As a result, for disjunctive *ML* and Π_1, satisfiability is a trivial constant-time affair once inconsistent conjunctions and $\bot$ have been simplified away. For other formulas, unsatisfiability can be caused by μ-operators that are syntactically inescapable, for example: $\mu X.\rightarrow\{X\}$ or $\nu X.\mu Y.(\rightarrow\{Y,X\} \vee \rightarrow\{Y\})$

In general, if the formula is seen as a parity game on subformulas where disjunctions belong to the Even player and modalities to the Odd player, an unsatisfiable formula is one where the Even player is not able to escape a cycle where the most significant fixpoint is a μ-operator. That is to say, an unsatisfiable disjunctive formula is one where the Odd player wins the natural parity game played on the formula, where a boolean subformula is winning for Even if it is satisfiable, and for Odd otherwise.

3.3 The transformation into disjunctive form

So far this chapter has considered properties of disjunctive formulas. This section shifts the focus onto how any non-disjunctive formulas can be turned into a disjunctive one. It is a detailed rewriting of the original proof from [JW95] that disjunctive form is as expressive as L_μ.

The transformation first takes the tableau of a formula, then finds a finite representation of the tableau from which a disjunctive formula with the same tableau (and therefore semantics) can be extracted. Informally, the tableau groups formulas into sets such that a strategy for Even in any model-checking game of the original formula Ψ reaches states at exactly the formulas in one of these sets. This first step, constructing the tableau of a formula, is perhaps the easier step: applying a set of tableau-rules yields an infinite (but regular, and therefore finitely representable) tree labelled with sets of formulas. The intuition in this step is that conjunctions are pushed into the formula, and only disjunctions and modalities cause branching. This first step alone gives a good idea of what the disjunctive counterpart of a formula will look like.

The second step turns a tableau back into a formula. The intuition is to abstract away from the subformulas of Ψ that the tableau is built around and retrieve the semantics of Ψ from just the propositional variables in the tableau and the parity of infinite paths. The most delicate operation along this process consists of deciding the appropriate fixpoint structure for the disjunctive formula. Once this is done, the final formula is easy to generate.

3.3.1 From formula to tableau

Definition 31. *(Tableau)* A tableau $\mathcal{T} = (T, L)$ for a formula Ψ consists of a potentially infinite tree T of which each node n has a label $L(n) \subseteq sf(\Psi)$. The labelling respects the following tableau rules, and the modal rule is applied only when it is the only choice.

$$\frac{\{\Gamma, \phi, \psi\}}{\{\Gamma, \psi \wedge \phi\}}\ (\wedge) \qquad \frac{\{\Gamma, \phi\} \qquad \{\Gamma, \psi\}}{\{\Gamma, \psi \vee \phi\}}\ (\vee) \qquad \frac{\{\Gamma, \phi\}}{\{\Gamma, \sigma X.\phi\}}\ (\sigma)\ \sigma \in \{\mu, \nu\}$$

$$\frac{\{\Gamma, \phi_X\}}{\{\Gamma, X\}}\ (X) \text{ where } \phi_X \text{ is the binding formula for } X$$

$$\frac{\{\psi\} \cup \{\bigvee \mathcal{F} | \rightarrow \mathcal{F} \in \Gamma, \mathcal{F} \neq \mathcal{F}'\} \text{ for every } \rightarrow \mathcal{F}' \in \Gamma, \psi \in \mathcal{F}'}{\{\Gamma\}}\ (\rightarrow)$$

For the multi-modal case, the $(\rightarrow)$ rule takes one a-successor node for each a that appears in the modal formulas of Γ – let these be $A_\Gamma \subseteq Act$. The a-successor must respect the following rule, for each a.

$$\frac{\{\psi\} \cup \{\bigvee \mathcal{F}_a | \xrightarrow{a} \mathcal{F}_a \in \Gamma, \mathcal{F}_a \neq \mathcal{F}'_a\} \text{ for every } \xrightarrow{a} \mathcal{F}'_a \in \Gamma, \psi \in F'_a, a \in A_\Gamma}{\{\Gamma\}}\ (\rightarrow)$$

Note that the order of applications of the tableau rules is non-deterministic: strictly speaking, a formula has more than one valid tableau. However, recall that the the modal rule is only applied if no other rule is applicable. This rule synchronises different tableaus of the same formula: irrespectively of the order in which rules are applied, a modal position will have the same set of next-step modal descendants. This means that a formula has a unique tableau, up to a notion of tableau equivalence – this is further discussed in section 3.3.1. Meanwhile, call nodes where the modal rule is applied *modal nodes* and note that they are the nodes fundamental to the structure of a tableau. The literals of a modal node are the literals which appear in the node's label; they also contribute to the semantics of the tableau.

Tableaus tend to be large and difficult to parse. In order to fit them onto a page and make parsing them easier, I will take some liberties when writing them out: The out-most curly brackets (present in Example 32) which indicate that the formulas at each node form a set can be done away with. Several non-modal steps may collapse into one, as done in Example 32, if the steps are of no particular interest. The type of

$$
\dfrac{
\dfrac{
\{A\,,\,B\} \qquad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{\dfrac{\mathbf{1}}{\{\to\{Y\}\}}\,(\to) \qquad \{B\}}{\{B \lor \to\{Y\}\}}
}{\{\nu Y.B \lor \to\{Y\}\}}\,(\nu)
}{\{Y\}\ (\mathbf{1})}\,(Y)
}{\{A\,,\,\to\{Y\}\}}\,(\to)
}{}
}{\{A\,,\,B \lor \to\{Y\}\}}\,(\lor)
\qquad
\dfrac{
\dfrac{
\dfrac{
\dfrac{
\dfrac{\dfrac{\mathbf{2}}{\{\to\{X\}\}}\,(\to) \qquad \{A\}}{\{A \lor \to\{X\}\}}\,(\lor)
}{\{\mu X.A \lor \to\{X\}\}}\,(\mu)
}{\{X\}\ (\mathbf{2})}\,(X)
}{\{B\,,\,\to\{X\}\}}\,(\to)
\qquad
\dfrac{\dfrac{\mathbf{3}}{\{Y\,,\,X\}}\,\{(X{,}Y)}{\{\to\{X\}\,,\,\to\{Y\}\}}\,(\to)
}{\{\to\{X\}\,,\,B \lor \to\{Y\}\}}\,(\lor)
}{
\dfrac{
\dfrac{\{A \lor \to\{X\}\,,\,B \lor \to\{Y\}\}}{\{\mu X.A \lor \to\{X\}\,,\,\nu Y.B \lor \to\{Y\}\}\ (\mathbf{3})}\,(\mu{,}\nu)
}{\{\mu X.A \lor \to\{X\} \land \nu Y.B \lor \to\{Y\}\}}\,(\land)
}\,(\lor)
$$

Figure 3.3.1: Tableau for $(\mu X.A \lor \to\{X\}) \land \nu Y.B \lor \to\{Y\}$

the applied rule will not always be spelt out. The boldfaced indices indicate points at which the tableau starts repeating itself and can be read as back-edges. I will also keep to unimodal examples, as the multimodal cases for this chapter are all straightforward generalisations of the unimodal case.

Example 32. Figure 3.3.1 shows a simple example of a tableau.

In this example, the formula is true if all branches reach A and are either infinite or reach B. Even's strategy in the disjunctive formula that will stem from this tableau will dictate for every branch in a structure whether she is attempting to show that A and B are reached simultaneously, whether A or B is reached first and then the other, and whether the branch is infinite or eventually reaches B.

The tableau construction has the effect of grouping subformulas into the labels of nodes. The significance of this grouping is that we can interpret a strategy σ of the even player in a parity game $\mathcal{M} \times \Psi$ as an assignment of a node n of the tableau to each state s of $\mathcal{M}$. The formulas at the node correspond exactly to the formulas the strategy reaches at s.

The intention of the tableau is to represent the semantics of a formula in a syntax-independent way. Thus, only data necessary to retain the semantics of the formula is important; its particular syntax is irrelevant. The semantics of the formula are encoded by the branching structure of the tableau, the literals at modal positions and the parity of infinite paths. The latter corresponds to whether in the original formula a strategy for Even that sees the sets of formulas on a path of the tableau is winning.

In the example, a strategy which sees the rightmost path (**3**) infinitely often corresponds to a play where neither A nor B is seen infinitely often. It is losing for Even, so this infinite path is of odd parity. The path that sees the node marked **1** infinitely often on the other hand is of even parity: A has been seen, and the path is infinite.

The next definitions formalise the parity of infinite paths. Informally, a trace is just a sequence of formulas along a path and its parity is defined as usual, by the parity of the most significant fixpoint seen infinitely often.

Definition 33. *(Traces)* Given an infinite branch in a tableau, that is to say a sequence $n_0n_1...$ of nodes starting at the root, where n_{i+1} is a child of n_i, a *trace* on it is an infinite sequence $f_0f_1...$ of formulas satisfying the following: each formula is taken from the label of the corresponding node, $f_i \in L(n_i)$ for all $i \geq 0$; if f_i is not the formula that the tableau rule from n_i to n_{i+1} acts on, then the formulas f_i and f_{i+1} are identical; if the tableau rule from n_i to n_{i+1} is a disjunction, conjunction, of fixpoint binding elimination acting on f_i, then f_{i+1} is an immediate subformula of f_i; if the tableau rule from n_i to n_{i+1} is a fixpoint regeneration acting on the fixpoint variable f_i, then f_{i+1} is the binding formula for f_i. A trace is a μ-trace if the most significant fixpoint variable seen infinitely often on it is a μ-variable.

Since labels are to be thought of as conjuncts, and fixing a strategy for Even only fixes which branch of the tableau is taken at disjunctive formulas, it is up to the Odd player to decide which trace is played on an infinite path. Following this intuition, it is sufficient for an infinite path in a tableau to allow one μ-trace for the infinite path to be winning for the Odd player.

Definition 34. *(Parity of infinite paths)* The parity of an infinite path in a tableau is said to be even if there are no μ-traces on it, otherwise it is said to be odd.

Definition 35. *(Tableau core)* A tableau core is $\mathcal{C} = (C, \Omega)$ where C is a potentially infinite tree of which the nodes are either modal nodes or disjunctive nodes and modal nodes are decorated with a set of literals. In the multimodal case, edges from a modal node to its children are decorated with an action from Act. Ω is a parity assignment with a finite prefix of $\mathbb{N}$ as co-domain. An infinite path in $\mathcal{C}$ is of the parity of the most significant priority seen infinitely often.

$\mathcal{C} = (C, \Omega)$ is a tableau core *for* $\mathcal{T} = (T, L)$ if there is a bijection b between C and the nodes of T which are either modal or the immediate successor of a modal node, which respects:

- The successor-relation: $b(i)$ is a child of $b(j)$ in C if and only if i is a child of j if j is modal, or the next modal descendant of j otherwise; In the multimodal case, the action between a modal node and its child must be respected;
- The node types: $b(i)$ is modal if and only if i is modal;
- The literals at modal nodes: i and $b(i)$ have the same set of literals;
- The parity of infinite paths: if nodes along a path in T map to a path in C, then they must be of the same parity.

In brief, the core of a tableau represents the elements of a tableau which are invariant under different orders of application of tableau rules.

Definition 36. *(Tableau equivalence)* Two tableaus $(\mathcal{T}_0, L_0)$ and $(\mathcal{T}_1, L_0)$ are equivalent if their cores are bisimilar with respect to their branching structure, whether nodes are disjunctive or modal, the literals at modal nodes, and the parity of infinite branches. Two formulas are tableau equivalent if they generate equivalent tableaus.

Theorem 37. *([JW95]) Tableau equivalent formulas are semantically equivalent.*

Tableau equivalence is a stricter notion than semantic equivalence: $\psi \vee \neg\psi$ and $\top$ have different tableaus for example. Tableau equivalence will be further studied in the next section which shows that it preserves alternation depth.

3.3.2 From tableau to formula

The second step of the transformation into disjunctive form consists of turning a tableau back into a formula. Two things are required: finding an appropriate finite representation of the tableau and then describing it using L_μ syntax. The first part is delicate: the finite representation has to capture the parity of infinite paths, extracted from the traces on a path. Section 3.3.3 describes the exact requirements for such a representation, and how the final step – generating a disjunctive formula – is easy once such a representation is found. The task of finding a good representation for a tableau is left to Section 3.3.4.

3.3.3 Encoding the parity of infinite paths

This section defines a criterion over trees with back edges representing a tableau for Ψ which guarantees that the tree with back edges can be transformed into a disjunctive formula. It then shows how any such tree with back edges generates a disjunctive formula and proves that this formula is equivalent to the original one. In the original description of the transformation into disjunctive form, this criterion is described as the existence of a magenta/navy colouring corresponding to what we here describe simply as the parity of nodes.

Definition 38. *(Head)* Given an infinite path in a tree with back edges, the head of the path is the node closest to the root that the path visits infinitely often. We say that the head of a path dominates it.

The head of a path is important: when the tree with back edges turns into a formula, this is where a fixpoint is bound. The parity of the fixpoint, so whether it is a μ-

or ν-binding, depends on the parity of the paths that the node dominates. Crucially, all the infinite paths that a node dominates have to have the same parity. This yields the criterion for a well-coloured tree with back edges: we must be able to assign each node a parity which must agree with the parity of all infinite paths that it dominates.

Definition 39. *(A well-coloured tree with back edges)* Call a tree with back edges that unfolds into a tableau well-coloured if every node dominates only infinite paths of one parity. In a well-coloured tree with back edges, call a node even if it only dominates paths without μ-traces and odd if it only dominates infinite paths with at least one μ-trace.

Often, given a tableau, the tree with back edges which identifies nodes with the first ancestor with the same label is good enough, as in the tableau of Figure 3.4.1: the node marked **1** dominates an even path while those marked **2** and **3** dominate paths of odd parity.

However, this is not always the case, as the following tableau demonstrates.

Example 40. Consider this tableau of $\nu Z.\mu X.\nu Y.(A\rightarrow\{X\} \vee \neg A\rightarrow\{Y\}) \wedge \rightarrow\{Z\}$, where ϕ stands for $(A\rightarrow\{X\} \vee \neg A\rightarrow\{Y\}) \wedge \rightarrow\{Z\}$:

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\phi\ (\mathbf{1})}{\mu X.\nu Y.\phi\ ,\ \nu Z.\mu X.\nu Y.\phi}}{X\ ,\ Z}}{A\rightarrow\{X\}\ ,\ \rightarrow\{Z\}} \qquad \dfrac{\dfrac{\dfrac{\phi(\mathbf{1})}{\nu Y.\phi\ ,\ \nu Z.\mu X.\nu Y.\phi}}{Y\ ,\ Z}}{\neg A\rightarrow\{Y\}\ ,\ \rightarrow\{Z\}}}{A\rightarrow\{X\} \vee \neg A\rightarrow\{Y\}\ ,\ \rightarrow\{Z\}}}{(A\rightarrow\{X\} \vee \neg A\rightarrow\{Y\}) \wedge \rightarrow\{Z\}(\mathbf{1})}}{\nu Z.\mu X.\nu Y.(A\rightarrow\{X\} \vee \neg A\rightarrow\{Y\}) \wedge \rightarrow\{Z\}}$$

In this example, all infinite paths are dominated by the same node. Paths that only sees the right hand branch infinitely often do not have odd traces on them, since both Y and Z are ν-bound. However, all paths that see the left hand branch infinitely often have a μ-trace on them which sees X infinitely often. In other words, the node marked **1** dominates both even and odd paths – this tree with back edges is not well-coloured

Section 3.3.4 will describe how to turn any tableau into a well-coloured tree with back edges. For now, assume that we have such a well-coloured tree to represent the tableau of a formula Ψ. Informally, to turn this tree into a disjunctive formula, the sources of back-edges become fixpoint variables, bound at the target of the back edge; the formula can then be read bottom-up from the tree, joining formulas with either disjunctions or modalities according to the type of the nodes.

Definition 41. *(The disjunctive formula from a well-coloured tree)* Let Ψ be a formula and let $\mathcal{T}$ be a well-coloured tree with back edges that unfolds into a tableau for Ψ. Let f be an assignment of formulas to the nodes of $\mathcal{T}$ satisfying the following: if n is a leaf with label $L(n)$, then $f(n)$ is the conjunction of literals in $L(n)$; if n is a disjunctive node with children n_0 and n_1, then $f(n) = f(n_0) \vee f(n_1)$; if n is the source of a back edge of which the target is m, then $f(n) = X_m$ where X_m is a fixpoint variable; if n is a modal node, then $f(n) = P \bigwedge_{a \in A} \xrightarrow{a} \mathcal{F}_a$ where P is the conjunction of the set of literals in $L(n)$ and $\mathcal{F}_a$ is the set of $f(n_i)$ for the n_i children of n via an a-edge and A is the set of actions a for which n has a-children; finally, if n is the target of a back edge, $f(n)$ is obtained as previously detailed but in addition, it binds the fixpoint variable X_n with a ν-binding if n is of even parity and with a μ-binding otherwise. Other nodes inherit the formula assigned to their unique child. If r is the root node of $\mathcal{T}$, then let $dis(\Psi) = f(r)$.

It should be clear that this method only builds disjunctive sentences $dis(\Psi)$. It then suffices to prove that $dis(\Psi) = \Psi$.

Lemma 42. *Given a modal μ formula Ψ and a well-coloured tree with back edges representing its tableau $\mathcal{T}$, the formula $dis(\Psi)$ derived from the tree with back edges is equivalent to Ψ.*

Proof. By construction these two formulas are tableau equivalent: The tableau $\mathcal{T}_{dis}$ for $dis(\Psi)$ is simply the one represented by the well-coloured tree with back edges with the assigned subformulas of Ψ as its labels. The tableaus $\mathcal{T}$ and $\mathcal{T}_{dis}$ are clearly bisimilar with respect to their branching structure, types of nodes and propositional variables. Furthermore, an infinite path in $\mathcal{T}$ is even if and only if its head is labelled with a ν-formula, which is the case if and only if it is the head of only even paths in $\mathcal{T}$. As a result, $\mathcal{T}$ and $\mathcal{T}_{dis}$ are also identical with respect to the parity of infinite paths. This establishes that $\mathcal{T}$ and $\mathcal{T}_{dis}$ are tableau equivalent, which in turn implies semantic equivalence. □

3.3.4 Generating a well-coloured tree with back edges from a tableau

We have so far shown that given a well-coloured tree with back edges representing a tableau, we can obtain a disjunctive formula with equivalent tableau. This section goes on to show how any tableau can be represented by a well-coloured tree with back edges. To do so, a deterministic parity automaton on words decides whether an infinite path in a tableau has a μ-trace. The labelling of the tableau by this automaton suffices to fold the tableau into a well-coloured tree with back edges.

Definition 43. *(Non-deterministic parity automaton on words)* A non-deterministic parity automaton on Σ^ω characterising a language of infinite words over the alphabet Σ can be written $A = (Q, \Sigma, \delta, q_I, \Omega)$ and consists of:

- A set of states Q with an initial state $q_I \in Q$,
- An alphabet Σ,
- A transition function $\delta : Q \times \Sigma \to \mathcal{P}(Q)$, and
- A priority function $\Omega : Q \to \omega$.

Such an automaton is said to be deterministic if $|\delta(q,a)| = 1$ for all $q \in Q$ and $a \in \Sigma$.

A run of such an automaton on an ω-word $w \in \Sigma^\omega$ consists of a potentially infinite sequence $\pi_w \in Q^\omega$ of states such that the initial state of the run is the initial state of the automaton, written $\pi[0] = q_I$, and thereafter every step in the run respects the transition function in the sense that $\pi[n+1] \in \delta(\pi[n], w[n])$ for all $n > 0$. A run sees a priority p if there is an n such that $\Omega(\pi[n]) = p$. A priority p is in $\mathit{inf}(\pi_w)$ if $\Omega(\pi[n]) = p$ for infinitely many distinct values of n. The word w is accepted by A if and only if there is a run π_w on A on which the most significant priority seen infinitely often is even, that is to say if the maximal element of $\mathit{inf}(\pi_w)$ is even.

The language of all the ω-words accepted by A is written $\mathcal{L}(A)$.

A deterministic parity automaton on words will yield a finite priority assignment to express the parity of the infinite paths of a tableau. This deterministic automaton is obtained by determinising a non-deterministic one, which is described below. Note that determinisation is a difficult problem with a wealth of literature of its own. Unfortunately this means that this transformation into disjunctive form depends on an opaque determinisation step, the details of which are outside the scope of this thesis. The interested reader will find a thorough exposition of the topic and further reference in [GTW02], and a discussion of more recent complexity results in [Var14].

Lemma 44. *(Complementation Lemma) [Pit06] For any non-deterministic parity automaton A on ω-words, there is a deterministic, effectively constructible parity automaton $\bar{A}$ which accepts a word w if and only if A does not.*

We can now build the non-deterministic parity automaton on words needed to recognise the parity of infinite paths. Its alphabet consists of the sets $S \in \mathcal{P}(\mathit{sf}(\Psi))$ of subformulas from Ψ. It operates on words $w \in \mathcal{P}(\mathit{sf}(\Psi))^\omega$ and accepts the sequences of sets of subformulas that contain a μ-trace. Its states are the subformulas of Ψ, and the transition function picks from the next set of subformulas the immediate subformulas of the current state. A run of this automaton on the labels of an infinite tableau path corresponds to a trace on that path. The parity condition recognises μ-traces, corresponding to an accepting run.

Formally the automaton is given by $A = (\mathit{sf}(\Psi), \mathcal{P}(\mathit{sf}(\Psi)), \delta, \Psi, \Omega)$ where the transition function δ is defined as:

- $\delta(\phi, S) = \{\phi\}$ if $\phi \in S$, otherwise
- $\delta(\phi_0 \wedge \phi_1, S) = \{\phi_0, \phi_1\}$
- $\delta(\phi_0 \vee \phi_1, S) = \{\phi_0, \phi_1\} \cap S$
- $\delta(\sigma X.\phi, S) = \{\phi\}$
- $\delta(X, S) = \{\phi_X\}$ where ϕ_X is the formula binding X
- $\delta(\xrightarrow{a}\mathcal{F}, S) = \{\bigvee \mathcal{F}\}$ if $\bigvee \mathcal{F} \in S$, else $S \cap \mathcal{F}$.

The priority function Ω assigns μ-variables even priorities and ν-variables odd priorities in accordance with the order of significance of the variables' bindings. Note that since we are accepting runs containing μ-traces, the parity of priorities is inverted.

Lemma 45. *The above automaton accepts an infinite sequence of labels from a tableau if and only if it has a μ-trace.*

Proof. Let π be an infinite sequence of labels from a tableau. It should be clear that the runs of A on π are simply the traces on π. Because of the parity inversion in Ω, a run is accepting exactly when the most significant fixpoint seen infinitely often is μ-bound. The path π of a tableau is then accepted if and only if it has an accepting run, that is to say if it has a μ-trace. $\square$

Note that this automaton is no larger than Ψ and has no more priorities than the number of priorities in the automaton for Ψ, that is to say the alternation depth of Ψ. The Complementation Lemma then gives us a deterministic parity automaton B which accepts branches of a tableau if and only if the branch does not have a μ-trace. We run this automata on each branch of the tableau of Ψ to obtain a labelling of the tableau with states of B.

It now suffices to argue that using the states assigned by B to the nodes of the tableau for Ψ, we can represent it with a well-coloured tree with back edges (see Definition 39).

For each node of a tableau, consider both its label and the state assigned by the automaton B. These augmented labels come from a finite set, so they must eventually start repeating themselves. Furthermore, two nodes with identical augmented labels are the root of identical, identically labelled subtrees. On each branch, one can therefore identify the first node v which has an identical ancestor w. Let d be the node between v and w which is assigned the state of B with the most significant priority. This node d will be the target of the back edge which will terminate this branch. The source of the back edge is the next node with identical augmented label to d. In a finite number of such operations, the infinite structure can be collapsed into a finite one. The operation preserves bisimulation, so the order in which the branches collapse is of no importance.

In this tree with back edges, the most significant B-priority on an infinite path is the B-priority of the node closest to the root. As a result, for any node, either all or none of the paths that it dominates are accepted by B, thus making the tree with back edges well-coloured. From the previous section, any such tree can be turned into a disjunctive formula with equivalent tableau.

To summarise, to build the disjunctive formula equivalent to Ψ one must:

- Build the tableau of Ψ, as per section 3.3.1;
- Collapse said tableau into a well-coloured tree with back edges;
- Retrieve a disjunctive formula from this tree, following Definition 41.

The second step can be somewhat opaque, as it uses an automaton of which only the complement has been explicitly described. However, typically it is not hard to find a suitable finite representation without requiring the help of this automaton. Quite often the simplest representation is suitable, as is the case for most, if not all examples in this thesis, with the exception of Example 40. Even for such examples, a well-coloured tree is easy to engineer without the help of the additional automaton. Hence, if one were to implement this transformation, experimenting with simple ways to find well-coloured trees would probably yield satisfactory results.

3.4 Disjunctive form and alternation depth

The previous section described in great detail how a formula is transformed into disjunctive form. This section discusses how this transformation affects the alternation depth of a formula. It begins with a discussion of tableau equivalence. As defined in the previous section, tableau equivalence is a syntax-based refinement on semantic equivalence which holds whenever two formulas have sufficiently similar tableaus. This section argues that tableau equivalence preserves alternation depth for disjunctive formulas and hence any transformation into a tableau equivalent disjunctive formula has the same effect on a formula's alternation depth. The proof is simple and the result is unsurprising – it simply establishes tableau equivalence as a robust equivalence class for disjunctive formulas which overlooks some syntactic details but preserves our notion of complexity. This is needed for the following results, proved in Sections 3.4.2 and 3.4.3, to be meaningful:

- Π^μ_2 preservation: a Π^μ_2 formula is tableau equivalent to a disjunctive Π^μ_2 formula;
- Σ^μ_2 non-preservation: some Σ^μ_2 formulas are tableau equivalent only to disjunctive formulas of arbitrarily high alternation depth.

A similar pattern is seen for non-deterministic automata: an alternating Büchi automaton (corresponding to index $\{2,1\}$ as for Π^μ_2 formulas), is equivalent to a non-deterministic Büchi automaton [MS95] while the same does not necessarily hold for alternating co-Büchi automata.

Note that the analysis of this section focuses on the transformation which turns a formula into a tableau equivalent formula. It is of course conceivable, albeit unlikely, that there exists a different transformation into disjunctive form which preserved alternation depth, but not tableau equivalence.

3.4.1 Tableau equivalence preserves alternation depth for disjunctive L_μ

This section argues that all disjunctive formulas generating the same tableau $\mathcal{T}$ have the same alternation depth. The structures used to identify the alternation depth, consisting of nested cycles of alternating parity, are similar to ones used to compute the Rabin–Mostowski index of a parity game [HKP12] and the ones (called flowers) used find the Rabin–Mostowski index of deterministic automata [NW05]. These structures witness that the priority assignment representing the tableau requires at least q priorities. I show that tableau equivalence preserves these structures and consequently also the alternation depth of disjunctive formulas.

Informally, each component of the witness of a priority assignment being minimal is a series of nested cycles of increasing alternating parity. There are two components – one to show the necessity of the minimal priority, the other to show the necessity of the maximal one.

Definition 46. *(I-witness)* Given an index $I = \{q,...,m\}$ where $m \in \{0,1\}$, an I-witness in a tree with back edges representing a tableau consists of two sets of nested cycles $e_m,...,e_{q-1}$ and $o_{m+1},...,o_q$ such that the cycles e_i and o_i are of the parity of i for all $i \in I$, and the cycles o_i and e_i are subcycles of o_{i+1} and e_{i+1} respectively, if they exist.

Lemma 47. *If a tree with back edges (A,Ω) has a I-witness, then the co-domain of the priority assignment Ω is no smaller than I.*

Proof. Given an I-witness $e_m...e_{q-1}$ and $o_{m+1},...,o_q$, the dominant priority of e_{i+1} must be strictly larger than the dominant priority of e_i for every $i < q$, since e_i and e_{i+1} are of different parity and e_i is contained in e_{i+1}. Similarly for the dominant priorities in the o_i cycles. There must therefore be at least $|I| - 1$ priorities in the cycles e_{q-1} and o_q which contain all the other cycles of the witness. Since e_m and e_{m+1} are of different priorities there must be at least $|I|$ priorities in total. $\square$

Lemma 48. *If a tree with back edges $\mathcal{A}$ representing a tableau $\mathcal{T}$ does not have an I-witness, then there is a tree with back edges $\mathcal{A}'$ which also represents $\mathcal{T}$ but has a priority assignment with fewer priorities.*

Proof. Assume a tree with back edges $\mathcal{A} = (A, \Omega)$ representing $\mathcal{T}$ with a priority assignment with co-domain $I = \{q, ..., m\}$ does not have an I-witness. Assume first it is missing the set of cycles $e_m, ..., e_{q-1}$. Define the following new priority assignment: first increase the priority of all nodes of priority m by 2. Wherever the priority of a subdominant node – *i.e.* nodes of second highest priority in a cycle – has been increased, increase by 2 the priority of all dominant nodes. Propagate in this manner the increase in priorities throughout. Since there is no set of e-cycles for the witness, this propagation does not increase the priority of any nodes of priority $q-1$. Then, the new minimal priority is greater than 1, every priority can be scaled down by 2. This parity assignment respects the parity of all cycles and is smaller than I.

Similarly, if there is no set of cycles $o_{m+1}, ..., o_q$ let the new priority assignment assign $q-2$ to nodes of priority q. This time the decrease propagates downwards to any subdominant nodes in cycles where the dominant priority has just decreased. Since there is no set of o-cycles for the I-witness, this does not propagate all the way down to $m+1$. As before, the parity of cycles is preserved, with a smaller priority assignment. □

Lemma 49. *All tableau equivalent trees with back edges have the same I-witnesses: for all I, either all or none of the trees with back edges representing a same tableau $\mathcal{T}$ have an I-witness.*

Proof. First we recall that if $\mathcal{A}$ is the finite representation of $\mathcal{T}$ induced by a disjunctive formula Ψ then the tableau of $\mathcal{T}$ is an infinite tree bisimilar to $\mathcal{A}$ with respect to node type, literals and parity of infinite branches. Hence any finite representation of $\mathcal{T}$ is bisimilar to $\mathcal{A}$. It then suffices to show that I-witnesses are preserved under bisimulation. This is straightforward: let $\mathcal{A}'$ be bisimilar to a finite tree with back edges $\mathcal{A}$ with respect to node type, literals at modal nodes and the parity of infinite paths. Then infinite paths in $\mathcal{A}$ are bisimilar to infinite paths in $\mathcal{A}'$. Since both $\mathcal{A}$ and $\mathcal{A}'$ are finite, an infinite path stemming from a cycle in $\mathcal{A}$ is bisimilar to a cycle in $\mathcal{A}'$. Both set of cycles in the I-witness contain at least one node which lies on all the cycles of that component of the witness. If $\mathcal{A}$ has a set of cycles c_i for $c \in \{e, o\}$, call the node on all of its cycles n and consider (one of) the deepest node(s) n' in $\mathcal{A}'$ bisimilar to n. That is to say, choose n' such that if another node bisimilar to n' is reachable from n', it must be an ancestor of n'. Since n' is bisimilar to n, there must be a cycle c_i' bisimilar to each c_i reachable from n'. Since n' is maximally deep, it is contained in each of these cycles c_i'. Then, the set of cycles can be reconstructed in $\mathcal{A}'$ by taking the cycle c_i', and then for each $i > 0$ the cycle consisting of all $c_j', j \leq i$. Since all c_i' cycles have n' in common, there is a cycle combining $c_j', j \leq i$ for any i. Since bisimulation respects the parity of cycles, this yields a set of cycles for an I-witness in $\mathcal{A}'$. □

Theorem 50. *All disjunctive formulas with tableau $\mathcal{T}$ have the same alternation depth.*

Proof. All trees with back edges representing the same tableau $\mathcal{T}$ have the same maximal witness, from the previous lemma, so from Lemma 48 they accept a minimal priority function with domain $\{0...q\}$. Since a disjunctive formula induces a tree with back edges with a minimal priority function corresponding to the formula's alternation depth, any two disjunctive formulas that are tableau equivalent must have the same alternation depth. $\square$

This concludes the proof that tableau equivalence preserves alternation depth on disjunctive formulas. The restriction to disjunctive formulas is crucial: as the next section shows, in the general case tableau equivalent formulas may have vastly different alternation depths.

As an immediate consequence, for any L_μ formula, the least alternation depth of a tableau equivalent disjunctive formula is decidable. This raises the question of whether the same is true if we lift the restriction to disjunctive form, but keep the restriction to tableau equivalence: for a L_μ formula, is the least alternation depth of any tableau equivalent formula decidable? Tableau equivalence is a stricter equivalence to semantic equivalence, so this problem is likely to be easier than deciding the alternation hierarchy with respect to semantic equivalence but it would still be a considerable step towards understanding accidental complexity in L_μ. The next sections answers this positively for formulas equivalent to disjunctive Π_2^μ formulas, but in general the question remains wide open.

3.4.2 Disjunctive form preserves Π_2^μ

In this section I argue that if a disjunctive formula is not syntactically in Π_2^μ – that is to say has a cycle which sees both an even priority and a more significant odd priority – then Ψ is not tableau equivalent to any Π_2^μ formula. As a consequence, a Π_2^μ formula remains in Π_2^μ when turned into disjunctive form.

Theorem 51. *All Π_2^μ formulas are tableau equivalent to disjunctive Π_2^μ formulas.*

Proof. Assume that a disjunctive formula Ψ is not in Π_2^μ and therefore has a cycle with a μ-bound fixpoint X which is more significant than a ν-bound fixpoint Y. Consider a tableau-equivalent formula Φ. Let $\mathcal{T}$ be a well-coloured tree with back edges which represents the common core of the tableau of Ψ and Φ, and let L_Ψ be the labelling of the nodes of $\mathcal{T}$ with subformulas of Ψ while L_Φ labels nodes of $\mathcal{T}$ with sets of subformulas of Φ, each according to their respective tableaus.

Since the two tableaus are tableau equivalent, they agree on the parity of infinite paths. In particular, the cycle dominated by Y according to L_Ψ is even, i.e. has no

μ-traces in either tableau, while all paths dominated by X according to L_Ψ have a μ-trace. We consider a path π_X which is dominated by X but sees a cycle dominated by Y many times – say n times – without seeing a more significant fixpoint in between. Choose such a path for n larger than the largest label in L_Φ. This ensures that any trace, in particular any μ-trace, on π_X reaches the node labelled by Y twice at the same subformula α while it goes through the Y-cycle n times. The highest priority p such a trace sees between the two instances of α has to be even, since the Y-cycle is even. However if the trace is a μ-trace, as at least one trace on π_X must be, then it has to be dominated by an odd priority that must therefore be more significant than p. Such traces do not exist in the tableaus of Π_2^μ formulas, so Φ is not Π_2^μ either.

Therefore, any Π_2^μ formula is only tableau equivalent to disjunctive Π_2^μ formulas.

□

3.4.3 Disjunctive form does not preserve Σ_2^μ

This section demonstrates that not only does disjunctive form not preserve alternation depth in general, but also that there is no hope for bounding the alternation depth of disjunctive formulas with respect to their semantic alternation depth: for any n there are one-alternation formulas which only are tableau equivalent to n-alternation disjunctive formulas. In other words, the alternation depth of a L_μ formula, when transformed into disjunctive form, can be arbitrarily large. Conversely, as shown in the second subsection, formulas of arbitrarily large alternation depth can be tableau equivalent to a disjunctive formula without alternations. Hence the alternation depths of tableau equivalent formulas are only related within the disjunctive fragment.

While the main theorem is proved by Example 55, the Examples 52 and 53 leading up to it should give the interested reader some intuition about the mechanics which lead the tableau of a formula to have higher alternation depth than one might expect.

Example 52. The first example is a rather simple one: a disjunctive formula with one alternation that can be expressed in non-disjunctive form without any alternations. The disjunctive formula $\nu Z.\mu W.(\neg A \wedge \rightarrow\{W\}) \vee (A \wedge \rightarrow\{Z\})$ signifies that all paths are infinite and A occurs infinitely often on all paths. Compare it to the formula $\nu Y.\rightarrow\{Y\} \wedge \mu X.(\neg A \wedge \rightarrow\{X\}) \vee A$ which is alternation free.

The tableaus of both these formulas are shown side by side in Figure 3.4.1. Both branches regenerate into either exactly the ancestral node marked * or a node that reaches a node identical to the one marked * in a single non-branching step.
The cores of the two tableaus, that is to say their branching nodes, are clearly isomorphic with respect to the node type and branching structure. Furthermore, for both

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{Y,X \quad \boldsymbol{W}}}{\to\{Y\},\neg A,\to\{X\} \quad \boldsymbol{\neg A,\to\{W\}}}}{\to\{Y\},\neg A\wedge\to\{X\} \quad \boldsymbol{\neg A\wedge\to\{W\}}} \qquad \dfrac{\dfrac{\dfrac{*}{Y \quad \boldsymbol{Z}}}{\to\{Y\},A \quad \boldsymbol{A,\to\{Z\}}}}{\to\{Y\},A \quad \boldsymbol{A\wedge\to\{Z\}}}}{* \quad \to\{Y\},(\neg A\wedge\to\{X\})\vee A \quad \boldsymbol{(\neg A\wedge\to\{W\})\vee(A\wedge\to\{Z\})}}}{\nu Y.\to\{Y\}\wedge\mu X.(\neg A\wedge\to\{X\})\vee A \quad \boldsymbol{\nu Z.\mu W.(\neg A\wedge\to\{W\})\vee(A\wedge\to\{Z\})}}$$

Figure 3.4.1: Side by side, the tableaus for $\nu Y.\to\{Y\}\wedge\mu X.(\neg A\wedge\to\{X\})\vee A$ and $\boldsymbol{\nu Z.\mu W.(\neg A\wedge\to\{W\})\vee(A\wedge\to\{Z\})}$ (in bold for readability)

formulas, there is μ-trace on any path that only goes through the left hand branch infinitely often. There is no μ-trace on *any* path that goes through the right hand path infinitely often, for either formula. As a result, both tableaus agree on the parity of infinite branches. The two formulas are tableau equivalent and therefore also semantically equivalent.

Observe that there is nothing obviously inefficient about how the disjunctive formula handles alternations. Simply inverting the order of the fixpoints yields a formula which cannot be expressed without an alternation: $\mu X.\nu Y.A\wedge\to\{X\}\vee\neg A\wedge\to\{Y\}$.

While the above example proves that disjunctive form does not preserve alternation, it must be noted that the alternating parity automata corresponding to these formulas require in both cases two priorities, although one of them is a weak automaton. The next example shows formulas in which the number of priorities is not preserved either.

Example 53. This example and the following ones will be built on Σ_2^μ formulas properly embedded in one another: all free variables within the inner formula are bound by fixpoint bindings within the inner formula. This means that the formula accepts a priority assignment with co-domain $\{0,1\}$. Without further ado, consider the formula α in question:

$$\mu X_0.\nu Y_0.(A\wedge\to\{X_0\})\vee(B\wedge\to\{Y_0)\wedge\mu X_1.\nu Y_1.(C\wedge\to\{X_1\})\vee(D\wedge\to\{Y_1\})\vee E$$

Lemma 54. *The formula α is tableau equivalent to a disjunctive formula which requires a parity assignment with co-domain $\{0...3\}$:*

$$\begin{aligned}\beta = \mu X_0.\nu Y_0.\mu X_1.\nu Y_1.(A\wedge C\wedge\to\{X_0\})\vee(A\wedge D\wedge\to\{X_0\})\vee(A\wedge E\wedge\to\{X_0\})\\ \vee(B\wedge E\wedge\to\{Y_0\})\vee(B\wedge C\wedge\to\{X_1\})\vee(B\wedge D\wedge\to\{Y_1\})\end{aligned}$$

Proof. The tableaus for both formulas are written out in Figures 3.4.2 and 3.4.3. The two tableaus are isomorphic with respect to branching structure, node type and the

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{Y_0,Y_1}}{B\,,\to\{Y_0\}\,,D\,,\to\{Y_1\}}}{B\wedge\to\{Y_0\}\,,D\wedge\to\{Y_1\}}\quad\dfrac{\dfrac{\dfrac{*}{Y_0,X_1}}{B\,,\to\{Y_0\}\,,C\,,\to\{X_1\}}}{B\wedge\to\{Y_0\}\,,C\wedge\to\{X_1\}}\quad\dfrac{\dfrac{\dfrac{*}{Y_0}}{B\,,\to\{Y_0\}\,,E}}{B\wedge\to\{Y_0\},E}}{B\wedge\to\{Y_0\}\,,(C\wedge\to\{X_1\})\vee(D\wedge\to\{Y_1\})\vee E}\quad\dfrac{\dfrac{\dfrac{\dfrac{*}{X_0\,,Y_1}}{A\,,\to\{X_0\}\,,D\,,\to\{Y_1\}}}{A\wedge\to\{X_0\}\,,D\wedge\to\{Y_1\}}\quad\dfrac{\dfrac{\dfrac{*}{X_0\,,X_1}}{A\,,\to\{X_0\}\,,C\,,\to\{X_1\}}}{A\wedge\to\{X_0\}\,,C\wedge\to\{X_1\}}\quad\dfrac{\dfrac{\dfrac{*}{X_0}}{A\,,\to\{X_0\}\,,E}}{A\wedge\to\{X_0\},E}}{(A\wedge\to\{X_0\}),(C\wedge\to\{X_1\})\vee(D\wedge\to\{Y_1\})\vee E}}{*(A\wedge\to\{X_0\})\vee(B\wedge\to\{Y_0\}),(C\wedge\to\{X_1\})\vee(D\wedge\to\{Y_1\})\vee E}}{\mu X_0.\nu Y_0.(A\wedge\to\{X_0\})\vee(B\wedge\to\{Y_0\})\wedge\mu X_1\nu Y_1(C\wedge\to\{X_1\})\vee(D\wedge\to\{Y_1\})\vee E}$$

Figure 3.4.2: Tableau for α

$$\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{\dfrac{*}{Y_1}}{B\,,D\,,\to\{Y_1\}}}{B\wedge D\wedge\to\{Y_1\}}\quad\dfrac{\dfrac{\dfrac{*}{X_1}}{B\,,C\,,\to\{X_1\}}}{B\wedge C\wedge\to\{X_1\}}\quad\dfrac{\dfrac{\dfrac{*}{Y_0}}{B\,,E\,,\to\{Y_0\}}}{B\wedge E\wedge\to\{Y_0\}}}{(B\wedge E\wedge\to\{Y_0\})\vee(B\wedge C\wedge\to\{X_1\})\vee(B\wedge D\wedge\to\{Y_1\})}\quad\dfrac{\dfrac{\dfrac{\dfrac{*}{X_0}}{A\,,D\,,\to\{X_0\}}}{A\wedge D\wedge\to\{X_0\}}\quad\dfrac{\dfrac{\dfrac{*}{X_0}}{A\,,C\,,\to\{X_0\}}}{A\wedge C\wedge\to\{X_0\}}\quad\dfrac{\dfrac{\dfrac{*}{X_0}}{A\,,E\,,\to\{X_0\}}}{A\wedge E\wedge\to\{X_0\}}}{(A\wedge E\wedge\to\{X_0\})\vee(A\wedge D\wedge\to\{X_0\})\vee(A\wedge C\wedge\to\{X_0\})}}{*\,(A\wedge E\wedge\to\{X_0\})\vee(A\wedge D\wedge\to\{X_0\})\vee(A\wedge C\wedge\to\{X_0\})\vee(B\wedge E\wedge\to\{Y_0\})\vee(B\wedge C\wedge\to\{X_1\})\vee(B\wedge D\wedge\to\{Y_1\})}}{\mu X_0.\nu Y_0.\mu X_1.\nu Y_1.(A\wedge E\wedge\to\{X_0\})\vee(A\wedge D\wedge\to\{X_0\})\vee(A\wedge C\wedge\to\{X_0\})\vee(B\wedge E\wedge\to\{Y_0\})\vee(B\wedge C\wedge\to\{X_1\})\vee(B\wedge D\wedge\to\{Y_1\})}$$

Figure 3.4.3: Tableau for β

literals at modal nodes. To prove their equivalence, it is therefore sufficient to argue that this isomorphism also preserves the parity of infinite branches, that is to say that there is a μ-trace in an infinite path of one if and only if there is a μ-trace in the corresponding infinite path of the other.

To do so, we look, case by case, at the combinations of branches that a path can see infinitely often and check which have a μ-trace in each tableau. First argue that the three right-most branches in both tableaus are such that any path that sees them infinitely often has a μ-trace. This is witnessed in both cases by the least fixpoint variable X_0 which will dominate any trace it appears on and appears on a trace on all paths going through one of these branches infinitely often. So, in both tableaus, any path going through one of the right-most branches infinitely often is of odd parity. Now consider the branch that ends in Y_0 before regenerating to the node marked * in both tableaus. All traces on paths that go infinitely often through this branch will see Y_0 regenerate infinitely often. Therefore in both tableaus, a path going through this branch infinitely has a μ-trace if and only if it also goes through one of the three rightmost branches infinitely often. Now consider the fifth branch from the right, the branch that regenerates Y_0, X_1 in one case and just X_1 in the other. In both tableaus, a path that goes through this branch infinitely often will have a μ-trace unless it goes through the Y_0 branch infinitely often and does not go through one of the three right-most branches infinitely often. Finally, in both tableaus, a branch that only sees the left-most branch infinitely often is of even parity since such a path does not admit any μ-traces. However, if a path sees this branch and some other branches infinitely often, its parity is determined by one of the previously analysed cases. Since we have analysed all the infinite paths on these tableaus and concluded that in each case the parity of a path is the same in both tableaus, this concludes the proof that the two tableaus are equivalent. □

The above example yields a disjunctive formula of alternation depth $\{0...3\}$ which semantically only requires alternation depth $\{0,1\}$. This proves that disjunctive form does not preserve the number of priorities that the model checking game of a formula requires.

The next step is to generalise the construction of Example 53 to arbitrarily many alternations to prove that there is no bound on the number of alternations of a disjunctive formula tableau equivalent to a non-disjunctive formula of n alternations. To do so, we will first define the Σ_2^μ formulas α_n inductively, based on the formula of Example 53. We then argue that the tableau of α_n admits a $(2n+1)$-witness, proving that α_n is not tableau equivalent to any disjunctive formula of less than $2n+1$ alternations. Due to the argument pertaining to traces in increasingly large tableaus, its details are, inevitably, quite involved. However, the mechanics of the tableaus of α_n are not difficult; writing down the tableau of α_2 and working out its disjunctive form should suffice to gain an intuition of the proof to follow.

Example 55. In order to define α_n for any n define:

$$a_1 = \mu X_1.\nu Y_1.((A_1 \wedge \rightarrow\{X_1\}) \vee (B_1 \wedge \rightarrow\{Y_1\}) \vee E_1) \wedge$$
$$\mu X_0.\nu Y_0.(A_0 \wedge \rightarrow\{X_0\}) \vee (B_0 \wedge \rightarrow\{Y_0\}) \vee E_0$$
$$a_{i+1} = \mu X_{i+1}.\nu Y_{i+1}.((A_{i+1} \wedge \rightarrow\{X_{i+1}\}) \vee (B_{i+1} \wedge \rightarrow\{Y_{i+1}\}) \vee E_{i+1}) \wedge a_i$$

Then, define:

$$\alpha_n = \mu X_n.\nu Y_n.((A_n \wedge \rightarrow\{X_n\}) \vee (B_n \wedge \rightarrow\{Y_n\})) \wedge a_{n-1}$$

In other words, nested formulas $\mu X_i.\nu Y_i.((A_i \wedge \rightarrow\{X_i\}) \vee (B_i \wedge \rightarrow\{Y_i\}) \vee E_i)$ are connected by conjunctions where the out-most clause does not have a $\vee E$.

As the formula grows, its tableau becomes unwieldy, but its structure remains constant: it is just as the tableau of α with more branches. Figure 3.4.2 can be used as reference.

The tableau of any α_n follows this structure:

- The first choice node $\{(A_n \wedge \rightarrow\{X_n\} \vee B_n \wedge \rightarrow\{Y_n\}), ..., (A_0 \wedge \rightarrow\{X_0\} \vee B_0 \wedge \rightarrow\{Y_0\} \vee E_0)\}$ branches into 2×3^n modal nodes – ignoring the modalities attached to each literal for a moment, this is the decomposition of $(A_n \vee B_n) \wedge (A_{n-1} \vee B_{n-1} \vee E_{n-1}) ... \wedge (A_0 \vee B_0 \vee E_0)$ into one large disjunction.
- Each choice leads to a modal node with some choice of propositional variables consisting of one of A_n and B_n and then for every $i < n$ one of A_i, B_i or E_i.
- These modal nodes have a single successor each, consisting of a set of fixpoint variables. In every case, one of these is Y_n or X_n and there is only ever at most one fixpoint variable out of $\{X_i, Y_i\}$ for each i. These nodes will be referred to as regeneration nodes. When a regeneration node does not contain X_i nor Y_i for some i, this corresponds to E_i having been chosen rather than A_i or B_i.
- Nodes consisting of a set of fixpoint variables all regenerate, give or take a couple of non-branching steps, into the same choice node, identical to the ancestral choice node labelled:

$$\{(A_n \wedge \rightarrow\{X_n\} \vee B_n \wedge \rightarrow\{Y_n\}), ..., (A_0 \wedge \rightarrow\{X_0\} \vee B_0 \wedge \rightarrow\{Y_0\} \vee E_0)\}$$

- An infinite trace in this tableau sees infinitely often only fixpoint variables Y_i and/or X_i for some i. As a consequence if a path goes infinitely often through a regeneration node which does not contain X_i or Y_i, then there is no trace that sees X_i infinitely often on that path.

Lemma 56. *The formula α_n is tableau equivalent only to disjunctive formulas which require a priority assignment with $2n+1$ priorities.*

Proof. Using the above observations, we will show that the tableau for this formula requires at least $2n+1$ alternating fixpoints. We describe a priority assignment to a subset of the nodes of the tableau of α_n such that on the paths within this subset, a path is even if and only if the most significant priority seen infinitely often is even. We then argue that this subset constitutes a $2n+1$-witness.

Consider the paths of the tableau which only contain the following regeneration nodes:

- For all i, the nodes regenerating exactly $Y_n Y_{n-1} ... Y_i$, and
- For all i the nodes regenerating exactly $Y_n ... Y_{i+1} X_i Y_{i-1} ... Y_0$.

For each i, assign priority $2i$ to the node regenerating $Y_n ... Y_i$ and $2i+1$ to the node regenerating $Y_n ... Y_{i+1}, X_i, Y_{i-1}, ... Y_0$. We now prove that this priority assignment is such that a path within this sub-tableau is even if and only if the highest priority seen infinitely often is even.

First consider the nodes $Y_n ... Y_i$, which have been assigned even priority. A path that sees such a node infinitely often can only have a μ-trace if it sees a node regenerating some X_j, $j > i$ infinitely often. Such a node would have an odd priority greater than $Y_n ... Y_i$. Therefore, if the most significant priority seen infinitely often is even, the path has no μ-trace. Conversely, if a path sees $Y_n ... X_i ... Y_0$ infinitely often and no $Y_n ... Y_j$ where $j > i$ infinitely often, then there is a trace which only regenerated X_i and Y_i infinitely often. This is a μ-trace since X_i is more significant than Y_i. This priority assignment therefore describes the parity of infinite paths on this subset of paths of $\mathcal{T}$.

Any assignment of priorities onto $\mathcal{T}$ should, on this subset of paths, agree in parity with the above priority assignment. However, in any tree with back edges generating this tableau, this subset of paths constitutes a $2n+1$ witness: c_0 is a cycle that only sees $Y_n ... Y_0$, c_1 contains c_0 and also sees $Y_n ... X_1 Y_0$ infinitely often and for all $i > 1$, the cycle c_{2i} is one containing c_{2i-1} and $Y_n ... Y_i$ while c_{2i+1} is one containing c_{2i} and $Y_n ... X_i ... Y_0$. Each cycle c_j is dominated by the priority j, making $c_0, ..., c_{2i+1}$ a $2i+1$-witness. Thus, using Theorem 50 any disjunctive formula with tableau $\mathcal{T}$ must require at least $2n+1$ priorities. □

This concludes the proof that for arbitrary n, there are Σ_2^μ formulas which are tableau equivalent to disjunctive formulas with n alternations.

3.4.4 Disjunctive formulas with small alternation depth

The previous section showed that transforming a formula into disjunctive form can increase its alternation depth. However, this does not mean that disjunctive form cannot also decrease the complexity of formulas. Indeed, the converse is very easy to show: there are very simple formulas for which the transformation into disjunctive form eliminates all alternations.

Lemma 57. *For any formula ψ, the formula $(\mu X.{\rightarrow}\{X\} \vee {\rightarrow}\{\}) \wedge \psi$ is tableau equivalent to a disjunctive formula without ν-operators.*

Proof. The semantics of $(\mu X.{\rightarrow}\{X\} \vee {\rightarrow}\{\}) \wedge \psi$ are that a structure must not have infinite paths and ψ must hold. Consider $\mathcal{T}$, the tableau for $(\mu X.{\rightarrow}\{X\} \vee {\rightarrow}\{\}) \wedge \psi$. It is easy to see that every modal node will either contain ${\rightarrow}\{X\}$ or ${\rightarrow}\{\}$. The latter case terminates that branch of the tableau, while the former will populate every successor node with X which will then regenerate into $({\rightarrow}\{X\} \vee {\rightarrow}\{\})$. As a result, all infinite paths have a μ-trace; there are no even infinite paths. Any disjunctive formula generating $\mathcal{T}$ will therefore only require the μ-operator. $\square$

If ψ is a formula of arbitrarily high alternation depth, $(\mu X.{\rightarrow}\{X\} \vee {\rightarrow}\{\}) \wedge \psi$ shows that there is no bound on how much the transformation into disjunctive form can reduce alternation depth. Together with the previous section, this concludes the argument that there are no bounds on the difference in alternation depth of tableau equivalent formulas.

To summarise, within the confines of the disjunctive fragment of L_μ, alternation depth is very well-behaved with respect to tableau equivalence: any two tableau equivalent disjunctive formulas have the same alternation depth. However, the story is quite different for L_μ without the restriction to disjunctive form: the alternation depth of a L_μ formula cannot be used to predict any bounds on the alternation depth of tableau equivalent disjunctive formulas.

3.5 Discussion

This chapter has presented an overview of the disjunctive fragment of L_μ and the transformation into it. The main contributions were:

- A brief survey of the various properties of disjunctive formulas.
- A detailed retelling of the transformation into disjunctive form.
- A proof that it preserves index for formulas in Π_2^μ; for other formulas, it can cause an arbitrarily large increase in the number of alternations.

This chapter has set the scene for using disjunctive form to study the alternation hierarchy: it is an expressively complete fragment of L_μ with useful structural properties. However, this comes at the cost of a potential increase in both formula size and alternation depth. For decidability, this does not prohibit turning input formulas into disjunctive form to analyse them. It does however mean that for the L_μ alternation hierarchy, it is not sufficient to decide the least number of alternations required to express a formula in disjunctive form, as this may be higher than the formula's true semantic alternation depth. Given the current state of understanding of the alternation hierarchy (or lack thereof), insights into any of these problems are of interest. I add to this list a more modest question: Given a formula, what is the least alternation depth of any *tableau equivalent* formula? From Section 3.4.1 we know that for the disjunctive fragment this is easy: a disjunctive formula is only tableau equivalent to disjunctive formulas of the same alternation depth. Section 3.4.3 tells us that for Π_2^μ this is decidable: a formula is tableau equivalent to a Π_2^μ formula only if its disjunctive counterpart is in Π_2^μ since a Π_2^μ formula cannot be tableau equivalent to a Σ_2^μ disjunctive formula. However, for the general case, this question is open and could provide some insight into the general index problem.

Chapter 4

The non-deterministic fragment of L_μ

The previous chapter discussed disjunctive L_μ in detail, laying out some of the tools that the forthcoming chapters will need. Before plunging into decision procedures in the next part, this chapter takes a short detour to consider a fragment of L_μ closely related to disjunctive L_μ: non-deterministic L_μ.

Disjunctive form was originally conceived to mirror non-determinism in automata theory [Wal95b] and hence, in many ways, the behaviour of disjunctive formulas is similar to that of non-deterministic automata. However, unlike L_μ and alternating parity automata, between which there is an index-preserving transformation, disjunctive L_μ and non-deterministic automata do not coincide with respect to the alternation depth: turning a disjunctive formula of index I into a non-deterministic automata may require an index much larger than I. This chapter defines the non-deterministic fragment of L_μ, which *does* have an index-preserving correspondence with non-deterministic automata. It can be seen as a natural generalisation of non-deterministic automata to unranked structures.

The first section of this chapter gives a syntactic characterisation of this fragment, defines a transformation into this fragment and studies its effect on the alternation depth of formulas. Incidentally, this yields a L_μ-theoretical alternative to the proof that alternating parity automata can be simulated by non-deterministic parity automata [MS95]. This section also clarifies the exact relation between non-deterministic automata and disjunctive L_μ.

The second section considers the index problem on the non-deterministic fragment. It argues that within this fragment, semantic equivalence between two formulas implies strong structural similarities between their tableaus. The index problem for this fragment therefore has a unique flavour, distinct in particular from the index problem for disjunctive formulas or non-deterministic automata: for this fragment, deciding the least index of a formula reduces to finding the best parity assignment for a family of structurally similar tableaus. This makes the non-deterministic fragment of L_μ particularly interesting: it is a powerful fragment, as witnessed by its correspondence to non-deterministic automata, yet its index problem seems much more accessible than that of other non-trivial fragments. The index-problem for this fragment isolates one of the challenges of the L_μ index problem: given the general structure of a formula, what is an optimal fixpoint assignment?

The overall aim of this chapter is to highlight some of the finer points in the relationship between L_μ and automata theory and emphasise the care needed when moving between the automata-theoretic and the L_μ frameworks: even though the objects under scrutiny are nearly identical, their respective index problems are drastically different.

4.1 Non-deterministic L_μ

Non-deterministic automata are a subclass of non-amorphous automata, *i.e.* defined on ranked structures only. These are discussed in this thesis because the Rabin–Mostowski index problem for both non-deterministic and alternating parity automata has received some attention in the past years, and some of the techniques developed there can be used in the L_μ framework. It is therefore useful to have a precise understanding of how to move between the L_μ and automata-theoretic frameworks when discussing the index problem. The main hurdle is that non-deterministic automata are defined for only a fraction of the structures L_μ operates on. This section defines what I will call the non-deterministic fragment of L_μ, designed to correspond exactly to non-deterministic automata on ranked structures. It can be seen as a generalisation of non-deterministic automata to unranked ones.

The intuition of the generalisation is natural: a transition in a non-deterministic automaton can be written down as $\bigvee_{i\in I}\{(a,q_{a,i}) \wedge (b,q_{b,i})\}$ to mean that the verifying player can choose a pair $q_{a,i},q_{b,i}$ to check at the unique a-successor and b-successor respectively. In an unranked structure, the a-successor and b-successor may not be unique, in which case the generalisation will require the verifying player to check $q_{a,i}$ and $q_{b,i}$ at *all* a- and b-successors, respectively. Using L_μ syntax, we can write this as $\bigvee_{i\in I}\{\xrightarrow{a}\{\phi_{a,i}\} \wedge \xrightarrow{b}\{\phi_{b,i}\}\}$. This idea yields the syntactic definition of non-deterministic L_μ: disjunctive L_μ that only allows singleton modalities $\xrightarrow{a}\{\phi\}$.

Definition 58. *(non-deterministic L_μ)* A L_μ formula Ψ is non-deterministic if it is in disjunctive form and for every modality $\xrightarrow{a}\mathcal{F}$ of Ψ, the set $\mathcal{F}$ is a singleton.

Lemma 59. *A non-deterministic L_μ formula is equivalent on ranked structures to a (non-amorphous) non-deterministic automaton of the same index and vice-versa.*

Proof. This follows from the observation that subformulas $\bigvee_{i\in I}\{\bigwedge_{a\in A_i} \xrightarrow{a}\{\phi_{a,i}\}\}$ can be written as a non-deterministic transition $\bigvee_{i\in I}\{\bigwedge_{a\in A_i}(a, q_{a,i})\}$ and vice-versa. The rest of the translation can be adapted from Wilke 2001 [Wil01] to fit any reasonable presentation of non-deterministic automata. □

A similar correspondence does not hold for disjunctive L_μ:

Example 60. A formula $\xrightarrow{a}\{\phi, \psi\}$ expresses that there is an a-successor satisfying ϕ, and one satisfying ψ, and all a-successors satisfy ϕ or ψ. On ranked structures, which only allow one a-successor, it is equivalent to $\xrightarrow{a}\{\phi \wedge \psi\}$. The corresponding automaton has universal branching, as witnessed by the conjunction.

On ranked structures, non-deterministic L_μ is expressively complete:

Theorem 61. *Given any L_μ formula Ψ, there is a non-deterministic L_μ formula Φ such that on ranked structures, Ψ agrees with Φ.*

Proof. On ranked structures, a modality $\xrightarrow{a}\mathcal{F}$ (where $\mathcal{F}$ is non-empty) is equivalent to $\xrightarrow{a}\{\bigwedge\mathcal{F}\}$, and $\xrightarrow{a}\{\}$ is equivalent to $\bot$, so any L_μ formula Ψ can easily be turned into a formula with singleton modalities of the same index, that agrees with Ψ on ranked structures. The transformation into disjunctive form, described in Section 3.3, turns a formula with only singleton modalities into a disjunctive formula with only singleton modalities. Then, to transform an arbitrary formula into a disjunctive formula with singleton modalities, it suffices to:

- Replace modalities $\xrightarrow{a}\mathcal{F}$ with the singleton modality $\xrightarrow{a}\{\bigwedge\mathcal{F}\}$, and
- Turn the resulting non-disjunctive formula into a disjunctive formula.

□

This transformation from L_μ formulas to non-deterministic L_μ gives an alternative, possibly simpler proof of the theorem that alternating parity automata can be simulated by non-deterministic automata [MS95].

As discussed in Section 3.4.3, the transformation into disjunctive form does not preserve alternation depth. Since the first step does not preserve disjunctive form, even disjunctive formulas can incur an increase in alternation depth when turned into non-deterministic formulas.

Example 62. Consider the formula:

$$\xrightarrow{a}\{\nu Y.\xrightarrow{a}\{Y\}, \mu X.(\neg P \wedge \xrightarrow{a}\{X\}) \vee P\}$$

On ranked structures, it is equivalent to the formula with singleton modalities:

$$\xrightarrow{a}\{(\nu Y.\xrightarrow{a}\{Y\}) \wedge \mu X.(\neg P \wedge \xrightarrow{a}\{X\}) \vee P\}$$

In disjunctive form, it yields (proof in Section 3.4.3):

$$\xrightarrow{a}\{\nu Z.\mu W.(\neg P \wedge \xrightarrow{a}\{W\}) \vee (P \wedge \xrightarrow{a}\{Z\})\}$$

Unlike the first two formulas, this last formula is not alternation-free. All of the examples in Section 3.4.3 can be similarly adapted to exemplify arbitrarily complex non-deterministic L_μ formulas that can be expressed with a single alternation by disjunctive formulas.

As a result, barring the unlikely discovery of an index-preserving transformation, non-deterministic L_μ is a weaker fragment than disjunctive L_μ: some properties of ranked structures may require a higher alternation depth if expressed with only singleton modalities. In the exact same sense, non-deterministic automata are weaker than disjunctive formulas restricted to ranked structures. On general unranked structures, non-deterministic L_μ is strictly less expressive than disjunctive L_μ, since it cannot even express $\xrightarrow{a}\{\phi, \top\}$, *i.e.* $\langle a\rangle\phi$.

4.2 The index problem for non-deterministic L_μ

As for any fragment of L_μ, there is an index problem for non-deterministic L_μ: Given a formula Ψ, what is the least index of any equivalent non-deterministic formula Φ?

When restricted to ranked structures, this is exactly the index problem for non-deterministic automata, known as the Rabin–Mostowski index problem. The Rabin–Mostowski index hierarchy is known to be strict; therefore the index hierarchy of non-deterministic L_μ is also strict. Note however that deciding the index hierarchy in either setting does not resolve it in the other: a non-deterministic automata may be equivalent to a simpler one on the ranked structures they are defined on, while the corresponding non-deterministic L_μ formulas may disagree on some unranked structures.

Example 63. $\xrightarrow{a}\{\phi\} \vee \xrightarrow{a}\{\neg\phi\}$ is semantically trivial on ranked structures while on unranked structures its complexity depends on ϕ.

In this section I study the index problem for non-deterministic L_μ on unranked structures. I begin by showing that non-deterministic L_μ is better behaved than both disjunctive L_μ and non-deterministic automata. The intuition is simple: Consider two equivalent disjunctions $\psi \vee \phi = \psi' \vee \phi'$. In general, such an equivalence does not imply any particular semantic relations between the subformulas ψ and ψ' nor between ϕ and ϕ'. However, for the particular case of disjunctions occurring in non-deterministic formulas, a disjunction such as this entails that either ψ implies ϕ' or it implies ψ', and similarly for ϕ.

Lemma 64. *If a subformula $\bigvee_{i\in I}\bigwedge_{a\in A_i}\xrightarrow{a}\{\phi_i^a\}$ of Φ implies $\bigvee_{j\in J}\bigwedge_{a\in A_j}\xrightarrow{a}\{\psi_j^a\}$, a subformula of Ψ, then for every $i\in I$, there is a $j\in J$ such that the disjunct $\bigwedge_{a\in A_i}\xrightarrow{a}\{\phi_i^a\}$ implies $\bigwedge_{a\in A_j}\xrightarrow{a}\{\psi_j^a\}$.*

Proof. Given $i\in I$ and $j\in J$ such that $\bigwedge_{a\in A_i}\xrightarrow{a}\{\phi_i^a\}$ does not imply $\bigwedge_{a\in A_j}\xrightarrow{a}\{\psi_j^a\}$, let for every action $a\in A_i$, the node s_j^a satisfy $\phi_i^a\wedge\neg\psi_j^a$ if $a\in A_j$ and it is satisfiable, and otherwise let s_j^a just satisfy ϕ_i^a. Note that each ϕ_i^a must be satisfiable and there must be some a such that $\phi_i^a\wedge\neg\psi_j^a$ is satisfiable, as otherwise the implication $\bigwedge_{a\in A_i}\xrightarrow{a}\{\phi_i^a\}\implies\bigwedge_{a\in A_j}\xrightarrow{a}\{\psi_j^a\}$ would hold.

Now assume that for some i, there is no j for which this implication holds. Then, consider the state s that has for successors s_j^a for each $a\in A_i$ and $j\in J$. This state s satisfies $\bigvee_i\bigwedge_{a\in A_i}\xrightarrow{a}\{\phi_i^a\}$ but not $\bigvee_j\bigwedge_{a\in A_j}(\xrightarrow{a}\{\psi_j^a\})$, a contradiction. □

This lemma captures the distinct behaviour of non-deterministic L_μ. Note that it only holds over unranked structures, as the proof relies on building structures with several successors for each action. In the rest of this section I will sketch out how this affects the index problem. The core idea is that semantic equivalence of two formulas implies semantic equivalences between their subformulas, so if a formula is equivalent to a formula of low index, it is equivalent to one that is structurally similar to itself.

Recall that a disjunctive formula is complete with respect to a set S of propositional variables if in every modality $P\bigwedge_{a\in A}\xrightarrow{a}\mathcal{F}_a$, the conjunction P contains for every propositional variable $p\in S$ either p or $\neg p$. Write the modalities in complete formulas as $\bigvee_{j\in J}\{P_j\wedge\bigvee_{i\in I}\bigwedge_{a\in A}\xrightarrow{a}\mathcal{F}_{a,i,j}\}$: that is to say, factorise the sets of propositional variables out so that they are not duplicated within the immediate subformulas of a disjunction. This just groups together modal formulas that Even can choose from at states satisfying P_j and makes the argument to follow slightly easier.

Definition 65. Let Ψ and Φ be equivalent non-deterministic complete formulas. Define $R\subseteq sf(\Psi)\times sf(\Phi)$ as follows:

- $(\Psi,\Phi)\in R$;
- If $(\nu X.\psi,\phi)\in R$ then $(\psi,\phi)\in R$;
- If $(\mu X.\psi,\phi)\in R$ then $(\psi,\phi)\in R$;
- If $(X,\phi)\in R$ then $(\psi_X,\phi)\in R$;
- If $(\psi,\nu X.\phi)\in R$ then $(\psi,\phi)\in R$;
- If $(\psi,\mu X.\phi)\in R$ then $(\psi,\phi)\in R$;

- If $(\psi, X) \in R$ then $(\psi, \phi_X) \in R$;
- If $(\bigwedge_{a\in A} \xrightarrow{a}\{\psi^a\}, \bigwedge_{a\in A} \xrightarrow{a}\{\phi^a\}) \in R$, then for each $a \in A$, $(\psi^a, \phi^a) \in R$;
- If $(\bigvee_{i\in I} D_i, \bigvee_{j\in J} D_j) \in R$ then $(D_i, D_j) \in R$ for all i, j such that D_i implies D_j;
- If $(P \wedge M, P \wedge M') \in R$ where P is a complete set of propositional variables, then $(M, M') \in R$

Note that it is always the case that if $(\phi, \psi) \in R$ then ϕ implies ψ. From the previous lemma this means that for pairs of disjunctions of modal subformulas in R, each disjunct of the first component implies at least one of the disjuncts of the second component. Similarly in the other direction, define $Q \subseteq sf(\Phi) \times sf(\Psi)$.

I will now show that in Ψ, only the subformulas in the co-domain of Q are necessary, and all other subformulas can be pruned.

Lemma 66. *Let Ψ and Φ be equivalent complete non-deterministic formulas and let R and Q be defined as above. Then Ψ is equivalent to Ψ where all subformulas which are not in the co-domain of Q are replaced with $\bot$, written $\Psi = \Psi \| Q$*

Proof. The direction $\Psi \| Q \implies \Psi$ is trivial. For the other direction, assume $\mathcal{M} \models \Psi$. Since Ψ and Φ are equivalent, there is a winning strategy σ for Even in $\mathcal{M} \times \Phi$. Note that for non-deterministic formulas, all strategies for Even are well-behaved.

Let $\mathcal{M}_\sigma$ be $\mathcal{M}$ pruned to only the nodes that σ reaches. We will now add to this tree some branches which will stop Even from using subformulas of Ψ that are not in the co-domain of Q when trying to prove that Ψ must also hold.

To do so, whenever σ reaches $v \times \phi$ in $\mathcal{M}_\sigma \times \Phi$, and ϕ is a disjunctive subformula $P \wedge \bigvee_{i\in I} \bigwedge_{a\in A_i} \xrightarrow{a}\{\phi_{a,i}\}$ where $v \models P$ and σ chooses $\bigwedge_{a\in A_i} \xrightarrow{a}\{\phi_{a,i}\}$ for some i, consider all disjunctions $\bigvee_{j\in J} \bigwedge_{a\in A_j} \xrightarrow{a}\{\psi_{a,j}\}$ for which

$$(\bigvee_{i\in I} \bigwedge_{a\in A_i} \xrightarrow{a}\{\phi_{a,i}\}, \bigvee_{j\in J} \bigwedge_{a\in A_j} \xrightarrow{a}\{\psi_{a,j}\}) \in Q$$

For each $j \in J$ such that $\bigwedge_{a\in A_i} \xrightarrow{a}\{\phi_{a,i}\}$ does not imply $\bigwedge_{a\in A_j} \xrightarrow{a}\{\psi_{a,j}\}$, but $A_j \subseteq A_i$, there is an $a \in A_i$ such that $\phi_{a,i}$ does not imply $\psi_{a,j}$. Add to v an a-successor $s_{a,j} \models \phi_{a,i} \wedge \neg\psi_{a,j}$. Extend σ to also play to $s_{a,j} \times \phi_{a,i}$ and use some winning strategy from there. Let this augmented $\mathcal{M}_\sigma$ be $\mathcal{T}_\sigma$.

Now consider $\mathcal{T}_\sigma \times \Psi$. Since σ is still a winning strategy in $\mathcal{T}_\sigma \times \Phi$, Even must also win $\mathcal{T}_\sigma \times \Psi$. Let us argue that she can win without leaving the co-domain of Q. She must start with Ψ itself, which is in the co-domain of Q. If σ reaches $v \times \phi$, and Even so far has played within the co-domain of Q, to $v \times \psi$ for some ψ such that $(\phi, \psi) \in Q$, then Even cannot from there play a subformula ψ' of ψ that is not in the co-domain of Q: at a disjunction $\bigvee_{j\in J}\{P_j \wedge \bigvee_{i\in I} \bigwedge_{a\in A} \xrightarrow{a}\mathcal{F}_{a,i,j}$ she has no choice but to play $P_j \wedge \bigvee_{i \in I} \bigwedge_{a \in A} \xrightarrow{a} \mathcal{F}_{a,i,j}$ for the P_j satisfied at the current state, and remain in the co-domain; at the choice of $\bigvee_{i \in I} \bigwedge_{a \in A} \xrightarrow{a} \mathcal{F}_{a,i,j}$, she can only choose i such that $\bigwedge_{a \in A} \xrightarrow{a} \mathcal{F}_{a,i,j}$ is in the co-domain of Q since the branches we have added to $\mathcal{T}_\sigma$ ensure that no other choice is viable. Furthermore, whichever subformula ψ' in the co-domain of Q she plays, it will be the case that either $(\phi, \psi') \in Q$, or $(\phi', \psi') \in Q$ for the next subformula ϕ' that σ plays. She therefore has a winning strategy restricted to $\Psi \| Q$.

This strategy is also a winning strategy in the original structure $\mathcal{M}$. □

Analogously, Φ can also be pruned to the co-domain of R. By iterating the pruning process, we obtain Ψ^*, Φ^*, R, and Q such that each subformula in Ψ^* is in the co-domain of Q and each subformula in Φ^* is in the co-domain of R.

So if Ψ, a complete non-deterministic formula that cannot be pruned, is equivalent to Φ, another complete non-deterministic formula of smaller index, they have near-identical structure: the structure of the parse-tree of the formula Φ, without the parity of its loops, is obtained from the parse-tree of Ψ by duplicating subformulas. This reduces the index problem for non-deterministic L_μ to deciding the optimal way of assigning parities onto formulas very similar to Ψ.

4.3 Discussion

This chapter defined the non-deterministic fragment of L_μ and discussed its index problem. Its primary goal was to make the relationship between disjunctive L_μ and non-deterministic automata explicit: non-deterministic automata are exactly disjunctive L_μ formulas with singleton modalities, restricted to ranked structures. When generalised to unranked structures, this corresponds to non-deterministic L_μ.

I then argued that this non-deterministic fragment is of independent interest: any two complete equivalent formulas of this fragment have almost identical structure. This makes this fragment an appealing non-trivial candidate for solving the index problem on. The problem seems easier on this fragment but still captures some of the challenges of the general index problem.

To conclude, I highlight a couple of interesting questions. First, having outlined the index problems for various fragments of L_μ and types of automata, and the differences between them, it seems likely that there could exist some reductions between some of these. Indeed, the model-checking problems for L_μ and alternating parity automata reduce to each other by encoding unranked structures by ranked structures and using an additional μ-operator to encode modalities; a similar reduction seems likely between model-checking disjunctive formulas and non-deterministic automata, although there might be a larger disrepancy between the indices. While these reductions are not sufficient to reduce the index problems to each other, it seems plausible that such reductions might exist.

Second, the theorem stating that non-deterministic L_μ is expressively complete on ranked structures is a logical equivalent of the simulation theorem [MS95] that shows that non-deterministic automata are as expressive as non-amorphous alternating parity automata. It is however not clear whether the two transformations yield a similar formula – in particular, the transformation here turns a formula with singleton modalities into a tableau-equivalent formula; does the automata-theoretic transformation also preserve tableaus? If not, can the transformation be rewritten into logical terms to provide an alternative structures to tableaus, yielding an alternative notion of formula equivalence?

Part II

In which I present decision procedures for ML, Π_1^μ, and Σ_2^μ

So far, I have considered some syntactic fragments of L_μ and how the index problems for these fragments differ. In this second part of the thesis, I shift the focus onto decision procedures for low alternation classes: Given an alternation class C, and a formula Ψ, is it decidable whether Ψ is equivalent to a formula in C?

For a while now the modal fragment, Σ_1^μ, and Π_1^μ have been the only levels of the L_μ alternation hierarchy known to be decidable. Otto showed in 1999 that modal properties are recognisable, and that the problem is EXPTIME-complete [Ott99]; in 2002 Küsters and Wilke showed the same for Σ_1^μ and Π_1^μ [KW02].

In both of these proofs, the focus is on the question of decidability, and little attention is given to how to turn the input formula into a formula in the target class. The first chapter of this part revisits the question of deciding ML, Σ_1^μ, and Π_1^μ, with a focus on generating the simplified formula. This is of course of interest for practical model-checking: formulas in ML, Σ_1^μ, and Π_1^μ are easy to model-check, so turning a formula into one in these classes, rather than just deciding the existence of a simple equivalent formula, is a sensible pre-processing step for model-checkers. On the theoretical side, these syntactic transformations give a more satisfying account of what unnecessary complexity looks like in L_μ and how it can be eliminated. The target formula is in both cases closely related to the original formula, obtained through simple formula manipulations.

Chapter 6 then climbs up the alternation hierarchy and addresses Σ_2^μ, a class for which, to the best of my knowledge, there are no previous decidability results. It shows that given a Π_2^μ formula, it is decidable whether it is equivalent to Σ_2^μ formula. Since $\Sigma_2^\mu \cap \Pi_2^\mu$ collapses to the alternation free class, this also decides whether Σ_2^μ and Π_2^μ formulas are semantically alternation-free. This decision procedure is also focused on the simplified formula and gives a syntactic account of how to turn a formula into an equivalent Σ_2^μ formula, if it exists. The core contribution of this chapter is a characterisation of Σ_2^μ consisting of a sequence of approximations of an input formula Ψ which eventually coincides with Ψ if and only if Ψ is semantically in Σ_2^μ.

Chapter 5

Eliminating recursion

This chapter considers how to eliminate syntactic complexity from semantically Π_1^μ or *ML* formulas. The core idea is that of *projections* into alternation classes: Given Ψ, its projection Ψ^C into C is a formula in the alternation class C which is equivalent to Ψ if and only if Ψ is semantically in C.

The lowest levels of the L_μ alternation hierarchy are by far the best understood: Unlike higher alternation classes, they are known to have simple characterisations. Modal properties, those that can be expressed without fixpoints, are *local* – they only describe an initial fragment of a structure. Σ_1^μ properties on the other hand are those with finite proofs while Π_1^μ properties have finite counter-proofs. This chapter uses these characterisations to define projections into the low alternation classes. These results have been published with Sandra Quickert at CSL 2015 [LQ15].

5.1 The Modal Fragment

The modal fragment of L_μ consists of formulas without fixpoint operators. Its syntax is in fact identical to Hennessy–Milner logic [HM85]. For modal formulas, model-checking consists of solving a reachability game, which is of linear complexity. The number of nested modal operators in such a formula – its *modal height* – determines how far into a structure a proof or counter-proof can reach. We can say that these formulas are local: whether they hold in a structure depends only on the initial fragment of the structure, up to the modal height of the formula. The same holds for semantically modal formulas, for which the *semantic modal height* determines the height of the initial fragment which can affect whether the formula holds in tree.

This section discusses how the property of being local can be recognised in syntactically complex formulas. Otto [Ott99] showed that properties in *ML* are recognisable via a reduction to S2S, the monadic second order logic over binary trees. Here, the same is achieved within L_μ, by defining a projection into ML: given a guarded formula Ψ, there is a formula Ψ^{ML} in ML such that Ψ is equivalent to Ψ^{ML} if and only if Ψ is semantically a modal formula.

The argument goes roughly as follows. If a formula Ψ is semantically ML, it is local: there is a depth m such that whether a structure $\mathcal{M}$ satisfies Ψ only depends on the initial fragment of $\mathcal{M}$, up to depth m. The value of m can be bounded by an exponential in the size of Ψ. I show that approximating all fixpoints to the m^{th} level and truncating the formula at modal depth m captures the semantics of a formula on structures of height m or less. Then, given the locality of semantically modal formulas, this captures their full semantics.

This projection is exponential in the size of the original formula. Example 81 shows that this is unavoidable: some modal formulas with large modal height can be expressed at least sub-exponentially more concisely using fixpoints.

5.1.1 A bound on modal height

The first two definitions fix the notation for measuring the depth of both structures and formulas. Since measuring how far into a structure a proof can reach is simpler on trees than graphs, in this section all structures are represented as potentially infinite trees.

Definition 67. *(height and depth)* The height of a state without successors is 0. The height of a state with finitely many successors is $r + 1$ where r is the supremum of the heights of the state's successors. The height of a finite tree is the height of its root and corresponds to the length of the longest path in the tree. An infinite tree does not have finite height.

The depth of the root of a structure is 0; for other states, the depth is one greater than the depth of its parent. Thus the depth of the deepest leaf in a structure of height r is r.

Definition 68. *(Modal height)* Given a formula without fixpoint operators, if it does not have any modalities, it is of modal height 0. A modal formula $\langle a\rangle\phi$ or $[a]\phi$ has modal height one greater than ϕ. Other modal formulas have the same modal height as their immediate subformula of the highest modal height. In other words, the modal height of a formula is the maximal number of nested modalities in it.

If a formula is equivalent to a modal formula, its *semantic* modal height is the least modal height of any equivalent modal formula. Only formulas which are semantically modal have a finite semantic modal height.

We use the following lemma, similar to one used by Otto [Ott99], to define *ML* formulas as those which are local. The intuition is that if a formula is equivalent to a *ML*-formula of modal height m, then what happens beyond depth m in a structure can have no effect on whether the formula holds or not.

Definition 69. *(Truncated structures $\mathcal{M}^m$)* Given a structure $\mathcal{M}$, and an integer m, let $\mathcal{M}^m$ be the maximal initial fragment of $\mathcal{M}$ of height m.

Lemma 70. *If ϕ is guarded of semantic modal height m, then $\mathcal{M} \models \phi$ if and only if $\mathcal{M}^m \models \phi$.*

Proof. If ϕ is guarded and of semantic modal height m, there is a formula ψ equivalent to ϕ of syntactic modal height m. Since ψ only has m nested modalities, and no fixpoints, a play in its model-checking game can visit at most the root of a structure and m other distinct states. Then, in the games $\mathcal{M} \times \psi$, only positions $v \times \phi$ where v is no deeper than m are reachable: the reachable fragment of $\mathcal{M} \times \psi$ is identical to $\mathcal{M}^m \times \psi$. Since ψ is equivalent to ϕ, $\mathcal{M} \models \phi$ if and only if $\mathcal{M}^m \models \phi$. □

The next definitions formalise the notion of evaluating fixpoints to their n^{th} level.

Definition 71. *(Approximants)* Approximants are defined syntactically:

$$\begin{aligned}
\mu X^0.\phi(X) &= \phi(\bot) \\
\mu X^n.\phi(X) &= \phi(\mu X^{n-1}.\phi(X)) \\
\nu X^0.\phi &= \top \\
\nu X^n.\phi(X) &= \phi(\nu X^{n-1}.\phi(X))
\end{aligned}$$

Then define ϕ_n to be the formula ϕ where every fixpoint binding μX or νX is substituted with the operator for the n^{th} approximation μX^n or νX^n. Using the notation $\phi[a/b]$ to mean ϕ where all instances of b are substituted with a, and $\mathit{fix}(\phi)$ is the set of fixpoint variables in ϕ, we can write this as

$$\phi_n = \phi[\nu X^n/\nu X; \mu X^n/\mu X]_{\forall X \in \mathit{fix}(\phi)}$$

All fixpoints are approximated simultaneously. For further details see *e.g.* [BS07].

Example 72. The approximant of a reachability condition, such as $\mu X.A \vee \Diamond X$ corresponds to reachability in a fixed number of steps: $A \vee \Diamond\bot$, $A \vee \Diamond(A \vee \Diamond\bot)$, $A \vee \Diamond(A \vee \Diamond(A \vee \Diamond\bot))$ and so on. Dually the n^{th} approximation of a safety condition only dictates safety for n steps.

Approximants can be easily understood in terms of the parity games they generate: instead of a fixpoint variable generating a cycle in a parity game, the parity game now only allows each fixpoint variable to be seen a set number of times n. If this is tracked with counters decremented each time their fixpoint is seen, seeing a more significant fixpoint will reset the counter value. If a counter reaches zero, that is to say a fixpoint is seen more than n times without a higher priority being seen in between, then the player of the parity of the fixpoint variable wins the play.

Then, for trees of bounded height n, ϕ agrees with its n^{th} approximant, ϕ_n.

Lemma 73. *If ϕ is guarded, ϕ and ϕ_n agree on trees of height bounded by n.*

Proof. Write all the fixpoint bindings ϕ_X in ϕ as $\phi_X(\phi_X(...\phi_X(\mu X.\phi_X(X))))$, to obtain a formula which only differs syntactically from ϕ_n at modal depth greater than n. On trees of height n or less, the model-checking game will never reach this point for either formula: the model-checking games for ϕ and ϕ_n are effectively identical on trees of height n or less. $\mathcal{M} \models \phi$ if and only if $\mathcal{M} \models \phi_n$ on n-height bounded structures $\mathcal{M}$.

□

To get a modal formula which agrees with the original formula on *all* trees, the approximated formula has to be truncated at modal depth m.

Let the formula ϕ^n be the one obtained from a modal formula ϕ by replacing subformulas $[a]\psi$ at modal depth n with $\top$ and $\langle a\rangle\psi$ at modal depth n with $\bot$.

Example 74. Consider the modal formula $\phi = A_0 \land \Diamond A_1 \land \Box(\Diamond\Diamond A_2 \lor \Box A_3) \land \Box\Box\Box A_4$. Then $\phi^2 = A_0 \land \Diamond A_1 \land \Box(\Diamond\bot \lor \Box A_3) \land \Box\Box\top$ is true in $\mathcal{M}$ if and only if its initial tree of height 2 satisfies ϕ. Similarly $\phi^1 = A_0 \land \Diamond A_1 \land \Box(\bot \lor \top) \land \Box\top$ is true in $\mathcal{M}$ if its initial tree of height 1 satisfies ϕ. Finally $\phi^0 = A_0 \land \bot \land \top = \bot$, which is consistent with the fact that the root of $\mathcal{M}$ cannot satisfy $\Diamond A_1$ without having any successors.

Lemma 75. *Let Φ be guarded. Then $\mathcal{M} \models \phi^n$ if and only if $\mathcal{M}^n \models \phi$ where $\mathcal{M}^n$ is the infinite tree of $\mathcal{M}$ truncated at depth n. That is to say, ϕ^n is true in $\mathcal{M}$ if and only if the initial tree of height n of $\mathcal{M}$ satisfies ϕ.*

Proof. First note that in the model checking parity games of modal formulas, a state at depth n can only be reached at a subformula that is itself at modal depth n. Assuming $\mathcal{M} \models \phi^n$, Even has a winning strategy σ in $\mathcal{M} \times \phi^n$ to prove it. This game is identical to $\mathcal{M} \times \phi$ until a position s at depth n is reached at $\bot$ or $\top$ instead of $\langle a\rangle\psi$ or $[a]\psi$ respectively. If Even can win, her strategy cannot reach any position $(s, \bot)$. The game $\mathcal{M}^n \times \phi$ is also identical to $\mathcal{M} \times \phi$ until a position at depth n is reached. The strategy σ is winning in $\mathcal{M}^n \times \phi$ since it can avoid $(s, \langle a\rangle\psi)$ positions where s is at depth n and positions $(s, [a]\psi)$ are automatically winning for states s at depth n since they have no successors in $\mathcal{M}^n$.

Conversely, assume Even has a winning strategy σ in $\mathcal{M}^n \times \phi$. She can use this strategy in $\mathcal{M} \models \phi^n$ until it reaches positions (s, ψ) where s is at depth n. Since these are leaves, her winning strategy does not reach any state $(s, \langle a \rangle \psi)$ where s is at depth n. In $\mathcal{M} \models \phi^n$ her strategy σ therefore only reaches final positions (s, P) and $(s, \top)$ where s is at depth n, which are winning for her. A strategy is therefore winning in $\mathcal{M} \times \phi^n$ if and only if it is winning in $\mathcal{M}^n \times \phi$ and therefore $\mathcal{M} \models \phi^n$ if and only if $\mathcal{M}^n \models \phi$.

□

It is time to show that if ϕ is of semantic modal height m, then it is equivalent to the formula ϕ^m_m, that is to say, the formula where fixpoints are approximated to the m^{th} stage of induction as detailed in Definition 71, then truncated at modal depth m.

Theorem 76. *If ϕ is guarded of semantic modal height m, then ϕ is equivalent to ϕ^m_m.*

Proof. Let ϕ be a guarded L_μ formula equivalent to a modal formula of modal height m. The following are equivalent:

(1) $\mathcal{M} \models \phi$

(2) $\mathcal{M}^m \models \phi$

(3) $\mathcal{M}^m \models \phi_m$

(4) $\mathcal{M} \models \phi^m_m$

The conditions (1) and (2) are equivalent since ϕ is semantically modal of height m, as per Lemma 70: ϕ is local, so only the initial tree of a structure up to the modal height affects whether ϕ holds. The conditions (2) and (3) are equivalent since ϕ and ϕ_m agree on structures of height m or less, from Lemma 73. Finally, the equivalence of (3) and (4) comes from Lemma 75. □

Next we aim to show that m can be calculated from ϕ, using a classical pumping argument. The argument relies on labelling the states of structures with the subformulas of ϕ they satisfy. Then, the successors of a state can freely be changed, as long as the set of successor-labels remains the same, without affecting the formulas the state satisfies. If two structures only differ at very high depth, but one satisfies ϕ and the other one does not, then the state labels must repeat themselves before the structures differ. We duplicate a portion of the branch leading to the difference in order to create structures which are differentiated even deeper. Since the sets of successor labels are kept constant during this transformation, it remains the case that only one of the two structures satisfies ϕ. Repeating this process creates pairs of structures with a larger and larger identical initial fragment, but which still disagree on ϕ, showing that ϕ is not semantically modal. Therefore, if a formula is modal, its modal height cannot be deeper than the point at which the state labels need to start repeating themselves. $2^{2|\phi|} + 1$ is an upper bound for that point.

The next lemma uses the fact that in any tree, a subtree rooted at s can be replaced with a distinct subtree rooted at r without affecting the subformulas of Ψ satisfied above depth s, as long as the subtrees rooted at s and r agree on all subformulas of Ψ. For a proof, see for example [Koz88]. The intuition is that whether a state satisfies a subformula of Φ depends only on the propositional variables that a state satisfies and the subformulas satisfied by its successor states.

Definition 77. Let $\mathcal{M}$ be an infinite tree and t be a state of $\mathcal{M}$, and let $\Psi \in L_\mu$. We denote by $\alpha^\Psi_\mathcal{M}(t)$ the set of subformulas of Ψ satisfied by the state t in $\mathcal{M}$.

Lemma 78. *(Consistent labelling)*

Let $\mathcal{M} = (M, E_M, P_M, L_M)$ be a tree rooted at i_M, and let r be the root of a disjoint structure $\mathcal{M}' = (M', E_{M'}, P_{M'}, L_{M'})$ such that $\alpha^\Psi_\mathcal{M}(s) = \alpha^\Psi_{\mathcal{M}'}(r)$ for some state s of $\mathcal{M}$. Let v be the predecessor of s in $\mathcal{M}$ via some action $a \in Act$.

Then, consider the tree $\mathcal{N}$ which is as $\mathcal{M}$, but with the subtree rooted at s replaced by the tree $\mathcal{M}'$ rooted at r. More precisely, $\mathcal{N} = (N, E_N, P_N, L_N)$ where:

- *$N = (M \setminus \{u \in M \mid u \text{ is } s \text{ or a descendant thereof}\}) \cup M'$;*
- *$E_N = (N \times N \restriction E_M) \cup E_{M'} \cup \{e'\}$ as set of edges, where $\restriction$ denotes restriction;*
- *L_N and P_N are inherited from $P_M, P_{M'}, L_M$ and $L_{M'}$ and $L_N(e') = L_M(e)$.*

Since $N \subseteq M \uplus M'$ (where $\uplus$ denotes disjoint union), the labelling $(\alpha^\Psi_M \cup \alpha^\Psi_{M'}) \restriction N$ is defined on all states of $\mathcal{N}$ as well. Then for all $s \in N$ we have $(\mathcal{N}, s) \models \phi$ if and only if $\phi \in ((\alpha^\Psi_M \cup \alpha^\Psi_{M'}) \restriction N)(s)$, meaning that $(\alpha^\Psi_M \cup \alpha^\Psi_{M'}) \restriction N$ is identical to α^Ψ_N.

Lemma 79. *Let ϕ be guarded and semantically modal, i.e. ϕ is equivalent to a formula in ML. Then the semantic modal height m of ϕ is bounded above by $2^{2|\phi|}$.*

Proof. Assume $m > 2^{2|\phi|}$ to be the semantic modal height of ϕ. Then there exists a tree $\mathcal{M}$ of height $2^{2|\phi|}$ which is the prefix of two models $\mathcal{M}_1$ and $\mathcal{M}_2$ such that $\mathcal{M}_1 \models \phi$ and $\mathcal{M}_2 \not\models \phi$. That is to say, for every state s of $\mathcal{M}$, there are states s_1 and s_2 in $\mathcal{M}_1$ and $\mathcal{M}_2$ respectively such that s, s_1, and s_2 agree on propositions and for all inner nodes of $\mathcal{M}$, s' is an a-successor of s if and only if s'_1 is an a-successor of s_1, if and only if s'_2 is an a-successor of s_2. If d is maximal such that $\mathcal{M}_1$ and $\mathcal{M}_2$ agree up to depth d, write $agree(\mathcal{M}_1, \mathcal{M}_2) = d$. To start with, $agree(\mathcal{M}_1, \mathcal{M}_2) > 2^{2|\phi|}$ since $\mathcal{M}_1$ and $\mathcal{M}_2$ agree on their prefix $\mathcal{M}$ of height $2^{2|\phi|}$.

Label every state s of $\mathcal{M}$ with a set $\alpha^\phi_{M_1}(s)$ consisting of subformulas of ϕ which are true in s_1 of $\mathcal{M}_1$ and a set $\alpha^\phi_{M_2}(s)$ consisting of subformulas of ϕ which are true in s_2 of $\mathcal{M}_2$. For each branch of $\mathcal{M}$, if the branch is longer than $2^{2|\phi|}$, there are two states a, b in $\mathcal{M}$ along the branch such that $\alpha^\phi_{M_1}(a) = \alpha^\phi_{M_1}(b)$ and $\alpha^\phi_{M_2}(a) = \alpha^\phi_{M_2}(b)$. For each branch i, choose b^i to be the first state on a branch which has an ancestor a^i such that

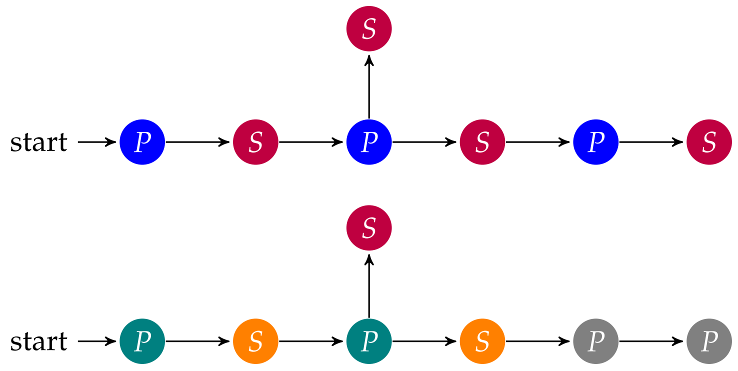


Figure 5.1.1: **Before:** The labels P and S represent the propositional variables that hold at a node. The two illustrations above represent the initial fragment of two structures which agree to start with, but disagree at some point. The colours represent different values of α_M^ϕ. The fragment between the second and fourth nodes can be duplicated to create structures which agree on a larger initial fragment.

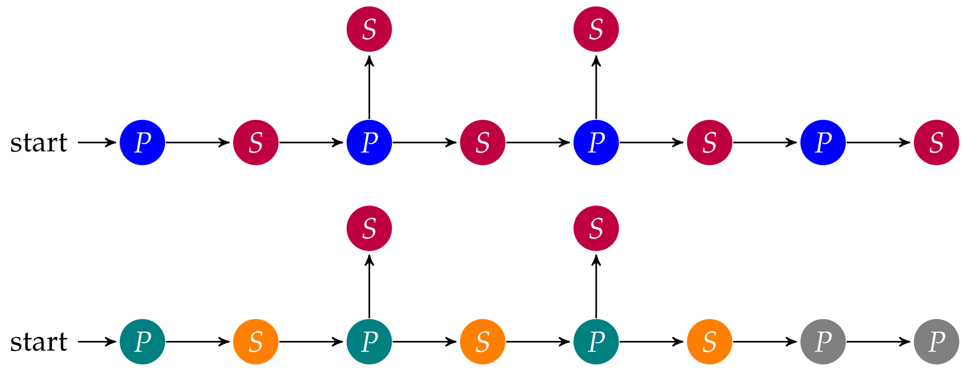


Figure 5.1.2: **After:** The new structures agree with their originals on all relevant formulas, but they agree with each other on a larger initial segment.

$\alpha^{\phi}_{M_1}(a^i) = \alpha^{\phi}_{M_1}(b^i)$ and $\alpha^{\phi}_{M_2}(a^i) = \alpha^{\phi}_{M_2}(b^i)$. Note that for any pair of branches i and j, either $b^i = b^j$ or b^i and b^j are not reachable from one another.

For each branch i and its states a^i and b^i, let ${a^i}'_1$ be the root of a distinct copy of the subtree in $\mathcal{M}_1$ rooted at a^i. Similarly, let ${a^i}'_2$ be the root of a distinct copy of the subtree rooted at a^i in $\mathcal{M}_2$. Let $\mathcal{M}'_1$ be obtained from $\mathcal{M}_1$ by replacing, for each branch i the state b^i with ${a^i}'_1$ and its induced subtree; similarly, let $\mathcal{M}'_2$ be obtained from $\mathcal{M}_2$ by replacing b^i with ${a^i}'_2$ and its induced subtree. Note that these transformations do not affect each other: each b^i is on a distinct branch and is replaced with a copy of a subtree of the original structure. Since $\alpha^{\phi}_{M_1}({a^i}'_1) = \alpha^{\phi}_{M_1}(b^i)$ and $\alpha^{\phi}_{M_2}({a^i}'_2) = \alpha^{\phi}_{M_2}(b^i)$, all states preserve their labels and we know that $\mathcal{M}'_1 \models \phi$ and $\mathcal{M}'_2 \not\models \phi$, from Lemma 78.

We now show that if $\mathcal{M}_1$ and $\mathcal{M}_2$ agree up to depth d, then $\mathcal{M}'_1$ and $\mathcal{M}'_2$ agree up to depth $d+1$. Let i be a branch in $\mathcal{M}$ of length d such that i is extended differently in models $\mathcal{M}_1$ and $\mathcal{M}_2$. Since $depth(b^i) > depth(a^i)$ the models $\mathcal{M}'_1$ and $\mathcal{M}'_2$ agree along all extensions of i to depth $d - depth(a^i) + depth(b^i) > d$. That is to say $\mathcal{M}'_1$ and $\mathcal{M}'_2$ agree at least up to $d+1$. This establishes $agree(\mathcal{M}'_1, \mathcal{M}'_2) > agree(\mathcal{M}_1, \mathcal{M}_2)$. In $z = m - d$ many steps, we will reach models $\mathcal{M}^z_1$ and $\mathcal{M}^z_2$ such that $agree(\mathcal{M}^z_1, \mathcal{M}^z_2) \geq m$ but $\mathcal{M}^z_1 \models \phi$ and $\mathcal{M}^z_2 \not\models \phi$. This contradicts m being the modal height of ϕ. □

Corollary 80. *Whether a guarded formula ϕ is equivalent to a modal formula can be decided by testing whether ϕ is equivalent to $\phi_{2^{|\phi|}}$.*

Proof. From the previous lemma, if a formula ϕ is modal, its semantic modal height m is no greater than $2^{2|\phi|}$. If $\phi \neq \phi_{2^{|\phi|}}$, then ϕ must disagree with $\phi_{2^{|\phi|}}$ on some structure $\mathcal{M}$. However, ϕ and $\phi_{2^{|\phi|}}$ agree on structures of height bounded by $2^{|\phi|}$ so $\mathcal{M}$ must be of height over $2^{|\phi|}$, which, since $m \leq 2^{|\phi|}$, contradicts the fact that the modal height of ϕ is m. □

This projection requires an exponential blow-up. Interestingly, this is necessary as formulas with fixpoints can indeed be at least sub-exponentially more compact than the equivalent modal formulas. The following example exhibits syntactically Σ^{μ}_1 but semantically modal formulas with modal height sub-exponential in the length of the Σ^{μ}_1 formula. The idea of these formulas is to require a series of propositional variables to occur at different frequencies until they all occur at the same time. The modal height of the formula is then the least common multiple of the frequencies, which grows exponentially.

Example 81. There is a family of formulas $\Phi(n) \in \Sigma^{\mu}_1$ which are semantically modal but have modal height $\Omega(2^{\sqrt{l}})$ in the length l of $\Phi(n)$.

Proof. Write $\Box^n$ for $\Box ... \Box$ repeated n times. The formula $\mu X. \Box^n (A \wedge (X \vee B))$ states that A occurs every n^{th} state on any path until B also occurs at a state whose depth is a multiple of n. By combining such formulas we can write $[\mu X.\Box^a A \wedge (X \vee (B \wedge C))] \wedge [\mu X.\Box^b B \wedge (X \vee (A \wedge C))] \wedge [\mu X.\Box^c C \wedge (X \vee (A \wedge B))]$ which sets the frequencies at which A, B and C are seen to a, b and c respectively, until they are seen simultaneously. This formula is modal since if it is true, at the latest at depth $a \times b \times c$, all of A, B and C are seen simultaneously. More precisely, its modal height is the least common multiple of a, b and c.

Generalising this, for a fixed n, let $\psi_d = \mu X.\Box^d(P_d \wedge X) \vee (\bigwedge_{i \leq n} P_i)$ be the formula stating that the proposition P_d occurs at frequency d until all propositions P_i for $i \leq n$ occur at the same time, at a depth multiple of d. Now, let $\Phi(n) = \bigwedge_{d \leq n} \psi_d$. The modal height of $\Phi(n)$ is the least common multiple of the integers up to n, written $lcm(n)$. For sufficiently large n, $lcm(n) > 2^n$ [Nai82] so the formula $\Phi(n)$ is of length $O(n^2)$ and has semantic modal height $\Omega(2^n)$. □

5.2 The Π_1^μ and Σ_1^μ fragments

This section takes a syntactic approach to eliminating complexity from semantically Π_1^μ or Σ_1^μ formulas. The main result is a projection into Π_1^μ: a transformation from an arbitrary formula Ψ into a syntactically Π_1^μ formula $\Psi^{\Pi_1^\mu}$ which is equivalent to Ψ if and only if Ψ is semantically in Π_1^μ.

The Σ_1^μ fragment of L_μ, which only allows the fixpoint μ, can be thought of as the formulas expressing reachability conditions of some sort. While *ML* formulas are local, or with bounded proofs, the key property of Σ_1^μ formulas is that they have *finite proofs*: if a Σ_1^μ formula holds in a structure $\mathcal{M}$, then some initial tree of $\mathcal{M}$ is sufficient to prove it. All completions of that initial tree must also satisfy the formula. Dually, Π_1^μ formulas have *finite counter-proofs*: an initial segment of a counter-model is sufficient to show that the property fails. Unlike in the modal case, the depth of these initial fragments is not bounded.

It is useful to distinguish between a formula having finite syntactic proofs, and finite semantic proofs: formulas syntactically in Σ_1^μ, such as $\mu X.A \vee \Diamond X$, have finite syntactic proofs – namely Even's winning strategy in the model checking game. However, there are also formulas such as $(\nu X.A \wedge \Box X) \vee (\mu Y.\Diamond Y \vee \neg A)$. This formula is tautological, yet in some models requires infinite proofs: if $\neg A$ is not reachable and the model has an infinite branch, then the proof must show that A always holds. Yet this formula clearly (and trivially) has the property that its models have an initial fragment of which all completions are also models. This property is a semantic characteristic of a formula, rather than just a syntactic one. Indeed, *semantically* Σ_1^μ properties correspond exactly to those for which all completions of a sufficiently large initial fragment of a model are also models.

The question is how to recognise this property in formulas without finite syntactic proofs. Better yet, how to turn a formula with finite semantic proofs into one with finite syntactic proofs?

The solution presented in this chapter is syntactic, based on disjunctive normal form. It turns out that once a formula is in disjunctive normal form, and unsatisfiable subformulas are eliminated, the property of having finite semantic counter-proofs – *i.e.* all completions of a large enough initial fragment of a counter-model are also counter-models – corresponds exactly to having finite syntactic counter-proofs. This is true even though semantically Π_1^μ formulas may have μ-operators. When this is a case for a disjunctive formula, the falsifying player, Odd, can always disprove a counter-model without using a μ-operator infinitely often. The only exception to this rule is when a μ binding is unsatisfiable, in which case it can be replaced with $\bot$.

This observation yields a very simple projection into Π_1^μ: every μ-operator is replaced with ν if it binds a satisfiable formula and $\bot$ otherwise.

Related work Küsters and Wilke showed [KW02] that the problem of deciding whether a property of L_μ can be expressed with only least fixpoints, or, by duality, only greatest fixpoints, is EXPTIME-complete. Their proof first constructs a bottom-up tree automaton the states of which correspond to sets of subformulas based on the L_μ formula. Roughly speaking, the bottom-up automaton accepts a structure if it has an initial fragment such that every completion admits a valid assignment of automaton states to its nodes. This automaton is closed under bisimulation if and only if it is Σ_1^μ definable and equivalent to the original formula.

Some elements of this proof are recognisable in the transformation presented in this section. The idea of finite semantic proofs – that all completions of a large enough initial fragment of a model are also models – is reminiscent of the acceptance condition of the bottom-up automata, and the power-set construction is mirrored in the usage of disjunctive formulas.

The rest of this section is organised as follows: Section 5.2.1 formalises the notion of finite counter-proofs and uses it to characterise Π_1^μ; Section 5.2.2 describes the transformation which turns a disjunctive formula into a syntactically Π_1^μ formula, and proves that it preserves meaning for semantically Π_1^μ properties.

5.2.1 Π^μ_1 has finite counter-proofs

This section characterises properties in Σ^μ_1 and Π^μ_1 as properties with finite proofs and counter-proofs respectively. Informally, μ-formulas express finite behaviour such as reachability – proofs of such properties are finite: once the desired state is reached, the rest of the structure is irrelevant. Dually, ν-formulas express infinite behaviour and if a structure fails to display infinite behaviour, the state at which it fails must be finitely reachable.

Lemma 82. *Let $\mathcal{M}$ be a structure with finite branching such that $\mathcal{M} \not\models \Psi$. If Ψ is semantically in Π^μ_1, then there is some n such that for any structure $\mathcal{M}'$, if $\mathcal{M}'$ agrees with $\mathcal{M}$ up to depth n, then $\mathcal{M}' \not\models \Psi$.*

Proof. Assume Ψ is semantically in Π^μ_1 and Φ is the equivalent formula with no least fixpoints. Since $\mathcal{M} \not\models \Phi$, Even has a winning strategy in $\mathcal{M} \times \neg\Phi$. The formula $\neg\Phi$, the negation of Φ, is a formula without greatest fixpoints. This means that Even has a strategy σ winning in $\mathcal{M} \times \neg\Phi$ which only agrees with finite plays, as an infinite play without ν-variables would be losing for Even. Let n be the depth of the furthest state in $\mathcal{M}$ which σ reaches – since $\mathcal{M}$ has finite branching, there is such an n. For any $\mathcal{M}'$ which agrees with $\mathcal{M}$ up to n, the strategy σ is still winning for Even, so $\mathcal{M}' \not\models \Psi$. □

This Lemma describes the characterisation of semantically Π^μ_1 formulas: A counter-model of a Π^μ_1 formula has an initial fragment such that all its completions are also counter-models.

5.2.2 The formula $\Psi^{\Pi^\mu_1}$

This section presents the projection into Π^μ_1 and proves the core theorem of this chapter: any semantically Π^μ_1 formula in disjunctive form can be transformed into an equivalent syntactically Π^μ_1 formula, by changing every occurrence of μ into either ν or $\bot$.

Example 83.

$$(\mu X.A{\rightarrow}\{X\}) \vee (\mu Y.B{\rightarrow}\{Y,\top\} \vee C) \vee (\nu Z.B{\rightarrow}\{Z,\top\})$$

The first disjunct $\mu X.A{\rightarrow}\{X\}$ is unsatisfiable, so it can be replaced with $\bot$. Then, looking at $(\mu Y.B{\rightarrow}\{Y,\top\} \vee C) \vee (\nu Z.B{\rightarrow}\{Z,\top\})$, the remaining formula is satisfied when C is reachable via a path along which B holds, or when there is an infinite path along which B holds. In the model-checking games of this formula, Odd never plays μ infinitely often to win the model-checking game: if there is a B path which never reaches C, such a structure is not a counter-model, since Even can play to $\nu Z.B{\rightarrow}\{Z,\top\}$ and win. The parity of the fixpoint binding Y is inconsequential, and μ can be replaced with ν.

This formula is therefore semantically Π_1^μ, and equivalent to:

$$\bot \vee (\nu Y.B{\rightarrow}\{Y,\top\} \vee C) \vee (\nu Z.B{\rightarrow}\{Z,\top\})$$

Or just $\nu Y.B{\rightarrow}\{Y,\top\} \vee C$ after simplifications.

To prove the main theorem, first select, for each μ-subformula $\mu X.\phi$ in disjunctive Ψ, a structure $\mathcal{M}$ such that whether $\mathcal{M}$ satisfies Ψ depends on certain states satisfying the formula $\mu X.\phi$. This structure exists due to the disjunctivity of Ψ, as shown in Lemma 84. Lemma 85 then shows that if a μ-subformula $\mu X.\phi$ of Ψ is satisfiable, but cannot be replaced with the corresponding ν-formula $\nu X.\phi$, then, for any n, we can build a twin structure for $\mathcal{M}$ agreeing with $\mathcal{M}$ up to n but disagreeing on Ψ. This implies that Ψ is not a Π_1^μ formula, as Lemma 82 shows that Π_1^μ formulas have finite counter-proofs. This leaves us with two scenarios: either the μ-subformula is unsatisfiable, in which case it can be replaced by $\bot$, using Lemma 86, or the μ-formula can be replaced with the corresponding ν-formula. In either case, any semantically Π_1^μ formula can be turned into an equivalent syntactically Π_1^μ formula, by replacing μ-subformulas with either $\bot$ or the dual ν-formula.

Notation Recall that $\Psi(\psi)$ indicates a formula Ψ which contains a subformula ψ and we write $\Psi(\psi')$ for the formula $\Psi[\psi'/\psi]$ in which ψ is substituted with the formula ψ'. This notation is used to define formulas related to Ψ in order to specify structures where the players' strategies must exhibit some desired behaviours. For example, if for some structure $\mathcal{M}$, in which Ψ holds Odd can win $\mathcal{M} \times \Psi(\bot)$, then Even can only win $\mathcal{M} \times \Psi(\mu X.\phi)$ by using a strategy which agrees with a play reaching $\mu X.\phi$.

In the following lemma we show that for a structure to satisfy $\neg\Psi(\phi) \wedge \Psi(\top)$ means that there is a set S of states such that Odd can win the game for $\Psi(\phi)$ only by playing a strategy reaching (s,ϕ) for some s in S. Then, if states of S are substituted with new substructures, Odd may only win if he can win from one of the new substructures at ϕ.

Recall that a structure is well-behaved with respect to a formula if the Even player can win with a strategy which only agrees with one play per branch. A model of a disjunctive formula is bisimilar to a well-behaved model.

Lemma 84. *Let Ψ be a guarded formula in disjunctive form with a subformula ϕ. If $\mathcal{M}$ is a structure such that $\mathcal{M} \models \neg\Psi(\phi) \wedge \Psi(\top)$ and $\mathcal{M}$ is well-behaved for $\Psi(\top)$, then there is a non-empty set of states S in $\mathcal{M}$ such that in $\mathcal{M} \times \Psi(\phi)$ each of Odd's winning strategies reaches (s,ϕ) for some $s \in S$ – that is to say, for each of Odd's winning strategies τ there is* *a counter strategy σ such that (s,ϕ) is on the play $\tau \times \sigma$ for some $s \in S$. Furthermore, if every state s_i of S is replaced with some state t_i, yielding a new model $\mathcal{M}'$, Odd only wins in $\mathcal{M}' \times \Psi(\phi)$ if Odd wins from (t_i,ϕ) in the same game for some t_i.*

Proof. If Even wins $\mathcal{M} \times \Psi(\top)$ but Odd wins $\mathcal{M} \times \Psi(\phi)$, then Odd cannot win $\mathcal{M} \times \Psi(\phi)$ with a strategy which avoids ϕ, otherwise the same strategy would be winning in $\mathcal{M} \times \Psi(\top)$. Let τ be one of Odd's winning strategies in $\mathcal{M} \times \Psi(\phi)$ and let σ be Even's well-behaved winning strategy in $\mathcal{M} \times \Psi(\top)$. Since $\mathcal{M} \times \Psi(\top)$ is identical to $\mathcal{M} \times \Psi(\phi)$ until a play reaches ϕ, the strategy σ is also an initial strategy in $\mathcal{M} \times \Psi(\phi)$, defined until ϕ is reached. The play $\tau \times \sigma$ must reach ϕ because otherwise it would be identical to a play respecting her winning strategy in $\mathcal{M} \times \Psi(\top)$, and therefore winning for Even. Let s_τ be the first state at which the play $\tau \times \sigma$ reaches ϕ in $\mathcal{M} \times \Psi(\phi)$. Then $S = \{s_\tau | \tau \text{ is a winning strategy for Odd } \}$ is the set such that in $\mathcal{M} \times \Psi(\phi)$ each of Odd's winning strategies reaches (s,ϕ) for some $s \in S$.

For the second part of the lemma, first observe that if Even wins from (t_i,ϕ) for all i, then Odd cannot use any of his winning strategies from $\mathcal{M} \times \Psi(\phi)$ to win in $\mathcal{M}' \times \Psi(\phi)$ since if Even initially plays according to σ, the play reaches (t_i,ϕ) from where Even has a winning strategy. As a result, Odd cannot avoid all t_i without losing. Since σ is well-behaved at least until it reaches ϕ, not only can Odd not avoid all t_i, Odd cannot avoid all (t_i,ϕ) without losing. Hence, if Odd loses from (t_i,ϕ) for all i, Odd loses in $\mathcal{M}' \times \Psi(\phi)$. □

We can now prove the main result: to obtain the syntactically Π_1^μ formula equivalent to a semantically Π_1^μ formula in guarded disjunctive form, it is sufficient to replace each least fixpoint with either $\bot$ or a greatest fixpoint. The crux is to show that each μ-binding in a semantically Π_1^μ formula can either be replaced by $\bot$ or ν. The following lemma identifies two cases. The first is that the subformula $\mu X.\phi$ is unsatisfiable in the sense that there is no structure $\mathcal{T}$ from the root of which Even can win at $\mu X.\phi$ in $\mathcal{T} \times \Psi(\mu X.\phi)$. Then it can be replaced with $\bot$. In the other case, $\mu X.\phi$ can be replaced with $\nu X.\phi$.

Lemma 85. *If $\Psi(\mu X.\phi)$, a guarded formula in disjunctive form with a subformula $\mu X.\phi$, is semantically in Π_1^μ, then either there is no structure $\mathcal{T}$ such that Even wins from $(r_0,\mu X.\phi)$ in $\mathcal{T} \times \Psi(\mu X.\phi)$ where r_0 is the root of $\mathcal{T}$ (*i.e.* $\mu X.\phi$ is unsatisfiable), or $\Psi(\mu X.\phi)$=$\Psi(\nu X.\phi)$.*

Proof. Assume that $\Psi(\mu X.\phi) \neq \Psi(\nu X.\phi)$ and that there is a structure $\mathcal{T}$ such that Even wins from $(r_0,\mu X.\phi)$ in $\mathcal{T} \times \Psi(\mu X.\phi)$ where r_0 is the root of $\mathcal{T}$. Since $\Psi(\mu X.\phi)$ implies $\Psi(\nu X.\phi)$ but not the other way around, there is a (finitely branching) structure $\mathcal{M}$ such that $\mathcal{M} \models \neg\Psi(\mu X.\phi) \wedge \Psi(\nu X.\phi)$. The structure $\mathcal{M}$ can be taken to be well-behaved with respect to $\Psi(\nu X.\phi)$. We will show that for any n there is a structure $\mathcal{M}'$ which agrees with $\mathcal{M}$ up to depth n but which satisfies $\Psi(\mu X.\phi)$. Using Lemma 82, this will contradict $\Psi(\mu X.\phi) \in \Pi_1^\mu$.

For any n, we can write $\Psi(\mu X.\phi)$ as $\Psi(\overbrace{\phi...\phi}^{n}(\mu X.\phi))$, where we drop some brackets for readability, so $\phi\phi(X)$ should be understood as $\phi(\phi(X))$. Then, the structure $\mathcal{M}$ satisfies $\neg\Psi(\overbrace{\phi...\phi}^{n}(\mu X.\phi)) \wedge \Psi(\overbrace{\phi...\phi}^{n}(\top))$ since $\overbrace{\phi...\phi}^{n}(\top)$ is implied by $\nu X.\phi$. Furthermore, $\mathcal{M}$ is well-behaved for $\Psi(\overbrace{\phi...\phi}^{n}(\top))$. From Lemma 84 we know that there is a set S of states in $\mathcal{M}$ such that for each $s_i \in S$, Even loses from $(s, \neg\mu X.\phi)$ and if each $s_i \in S$ is replaced with r_0, the root of $\mathcal{T}$, to yield a new model $\mathcal{M}$, then Ψ holds in $\mathcal{M}'$. Furthermore, since X is guarded in $\mu X.\phi$, a play can only reach $\mu X.\phi$ from $\overbrace{\phi...\phi}^{n}(\mu X.\phi)$ at depth at least n: each $s_i \in S$ is at least at depth n therefore $\mathcal{M}'$ agrees with $\mathcal{M}$ up to depth n. We have built for any n, a structure that agrees with $\mathcal{M}$, a counter-model of Ψ, up to n but satisfies Ψ. This contradicts the assumption that Ψ is in Π_1^μ, and has finite counter-proofs using Lemma 82. □

It now suffices to show that if a subformula $\mu X.\phi$ is unsatisfiable in the sense that there is no structure $\mathcal{T}$ from the root of which Even can win at $\mu X.\phi$ in $\mathcal{T} \times \Psi(\mu X.\phi)$, then $\mu X.\phi$ can be replaced with $\bot$. This is trivial if $\mu X.\phi$ has no free variables, as then it is undoubtedly equivalent to $\bot$. Here this is generalised to $\mu X.\phi$ where ϕ may have free variables bound higher up in Ψ.

Lemma 86. *If there is no structure $\mathcal{T}$ rooted at t_0 such that Even wins from $(t_0, \mu X.\phi)$ in $\mathcal{T} \times \Psi(\mu X.\phi)$, then $\Psi(\mu X.\phi) = \Psi(\bot)$.*

Proof. If Even wins $\mathcal{M} \times \Psi(\mu X.\phi)$ but there is no $\mathcal{T}$ rooted at t_0 such that Even wins from $(t_0, \mu X.\phi)$, then Even's winning strategy cannot reach any position $(s, \mu X.\phi)$. Then the same strategy can be used in $\mathcal{M} \times \Psi(\bot)$ to avoid any position $(s, \bot)$. Since these two games are identical up until $\mu X.\phi$ or $\bot$ is reached, Even also wins in $\mathcal{M} \times \Psi(\bot)$. This shows $\Psi(\mu X.\phi)$ implies $\Psi(\bot)$. The other direction is trivial since $\bot$ implies $\mu X.\phi$ and L_μ is monotone. □

Theorem 87. *If Ψ is a formula in guarded disjunctive form and semantically in Π_1^μ, then either $\Psi = \Psi[\bot/\mu X.\phi]$ or $\Psi[\nu X.\phi/\mu X.\phi]$ for any subformula $\mu X.\phi$ of Ψ.*

Proof. If there is no structure $\mathcal{T}$ such that Even wins from $(r_0, \mu X.\phi)$ in $\mathcal{T} \times \Psi(\mu X.\phi)$ where r_0 is the root of $\mathcal{T}$, then from the previous lemma, $\Psi = \Psi[\bot/\mu X.\phi]$. If there is such a structure, then from Lemma 85 we know that $\Psi = \Psi[\nu X.\phi/\mu X.\phi]$. □

Corollary 88. Π_1^μ *and by duality* Σ_1^μ *are decidable.*

Proof. Any formula Ψ of L_μ can be turned into a guarded formula in disjunctive form. Then, if Ψ is semantically in Π^μ_1, every occurrence of $\mu X.\phi$ can be eliminated either by replacing it with $\bot$ or $\nu X.\phi$. Hence to decide whether a formula is semantically in Π^μ_1, it is sufficient to decide whether it is equivalent to the formula where each satisfiable $\mu X.\phi$ formula is replaced with $\nu X.\phi$.

If a formula is in a alternation class Π^μ_n, its syntactic negation is in Σ^μ_n and vice-versa. Therefore, to decide whether a formula is semantically in Σ^μ_1 it is sufficient to decide whether its negation is in Π^μ_1. If this is the case, the Π^μ_1 formula can be syntactically negated to yield a formula in Σ^μ_1. □

This concludes the argument that Π^μ_1 is decidable via formula construction. The projection into Π^μ_1 preserves or reduces the size of a disjunctive formula, although the transformation into disjunctive form can incur an exponential blow-up in the size of the formula.

5.3 Discussion

This chapter has provided two novel, formula-focused decision procedures for the first two levels of the alternation hierarchy. The procedure for Π^μ_1 is particularly interesting because of how simple it is. Discounting unsatisfiable subformulas, all fixpoints in a disjunctive formula are just turned into ν and the resulting formula is equivalent to the original if and only if the original formula is semantically in Π^μ_1.

As this chapter demonstrates, disjunctive form is a remarkably powerful tool for looking at the alternation hierarchy. Perhaps it is then unsurprising that the Rabin–Mostowski index problem for non-deterministic parity automata, the automata-theoretic cousin of disjunctive form, has seen more progress [CL08] than the index problem for non-amorphous alternating parity automata. In the chapters to follow, disjunctive form will remain a key ingredient.

Chapter 6

Σ_2^μ is decidable for Π_2^μ

After considering syntactic alternatives to the existing decision procedures for ML, Σ_1^μ and Π_1^μ, it is time to move onto the next alternation level, Σ_2^μ, for which the decidability so far has been open. This chapter argues that given a Π_2^μ formula, it is decidable whether an equivalent Σ_2^μ formula exists.

The approach to decidability is still in many ways similar to the one in the last chapter: characterising the target class by defining a projection. However, given the increased complexity of the problem, the projection is parameterised. That is to say, for any L_μ formula Ψ, this chapter defines a projection $\Psi^{\Sigma_2^\mu}(n)$ such that that Ψ is semantically in Σ_2^μ if and only if it is equivalent to $\Psi^{\Sigma_2^\mu}(n)$ for some n. The idea is to construct a formula equivalent to Ψ in which all alternations between μ and a more significant ν are captured by one single such alternation. Then, whether Ψ is semantically in Σ_2^μ depends exactly on whether this alternation is necessary, or whether it can be approximated.

The decision procedure for Π_2^μ input formulas comes from generalising techniques from automata theory to bound the the parameter n, corresponding to the depth of the approximation. These techniques come from a line of inquiry closely related to the L_μ alternation hierarchy: the Rabin–Mostowski index hierarchy of non-deterministic automata. As detailed in Chapter 4, the Rabin–Mostowski hierarchy problem is not the same problem as the L_μ index problem; however, in this instance, automata-theoretic techniques generalise well into the L_μ setting.

Colcombet and Lödig reduced the Rabin–Mostowski index problem to the uniform universality problem for distance parity automata [CL08]. This was in turn solved for Büchi definable languages [CKLV13]: given a Büchi definable language, it is decidable whether it can be described by a co-Büchi automaton. Skrzypczak and Walukiewicz gave an alternative proof of the same result [SW16], while adding a topological characterisation of recognised languages.

In L_μ-terms, this result corresponds to deciding, given a Π^{μ}_{2} formula, whether it is equivalent, with respect to ranked trees only, to a Σ^{μ}_{2} formula. This section lifts this result to deciding whether a Π^{μ}_{2} formula is equivalent to a Σ^{μ}_{2} formula. In other words, it provides a novel reduction from the decidability of Σ^{μ}_{2} to a boundedness question, then decides the boundedness question for Π^{μ}_{2} inputs by applying automata-theoretic techniques in the L_μ setting.

The proof consists of two parts: (a) defining for any input formula Ψ asequence of Σ^{μ}_{2} formulas $\Psi(n)$ which approximate Ψ and which, if Ψ is semantically Σ^{μ}_{2}, eventually coincide with Ψ, and (b) bounding the parameter n such that $\Psi(n) = \Psi$ for Ψ in Π^{μ}_{2}. Part (a) is more general than required for the namesake result of this chapter. This means that generalising (b) to arbitrary input formulas would suffice to decide Σ^{μ}_{2}.

The contributions of this chapter can be summarised as follows:

- A game construction similar to those found in the literature [CKLV13, SW16], generalised into the L_μ setting and arbitrary formulas;
- A reduction of the decidability of Σ^{μ}_{2} to deciding whether the sequence of ${}^{\mu}_{2}$ approximants of Ψ eventually coincides with Ψ, or, equivalently, whether for some n the n-challenge game construction corresponds to the model-checking game of a formula. This yields a comparatively simple and more general alternative for the most involved part of the proof by Skrzypczak and Walukiewicz [SW16];
- A generalisation of the main theorems from Colcombet and Lödig 2013 [CKLV13] and Skrzypczak and Walukiewicz 2016 [SW16] to unranked structures: it is decidable whether a Σ^{μ}_{2} formula is equivalent to a Π^{μ}_{2} formula, and vice versa.

The first two points make up the core technical contribution of this chapter. They provide a more general account of the decidability of Σ^{μ}_{2} than had previously been seen. The third point simply shows how this can be used to generalise previous automata-theoretic work to yield the namesake result of this chapter.

The pattern of this proof, as well as those in the previously cited works, is familiar: first, the semantic complexity of the language is reduced to a boundedness question, which is then solved for a fragment of the input space. The same schema also describes the decision procedure for *ML* from the previous chapter. This idea of boundedness can be traced all the way back to datalog [Var88], and more recently the star-height problem [Kir05], another cousin of the L_μ-index problem. While this chapter focuses on deciding Σ^{μ}_{2} for input in Π^{μ}_{2}, the final part of this thesis will revisit the question of boundedness as a uniform syntactic approach to the L_μ alternation hierarchy, featuring in particular a generalisation of the construction of Section 6.1 to characterise arbitrary alternation classes of disjunctive L_μ.

The work in this chapter is based on joint work with Sandra Quickert.

6.1 A challenge game

The first part of this chapter's decidability proof defines a parametrised game called an *n-challenge game* on a parity game arena. This game consists of a parity game augmented with some additional structure intended to capture whether Odd can force alternations between even and odd priorities. For each finite n, the n-challenge game is described by a Σ_2^μ formula $\Psi(n)$ which holds in $\mathcal{M}$ if and only if Even wins the n-challenge game on $\mathcal{M} \times \Psi$. Furthermore, a disjunctive formula Ψ is semantically in Σ_2^μ if and only if there is some m such that Ψ is equivalent to $\Psi(m)$. In other words, the sequence of approximations $\Psi(m)$ converges if and only if Ψ is in Σ_2^μ.

Similar constructions for non-deterministic Büchi input automata on labelled binary trees are found in the literature [CKLV13, SW16]. The construction as described here is more general as it allows, for now, arbitrary input formulas and operates on unranked structures (see Section 2.1.3). For the main result of this chapter, the construction for Π_2^μ formulas is sufficient. However, the proposed generalisation may be of interest, in particular if the second part of the proof can also be generalised to arbitrary input formulas. This would decide Σ_2^μ in general, and by duality Π_2^μ.

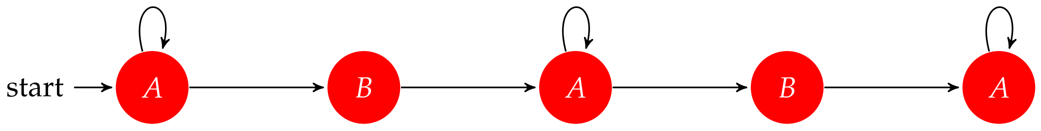
The proof of Theorem 94 is based on previous unpublished work by Sandra Quickert on the challenge game for Π_2^μ formulas; the version here, generalised to arbitrary input, stems from joint work.

Fix a formula Ψ in disjunctive form, of index $\{q,...,0\}$. Let $I = \{q,...,0\}$ if q is even and $\{q+1,q,...,0\}$ otherwise. Write I_e for the even priorities in I.

The n-challenge game consists of a normal parity game augmented with a set of challenges, one for each even priority i. A challenge can either be *open* or *met* and has a counter c_i attached to it. Each counter is initialised to n, and decremented when the corresponding challenge is opened. The Odd player can at any point open challenges of which the counter is non-zero, but he must do so in decreasing order: an i-challenge can only be opened if every j-challenge for $j > i$ is opened. When a play encounters the priority j while the j-challenge is open, the challenge is said to be met. All i-challenges for $i < j$ are then *reset*. This means that the counters c_i are set back to n and marked *met*.

A play of this game is a play in a parity game, augmented with the challenge and counter configuration at each step. A play with dominant priority d is winning for Even if either d is even or if every opened $d+1$ challenge is eventually met or reset.

Example 89. Consider the formula $\nu Y.\mu X.(A \wedge \Diamond X) \vee (B \wedge \Diamond Y)$ which is true if on some path B always eventually holds. This formula does not hold in the following structure:



However, Even wins the 1- and 2-challenge games: her strategy is to loop in the current state until Odd opens a 2-challenge, then meet the challenge by moving to the next state, as seeing a B corresponds to seeing 2. Odd will run out of challenges before reaching the last state. Although Odd wins the 3-challenge game in this structure, for any m it is easy to construct a similar structure in which he loses the m-challenge game but wins the parity game. This section argues that this is sufficient to show that $\nu Y.\mu X.(A \wedge \Diamond X) \vee (B \wedge \Diamond Y)$ is not equivalent to any Σ_2^μ formula.

In contrast, in the formula $\nu Y.\mu X.(A \wedge \Box X) \vee (B \wedge \Diamond Y)$, Odd wins the 1-challenge game whenever he wins the parity game: he can open the challenge when his strategy in the parity game reaches the point at which he can avoid B. This formula is therefore equivalent to a Σ_2^μ formula, namely the alternation free formula $\nu Y.((A \wedge \Box Y) \vee (B \wedge \Diamond Y)) \wedge \mu X.(A \wedge \Box X) \vee B$.

Definition 90. *(Challenge games)* A configuration $(v,p,\bar{c},r)$ of the n-challenge game on a parity game G of index $\{q,...,0\}$ where q is even consists of:

- a position v in the parity game;
- an even priority p indicating the least significant priority on which a challenge is open or $p = q + 2$ if all challenges are currently met;
- $\bar{c} = (c_0, c_2, \ldots, c_q)$ a collection of counter values c_i, initialised to n, for each even priority $i \in I_e$.
- $r \in \{0,1\}$ indicating the round of the game: 1 for Odd's turn to open challenges, 0 for a turn in the parity game.

At configuration $(v,p,\bar{c},1)$, corresponding to Odd's turn, he can open challenges up to any $p' \leq p$, as long as $c[i] > 0$ for each i such that $p' \leq i < p$. Then the configuration becomes $(v,p',\bar{c}',0)$ where $c'[i] = c[i] - 1$ for all newly opened challenges i, that is to say i such that $p' \leq i < p$,and $c'[i] = c[i]$ for all other i.

At configuration $(v,p,\bar{c},0)$, the player whose turn it is in the parity game decides the successor position v' of v. If this position is terminal, the player winning the parity game also wins the challenge game. Else, the configuration is updated to $(v',p',\bar{c}',1)$ according to the priority i of v' as follows:

- If $i \geq p$ then $p' = i + 2$ if i is even, $p' = i + 1$ otherwise. This indicates which challenges have been met. Note that if all challenges are met, $p = q + 2$.

- If $i < p$ then $p' = p$.
- For each $j < i$, the counter value c_j is reset to n.
- If i is even and $c_i = 0$, then the game ends immediately with a win for Even.

A play starts at $(v_\iota, q+2, (n,...,n), 1)$ where v_ι is the initial position of the parity game. A play is a potentially infinite sequence of configurations. An infinite play is winning for Even if the dominant priority on the sequence of parity game positions is d but the game reaches infinitely many configurations $(v, p, \bar{c}, 0)$ where $p > d+1$. This is the case if d is even or if all $d+1$ challenges set by Odd are either met or reset.

A strategy for Odd in a challenge game consists of two parts: a strategy for opening challenges, and a strategy for the underlying parity game. Even only has a parity game strategy. Both players' strategies may of course depend on the challenge configuration as well as the parity game configuration. Given a challenge-game strategy σ for Even, each challenging strategy γ for Odd induces a normal parity game strategy σ_γ for Even which does not depend on the challenge configuration, up to the point where Even meets a challenge with a counter value 0.

I now show that for all Ψ and finite n, there is a formula $\Psi(n)$ which holds in $\mathcal{M}$ if and only if Even wins the n-challenge game on $\mathcal{M} \times \Psi$. For clarity, $\Psi(n)$ is described via the corresponding L_μ-automaton on unranked trees. From Theorem 18, this is equivalent to describing a L_μ formula. For readers who prefer formulas to automata, an intuition of the syntax of this formula is given in Example 92.

Definition 91. Let $A = (S, s_i, \delta, \Omega)$ be the L_μ-automaton for Ψ. Let us build the automaton A^n for $\Psi(n)$ using distinct copies of A for each possible challenge configuration $(p, \bar{c})$. For $p = q+2$ and each even priority p, and counter values $\bar{c} \in [n]^{I_e}$, the copy $A(p, \bar{c})$ of A corresponds to p being the least significant open priority and the counter values being $\bar{c}$. These components will then be combined into the automaton A^n. The *original priority* of a position in a component refers to the priority assigned to the corresponding node in S by Ω.

$A(p,c) = (S^{(p,c)}, s_i^{(p,c)}, \delta^{(p,c)}, \Omega^{(p,c)})$ is based on A: $S^{(p,c)}$, $s_i^{(p,c)} \in S^{(p,c)}$ and $\delta^{(p,c)}$ are disjoint copies of S, s_i and δ respectively. The priority function is g $\Omega^{(p,c)}$:

- If $\Omega(s) \geq p-1$ then $\Omega^{(p,c)}(s) = 1$;
- If $\Omega(s) < p-1$ then $\Omega^{(p,c)}(s) = 0$;

Then, the components $A(p,c)$ are linked in $A^n = (S^n, s_i^n, \delta^n, \Omega^n)$ consisting of:

- The disjunct union of all component state spaces: $S^n = \biguplus_{p \in I_e, c \in [n]^{I_e}} S^{(p,c)}$;
- The initial state $s_i^n = s_i^{(q+2,\bar{n})}$ of $A(q+2, \bar{n})$ where $\bar{n}_i = n$ for all i;
- Ω^n defined by $\Omega^n(s) = \Omega^{(p,c)}(s)$ where s is a state of the component $A(p,c)$;
- For states s in $A(p, \bar{c})$ of original priority $j \geq p$, let $\delta^n(s, A) = \top$ if $c_j = 0$. This corresponds to Even having met all n challenges. Otherwise, let $\delta^n(s, A) = s'$

such that: s' is the copy of s in $A(k,\bar{c}')$ where $k = j+2$ if j is even and $k = j+1$ otherwise, and $\bar{c}'[i] = n$ for $i < j$ and $\bar{c}'[i] = \bar{c}[i]$ for other i. This corresponds to the open j-challenge being met and all counters below j being reset.

For every state s in $A(p,c)$ with original priority $j < p$, if K is the set of even priorities k smaller than p such that for all $i.p > i \geq k, \bar{c}[i] > 0$, let $\delta^n(s,P)$ be $\delta^{(p,c)}(s,P) \wedge \bigwedge_{k \in K} s_k$ where s_k is the copy of s in $A(k,\bar{c}')$, and $\bar{c}'[i] = \bar{c}[i] - 1$ for i such that $k \leq i < p$ and $\bar{c}'[i] = \bar{c}[i]$ otherwise. In other words, Odd can open challenges k below p if counter-values up to k are non-zero, by moving to the component $A(k,\bar{c}')$ which reflects the new challenge configuration.

This automaton only has priorities $\{0,1\}$ so the corresponding formula, $\Psi(n)$, is in Σ_2^μ. It therefore suffices to argue that this automaton describes the challenge game.

A game in A^n maps to a game in A, augmented with challenge configurations $(p,\bar{c})$ at each state, corresponding to the component in which a state is played. Transitions between components account for challenges being opened, met, and reset according to the rules of the game. Then, let us check that Ω^n accounts for the winning conditions of the challenge game.

Opening challenges in A^n makes the play move to lower components $A(p,c)$, as measured by p; seeing high *original* priorities makes the play move up to higher components. If the dominant original priority d is even, then eventually the play can no longer move up to components $A(p',c')$ with $p' \geq d+2$ from components $A(p,c)$ where $p < p'$. Such plays eventually settle into some $A(p,c)$ where $p \geq d+2$ since Odd can not challenge d infinitely often if a higher priority is not also seen infinitely often. From Ω^n, such a play is winning for Even since nodes of orinigal priority d and lower in components $A(p,c)$ where $p \geq d+2$ are of priority 0.

If d is odd, then Even wins only if the play settles into a component $A(p,c)$ where $p > d+1$ since those are the components in which d and lower priorities are replaced with 0 – this corresponds to Odd eventually not opening the challenge on $d+1$ after it has been met, causing him to lose. If the minimum challenged priority never settles, in A^n such a play sees 1 infinitely often, and this means that a $d+1$-challenge is not met – that is to say, Odd wins the challenge game.

Example 92. Given a formula $\Psi = \nu Y.\mu X.\phi(X,Y)$ where X and Y are the only fixpoint variables, the corresponding Σ_2^μ formula is

$$\nu W^n.\nu Y.\phi(Y,Y) \wedge \mu X.\phi(X,W)$$

for some sufficiently large value of n. The intuition of the formula is that the first copy of ϕ corresponds to the challenge game when the challenge is met and Odd has to open a challenge by moving to the second copy of ϕ. Meeting the challenge is encoded by the fixpoint W, and the approximation n accounts for the counting of challenges. For inputs with further fixpoints, the formula becomes more cumbersome, but the idea remains the same.

The following Theorem and its Corollary constitute the core of this chapter, and one of the most significant results of the thesis. They represent a novel characterisation of Σ_2^μ, based on the sequence of approximation $\Psi(n)$ of Ψ.

Theorem 93. *If a disjunctive formula* Ψ *is semantically in* Σ_2^μ*, then there is a finite* m *such that for all* $\mathcal{M}$ *it is the case that* $\Psi \Leftrightarrow \Psi(m)$*.*

Proof. Assume that Ψ is in Σ_2^μ, *i.e.* equivalent to some Φ of index $\{1,0\}$ and that for all m, $\Psi \not\Leftrightarrow \Psi(m)$: there is a structure $\mathcal{M}$, such that Odd wins the parity game $\mathcal{M} \times \Psi$ but Even wins the m-challenge game on $\mathcal{M} \times \Psi$. Fix $m > 2^{|\Psi|+|\Phi|}$. Without loss of generality, take $\mathcal{M}$ to be finitely branching. The overall structure of this proof is to first use a winning strategy τ for Odd in $\mathcal{M} \times \Phi$ to define a challenging strategy γ for him in the m-challenge game on $\mathcal{M} \times \Psi$ (Part I). It then adds back-edges to $\mathcal{M}$ using Even's winning strategy σ against γ in the challenge game (Part II). This turns $\mathcal{M}$ into a new structure $\mathcal{M}'$ which preserves Odd's winning strategy τ in $\mathcal{M}' \times \Phi$ but turns σ_γ into a winning strategy in $\mathcal{M}' \times \Psi$ (Part III). This contradicts the equivalence of Φ and Ψ.

Part I. Let τ be Odd's winning strategy in $\mathcal{M} \times \Phi$. Since $\mathcal{M}$ is finitely branching, for any node v reachable via τ, there is a finite bound i such that any play that agrees with τ sees 1 within i modal steps of any position $v \times \alpha$ that it reaches (König's Lemma). For a branch b of $\mathcal{M}$, on which τ reaches a node v, indicate by $next(b,v)$ the i^{th} node on b from v. This node has the property that any play on the branch b agreeing with τ must see a 1 between v and $next(b,v)$. If τ does not agree with any plays on the branch b, then let $next(b,v)$ be a node on b which τ does not reach.

Now consider the m-challenge game on the arena $\mathcal{M} \times \Psi$. Let Odd's challenging strategy γ be: to open all challenges at the start of the game, and whenever its counter is reset; if a challenge for a priority i is met at v, and its counter c_i is not at 0, to open the next challenge when the play reaches a node $next(b,v)$ for any branch b, unless the counter is reset before then (*i.e.* a higher priority is seen).

Part II. Even wins the m-challenge game on $\mathcal{M} \times \Psi$, so let σ be her winning strategy. Recall that σ_γ is a strategy for Even in the parity game $\mathcal{M} \times \Psi$ up to the point where an m^{th} challenge in the original challenge game is met, and undefined thereafter. Since Ψ is disjunctive, we can adjust $\mathcal{M}$ into a bisimilar structure in which the pure parity game strategy σ_γ is well-behaved wherever it is defined – it reaches each position of $\mathcal{M}$ at either one subformula, or none.

The strategy σ_γ is winning in the challenge game against any strategy for Odd which uses the challenging strategy γ. Since Odd always eventually opens the next challenge, the only way for him to lose is to either reach a losing position in the underlying parity game, or reaches a position of priority p when $c_p = 0$. Thus, every play that agrees with σ_γ is finite.

Since σ_γ is well-behaved, each branch carries at most one play. For every branch b the finite play it may carry must either end in a position winning for Even in the underlying parity game, or end in a long streak in which the highest priority seen is some even p, and it is seen at least m times, corresponding to every instance of Even meeting a p-challenge. As long as m is sufficiently large, on every such branch there are two nodes v and its descendant w, at which Odd opens challenges on p, which agree on the set of subformulas that σ_γ reaches there in $\mathcal{M} \times \Psi$ and that τ reaches there in $\mathcal{M} \times \Phi$. We now consider the structure $\mathcal{M}'$, which is as $\mathcal{M}$ except that the predecessor of each w-node has a back-edge to v instead. The strategies σ_γ and τ transfer in the obvious way to $\mathcal{M}'$.

Part III. We now claim that τ is winning in $\mathcal{M}' \times \Phi$ and that σ_γ is winning in $\mathcal{M}' \times \Psi$. Starting with σ_γ, all finite plays end in positions that are winning for Even, as in the underlying parity game of the challenge game on $\mathcal{M} \times \Psi$. Any infinite play in $\mathcal{M} \times \Psi$ that agrees with σ_γ sees both v and w infinitely often. Any such play is dominated by an even priority between v and w. Then, as w and v agree at which subformula σ_γ reaches them, an even priority dominates any play that goes through back edges in $\mathcal{M}' \times \Psi$ infinitely many times. The strategy σ_γ is therefore winning in $\mathcal{M}' \times \Psi$.

Now onto τ in $\mathcal{M}' \times \Phi$. If a branch is unchanged by the transformation, then any play on it is still winning for τ. If a branch that τ plays on has been changed, then consider in $\mathcal{M}$ the two nodes v and w at which the transformation is done. These are both nodes at which Odd opens challenges according to γ, therefore, from the definition of *next* and γ, the highest priority seen between them by any play agreeing with τ is 1. Since v and w agree at which subformulas τ reaches them, any play in $\mathcal{M}' \times \Phi$ which goes through a back-edge infinitely often sees 1 infinitely often and is therefore winning for Odd.

This contradicts the equivalence of Ψ and Φ. Therefore, if Ψ is semantically in Σ_2^μ, then for all structures $\mathcal{M}$ the m-challenge game and the parity game on $\mathcal{M} \times \Psi$ have the same winner for $m > 2^{|\Phi|+|\Psi|}$. □

Corollary 94. *A disjunctive formula Ψ is semantically in Σ_2^μ if and only if there is some finite m such that $\Psi \Leftrightarrow \Psi(m)$.*

This is the core contribution of this chapter, and one of the most significant results of this thesis. It proves that the sequence of formulas $\Psi(n)$ approximating Ψ eventually coincides with Ψ as long as Ψ is semantically in Σ_2^μ. These approximations can be understood as a a way to capture all Π_2^μ alternations in Ψ in a single such alternation which can then be approximated whenever Ψ is semantically Σ_2^μ. The core novel technique introduced here is the n-challenge game on arbitrary parity games – as mentioned previously, the construction for $\{1,2\}$ can be found in the automata-theoretic literature – and the characterisation of Σ_2^μ formulas as those for which the model-checking game coincides with one of the n-challenge games.

6.2 A tree-building game

This section applies a proof technique from Skrzypczak and Walukiewicz 2016 [SW16] to the L_μ setting in order to complete the proof of decidability for Π_2^μ input formulas.

The idea is to define another game, the *n-tree-building game* $\mathcal{F}(n)$, such that for finite n Odd wins $\mathcal{F}(n)$ if and only if Ψ is equivalent to $\Psi(n)$. This reduces the decidability of Σ_2^μ to deciding whether there is an n such that Odd wins the n-tree-building game. A win for Even in the ω-tree-building game will imply a win in $\mathcal{F}(n)$ for every n; this reduces the decidability of Σ_2^μ to the decidability of the winner of the ω-tree-building game. This game is generalised from Skrzypczak and Walukiewicz 2016 [SW16] to deal with unranked trees and arbitrary input formulas. The proof strategy to show the decidability of $\mathcal{F}(\omega)$ follows closely the original proof. I adhere to the original presentation whenever reasonable, but describe it within the L_μ framework.

Informally, the game consists of the players building a structure annotated with traces from both the challenge game for Ψ and the parity game for $\neg\Psi$.

Let Ψ be a Π_2^μ formula in disjunctive form, of index $I = \{2,1\}$. Let $A \subset Act$ be the finite set of action labels appearing in Ψ. Fix $\neg\Psi$, the negation of Ψ in disjunctive form, which may have a different index from Ψ. The previous section defined the challenge game for arbitrary L_μ formulas; however, when restricted to Π_2^μ, there is only one challenge. A binary state $\{open, met\}$ and one counter suffice to represent the challenge configuration. Indicate by $\omega + 1$ the set containing all natural numbers and ω.

Definition 95. A position (S,ϕ,κ,r) of $\mathcal{F}(\beta)$ for $\beta \in \omega + 1$ is:

- S a set of *active states*: pairs (f,p) where $f \in sf(\Psi)$ and $p \in \{open, met\}$.
- $\phi \in sf(\neg\Psi)$;
- $\kappa : S \to (\beta + 1)$ a function that assigns to each active state a counter value.
- $r \in \{0,1\}$ a sub-round number.

The initial position is $(\{(\Psi, q+2)\}, \neg\Psi, \kappa, 0)$ where $\kappa(\Psi, q+2) = \beta$.

The active states and κ describe positions of the challenge game in a structure built on the fly, reached by a strategy specified by Even. Along a sequence of positions, the subformulas ϕ of $\neg\Psi$ form a strategy for Even in the parity game for $\neg\Psi$.

Then, define multi-transitions.

Definition 96. A multi-transition from a position (S, ϕ, κ, r) to (S', ϕ', κ', r') consists of:

- The pre- and post-states (S, ϕ, κ, r) and (S', ϕ', κ', r') where $r' = r + 1 \texttt{ mod } 2$;
- a set e of edges from the active states in S to the active states in S', labelled with an intermediate modal formula;
- a set $\neg e \subseteq e$ of boldfaced edges, where exactly one ends at each $(f, p) \in S'$.

These multi-transitions specify not only the next position, but also the origin of each new active state. Since an active state can potentially be reached from several previous active states, the boldfaced edges decide on one of these. The function κ' will then describe the counter values along the boldfaces traces.

The intention of the game $\mathcal{F}(n)$ is to let Even win if and only if there is a model for $\neg\Psi \wedge \Psi(n)$. The positions can be seen as attempts to build a branch of such a model, annotated with a witness strategy in the $\neg\Psi$ parity game and tracking potential opened challenges for $\Psi(n)$. The edges e denote potential strategies for Odd in the challenge game, and the boldface edges finally ask Odd to decide the counter value for a given active state.

During round 0, the Odd player is given the choice to restrict what challenges he may open in the actual challenge-game. This is only relevant in $\mathcal{F}(\omega)$ where he has to avoid opening infinitely many challenges on any trace; in $\mathcal{F}(n)$ for finite n, it is always in Odd's interest to allow challenges to be opened at any time – see the winning conditions. Then, in round 1, Even decides on the propositional variables true in the current state and a finite set of successor states. She also extends her strategies on Ψ and $\neg\Psi$ to those successors. Odd then chooses a successor, which induces a new set of active states. Since the same active state may be reached from more than one predecessor state, he also specifies boldfaced edges to each new active state. The challenge-configuration is updated to reflect any challenges met or reset on the traces along boldfaced edges.

More formally, if the current configuration is (S, ϕ, κ, r), then the players construct a multi-transition to a new configuration in the following ways:

- (R0) $r = 0$. Odd chooses a set C of pairs (f, open) such that $(f, \mathit{met}) \in S$, and $\kappa(f, p) > 0$. The edges e are $((f, p), (f, p))$ for $(f, p) \in S$ and $((f, \mathit{met}), (f, \mathit{open}))$

for $(f,open) \in C$. For each $(f,p) \in S$ Odd must specify exactly one bold-edges edge $(s,(f,p)) \in e$ For each such new state $(f,open)$ with boldfaced predecessor (f,met), set $\kappa'(f,open) = \kappa(f,met) - 1$ if $\beta \neq \omega$; if $\beta = \omega$ then κ' is always the constant ω; for other (f,p) let $\kappa'(f,p) = \kappa(f,p)$. The new configuration is $(S',\phi,\kappa',1)$.

- (R1) $r = 1$. Even chooses:
  1. a set of propositional variables P,
  2. a set of successors $N_a = \{s_0,...,s_n\}$ for each $a \in A$, each no larger than $|\Psi| + |\neg\Psi|$,
  3. a next modal formula $P_\phi \bigwedge_{a\in A'} \xrightarrow{a} B_a$ of ϕ where P_ϕ respects P,
  4. a surjection $g_a : N_a \to B_a$ for each $a \in A'$,
  5. a set D consisting of a pair (f',p') for every $(f,p) \in S$ where f' is a next modal formula $f' = P_f \bigwedge_{a\in A'} \xrightarrow{a} B_a$ such that P_f respects P, and if the trace from f to f' sees a ν-bound variable, then $p' = met$, otherwise $p' = p$.
  6. for each chosen (f',p'), where $f' = A_f \bigwedge_{a\in A'} \xrightarrow{a} B_a$ surjections $g_{(f',p'),a} : N_a \to B_a$ for each $a \in A'$.

  Odd responds by choosing s' out of the successors. This induces a new set of active states: if s' is an a-successor, the set S' consists of $(g_{(f,p),a}(s'),p')$ for each $(f,p) \in D$ such that if $g_{(f,p),a}$ is a ν-bound variable, then $p' = met$, else $p' = p$.
  The set of edges is built as follows: there is an edge from (f,p) to (f'',p'') labelled with (f',p') if for point (5) Even chooses $(f',p') \in D$ for $(f,p) \in S$ and $f'' = g_{(f',p'),a}(s')$ and p'' is met or p' according to whether a ν-bound variable is seen along the trace from f' to f''.
  Finally, Odd also chooses for each $(f',p') \in S'$ an edge $((f,p),(f',p'))$ to make boldfaced. The new configuration is then $(S',g_a(s'),\kappa,0)$. If either player fails to perform one of these steps, that player loses immediately. For example even might fail because two active states don't have any next modal subformulas with compatible sets of propositional variables.

A play is a sequence of game configurations, linked by multi-transitions. This presentation differs slightly from the original [SW16]. First, it updates the challenge configuration on the fly rather than as a separate round. The traces built during this game can then be directly read as traces in the challenge game. The concept of *flush*, which forces Odd to synchronise his challenges between different traces, is not used here until the proof of Lemma 98. Finally, this definition deals with non-binary branching and disjunctive formulas: instead of building a right-successor and a left-successor, Even builds sets of successors via each label. The size of these sets is bounded.

A play carries one $\neg\Psi$-trace and one or several Ψ challenge traces, some of which are boldfaced. Even wins a play if:

1. for every infinite Ψ-challenge trace Even meets every challenge opened by Odd, and
2. at least one of the following is true:
   (a) on some boldfaced trace, infinitely many challenges are opened and met, or
   (b) the $\neg\Psi$ trace is winning for Even.

If Even wins $\mathcal{F}(n)$ with conditions 1) and 2b), this will give rise to a model of $\neg\Psi \wedge \Psi(n)$. Conditions 1) and 2a) can only be satisfied in $\mathcal{F}(\omega)$ – these will help establish the winner in $\mathcal{F}(n)$ for large n.

Lemma 97. *For finite n, Even wins the $\mathcal{F}(n)$ game if and only if $\Psi \not\Leftrightarrow \Psi(n)$.*

Proof. First assume that Even wins $\mathcal{F}(n)$ for some finite n. Consider the following family of strategies for Odd in $\mathcal{F}(n)$: at round 0, he chooses C to include every pair (f, p') such that $(f, p) \in S$ and $\kappa(f, p) > 0$. That is to say, he allows himself to set challenges whenever the counter values permit it. This means that in the structure built from Even's response to these challenges, Even will have to have a strategy against all possible challenging strategies. He also always chooses a boldfaced edge inducing the largest κ.

This defines Odd's strategy apart from the choice of direction. Such a partial strategy, combined with Even's winning strategy σ in $\mathcal{F}(n)$ induces a structure $\mathcal{M}$, built by σ in response to every possible choice of successor odd can make. Even's winning strategy σ in $\mathcal{F}(n)$ induces a strategy in the n-challenge game on $\mathcal{M} \times \Psi$. From the winning condition 1) of $\mathcal{F}(n)$, every play in the challenge game on $\mathcal{M} \times \Psi$ which agrees with this strategy must be winning for Even. On the other hand, Odd's challenging strategy does not open infinitely many challenges on any boldfaced trace, so 2a) can not hold. Thus, 2b) holds, and Even's strategy in $\mathcal{M} \times \neg\Psi$ induced by σ must be winning. Odd therefore wins $\mathcal{M} \times \Psi$ and $\mathcal{M}$ witnesses that $\Psi \not\Leftrightarrow \Psi(n)$.

For the other direction assume that there is a structure $\mathcal{M}$ such that Odd wins $\mathcal{M} \times \Psi$ but Even wins the n-challenge game on the same arena with a strategy σ. Let $\mathcal{M}$ and σ be such that for each Even's strategies σ_γ, for all challenging strategies γ, in $\mathcal{M} \times \Psi$ the strategy only agrees with one play per branch; let the same be true for the winning strategy $\bar{\sigma}$ in $\mathcal{M} \times \neg\Psi$. This is possible due to both Ψ and $\neg\Psi$ being in disjunctive form. Note that $\mathcal{M}$ needs no higher branching arity than $|A|(|\Psi| + |\neg\Psi|)$.

Her strategy in $\mathcal{F}(n)$ is to build $\mathcal{M}$. The initial position corresponds to the root of $\mathcal{M}$ and at each game configuration thereafter, she keeps track of the state v in $\mathcal{M}$ that it corresponds to. At $(S, \phi, \kappa, 1)$ she then plays:

- the set of propositional variables at v;
- the sets N_a of a-successors of v for each $a \in A$;
- the set D of pairs (f', p') for each $(f, p) \in S$ such that f' is the next modal formula of f which her winning strategy plays at $v \times f$ if the current challenging configuration is p with counter value $\kappa(f, p)$; $p' = p$ if no ν-variable is seen along these steps, otherwise $p' = met$;
- for each $(P \bigwedge_{a \in A'} \xrightarrow{a} B_a, p) \in D$ and $a \in A'$, the surjection $g_{(f',p),a}$ which map each a-successor v' to the unique formula $b \in B_a$ such that σ plays $v' \times b$ from $v \times P \bigwedge_{a \in A'} \xrightarrow{a} B_a$ when the challenge configuration is p with counter $\kappa(f, p)$
- the next modal formula ϕ' her winning strategy $\bar{\sigma}$ in $\mathcal{M} \times \neg\Psi$ plays at $v \times \phi$;
- for the modal formula ϕ of $\neg\Psi$, the surjections g_a which map each a-successor v' to the unique subformula ϕ' such that $\bar{\sigma}$ plays $v' \times \phi''$ from $v \times \phi'$.

The Ψ-traces in any play that agrees with this strategy correspond to plays agreeing with σ, which guarantees that the $\mathcal{F}(n)$-play satisfies the winning condition 1). The $\neg\Psi$ trace corresponds to a play that agrees with $\bar{\sigma}$, satisfying winning condition 2b.

□

It then remains to be shown that Even wins $\mathcal{F}(\omega)$ if and only if she wins $\mathcal{F}(n)$ for all n and that the winner of $\mathcal{F}(\omega)$ is decidable.

Lemma 98. *There is a finite value K_0, computable from Ψ and $\neg\Psi$, such that Even wins $\mathcal{F}(\omega)$ if and only if she wins $\mathcal{F}(K_0)$.*

Proof. First, note that if Odd wins $\mathcal{F}(\omega)$, he can win with a strategy which in round 0 chooses C to be the empty set whenever S contains an active state $(f, open)$. In other words, Odd can always wait for *all* traces to meet opened challenges before opening a new challenge. Let such a strategy be called *patient*. Note that $\mathcal{F}(\omega)$ is a finite game with a regular winning condition. Its winner therefore has a finite memory winning strategy. Suppose that Odd wins the game. Let M be the size of the memory of Odd's patient winning finite memory strategy τ in $\mathcal{F}(\omega)$. Let K_0 be the product of M, the number of configurations of $\mathcal{F}(\omega)$ and the set of possible active states.

Next argue that τ is a winning strategy in $\mathcal{F}(K_0)$. First, we have to show that it is a valid strategy, i.e., Odd never tries to open a challenge with an empty counter. This could only occur if some boldfaced trace opened K_0 challenges. If that was the case then, K_0 being very large and Odd's memory being only M, there would be a looping fragment along this play in which on a boldfaced trace a challenge is both opened and met. Following this boldfaced trace, a challenge would be opened and met infinitely often. Furthermore, since τ is patient, winning condition 1 would also hold on that branch, since Odd only opens challenges when all traces are in the *met* state. This contradicts the assumption that τ is winning in $\mathcal{F}(\omega)$.

Then argue that if Odd plays using τ, this is a winning strategy for Odd in $\mathcal{F}(K_0)$. Counting challenges does not affect the first winning condition whereby if Even is to win, in every infinite trace, Even must meet every challenge. So if a play that agrees with τ is winning for Odd in $\mathcal{F}(\omega)$ because on some trace, Even fails to meet some challenge, then the same is true in $\mathcal{F}(K_0)$. Furthermore, as argued above, τ does not open more that K_0 challenges, so in no play does condition 2a) hold. Finally, condition 2b is not affected by the addition of counters and inherits the winner from the $\mathcal{F}(\omega)$ game. As a result, τ is winning in $\mathcal{F}(K_0)$.

For the case that Even wins the $\mathcal{F}(\omega)$ game, note that a winning strategy for Even in $\mathcal{F}(\omega)$ is a winning strategy in any $\mathcal{F}(n)$ for finite n. In particular, she would win $\mathcal{F}(K_0)$. □

Corollary 99. *Even wins $\mathcal{F}(\omega)$ if and only if she wins $\mathcal{F}(n)$ for all n.*

Theorem 100. *It is effectively decidable whether any given Π_2^μ formula is equivalent to a Σ_2^μ formula. By duality, it is also effectively decidable whether any given Σ_2^μ formula is equivalent to a Π_2^μ formula.*

Proof. Given any Π_2^μ formula, it can be effectively turned into a disjunctive formula Ψ also in Π_2^μ (Lemma 51). Then, Theorem 94 yields that Ψ is semantically in Σ_2^μ if and only if it is equivalent to $\Psi(n)$ for some n. From Lemma 97, $\Psi \not\Leftrightarrow \Psi(n)$ if and only if Even wins $\mathcal{F}(n)$, so Ψ is semantically in Σ_2^μ if and only if Odd wins $\mathcal{F}(n)$ for some n. From Lemmas 98 and 99, this is true if and only if he wins $\mathcal{F}(K_0)$, i.e., if $\Psi \Leftrightarrow \Psi(K_0)$ and K_0 depends only on Ψ.

Given any Σ_2^μ formula, it can also be decided whether it is equivalent to a Π_2^μ formula, via checking whether its negation is equivalent to a Σ_2^μ formula. □

Note that for Π_2^μ input formulas, the formula $\Psi(n)$ is in fact alternation free: each component of the L_μ automaton either only has priorities 1 or only priorities 0; since no counters can be reset, there are no loops through multiple components. Hence for Π_2^μ input formulas equivalent to Σ_2^μ formulas, the equivalent Σ_2^μ formula is in fact alternation free. This is unsurprising, since the intersection of semantically Π_2^μ and Σ_2^μ formulas is exactly alternation-free L_μ.

6.3 Discussion

This chapter has shown that given any L_μ formula in Π_2^μ, it can be effectively decided whether it is equivalent to a Σ_2^μ formula. The first part of the proof provides a sequence of formulas that approximate an input formula Ψ. This sequence is shown to coincide semantically with Ψ if and only if Ψ is semantically Σ_2^μ. Equivalently, it reduces the decidability for Σ_2^μ for arbitrary L_μ formulas to deciding whether there is an n such that the n-challenge game is the model-checking game fo a formula. The proof strategy proposes a formula which encapsulates all Π_2^μ-type alternations into one such alternation, and then asks whether this single alternation is truly necessary, or whether it can be finitely approximated.

Unfortunately, the second part which applies automata-theoretic techniques to the L_μ setting to decide at which stage the sequence of approximations coincides with the input formula, is less general and only admits input formulas in Π_2^μ. If this could also be generalised to arbitrary formulas, this would yield a decidability proof for Σ_2^μ.

This concludes the second part of this thesis, concerned with decision procedures for low levels of the L_μ alternation hierarchy. The final part will generalise some of the constructions seen in this chapter to achieve parametrised characterisations of all disjunctive L_μ alternation classes. I will tie these in with the descriptive complexity of model-checking games and show how all of the decision procedures in the preceding chapters, despite their distinct flavours, can be understood in such terms.

Part III

In which descriptive complexity comes to the service of the index problem.

Up to this point, parity games have mainly appeared as a tool for model checking L_μ formulas. However, the relationship between L_μ and parity games is much deeper. A parity game can itself be viewed as a transition system in which L_μ formulas may or may not hold. Then, given a class of parity games, one can study its *descriptive complexity*: how complex is a L_μ formula which holds in exactly those games that are winning for Even? The L_μ literature tells us that the winning regions of the class of parity games with priorities I can be described by a formula of index I [EJ91, Wal02]. In particular, the winning regions of the model checking games of a formula can be described with a formula of the same syntactic complexity.

So far, only the syntactic complexity of formulas has entered the equation. This part considers how the *semantic* complexity of a formula relates to the descriptive complexity of the winning regions of its model-checking parity games. The core result of Chapter 7 is that while the syntactic complexity of a formula Ψ is an upper bound to the descriptive complexity of Ψ's model checking games, the semantic complexity of Ψ provides a lower bound. This means that if there is a formula Φ of lower complexity than Ψ which describes the winning regions of the model-checking games of Ψ – written Φ *interprets* Ψ – then Ψ must be semantically in the alternation class of Φ. This is a natural extension of the known harmony between L_μ and parity games.

Chapter 8 considers the converse: if a formula Ψ is semantically in a class C, can the winning regions of the parity games generated by Ψ be described by a formula in C? If so, by what formula? This amounts to asking whether the semantic complexity of a formula is also an upper bound to the descriptive complexity of its model-checking games. I answer this question positively for ML, Π_2^μ and Σ_2^μ, as well as all disjunctive alternation classes. As a result, at least for formulas of semantic complexity up to Σ_2^μ, the descriptive complexity of the model-checking games corresponds exactly to the semantic complexity of the formula, rather than its syntactic complexity. In other words, the relationship between parity games and L_μ is even more robust than previously thought.

Both chapters also discuss the practical implications of these results. Chapter 7 proposes a methodology for finding formula optimisations: some syntactic features are provably indicators of low descriptive complexity for the model-checking games, and therefore of low semantic complexity for the formula. As a proof of concept, I present such features for the alternation-free class $\Pi_2^\mu \cap \Sigma_2^\mu$ and Π_2^μ. Chapter 8 on the other hand, argues that the known decision procedures for ML, Π_1^μ and Σ_2^μ for Π_2^μ, despite their distinct flavours, can all be understood in terms of describing the model checking parity games with formulas in the target class. I propose this as a unified approach to addressing the alternation hierarchy.

Chapter 7

Descriptive Complexity of Model-Checking Games

This chapter revisits one of the fundamental concepts of L_μ literature: the relationship between L_μ formulas and parity games. Recall that the model checking games for L_μ formulas of index I are parity games with priorities I, and that an L_μ formula of index I suffices to describe the winning regions of such games [EJ91, Wal02]. Here I argue that this relation extends, at least in one direction, to the *semantic* complexity of the formula: if the winning regions of the model checking games for Ψ, are described by Φ , written Φ interprets Ψ, then Ψ is semantically no more complex than Φ. In other words, the descriptive complexity of the model-checking parity games of Ψ is an upper bound on the semantic complexity of Ψ.

This means that analysing the parity games that a formula generates is a valid strategy for understanding a formula's semantic complexity. As a proof of concept, the rest of this chapter exemplifies how the idea of interpretation can be used to identify semantically alternation free or Π_2^μ formulas. It defines features which indicate that a formula can be interpreted by an alternation free formula, or a Π_2^μ formula. Unlike the Σ_2^μ decidability result of the previous chapter, these features are applicable to inputs of any complexity. For example, all disjunctive formulas in which no ν-cycle contains a universal modality, such as $\mu X.\nu Y.(A \wedge \Box X) \vee (B \wedge \Diamond Y)$, are in fact semantically alternation-free.

In short, this Chapter proposes studying the descriptive complexity of model-checking games as an approach to simplifying formulas. The first section discusses the technicalities of treating model-checking games as structures that formulas operate on. Section 7.2 shows that if Ψ is interpreted by Φ, then Ψ is semantically no more complex than Φ. Finally, Section 7.3 demonstrates how this can be used to identify semantically simple formulas.

7.1 Encoding parity games as structures

So far, the model-checking parity game for a structure $\mathcal{M}$ and a formula Ψ has been treated as an object with positions $s \times \phi$ where s is a state of $\mathcal{M}$ and ϕ is a subformula of Ψ. The subformula ϕ dictates both the owner of the position and its priority. In this chapter, parity games – and in particular model-checking parity games – are themselves treated as a class of structures. This section refines the definition of the model-checking parity game $\mathcal{M} \times \Psi$ to fit this view. In practice, this means defining propositional variables to encode the ownership and priority of nodes, and assigning those variables to the appropriate states.

Definition 101. *(Parity games as structures)* A parity game with priorities I is a transition system of which the states are labelled with exactly one of T, F, E_i or O_i, for $i \in I$ standing for True, False, Even's position of priority i, and Odd's position of priority i respectively. Only nodes labelled T or F are terminal.

Any parity game $\mathcal{P}$ can be encoded as a structure by marking nodes of priority i belonging to Even with E_i, and those belonging to Odd as O_i. Terminal nodes that are winning for Even are marked T and terminal nodes winning for Odd are marked F.

The winning regions of a parity game $\mathcal{P}$ are described by a formula of which the alternation depth depends on the number of priorities in $\mathcal{P}$. These formulas [EJ91, Wal02] are of particular importance to the L_μ literature as they are the archetypal complete formulas for their respective alternation classes [Bra99]: the formula describing the winning regions of parity games of priorities I is not equivalent to any formula of index smaller than I. In contrast, this chapters will consider fragments of L_μ for which this completeness result does not hold, *i.e.* formulas with model-checking games in which simpler formulas can describe the winning regions.

For the parity games formalised as above, the parity game formula for priority set $I = \{i, ..., m\}$ can be written down as follows.

Definition 102. *(Parity game formula)*

$$\text{Parity}_{i,m} = \gamma_i X_i . \gamma_{i-1} X_{i-1} ... \gamma_m X_m . T \vee \bigvee_{j \in I} (E_j \rightarrow \{X_j, \top\}) \vee (O_j \rightarrow \{X_j\})$$

where γ_j is ν for even j and μ otherwise. This formula is simply saying that at positions belonging to Even, marked E_j for some j, Even gets to choose the successor – with the modality $\rightarrow\{X, \top\}$ – while at positions marked O_j, Odd chooses the successor. The priorities seen along the way are reflected in the fixpoints: after a position E_j, corresponding to a priority j, a play sees fixpoint X_j. Odd priorities lead to least fixpoints while even ones lead to greatest fixpoints, thus respecting the winning condition of plays.

Theorem 103. *[AG11] Let $\mathcal{P}$ be a parity game with priorities $\{i,...,m\}$. Then $\mathcal{P} \models \mathrm{Parity_{i,m}}$ if and only if $\mathcal{P}$ is winning for Even.*

This is true for all parity games, regardless of whether they are model-checking games or arbitrary parity games. However, model-checking parity games have more structure than arbitrary parity games – even more so for games which stem from formulas with syntactic restrictions, such as disjunctive formulas. In this chapter we will exploit such additional structure to find tighter definitions for the winning regions of these games. It will be helpful to encode some additional data into the transition systems which represent model-checking parity games. Instead of just considering position ownership and priority, the next definition adds propositional variables to encode what type of subformula a position stems from.

Definition 104. *(Model-checking parity games as transition systems)* A model-checking parity game with a set of priorities I is a transition system of which each state is labelled with exactly one of the following propositional variables: T, F, C, D, E, O and V_α for α a fixpoint variable name. These stand for True, False, Conjunction, Disjunction, Even's modality, Odd's modality, and fixpoint Variable α respectively. The subscript α differentiates fixpoints – it can be simply $i \in I$, the priority of a fixpoint, or it can be a name distinguishing a fixpoint variable from others of the same priority. As before, only nodes marked T or F are terminal while those marked V_α have a single successor. The priority assignment is given by a function Ω which assigns to each V_α a priority in I.

The formula to describe the winning regions of model-checking games with priorities $I = \{i,...,m\}$ is then as follows.

Definition 105.

$$\mathrm{ModelChecking}_{i,m} = \gamma_i X_i.\gamma_{i-1} X_{i-1}...\gamma_m X_m.\gamma_m Y.$$
$$T \vee (E \to \{Y \top\}) \vee (O \to \{Y\}) \vee (D \to \{Y, \top\}) \vee (C \to \{Y\}) \vee \bigvee_\alpha V_\alpha \to \{X_{\Omega(\alpha)}\}$$

where again γ_j is ν for even j, and μ otherwise. This formula is morally the same as Definition 102, only it now incorporates the additional data about what type of formula a position stems from. With this formula, non-fixpoint positions use a special fixpoint Y of the lowest priority, while positions marked V_α use the fixpoint $X_{\Omega(\alpha)}$, of the right significance, dictated by the priority of α.

Definition 101 of parity games as structures subsumes Definition 104, of model-checking parity games as structures. A model-checking game can be treated as a plain parity game by setting $E_m = E \vee D \vee \bigvee_{\{\alpha | \Omega(\alpha) = m\}} V_\alpha$ to be all the nodes belonging to

Even, $O_m = O \vee C$, the nodes belonging to Odd, and $E_i = \bigvee_{\{\alpha|\Omega(\alpha)=i\}} V_\alpha$ for $i \neq m$, the fixpoint positions of priority higher than m. From now on, unless specified otherwise $\mathcal{M} \times \Psi$ refers to the structure representing the model-checking parity game. On occasions, it may also be treated as a plain parity game, for instance by writing $\mathcal{M} \times \Psi \models \mathrm{Parity}_{\mathrm{i,m}}$, in which case propositional variables should be understood using this previously stated equivalence.

Theorem 106. *Let $\mathcal{M}$ be a labelling system and Ψ a formula with priority assignment $\{i, ..., m\}$. Then $\mathcal{M} \times \Psi \models \mathrm{ModelChecking}_{i,m}$ if and only if $\mathcal{M} \models \Psi$.*

Proof. By setting the equivalences $E_m = E \vee D \vee \bigvee_{\{\alpha|\Omega(\alpha)=m\}} V_\alpha$, $O_m = O \vee C$, and $E_i = \bigvee_{\{\alpha|\Omega(\alpha)=i\}} V_\alpha$ for $i \neq m$, the formula $\mathrm{ModelChecking}_{i,m}$ is equivalent to $\mathrm{Parity}_{i,m}$. □

From now, the phrase Φ *interprets* Ψ will be used to mean that for all structures $\mathcal{M}$, it is the case that $\mathcal{M} \times \Psi \models \Phi$ if and only if $\mathcal{M} \models \Psi$. In other words, Φ interprets Ψ is short for "Φ describes the winning regions of the model-checking games generated by Ψ".

7.2 Descriptive complexity of games: a lower bound

The last section reviewed the idea that to interpret a formula, a formula of the same alternation depth is sufficient: the syntactic complexity of Ψ is an upper bound on the descriptive complexity of its model-checking game. In this section, I show that the semantic complexity of a L_μ formula is a lower bound to the descriptive complexity of its model-checking games: if a formula in an alternation class C interprets Ψ, then the formula Ψ is equivalent to one in C.

Theorem 107. *Let Ψ be a formula of L_μ. If for some formula $\mathrm{WinningRegion}$ and all structures $\mathcal{M}$ we have $\mathcal{M} \times \Psi \models \mathrm{WinningRegion}$ if and only if $\mathcal{M} \models \Psi$, that is to say $\mathrm{WinningRegion}$ interprets Ψ, then Ψ is equivalent to a formula $\Psi \times \mathrm{WinningRegion}$ which has the same alternation depth as $\mathrm{WinningRegion}$.*

The proof is very simple: there is a natural, product-like operation on formulas which, if $\mathrm{WinningRegion}$ interprets Ψ, yields a formula $\Phi \times \mathrm{WinningRegion}$, equivalent to Ψ, with the alternation depth of $\mathrm{WinningRegion}$.

The intention of the construction is that the parity game $(\mathcal{M} \times \Psi) \times \mathrm{WinningRegion}$ is exactly the same as $\mathcal{M} \times (\Psi \times \mathrm{WinningRegion})$. The choice of overloading the notation $\times$ is meant to emphasize this associativity. Note however the type of the objects in these statements: $\Psi \times \mathrm{WinningRegion}$ is a formula if Ψ and $\mathrm{WinningRegion}$ are both formulas while $\mathcal{M} \times \Psi$ is a transition system if $\mathcal{M}$ is a transition system.

To preserve the intuition in this construction, the definition of the formula $\Psi \times \mathrm{WinningRegion}$ defines the priority $\Omega(X)$ of each fixpoint X explicitly. As a result it may be the case that Ω is not an order-preserving priority assignment: a fixpoint X might be free in a binding $\nu Y.\phi_Y$, but $\Omega(X) < \Omega(Y)$. This can easily be fixed by re-ordering the bindings so that Ω is order-preserving: if Y is free in ϕ_X and $\Omega(Y) < \Omega(X)$ then replace Y with its binding $\nu Y.\phi_Y[X/\mu X.\phi_X]$ or $\mu Y.\phi_Y[X/\mu X.\phi_X]$. This presentation is chosen to emphasize that the fixpoint structure is the same in $\Psi \times \mathrm{WinningRegion}$ and in $\mathrm{WinningRegion}$.

Definition 108. *(*$\Psi \times \mathrm{WinningRegion}$*)* Let Ψ be a formula of L_μ, $\mathrm{WinningRegion}$ a uni-modal formula over propositional variables $\{T, C, D, O, E, V_\alpha\}$ for α in the set of fixpoint variables of Ψ, with priority assignment $\Omega_{\mathrm{WinningRegion}}$ with co-domain I. We define $\Psi \times \mathrm{WinningRegion}$ as follows:

$$
\begin{aligned}
P \times T &= P \text{ where } P \text{ is a conjunction of literals;}\\
(P \wedge \phi) \times \psi &= P \wedge (\phi \times \psi) \text{ where } P \text{ is a conjunction of literals;}\\
\phi \wedge \psi \times C &= \top;\\
\phi \vee \psi \times D &= \top;\\
\langle a \rangle \times E &= \top;\\
[a] \times O &= \top;\\
\top \times T = \bot \times F &= \top;\\
\alpha \times V_\alpha &= \top;\\
\phi \times P &= \bot \text{ for } P \in \{T, C, D, O, E\} \cup \{V_\alpha\}_\alpha \text{ otherwise.}
\end{aligned}
$$

$$
\begin{aligned}
[a]\phi \times \Box\psi = \langle a \rangle \times \Box\psi &= [a](\phi \times \psi);\\
\phi \times \Box\psi &= \bigwedge_{\phi' \in im(\phi)} \phi' \times \psi \text{ where } im(\phi) \text{ is the set of immediate subformulas of } \phi;\\
[a]\phi \times \Diamond\psi = \langle a \rangle \times \Diamond\psi &= \langle a \rangle(\phi \times \psi);\\
\phi \times \Diamond\psi &= \bigvee_{\phi' \in im(\phi)} \phi' \times \psi \text{ where } im(\phi) \text{ is the set of immediate subformulas of } \phi;\\
\phi \times X &= W_{\phi \times X} \text{ if reached computing a subformula of } \gamma W_{\phi \times X}.(\phi \times \psi_X)\text{; else}\\
&= \mu W_{\phi \times X}.(\phi \times \psi_X) \text{ if } X \text{ is a } \mu \text{ variable;}\\
&= \nu W_{\phi \times X}.(\phi \times \psi_X) \text{ if } X \text{ is a } \nu \text{ variable.}\\
\phi \times \mu X.\psi_X &= \mu W_{\phi \times X}.\phi \times \psi_X;\\
\phi \times \nu X.\psi_X &= \nu W_{\phi \times X}.\phi \times \psi_X;\\
\phi \times \psi \wedge \psi' &= (\phi \times \psi) \wedge (\phi \times \psi');\\
\phi \times \psi \vee \psi' &= (\phi \times \psi) \vee (\phi \times \psi').
\end{aligned}
$$

where $\gamma \in \{\mu, \nu\}$, and $\Omega_{\Psi \times \mathrm{WinningRegion}}(W_{X \times \phi}) = \Omega_{\mathrm{WinningRegion}}(X)$.

The construction terminates since the second time any branch is about to compute $\phi \times \psi_X$, the fixpoint variable $W_{\phi \times X}$ is bound instead.

Proof. (Theorem 107) To prove the correctness of this construction, it is sufficient to compare rule by rule the parity games $(\mathcal{M} \times \Psi) \times \mathrm{WinningRegion}$ and $\mathcal{M} \times (\Psi \times \mathrm{WinningRegion})$. For all ϕ, ψ, it is indeed the case that $(s \times \phi) \times \psi$ is the same parity game as $s \times (\phi \times \psi)$. The formula $\Psi \times \mathrm{WinningRegion}$ inherits its priority assignments from $\mathrm{WinningRegion}$, hence it is in the same alternation class as $\mathrm{WinningRegion}$.

Since $\mathcal{M} \models \Psi \Leftrightarrow \mathcal{M} \times \Psi \models \mathrm{WinningRegion} \Leftrightarrow \mathcal{M} \models \Psi \times \mathrm{WinningRegion}$, the formula $\Psi \times \mathrm{WinningRegion}$, of the index of $\mathrm{WinningRegion}$, is equivalent to Ψ. □

This concludes the argument that if Ψ can be interpreted by a formula in an alternation class C, then Ψ is semantically in the class C. Since we have an effective transformation that turns Ψ into a formula which is syntactically in C, this allows for formulas to be simplified whenever we can find a suitable interpreting formula. Of course, deciding the descriptive complexity of a class of parity games is not necessarily easy. In the rest of this chapter, I study circumstances in which this can be done.

7.3 Low descriptive complexity games

This section considers how to find interpreting formulas by looking at a formula's syntax. It can be read as an advanced tutorial in L_μ-hacking on how to spot semantically simple formulas and prove that they can be simplified. The syntactic features studied here identify some alternation free and Π_2^μ formulas, but the methodology is more general: if one can prove that a syntactic feature make the model-checking game easy to describe, this means that the syntactic feature can be used to identify semantically simple formulas.

Example 109. The Σ_2^μ formula $\Psi = \mu X.\nu Y.A \rightarrow \{X\} \vee B \rightarrow \{Y, \top\}$ expresses that eventually, if Even chooses a successor whenever B holds, and Odd when only A holds, Even can force a play to reach a position from where there is an infinite path on which B is always true. Chapter 3, Section 3.4.3 proves that this formula is tableau equivalent to $\Phi = \mu X.A \rightarrow \{X\} \vee B \rightarrow \{Y, \top\} \vee \nu Y. \rightarrow \{Y, \top\}$. It does not however provide an effective transformation from the original formula into an alternation-free one.

This section will show that the syntax of Ψ, in particular its even cycles, is so simple that its model-checking games have low descriptive complexity. Then, Theorem 107 gives the transformation of the formula into a semantically alternation-free one which is morally the same as Φ, although its syntax may require some tidying up.

7.3.1 Restricted strategies

Recall that Chapter 5, discussed the characterisations of modal formulas as those which are local, and Σ_1^μ formulas as those which have finite proofs. The syntactic characteristic of having finite or bounded proofs can be understood as a restriction on Even's strategies: if Even can win, she can win with a strategy that only agrees with finite plays or plays of bounded length. Here I propose a similar approach to understanding Even's strategies in the model-checking games of alternation free and Π_2^μ formulas. The first type of strategies considered here, *eventually even strategies*, is extremely restrictive: every play has to reach a point from where odd priorities are no longer reached by the strategy. Syntactically alternation free formulas do not all have this property; only those in the first level of the weak alternation hierarchy do.

Definition 110. A strategy σ for Even is *eventually even* if every even play which agrees with σ eventually reaches a *even-critical point* – that is to say, a point from which σ does not reach odd fixpoints.

Lemma 111. *The winning regions of parity games in which eventually even strategies are sufficient for Even – i.e. if Even wins, she can win with an eventually even strategy – are described by an alternation free formula in the first level of the weak alternation hierarchy.*

Proof. I will show that on classes of parity games with priorities $\{q,...,0\}$ on which Even can win with eventually even strategies, the formula $\mathrm{Parity}_{q,0}$ is equivalent to $\mathrm{EventuallyEven}_{q,0}$:

$$\mu X.T \vee \bigvee_{i\leq q} E_i{\to}\{X,\top\} \vee O_i{\to}\{X\} \vee$$

$$\nu Y.T \vee \bigvee_{i\in\{i\leq q|even(i)\}} E_i{\to}\{Y,\top\} \vee O_i{\to}\{Y\}$$

The first part of the formula is identical to the standard formula to describe winning regions of parity games except that every fixpoint encountered is μ-bound. The second part is identical, but allows only the encounter of Even priorities; the fixpoints associated with these positions are ν-bound. Given an eventually even winning strategy σ for Even, the formula fragment $\nu Y.T \vee \bigvee_{i\in\{i\leq q|even(i)\}} E_i{\to}\{Y,\top\} \vee O_i{\to}\{Y\}$ holds at even-critical points since σ no longer reaches odd priorities.

It follows that $P \models \mathrm{Parity}_{q,0}$ implies $P \models \mathrm{EventuallyEven}_{q,0}$ whenever Even has an eventually even winning strategy σ in a parity game P: To win in $P \times \mathrm{EventuallyEven}_{q,0}$ it suffices for her to copy σ while staying in the first part of the formula to start with. The game is then identical to $P \times \mathrm{Parity}_{q,0}$ except that all priorities encountered are odd. Since σ is eventually even, every play reaches a critical point. Even can then win by moving to the second part of the formula.

The direction $\mathrm{EventuallyEven}_{q,0}$ implies $\mathrm{Parity}_{q,0}$ is trivial as it is true on all parity games: a winning strategy in $\mathrm{EventuallyEven}_{q,0}$ directly translates into a winning strategy in $\mathrm{Parity}_{q,0}$. □

In exactly the same way, although with a slightly more involved formula, we define *eventually* Π_2^μ strategies and an interpreting formula.

Definition 112. A strategy σ for Even is *eventually* Π_2^μ if every play of dominant priority d which agrees with σ eventually reaches a *d-critical point* – that is to say, a point from which σ does not reach fixpoints more significant than d. While any play by definition eventually no longer reaches priorities greater than the dominant one, here we require that eventually no play branching from p reaches such a priority either.

Lemma 113. *The winning regions of parity games in which eventually* Π_2^μ *strategies are sufficient for Even – i.e. if Even wins, she can win with an eventually* Π_2^μ *strategy – are described by a* Π_2^μ *formula.*

Proof. We define the interpreting formula inductively:

$$\mathrm{EventuallyPi}_{2,1} = \nu Y.\mu X.T \vee E_2{\rightarrow}\{Y,\top\} \vee O_2{\rightarrow}\{Y\} \vee E_1{\rightarrow}\{X,\top\} \vee O_1{\rightarrow}\{X\}$$

$$\mathrm{EventuallyPi}_{q,1} = \nu Y.\mu X.T \vee E_q{\rightarrow}\{Y,\top\} \vee O_q{\rightarrow}\{Y\} \vee \bigvee_{i<q} E_i{\rightarrow}\{X,\top\} \vee O_i{\rightarrow}\{X\}$$
$$\vee\ \mathrm{EventuallyPi}_{q-2,1}$$

$\mathrm{EventuallyPi}_{2,1}$ is just $\mathrm{Parity}_{2,1}$. I now show that for all q, on parity games in which eventually Π_2^μ strategies are sufficient for Even, $\mathrm{EventuallyPi}_{q,1}$ is equivalent to $\mathrm{Parity}_{q,1}$. The direction $\mathrm{EventuallyPi}_{q,1}$ implies $\mathrm{Parity}_{2,1}$ for all q is easy: for all parity games P, a winning strategy for Even in $P \times \mathrm{EventuallyPi}_{q,1}$ directly translates into a winning strategy in $P \times \mathrm{Parity}_{q,1}$.

For the other direction, assume that Even has an eventually Π_2 winning strategy in $P \times \mathrm{Parity}_{q,1}$. Then Even's strategy in $P \times \mathrm{EventuallyPi}_{q,1}$ is as follows: copy the modal choices of σ while staying in the first part of the formula. When a play reached a d-critical point, σ no longer reaches any priorities higher than d, so Even moves to the formula $\mathrm{EventuallyPi}_{d,1}$. Every play eventually reaches a critical point, and after a d-critical point, only p-critical points with $p \leq d$ can be reached.This strategy is winning since a play with dominant priority d will eventually be played in $\mathrm{EventuallyPi}_{d,1}$ where d is associated with the most significant ν-variable. □

This gives us two interpreting formulas to describe the winning regions of formulas which generate parity games in which Even does not need complex winning strategies. These are only two examples of how looking at necessary strategies can help identify semantically simple formulas. A similar argument can be made for strategies corresponding to each level of the weak alternation hierarchy for example; we could also look at the dual, restricted strategies for Odd, or describe strategies of higher complexity. For now, however, I only consider these two examples and look at the syntactic characteristics which imply that such strategies are sufficient for Even.

7.3.2 Syntactic features

The syntactic features considered here focus on the winning cycles in a formula, *i.e.* the paths through a formula's parse tree which correspond to winning plays for Even. I will define two syntactic characteristics of these cycles, which can be used to recognise semantically alternation-free and Π_2^{μ} formulas respectively.

Definition 114. A cycle in a formula is a cycle in the parse-tree of the formula, with back edges from fixpoint variables X to the formula binding them $\nu X.\phi_X$ or $\mu X.\phi_X$. Note that a cycle may go through the same subformula more than once. A cycle is a ν-cycle, or an even cycle, if the most significant fixpoint along it is ν-bound. Otherwise it is a μ-cycle, or odd.

Recall that the first level of the weak alternation hierarchy, or the alternation-free hierarchy, consists of formulas such as $\mu X.\Diamond X \vee \nu Y.B \wedge \Box Y$, where there are no true alternations and only one weak alternation. A formula which is syntactically in the first level of the alternation-free hierarchy clearly only allows eventually even winning strategies for the even player, since μ-bound fixpoints are not free in ν-bound formulas. However, an arbitrary formula of any alternation depth can have an equivalent syntactic property, described in the following definition:

Definition 115. (*eventually even winning cycles*) A formula has *eventually even winning cycles* if for every ν-cycle c in Ψ, it is the case that:

- at a conjunction $\alpha \wedge \beta$ in c, if β is the next formula in c, then no μ-bound fixpoint is reachable from α;
- at a universal modality $\Box\phi$, only ν-bound fixpoints are reachable from ϕ.

Recall that a universal modality is $\Box\phi$ or $\rightarrow B$ where $B \neq \{\phi, \top\}$, while $\Diamond\phi$, or equivalently $\rightarrow\{\phi, \top\}$, is an existential modality.

The intuition is that conjunctions and universal modalities in a winning cycle result in branching in a winning strategy, so to guarantee that least fixpoints are eventually not reachable in a winning strategy, the syntax has to restrict the behaviour on the plays which branch from the winning cycle. Note that this criterion is very restrictive. For instance if a least fixpoint is reachable from a ν-bound variable Y, then no cycle dominated by Y can pass through a universal modality.

Example 116. The formula $\Psi = \mu X.\nu Y.A \wedge \Box X \vee B \wedge \Diamond Y$ has eventually even winning cycles since its winning cycles have no conjunctions nor universal modalities.

The following definition is slightly less restrictive. We will see that it is a sufficient criterion for a formula being semantically Π_2^μ.

Definition 117. (Π_2 *winning cycles*) A formula has Π_2 *winning cycles* if for every ν-cycle c in Ψ, of dominant greatest fixpoint variable Y, it is the case that:

- at a conjunction $\alpha \wedge \beta$ in c, if β is the next formula in c, then in α no fixpoint more significant than Y is reachable;
- at a universal modality $\Box\phi$, no fixpoint more significant than Y is reachable.

These definitions restrict behaviour at conjunctions. This is due to the fact that we are considering the complexity of Even's winning strategies, which branch at conjunctions. These definitions can of course be dualised, to address disjunctions.

We can then prove the theorems stating that these syntactic features imply semantic simplicity.

Theorem 118. *If Ψ has eventually even winning cycles, then the winning regions of the model-checking games of Ψ can be described by an alternation-free formula in the first level of the weak alternation hierarchy.*

Theorem 119. *If Ψ has Π_2 winning cycles, then the winning regions of the model-checking games of Ψ can be described by a Π_2^μ formula.*

To prove these, it suffices to note that the syntactic criteria imply that the corresponding strategies are sufficient for Even to win in the model checking game of any model. Then, using the Lemmas from the previous section, this implies that the formula is interpreted by an alternation free or Π_2^μ formula.

Lemma 120. *If a formula Ψ has eventually even winning cycles, then for all $\mathcal{M}$ such that $\mathcal{M} \models \Psi$, Even can win in $\mathcal{M} \times \Psi$ with a strategy which is eventually even.*

Proof. If a strategy σ is winning for Even in $\mathcal{M} \times \Psi$, then every infinite play has to eventually enter a cycle c of Ψ, dominated by some Even fixpoint. Since Ψ has eventually even winning cycles, and σ only branches at conjunctions and universal modalities, σ can not reach any odd priorities after entering its final winning cycle. □

Lemma 121. *If a formula Ψ has Π_2 winning cycles, then for all $\mathcal{M}$, such than $\mathcal{M} \models \Psi$, Even can win $\mathcal{M} \times \Psi$ with a strategy which is eventually Π_2.*

Proof. If a strategy σ is winning for Even in $\mathcal{M} \times \Psi$, then every infinite play has to eventually enter a cycle c of Ψ, dominated by some Even fixpoint Y of priority d, and remain there. Since Ψ has Π_2^μ winning cycles, and σ only branches at conjunctions and universal modalities, σ does not reach a more significant priority than d after a play has entered its final cycle dominated by d. $\square$

Theorem 118, follows directly from the fact that eventually even strategies suffice in the model checking games of those formulas (Lemmas 120) and that such games have low descriptive complexity (111). Similarly, Theorem 119,the corresponding result for Π_2^μ, follows from Lemmas 121 and 113. This concludes the proof that a formula is semantically in the first level of the weak alternation hierarchy, or in Π_2^μ, whenever its winning cycles are eventually even or eventually Π_1^μ respectively.

To summarise, some semantically simple formulas can be recognised by simple syntactic features that imply simple model-checking games. Formulas without these features can still have simple model-checking games. A hierarchy begins to form to distinguish how easily unnecessary complexity is recognised: some formulas are already syntactically of low index; some are not yet of low index but their syntax betrays their semantic index; some formulas may not be recognisable as simple from their syntax directly, but have simple model-checking games and are therefore interpreted by simple formulas. The next chapter considers whether some formulas have low semantic index despite producing model-checking games of high descriptive complexity.

7.4 Discussion

This chapter has revisited some of the foundations of L_μ theory in light of the index problem: the relationship between a formula's complexity and the descriptive complexity of its model-checking games is not only true at the syntactic level, but extends to the semantic level: the descriptive complexity of the model-checking games of a formula Ψ is an upper bound on the semantic complexity of Ψ. This can be used to simplify formulas: one can come up with semantic and syntactic features which provably indicate low descriptive complexity for the model-checking games of a formula.

While such syntactic features may be easy to check, they are by no means complete characterisations: plenty of semantically Π_2^μ formulas require Even to use complex strategies and are therefore not interpreted by EventuallyPi. A trivial example is $\psi \vee \neg\psi$ for any complex enough ψ. The question tackled in the next chapter is whether there is another Π_2^μ formula to interpret such formulas. More generally, is the semantic complexity of a formula an upper bound on the descriptive complexity of the formula's model-checking games?

Chapter 8

Interpretation theorems

Recall Theorem 107 which states that if a formula Ψ is interpreted by a formula in C, then Ψ is itself semantically in C. This section considers the converse: when is it the case that a formula semantically in C can be interpreted by a formula syntactically in C? The following conjecture can be studied with respect to any class C:

Conjecture 122. *If Ψ is semantically in an alternation class C, then there is a formula Φ syntactically in C such that for all structures $\mathcal{M}$, $\mathcal{M} \times \Psi \models \Phi$ if and only if $\mathcal{M} \models \Psi$.*

If this conjecture holds in general, then the descriptive complexity of the model-checking games of a formula is exactly its semantic complexity.

For classes C in which this conjecture holds, write that they have an *interpretation theorem*. Then, we are interested in whether the interpreting formula Φ can be decided from Ψ, *i.e.* whether the class C has an *effective* interpretation theorem.

This section considers this conjecture in light of what we know about deciding ML,Π^{μ}_{1} and Σ^{μ}_{2}.

In brief, it shows that:

- ML has an effective interpretation theorem;
- Π^{μ}_{1} has an effective interpretation theorem for disjunctive L_μ;
- Σ^{μ}_{2} has an interpretation theorem for disjunctive L_μ, and an effective one for Π^{μ}_{2};
- Disjunctive L_μ alternation classes all have an interpretation theorem for input in co-disjuntive form and vice-versa.

An interpretation theorem makes the semantic index of a formula an upper bound on the complexity of its model-checking games. It also establishes a structural relationship between a formula and an equivalent formula in its semantic alternation call. In general, if a formula Ψ is semantically in an alternation class C, the equivalent formula in C might not bear any relation to Ψ. However, an interpretation theorem will establish that an equivalent formula in C can be reached from Ψ, using the product-like transformation described in the previous chapter. The interpretation theorems in this chapter go further and give for an input formula Ψ a parameterised target formula $\Psi^C(n)$ in the target class C such that Ψ is semantically in C if and only if Ψ is equivalent to $\Psi^C(n)$ for some n.

This links the index problem back to a recurring theme of boundedness, in a similar spirit to the reduction [CL08] of the index problem for non-deterministic automata to the boundedness question for distance-parity automata. This result seems to be a cousin of the general interpretation theorem presented here for disjunctive alternation classes on co-disjunctive input formulas. In the framework of ordered structures, it does not demand that inputs are in the dual of non-deterministic form – this leaves some hope that perhaps the interpretation theorem of Section 8.4 could also be freed of such restrictions, at least on non-deterministic L_μ.

8.1 Interpretation theorem for ML

This section shows that conjecture 122 holds for semantically modal formulas:

Theorem 123. *(Effective Interpretation Theorem for ML) Let Ψ be a semantically modal formula, then there exists effectively a modal formula Φ such that Φ interprets Ψ,* i.e. *for all structures $\mathcal{M}$, it is the case that $\mathcal{M} \times \Psi \models \Phi$ if and only if $\mathcal{M} \models \Psi$.*

Recall from Chapter 5 that semantically modal formulas are equivalent to one of their finite approximations truncated at a suitable modal depth. More precisely, for a semantically modal formula, of modal rank m, for all structures $\mathcal{M}$, Even wins in $\mathcal{M} \times \Psi$ if and only if she wins in $\mathcal{M} \times \Psi^m_m$ where Ψ^m_m is Ψ where all fixpoints are approximated to the m^{th} level and truncated at modal depth m.

Definition 124. *(Bounded parity games)* An n-bounded parity game is a parity game with a set M of marked positions and a counter. Whenever a play reaches a position in M, the counter is decremented. Then, the winner of the game is decided as in a normal parity game, except that upon reaching a position in M at counter 0, the owner of the position loses.

Lemma 125. *Given a formula Ψ, the formula Ψ^n_n, that is to say Ψ, approximated n times and truncated at modal depth n, holds in a structure $\mathcal{M}$ if and only if Even wins the n bounded parity games on $\mathcal{M} \times \Psi$ where M is the set of modal positions,* i.e. *positions $v \times \Diamond\phi$ or $v \times \Box\phi$.*

Proof. The model checking game $\mathcal{M} \times \Psi^n_n$ has the same winner as the n-bounded parity game on $\mathcal{M} \times \Psi$: they are effectively identical until n modal positions are seen; then, in both games the owner of the next would-be modal position loses. $\square$

Definition 126. If the longest path between two positions in M is p, the winning regions of these games are described by the modal formula $\mathsf{Bounded}_I^{p,m}$ defined inductively as follows:

$$\begin{aligned}
\mathsf{Bounded}_I^{0,b} =& \bot \\
\mathsf{Bounded}_I^{a,0} =& (E_i \wedge \neg M \wedge \Diamond \mathsf{Bounded}_I^{a-1,0}) \vee \\
& (E_i \wedge M \wedge \bot) \vee \\
& (O_i \wedge \neg M \wedge \Box \mathsf{Bounded}_I^{a-1,0}) \vee \\
& (O_i \wedge M \wedge \top) \\
\mathsf{Bounded}_I^{a,b} =& (E_i \wedge \neg M \wedge \Diamond \mathsf{Bounded}_I^{a-1,b}) \vee \\
& (E_i \wedge M \wedge \Diamond \mathsf{Bounded}_I^{p,b-1}) \vee \\
& (O_i \wedge \neg M \wedge \Box \mathsf{Bounded}_I^{a-1,b}) \vee \\
& (O_i \wedge M \wedge \Box \mathsf{Bounded}_I^{p,b-1})
\end{aligned}$$

Proof. (of Theorem 123)

Let Ψ of index I be equivalent to a modal formula of modal depth m. Let p be the longest path in the parse-tree of Ψ (where ϕ_X is taken to be the child of X) without modalities. Since Ψ is taken to be guarded, p is finite. Then, the model-checking games of Ψ are m-bounded parity games where M is the set of positions $s \times \phi$ where ϕ is a modality. Then, noting that the longest path between two position in M in any model-checking parity game of Ψ is of length p, $\mathsf{Bounded}_I^{p,m}$ interprets Ψ. □

Hence the modal fragment of L_μ has an effective interpretation theorem: any semantically modal formula can be interpreted by a syntactically modal formula. The interpreting formula depends on the size of Ψ as well as its index, but is computable from the formula. This means that all semantically modal formulas, no matter how high their syntactic index, can only generate a class of model-checking parity games with modal descriptive complexity.

8.2 Interpretation theorem for Π_1^μ

The conjecture 122 also holds for *disjunctive formulas* that are semantically Π_1^μ.

Theorem 127. *(Effective Interpretation for Π_1^μ, on disjunctive formulas)*

Let Ψ be a disjunctive formula which is semantically in Π_1^μ. Then there exists effectively a Π_1^μ formula Φ such that Φ interprets Ψ.

As seen in Chapter 5, in disjunctive semantically Π_1^μ formulas, every μ-subformula can either be replaced with the same subformula bound by ν, or by $\bot$. This can be translated into the formula describing the winning regions of the model-checking games of the formula, simply by substituting the least fixpoint in the parity game formula corresponding to a subformula $\mu X.\phi$ with either $\bot$ or a greatest fixpoint, depending on whether the subformula is unsatisfiable.

For example, the appropriate formula describing the winning regions of a disjunctive formula in which all μ-bound subformulas are satisfiable would simply be:

$$\nu Y.E_0 \rightarrow \{Y, \top\} \vee E_1 \rightarrow \{Y, \top\} \vee O_0 \rightarrow \{Y\} \vee O_1 \rightarrow \{Y\} \vee T$$

For μ-bound subformulas which are unsatisfiable, it is enough to turn the clause $E_{X_i} \rightarrow B$ corresponding to the μ-variable X_i in question into $E_{X_i} \wedge \bot$.

Note that for modal formulas, the interpreting formula depends only on the modal rank of the formula and the length of the longest path without fixpoints in the formula. Since both of these can be bound in relation to the size of the formula, all semantically modal L_μ formulas of the same size can be interpreted by the same formula. This cannot be said for semantically Π_1^μ formulas: the interpreting formulas depend on which fixpoint subformulas are unsatisfiable and which are interchangeable with ν. It remains an open question whether a *uniform* interpreting formula could be devised for all semantically Π_1^μ formulas in disjunctive form.

Both of the modal and Π_1^μ cases are easy to spell out, with the benefit of hindsight, once we know how to simplify semantically modal or Π_1 formulas. In both the modal and Π_1^μ case, the syntactic optimisations that we perform on the formulas can instead be encoded into the interpreting formula.

However, the Π_1^μ transformation requires formulas to be in disjunctive form. What can be said of the conjecture for formulas which are not in disjunctive form? Consider the following example.

Example 128.

$$((\Box(\mu X.A \vee \Diamond X) \vee F) \wedge (C \vee \neg C \wedge \Box \nu Y.\neg A \wedge \Box Y)) \vee C \wedge \Box \nu X.\Diamond X$$

First let us argue that this formula is semantically in Π_1^μ. The important thing to note about this formula is that in the tableau, the least fixpoint subformula $\mu X.A \vee \Diamond X$ appears once in conjunction with $\nu Y.\neg A \wedge \Box Y$ and once with C. The former conjunction is equivalent to $\bot$. The other occurrence can be replaced with $\nu X.A \vee \Diamond X$ because of the disjunct $C \wedge \nu X.\Diamond X$: indeed, $\nu X.A \vee \Diamond X \wedge \neg \mu X.A \vee \Diamond X$ implies $\nu X.\Diamond X$.

However, it is not clear how to turn this formula into a Π_1^μ formula without first turning it into disjunctive form. Similarly, the previously described method does not work for interpreting this formula with a Π^μ_1 formula: $\mathcal{M} \times \Psi$ will contain some positions $v \times \mu X.\Diamond X \vee A$ which should be interpreted as $v \times \bot$, namely those occurring after positions in which C does not hold; other positions $w \times \mu X.\Diamond \vee A$ should be interpreted as an even priority position instead.

While this example does of course not rule out the existence of a Π^μ_1 formula to describe the parity games stemming from this formula, it gives some indication that such a formula will not be as simple to concoct as the previous ones.

8.3 Interpretation theorem for Σ^μ_2

The previous sections establish effective interpretation theorems for Π^μ_1, restricted to disjunctive formulas, and for the modal fragment of L_μ. These can be treated as exercises in understanding Conjecture 122, but on their own, they don't provide any further insight into deciding the alternation hierarchy, since both ML and Π^μ_1 are already known to be decidable. This section shifts the focus onto an alternation class not known to be decidable, Σ^μ_2: If Ψ is semantically Σ^μ_2, is it interpreted by some Φ in Σ^μ_2? If so, what does Φ look like?

In this section I argue that the first part of the proof of decidability of Σ^μ_2 for Π^μ_2 presented in Chapter 6 translates into an interpretation theorem for Σ^μ_2.

Theorem 129. *If a disjunctive formula Ψ is semantically in Σ^μ_2, then it is interpreted by a Σ^μ_2 formula.*

Proof. Recall that for the modal fragment of L_μ, we could define a family of formulas $\mathrm{Bounded}^{p,m}_I$ such that if a formula Ψ with index I is semantically modal, then $\mathrm{Bounded}^{p,m}_I$ interprets Ψ for some m. The value m can be bounded by an exponential in the size of Ψ, while p is no larger than $|\Psi|$, giving us the effective interpretation theorem for the modal fragment of L_μ.

In a similar vein, Chapter 6 already argued that for semantically Σ^μ_2 disjunctive formulas Ψ, with index I, the model-checking game corresponds to the n-challenge game, for some n: $\mathcal{M} \models \Psi$ if and only if even wins the n-challenge game on $\mathcal{M} \times \Psi$. Furthermore, for every Ψ, the formula $\Psi(n)$ holds in a structure $\mathcal{M}$ whenever Even wins the model-checking n-challenge game on $\mathcal{M} \times \Psi$. Then, in the same way we can construct $\mathrm{Challenge}^n_I = \mathrm{Parity}_I(n)$ which is true in a parity game G if and only if Even wins the n-challenge game on $\mathcal{G} \times \mathrm{Parity}_I$. However, G and $\mathcal{G} \times \mathrm{Parity}_I$ are the same parity game, modulo some tidying up [Grä11] which does not affect the winner of the challenge game. Therefore $\mathcal{M} \times \Psi \models \mathrm{Challenge}^n_I \iff \mathcal{M} \models \Psi(n) \iff \mathcal{M} \models \Psi$.

This makes $\mathrm{Challenge}_I^n$ the parameterised family of interpreting formulas for semantically Σ_2^μ formulas with index I. □

As noted in Chapter 6, a bound on n would yield a decidability theorem for Σ_2^μ.

This concludes the argument that the decidability proofs known so far for levels of the L_μ alternation hierarchy can be understood in terms of the descriptive complexity of model-checking games. In other words, semantically low alternation formulas, at least in disjunctive form, generate parity games of low descriptive complexity. The difficulty of establishing such theorems for non-disjunctive formulas highlights disjunctive form's crucial role in the index problem.

8.4 General interpretation theorem for disjunctive L_μ

Given that all existing decision procedures for L_μ alternation classes can be framed in terms of an interpretation theorem, it seems plausible that this could also be the case for higher alternation classes. This section generalises the argument for Σ_2^μ as far as possible with our current tools. One of the difficulties of generalising the Σ_2^μ construction to more priorities is that beyond $\{0,1\}$ König's lemma can no longer be used to find a finite bound after which the higher priority is seen on all plays on a branch. Instead, some concessions over the structure of target formulas, that is to say only considering disjunctive alternation classes, allows us to only have one play per branch. Then the proof generalises and we get an interpretation theorem for co-disjunctive formulas. Although the generality of this result is appealing, the restriction to disjunctive form in the target class and co-disjunctive form in the inputs highlights some of the difficulties that will probably need to be tackled to solve the index problem.

8.4.1 Generalised challenge game

The n-challenge game used in relation to the decidability of Σ_2^μ has one challenge per even priority of the original formula. This section generalises the construction for any target alternation class, by considering a set of challenges per priority rather than a single challenge. More precisely, fix J to be the *input* index, *i.e.* the index of the input formula, while I is the *target* index, *i.e.* the index of the target alternation class. Then for the pair J, I, this section defines a parameterised challenge game such that an input formula Ψ in co-disjuntive form of index J is interpreted by the formula (of index I), describing the winning regions of these games if and only if Ψ is equivalent to a disjunctive formula of index I.

As usual, this game is played on a parity game arena with priorities from J. Unlike for Σ_2^μ, the challenging player is now Even. This is because we are looking at target formulas in disjunctive form, which yield well-behaved strategies for Even. The input formula on the other hand is in co-disjunctive form, to yield well-behaved strategies for Odd. Instead of a single challenge per priority, Even can open a challenge on each odd priority of J at different levels, corresponding to priorities in I.

Definition 130. *(Challenge configuration)* A challenge configuration $(\bar{a},\bar{c})$ consists of $|I_o| \times |J_o|$ challenges, where I_o and J_o are the odd priorities in I and J respectively, each of which can be *met* or *open*, and their counters. Write $a_{i,j} = met$ or $open$ for $i \in I$ and $j \in J$ to indicate whether the i-level challenge on j is open. Each challenge $a_{i,j}$ is attached to a positive integer counter value $c_{i,j}$, bounded by n.

Given a configuration $(\bar{a},\bar{c})$, the least j for which $a_{i,j}$ is open for some i is the *priority* of the challenge configuration, and the highest level i at which $a_{i,j}$ is open is its *level*.

A valid challenge configuration respects the following constraint: if $a_{i,j} = open$ then $a_{i,k} = open$ for all $k > j$. That is to say, challenges are opened in decreasing order.

The game configuration consists of a position in the parity game and a challenge configuration. The progress of the game can be divided into two rounds: in the first round Even opens or resets challenges, while the second round is a step in the parity game. In the first round her possible actions are:

- To *k-reset* for any odd $k \in I_o$, setting $a_{i,j} := met$ and $c_{i,j} := n$ for all $i \leq k$;
- To open at any level i, challenges up to any p, as long as the counters allow it: for all $j \geq p$ such that $a_{i,j} = met$, set $a_{i,j} := open$ and $c_{i,j} := c_{i,j} - 1$ if $c_{i,j} > 0$.

Then, in the second round, the player whose turn it is in the parity game picks a successor position. If the underlying parity game ends in a terminal state, then the winner of the underlying parity game immediately wins the challenge game, too. The challenge configuration is updated according to the priority p of this new position:

- $a_{i,j} := met$ for all $j \leq p$, all $i \in I_o$;
- $c_{i,j} := n$ for all $j < p$, all $i \in I_o$.

If $c_{i,p} = 0$ for some i, then the game ends immediately with a win for Odd.

A play is a potentially infinite sequence of game configurations: an underlying parity game play augmented with challenge configurations. The dominant priority of the parity game play is also the dominant priority of the challenge game play. An infinite play with dominant priority d is winning for Odd if: d is odd, or all $a_{i,d+1}$ challenges are in the *met* state infinitely often.

Lemma 131. *The winning regions of a generalised n-challenge game for J, I are described by a L_μ formula with index I.*

For clarity, I describe the formula $\mathrm{GChallenge}^n_{J,I}$ as a L_μ-automaton.

Let A be the L_μ-automaton of $\mathrm{Parity}_{\mathrm{J}}$. For each valid challenge configuration $(\bar{a},\bar{c})$, of priority p and level k, let $A(\bar{a},\bar{c})$ be a copy of A with the following modifications:

- $\Omega_{(\bar{a},\bar{c})}(q) = k$ (odd) if $\Omega_A(q) < p-1$;
- $\Omega_{(\bar{a},\bar{c})}(q) = k+1$ (even) if $\Omega_A(q) \geq p-1$;

The components $A(\bar{a},\bar{c})$ are joined to form one automaton by adding transitions as follows:

- For a state q in $A(\bar{a},\bar{c})$ with $\Omega_A(q) = d \geq p$ and some i such that $a_{i,p} = \mathit{open}$ and $c_{i,p} = 0$, there is a unique transition to $\bot$, indicating an immediate win for Odd; else
- A state q in $A(\bar{a},\bar{c})$ with $\Omega_A(q) = d \geq p$ has a transition to the copy of q in $A(\bar{a}',\bar{c}')$ where: $a'_{i,j} = \mathit{met}$ for all i and $j \leq d$, $a'_{i,j} = a_{i,j}$ otherwise, and $c'_{i,j} = n$ for all i and $j < d$, and $c'_{i,j} = c_{i,j}$ otherwise; this indicates all challenges up to d being met.
- All other states q have a transition $\delta(q) = \bigvee Q_{open} \cup Q_{reset}$ where Q_{open} and Q_{reset} are as follows. Q_{open} is the set of states corresponding to q in the components $A(\bar{a}',\bar{c}')$ such that if $a_{i,j} = \mathit{open}$, then $a'_{i,j} = \mathit{open}$ and whenever $a_{i,j} = \mathit{met}$ but $a'_{i,j} = \mathit{open}$, then $c'_{i,j} = c_{i,j} - 1 \geq 0$; this corresponds to Even opening some challenges. Q_{reset} is the set of states corresponding to q in the components $A(\bar{a}',\bar{c}')$ such that for some $k \in I_o$, $a'_{i,j} = \mathit{met}$ and $c'_{i,j} = n$ for all $i \leq k$ and $a'_{i,j} = a_{i,j}$ and $c'_{i,j} = c_{i,j}$ otherwise; This corresponds to Even resetting counters up to level k.

A play of this automaton can be read as a play of the automaton A augmented with the challenge configurations $(\bar{a},\bar{c})$ corresponding to what component a state is visited in. The original priorities of states, as defined by Ω_A, correspond to the priorities in the underlying game. The moves between the components describe how the challenge configuration evolves as the Even player sets and resets challenges and as these are met. It suffices to check that the winning conditions of the challenge game are respected by the new parity assignment.

A play that reaches a terminal position corresponds to either a finite play in the parity game, or a challenge on p being met when a counter $a_{i,p}$ is at 0. We then need to consider three cases.

First consider infinite plays, dominated in the underlying parity game by an even priority d, where some $d+1$ challenge $a_{i,d+1}$ is eventually always *open*. Such a play should, according to the rules of the challenge game, be winning for Even. In the automaton of $\mathrm{GChallenge}^{\mathrm{n}}_{\mathrm{J,I}}$, the play eventually gets stuck in automaton components at some level k, corresponding to the most significant level $d+1$ is opened but not met at. The play visits a component with level k and priority $d+1$ infinitely often, at least whenever d is seen. The play will therefore see the even priority $k+1$ infinitely often. This will be the dominant priority since it is the largest priority within components of level k. Such a play is therefore winning for Even, as it should be.

Now consider infinite plays, dominated in the underlying parity game by an even priority d, where all $d+1$ challenges are always eventually *met*. Such a play should be winning for Odd according to the challenge game specifications. We consider the case that $d+1$ is opened finitely many times first. Then, the game eventually gets stuck in components of priority $d+3$ or more. In such components priorities d and lower receive Odd priorities, so Odd wins this game.

If $d+1$ is opened infinitely often at some maximal level k, Even must be resetting level k or higher infinitely often. The play therefore visits a component of level $k+2$ or higher with priority $d+3$ infinitely often, where it must see the odd priority k. The play can see no higher even priority (available only when $a_{i,j}$ is open for $i \geq k+2$ when a priority equal or higher to $j-1$ is seen) infinitely often since the $d+1$ challenge is not opened at a more significant level infinitely often. The play is therefore winning for Odd, as required.

Finally, if an infinite play is dominated by an odd priority d, all d challenges are always eventually *met*. Again, we can consider both cases where a d challenge is opened finite and infinitely often. If such challenges are only opened finitely often, the game gets stuck in components of priority $d+2$ where all nodes of original priorities d and lower receive an odd priority. If a d challenge is opened infinitely often at the maximal level k, then Even must be resetting level k or higher infinitely often, which means that the play visits a node of priority k infinitely often and, as above, cannot visit a higher even priority.

Lemma 132. *A co-disjunctive L_μ formula Ψ of index J is interpreted by $\mathrm{GChallenge}^n_{J,I}$ if it is equivalent to a disjunctive formula of index I.*

Proof. The proof structure is similar to the proof of Theorem 94 in Chapter 6 – here it just incorporates the additional challenges and resets. Furthermore, due to the added priorities, we have to constrain the target formula to be in disjunctive form and the input formula to be in co-disjunctive form, rather than invoking König's Lemma.

Assume that Ψ is equivalent to some disjunctive Φ of index I and that for all m, in particular a fixed $m > (|\Psi| + |\Phi|)^2$, there is a structure $\mathcal{M}$, such that Even wins the parity game $\mathcal{M} \times \Psi$ but Odd wins the generalised m-challenge game on $\mathcal{M} \times \Psi$. Without loss of generality, take $\mathcal{M}$ to be finitely branching. As for the proof of Theorem 94, this proof first uses a winning strategy σ for Even in $\mathcal{M} \times \Phi$ to define a challenging strategy γ for her in the generalised m-challenge game on $\mathcal{M} \times \Psi$ (Part I). Then Odd's winning strategy τ is used to add back edges to $\mathcal{M}$ (Part II), turning it into a new structure $\mathcal{M}'$ which preserves Even's winning strategy σ in $\mathcal{M}' \times \Phi$

while turning τ_γ into a winning strategy in $\mathcal{M}' \times \Psi$ (Part III). This contradicts the equivalence of Φ and Ψ.

Part I. Let σ be Even's well-behaved winning strategy in the parity game $\mathcal{M} \times \Phi$. For a branch b of $\mathcal{M}$, on which σ reaches a node v, indicate by $next_k(b,v)$, for even $k \in I$, the next node along b at which k or a higher even priority is seen. If k or higher is not seen after v, leave it undefined. For each odd $k \in I$ define R_k the set of nodes at which σ sees the odd priority k.

Now consider the generalised m-challenge game on the arena $\mathcal{M} \times \Psi$. Let Even's challenging strategy γ be: to open all challenges at the start of the game, and whenever its counter is reset; if challenges for a priority j is met at v, and none of its counters $c_{i,j}$ is at 0, open the next challenge $a_{i,j}$ when the play reaches a node $next_{i-1}(b,v)$ for any branch b, unless the counter is reset before then. Even resets level k upon reaching a position in R_k.

Part II. Odd wins the generalised m-challenge game on $\mathcal{M} \times \Psi$, so let τ be his winning strategy. Recall that τ_γ is an Odd's strategy for Ψ up to the point where an m^{th} challenge is met, and undefined thereafter. Since Ψ is co-disjunctive, we can adjust $\mathcal{M}$ into a bisimilar structure in which the pure parity game strategy τ_γ is well-behaved wherever it is defined – it reaches each position of $\mathcal{M}$ at either one subformula, or none.

The strategy τ_γ is winning in the challenge game against any strategy for Even which uses the challenging strategy γ. Even always eventually opens a challenge on every priority, at some level, and only resets level k infinitely often on a branch if all challenges at level $k+2$ infinitely often are *open* infinitely often. Hence the only way for her to lose is to either lose finitely in the underlying parity game or for the play to reach a position of priority p when for some i, $c_{i,p} = 0$. Thus, every play in the challenge game that agrees with τ_γ and the challenging strategy γ is finite.

Since τ_γ is well-behaved, each branch carries at most one play of the parity game. For every branch b with a play on it, the play either ends in a winning position for Odd in the underlying parity game, or in a long streak in which the highest priority seen is some odd p, and it is seen at least m times, corresponding to every instance of Odd meeting a p-challenge, set on some level k. As long as m is sufficiently large, on every such branch there are two nodes v and its descendant w, at which Even opens challenge $a_{k,p}$, which agree on the set of subformulas that τ_γ reaches there in $\mathcal{M} \times \Psi$ and that σ reaches there in $\mathcal{M} \times \Phi$. Now consider the structure $\mathcal{M}'$, which is as $\mathcal{M}$ except that the predecessor of each w-node has an edge to v instead. The strategies τ_γ and σ transfer in the obvious way to $\mathcal{M}'$.

Part III. Finally, let us show that σ is winning in $\mathcal{M}' \times \Phi$ and that τ_γ is winning in

$\mathcal{M}' \times \Psi$. Starting with τ_γ, consider plays that do not go through back edges infinitely often. They must be finite, and end in a position that is winning for Odd, as in the challenge game on $\mathcal{M} \times \Psi$. Any play in $\mathcal{M} \times \Psi$ that agrees with τ_γ which sees both v and w is dominated by an odd priority between v and w. Then, as the w and v agree on which subformula σ_γ reaches them at, an odd priority dominates plays that go through back edges in $\mathcal{M}' \times \Psi$ infinitely often: strategy τ_γ is winning in $\mathcal{M}' \times \Psi$.

Now onto σ in $\mathcal{M}' \times \Phi$. If a branch is unchanged by the transformation, then any play on it is still winning for σ. If a branch that σ plays on has been changed, then consider in $\mathcal{M}$ the two nodes v and w at which the transformation is done. These both are nodes at which according to γ, Even opens a $a_{k,p}$ challenge for the k and p for which Even runs out of challenges in the challenge game on $\mathcal{M}$. Therefore, from the definition of $next_{k-1}$ and γ, σ sees $k - 11$ between them and no odd priority k or larger, as witnessed by the lack of reset of k-counters. Since v and w agree on which subformula σ reaches them at, any play in $\mathcal{M}' \times \Phi$ which goes through a back-edge infinitely often is dominated by the even priority $k - 1$.

This contradicts the equivalence of Ψ and Φ. Therefore, if Ψ is semantically equivalent to a disjunctive formula of index I, then for all structures $\mathcal{M}$ the m-challenge game and the parity game on $\mathcal{M} \times \Psi$ have the same winner for $m > (|\Phi| + |\Psi|)^2$, *i.e*, Ψ is interpreted by $\mathrm{GChallenge}^m_{J,I}$ □

This yields an interpretation theorem for all disjunctive Σ^μ alternation classes, for co-disjunctive input formulas: I is an upper bound on the descriptive complexity of the model checking games generated by the co-disjunctive form of Ψ.

8.5 Discussion

After the last chapter showed that the descriptive complexity of the model-checking games of Ψ is an upper bound on the semantic complexity of Ψ, this chapter studied in what circumstances the converse is also true: when is the semantic complexity of Ψ an upper bound on the semantic complexity of the model-checking games of Ψ. That is to say, are formulas semantically in a class C interpreted by a formula in C? For classes for which this is the case, the descriptive complexity of model-checking games corresponds exactly to the semantic complexity of formulas. Furthermore, the interpreting formulas are an indication of what the equivalent formula in the low alternation class looks like, if it exists.

For low alternation classes ML,Π^μ_1 and Σ^μ_2, the existing decision procedures provided clues on how to construct the interpreting formula. For both Π^μ_1 and Σ^μ_2, the interpretation theorem only applies for disjunctive input formula. This once again highlights the importance of disjunctive form for the index problem: the descriptive complexity of the model-checking games of disjunctive formulas seems to be lower than for general L_μ formulas.

The last section generalises the challenge-game construction from Chapter 6 to any disjunctive alternation class, and shows that any co-disjunctive formula equivalent to a disjunctive formula in a class C is interpreted by a disjunctive formula in C.

The generality of this result comes at a cost. Since it is restricted to disjunctive target classes and co-disjunctive input classes, it is less powerful than the interpretation theorems for L_μ described in the previous section. In particular, it is not constructive, in the sense that if a formula is indeed equivalent to a disjunctive formula of index I, even if we find the interpreting disjunctive formula of index I, this only allows us to construct a formula of index I, which may not itself be disjunctive.

The current techniques meet their limits as the definition of $next_k$ in the proof of Lemma 132 requires us to be able to pinpoint a moment when it is safe for the challenger to open the next challenge. In the Σ_2^μ case we could use König's lemma to achieve this while in the generalised version disjunctive form makes this straightforward. Generalising the non-disjunctive case requires a novel strategy.

If the recent work on the Rabin–Mostowski index problem for parity automata [CKLV13, SW16] is the automata-theoretic cousin of the decidability proof of Σ_2^μ for Π_2^μ in Chapter 6, then the proof of the last section is related to the reduction of the Rabin–Mostowski index problem of non-deterministic automata to the boundedness of distance-parity automata [CL08]. The presentation is of course very different, and the proof methods are not obviously recognisable, but the underlying idea of reducing an index problem to a question of boundedness is identical. Interestingly, in the framework of automata and ordered structures, the restrictions to co-disjunctive input formulas does not seem to be necessary. This could either be a symptom of the difference between the index problems for L_μ and automata, or a sign that there is hope for an interpretation theorem for disjunctive alternation classes on disjunctive formulas, rather than co-disjunctive formulas. It may also be interesting to consider these questions on the non-deterministic fragment of L_μ. It is likely that for this fragment the restriction to co-disjunctive input could be lifted; furthermore, perhaps the outstanding boundedness questions are easier to answer on this well-behaved fragment.

This concludes the last part of the thesis, which argued that the fundamental relationship between formula complexity and the descriptive complexity of parity games does not only hold with respect to syntactic complexity, but also extends to the semantic complexity of formulas.

Chapter 9

Conclusions and outlook

9.1 Summary and contributions

This thesis has revisited the L_μ index problem, with a focus on syntactic simplifications. Part I studied two syntactically defined fragments of L_μ – disjunctive L_μ and non-deterministic L_μ – and how the restrictions defining these fragments affect the index problem. The main contributions were:

1. A survey on the benefits of disjunctive form and a detailed rewriting of the proof that disjunctive form is expressively complete;
2. Proof that the transformation into disjunctive form preserves Π^μ_2 but not Σ^μ_2;
3. The introduction of the non-deterministic fragment of L_μ and a preliminary study of its index problem;
4. An alternative proof that non-deterministic automata are as expressive as alternating parity automata; and
5. A precise exposition of the relation between various fragments of L_μ and non-deterministic parity automata, highlighting how the index problems differ in each case.

Part II provided three novel decision procedures for low L_μ alternation classes. Each of these is focused on generating the simplified formula, rather than just deciding its existence. The contributions of Part II were:

1. An alternative proof that *ML* is decidable, focused on finding the equivalent modal formula;
2. An alternative proof that Π^μ_1 is decidable, also focused on the target formula;
3. A proof that given a formula in Π^μ_2, it is decidable whether it is in Σ^μ_2. For arbitrary input, Chapter 5 gives a sequence of formulas $\Psi(n)$ such that Ψ is semantically in Σ^μ_2 if and only if this sequence eventually coincides with Ψ, *i.e.* Ψ is equivalent to $\Psi(n)$ for some n.

Finally, Part III tied the index problem to the descriptive complexity of parity games. Most significantly, it revisits the fundamental relationship between formula complexity and parity game complexity, and shows that this relationship is more robust than previously thought with respect to semantic complexity. It then uses this connection to present a uniform approach to the alternation hierarchy, which subsumes all the decision procedures of Part II. The core contributions are:

1. A proof that the descriptive complexity of the model-checking games of Ψ is an upper bound to the semantic index of Ψ;
2. A proof that the converse is true for ML, Π^{μ}_{1} and Σ^{μ}_{2}: for these classes, the winning regions of the model-checking games of a disjunctive formula semantically in C are described by a formula in C;
3. A proof that the winning regions of the model-checking games of a co-disjunctive formula that is equivalent to a disjunctive formula in an alternation class C are described by a disjunctive formula in C;
4. A study of how to identify formulas with model-checking games of low descriptive complexity.

One of the aims of this thesis has been to develop strategies to analyse and simplify L_{μ} formulas. It therefore seems fitting to include a brief guide to approaching seemingly complex L_{μ} formulas, which summarises the practical contributions of this thesis. This document is in the appendices and can be read as an inventory of tools with which to analyse a formula in the hopes of finding simplifications.

9.2 Directions for further work

I leave the reader with ten open questions, each of which I find interesting and believe to be useful for furthering out understanding of the L_{μ} alternation hierarchy.

9.2.1 Fragments of L_{μ}

This thesis has demonstrated the importance of disjunctive form. Perhaps further research on disjunctive L_{μ} could overcome some of the challenges of the index problem.

1. **Deciding the index problem with respect to tableau equivalence.** Instead of asking for the lowest index of any formula equivalent to Ψ, we can ask for the lowest index of any *tableau equivalent* formula. Chapter 3 begins to address this question by studying differences in indices for tableau equivalent formulas: Π^{μ}_{2} is closed with respect to tableau equivalence; Σ^{μ}_{2} is not. Developing methods for finding low-index tableau equivalent formulas to Σ^{μ}_{2} formulas and higher would

further our understanding of both disjunctive form and the index problem. In particular, research in this direction could have implications for generalising the Σ_2^μ decision procedure to non Π_2^μ input.

2. **Interpretation theorems for non-disjunctive formulas.** Most of the interpretation theorems in Chapter 8 use formulas in disjunctive or co-disjunctive form. Do disjuntive formulas truly generate model-checking games of lower descriptive complexity than arbitrary formulas?
3. **Non-deterministic L_μ.** Chapter 4 introduced the non-deterministic fragment of disjunctive L_μ; it calls for further investigations. As Chapter 4 discusses, the index problem for non-deterministic L_μ seems simpler than the general index problem. To the best of my knowledge there are no non-trivial fragments of L_μ for which the index-problem is solved; I propose non-deterministic L_μ as a candidate for such a milestone. It may also be worth investigating whether the methods of Chapter 6, 7 and 8 can be pushed any further on this fragment.

9.2.2 The index problem

The index problem itself is of course still open. Chapters 6 to 8 suggest some directions for further investigation.

4. **Deciding Σ_2^μ.** Chapter 6 presents formulas $\Psi(n)$ such that Ψ is semantically Σ_2^μ if and only if it equivalent to $\Psi(n)$ for some n. While this parameter is bounded for input formulas in Π_2^μ, finding a general bound would immediately decide Σ_2^μ for all of L_μ.
5. **Finding characterisations for higher alternation classes.** Chapters 5, 6 and 7 all study what fundamentally characterises an alternation class. Higher alternation classes are more difficult to understand, and characterisations are more complex. The methodology of Section 7.3 however allows for partial characterisations. Describing strategies characteristic of Π_3^μ parity games may be a way to begin understanding how higher-index formulas could be characterised.

9.2.3 Interpretation theorems

Chapters 7 and 8 studied the relationship between the semantic index of a formula and the descriptive complexity of its model-checking games. Some of the avenues for further work are:

6. **Finding bounds.** The interpretation theorems for Σ_2^μ and disjunctive alternation classes of Chapter 8 involve parameterised interpreting formulas. Finding the bounds for this parameter remains one of the main challenges of the index problem.

7. **Counter-examples.** Formulas for which this parameter value is higher than 1 are difficult to find and seem to require intricate constructions. There is therefore a large gap in our understanding: we don't seem to be able to find an upper bound for this parameter, yet finding examples for which values larger than 1 is necessary also seems difficult. Perhaps finding such examples for Σ_2^μ and beyond could help us understand the truly difficult instances of the index problem.
8. **A generalised interpretation theorem.** The interpretation theorem for all disjunctive alternation classes considers input formulas in co-disjunctive form. An equivalent theorem for disjunctive input would be stronger as it would allow the low complexity disjunctive formula to be built from the input formula. It would also bring the L_μ theory in line with the automata-theoretic counterpart, which reduces the Rabin-Mostowski index problem to a boundedness question on distance-parity automata.

9.2.4 The L_μ index problem and automata

9. **Comparing simulation theorems.** In Chapter 4, I present a transformation from L_μ formulas to non-deterministic L_μ formulas that agree on all ranked structures. This is a logical equivalent to the automata-theoretic result that non-deterministic parity automata are as expressive as alternating ones [MS95]. It is not clear to what extent the two transformations are similar. In particular, given a disjunctive formula Ψ, the L_μ transformation yields a tableau equivalent non-deterministic formula. Does the automata-theoretic transformation turn the automaton corresponding to Ψ into a non-deterministic automaton corresponding to a tableau equivalent formula?
10. **Comparing index-problems.** Throughout this thesis, I have discussed the differences between L_μ and automata: the existence of an index-preserving transformation between disjunctive L_μ and non-deterministic automata seems highly unlikely; the index problems for disjunctive L_μ, non-deterministic L_μ and non-deterministic automata are all distinct. In a similar vein, the index problem for L_μ and non-amorphous alternating parity automata are also distinct: despite the existence of an index-preserving translation, a formula can have higher semantic index than the corresponding non-amorphous automaton. However, considering the similarities between the L_μ and automata-theoretic versions of the index problem, it would be interesting to either (a) find a reduction between them, or (b) understand the cases in which fundamentally different tools are needed.

Appendix A

A guide to simplifying L_μ formulas

Although decision procedures for levels of the L_μ alternation hierarchy remain sparse, there are plenty of ways to analyse L_μ formulas and eliminate unnecessary complexity. This document is an inventory of tools that can be used to simplify L_μ formulas.

A.1 Decision procedures

The most obvious way of simplifying L_μ formulas is to use the available decision procedures to check whether a formula is equivalent to a low-alternation formula. Chapter 5 describes decision procedures for ML and Π^μ_1: given a formula Ψ, they describe formulas Ψ^{ML} and $\Psi^{\Pi^\mu_1}$ such that Ψ is equivalent to one of these if and only if it is in the corresponding alternation class. The formulas Ψ^{ML} corresponds to Ψ in which fixpoints are approximated to a large enough level, corresponding to the semantic modal rank of a formula.

Example 133. The formula $\Box\Box\Box\bot \wedge \mu X.A \vee \Diamond X$ is modal of rank 3 and equivalent to:

$$\Box\Box\Box\bot \wedge A \vee \Diamond(A \vee \Diamond(A \vee \Diamond(A \vee \bot)))$$

The semantic modal rank can be at least sub-exponentially larger than the formula itself, but it is bounded above by an exponential. There is therefore little hope for a more efficient tranformation.

The formula $\Psi^{\Pi^\mu_1}$ requires Ψ to be in disjunctive form. It consists of:

$$\Psi[\bot/\mu X.\phi]_{\forall X\in U}[\nu X.\phi/\mu X.\phi]_{\forall X\notin U}$$

where X ranges over μ-bound fixpoint variables and U is the set of μ-bound fixpoint variables that are bound by unsatisfiable formulas.

Example 134.

$$(\mu X.A{\rightarrow}\{X\}) \vee (\mu Y.B{\rightarrow}\{Y,\top\} \vee C) \vee (\nu Z.B{\rightarrow}\{Z,\top\})$$

The first disjunct $\mu X.A{\rightarrow}\{X\}$ is unsatisfiable, so it can be replaced with $\bot$. Then, looking at $(\mu Y.B{\rightarrow}\{Y,\top\} \vee C) \vee (\nu Z.B{\rightarrow}\{Z,\top\})$, the remaining formula is satisfied when C is reachable via a path along which B holds, or when there is an infinite path along which B holds. In the model-checking games of this formula, Odd never plays μ infinitely often to win the model-checking game: if there is a B path which never reaches C, such a structure is not a counter-model, since Even can play to $\nu Z.B{\rightarrow}\{Z,\top\}$ and win. The parity of the fixpoint binding Y is inconsequential, and μ can be replaced with ν.

This formula is therefore semantically Π^μ_1, and equivalent to:

$$\bot \vee (\nu Y.B{\rightarrow}\{Y,\top\} \vee C) \vee (\nu Z.B{\rightarrow}\{Z,\top\})$$

Or just $\nu Y.B{\rightarrow}\{Y,\top\} \vee C$ after simplifications.

For input formulas in Π^μ_2, Chapter 6 shows a decision procedure for Σ^μ_2. The Σ^μ_2 formula, which is in fact alternation-free, is designed based on the challenge-game, described in Chapter 6.

Example 135. The projection $\Psi^{\Sigma^\mu_2}$ for $\Psi = \nu Y.\mu X.\phi(X,Y)$ is

$$\nu W^n.\nu Y.\phi(Y,Y) \wedge \mu X.\phi(X,W)$$

for some sufficiently large value of n. The intuition of the formula is that the first copy of ϕ corresponds to the challenge game when the challenge is met and Odd has to open a challenge by moving to the second copy of ϕ. Meeting the challenge is encoded by the fixpoint W, and the approximation n accounts for the counting of challenges.

In each of these cases the result formula is closely related to the original formula so these transformations give a clear account of how the incidental complexity is eliminated. By duality, we can also recognise semantically Σ^μ_1 formulas and syntactic Σ^μ_2 formulas that are semantically in Π^μ_2.

A.2 Disjunctive form

We have a much wider range of tools at our disposal than just the decision procedures for the low levels of the alternation hierarchy. Throughout the thesis disjunctive form has been used in many proofs and manipulations; however, at its simplest, just turning a formula into disjunctive form can reduce its index.

Example 136. This formula is in Σ_1^μ when turned into disjunctive form, regardless of the complexity of ψ:

$$\mu X.\Box x \wedge \psi$$

On the other hand, unless a formula is already in Π_2^μ, the transformation into disjunctive form can also increase the index of a formula an arbitrarily large amount. Chapter 3 both proves that this increase does not happen for Π_2^μ and shows examplse of Σ_2^μ formulas with much higher index in disjunctive form.

For some formulas, co-disjunctive form may be more useful.

A.3 Descriptive complexity

Chapter 7 revisited the connection between the index of a formula and the descriptive complexity of its model-checking games. The main result is that the descriptive complexity of the model-checking games of Ψ is an upper bound on the semantic complexity of Ψ. This gives us an additional tool for analysing formulas: analysing the parity games a formula can generate. Chapter 7 had a closer look at analysing the complexity of the even cycles of a formula, and outlining some syntactic features which imply semantic simplicity.

Example 137. Consider the formula $\Psi = \mu X.\nu Y.A{\rightarrow}\{X\} \vee B{\rightarrow}\{Y,\top\}$. Its winning cycles have no conjunctions nor universal modalities, it is therefore semantically alternation-free. The equivalent alternation-free formula is:

$$\mu X.A{\rightarrow}\{X\} \vee B{\rightarrow}\{X,\top\} \vee \nu Y.B{\rightarrow}\{Y,\top\}$$

More generally, if a syntactic feature can be shown to restrict the model-checking games sufficiently to guarantee that their winning regions can be described by a simple formula, then this yields a formula simplification.

A.4 Parameterised projections

The decision procedure for Σ_2^μ of Chapter 6 described a parameterised family of formula $\Psi(n)$ in Σ_2^μ such that Ψ is semantically in Σ_2^μ if and only if it is equivalent to $\Psi(n)$ for some n. Chapter 6 only establishes a bound for n if the input formula is in Π_2^μ. However, examples that require high n are very complex to construct. For practical purposes, testing whether Ψ is equivalent to $\Psi(n)$ for a low n is likely to suffice. In fact, even the bound in Chapter 6 for Π_2^μ input formulas seems higher than necessary for formulas that have not been constructed specifically to display pathological behaviour.

Bibliography

[AG11] Krzysztof R Apt and Erich Grädel. *Lectures in game theory for computer scientists*. Cambridge University Press, 2011.

[AN07] André Arnold and Damian Niwiński. Continuous separation of game languages. *Fundam. Inf.*, 81(1-3):19–28, January 2007.

[AU79] Alfred V Aho and Jeffrey D Ullman. Universality of data retrieval languages. In *Proceedings of the 6th ACM SIGACT-SIGPLAN symposium on Principles of programming languages*, pages 110–119. ACM, 1979.

[BDM16] Massimo Benerecetti, Daniele Dell'Erba, and Fabio Mogavero. Solving parity games via priority promotion. In *International Conference on Computer Aided Verification*, pages 270–290. Springer, 2016.

[BG93] Orna Bernholtz and Orna Grumberg. Branching time temporal logic and amorphous tree automata. In *International Conference on Concurrency Theory*, pages 262–277. Springer, 1993.

[Bra98] Julian C. Bradfield. The modal mu-calculus alternation hierarchy is strict. *Theoretical Computer Science*, 195(2):133–153, 1998.

[Bra99] Julian C Bradfield. Fixpoint alternation: Arithmetic, transition systems, and the binary tree. *RAIRO-Theoretical Informatics and Applications*, 33(4-5):341–356, 1999.

[BS07] Julian C. Bradfield and Colin Stirling. Modal mu-calculi. *Handbook of modal logic*, 3:721–756, 2007.

[CJK$^+$16] Cristian S Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasipolynomial time, 2016.

[CKLV13] Thomas Colcombet, Denis Kuperberg, Christof Löding, and Michael Vanden Boom. Deciding the weak definability of Büchi definable tree languages. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 23. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2013.

[CL08] Thomas Colcombet and Christof Löding. The non-deterministic Mostowski hierarchy and distance-parity automata. In *International Colloquium on Automata, Languages, and Programming*, pages 398–409. Springer, 2008.

[DMPV16] Antonio Di Stasio, Aniello Murano, Giuseppe Perelli, and Moshe Y. Vardi. *Solving Parity Games Using an Automata-Based Algorithm*, pages 64–76. Springer International Publishing, Cham, 2016.

[EC82] E Allen Emerson and Edmund M Clarke. Using branching time temporal logic to synthesize synchronization skeletons. *Science of Computer programming*, 2(3):241–266, 1982.

[EJ88] E Allen Emerson and Charanjit S Jutla. The complexity of tree automata and logics of programs. In *Foundations of Computer Science, 1988., 29th Annual Symposium on*, pages 328–337. IEEE, 1988.

[EJ91] E Allen Emerson and Charanjit S Jutla. Tree automata, mu-calculus and determinacy. In *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, pages 368–377. IEEE, 1991.

[FL79] Michael J Fischer and Richard E Ladner. Propositional dynamic logic of regular programs. *Journal of computer and system sciences*, 18(2):194–211, 1979.

[Flo67] Robert W Floyd. Assigning meanings to programs. *Mathematical aspects of computer science*, 19(19-32):1, 1967.

[FMS13] Alessandro Facchini, Filip Murlak, and Michal Skrzypczak. Rabin-mostowski index problem: a step beyond deterministic automata. In *Proceedings of the 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, pages 499–508. IEEE Computer Society, 2013.

[GPSS80] Dov Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi. On the temporal analysis of fairness. In *Proceedings of the 7th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 163–173. ACM, 1980.

[Grä11] Erich Grädel. Back and forth between logic and games. *Lectures in game theory for computer scientists*, pages 99–145, 2011.

[GS86] Yuri Gurevich and Saharon Shelah. Fixed-point extensions of first-order logic. *Annals of pure and applied logic*, 32:265–280, 1986.

[GTW02] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata Logics, and Infinite Games: A Guide to Current Research.* Springer-Verlag New York, Inc., New York, NY, USA, 2002.

[HKP12] Michael Huth, Jim Huan-Pu Kuo, and Nir Piterman. *The Rabin Index of Parity Games*, pages 259–260. Springer Berlin Heidelberg, Berlin, Heidelberg, 2012.

[HM80] Matthew Hennessy and Robin Milner. On observing nondeterminism and concurrency. In *International Colloquium on Automata, Languages, and Programming*, pages 299–309. Springer, 1980.

[HM85] Matthew Hennessy and Robin Milner. Algebraic laws for nondeterminism and concurrency. *Journal of the ACM (JACM)*, 32(1):137–161, 1985.

[HMP77] David Harel, Albert R Meyer, and Vaughan R Pratt. Computability and completeness in logics of programs (preliminary report). In *Proceedings of the ninth annual ACM symposium on Theory of computing*, pages 261–268. ACM, 1977.

[Hoa69] Charles Antony Richard Hoare. An axiomatic basis for computer programming. *Communications of the ACM*, 12(10):576–580, 1969.

[Imm82] Neil Immerman. Upper and lower bounds for first order expressibility. *Journal of Computer and System Sciences*, 25(1):76–98, 1982.

[JW95] David Janin and Igor Walukiewicz. *Automata for the modal μ-calculus and related results*. Springer, 1995.

[JW96] David Janin and Igor Walukiewicz. On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In *International Conference on Concurrency Theory*, pages 263–277. Springer, 1996.

[Kir05] Daniel Kirsten. Distance desert automata and the star height problem. *RAIRO-Theoretical Informatics and Applications*, 39(3):455–509, 2005.

[Koz83] Dexter Kozen. Results on the propositional μ-calculus. *Theoretical computer science*, 27(3):333–354, 1983.

[Koz88] Dexter Kozen. A finite model theorem for the propositional μ-calculus. *Studia Logica*, 47(3):233–241, 1988.

[Kri63] Saul A Kripke. Semantical analysis of modal logic i normal modal propositional calculi. *Mathematical Logic Quarterly*, 9(5-6):67–96, 1963.

[KV98] Orna Kupferman and Moshe Y Vardi. Weak alternating automata and tree automata emptiness. In *Proceedings of the thirtieth annual ACM symposium on Theory of computing*, pages 224–233. ACM, 1998.

[KVW00] Orna Kupferman, Moshe Y Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM (JACM)*, 47(2):312–360, 2000.

[KW02] Ralf Küsters and Thomas Wilke. Deciding the First Level of the μ-calculus Alternation Hierarchy. (0209), 2002.

[Leh15] M Karoliina Lehtinen. Disjunctive form and the modal μ alternation hierarchy. *FICS15, The 10th International Workshop on Fixed Points in Computer Science, EPTCS 191*, page 117, 2015.

[Len96] Giacomo Lenzi. A hierarchy theorem for the μ-calculus. In Friedhelm Meyer and Burkhard Monien, editors, *Automata, Languages and Programming*, volume 1099 of *Lecture Notes in Computer Science*, pages 87–97. Springer Berlin Heidelberg, 1996.

[LQ15] Karoliina Lehtinen and Sandra Quickert. Deciding the first levels of the modal μ alternation hierarchy by formula construction. In *LIPIcs-Leibniz International Proceedings in Informatics*, volume 41. Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, 2015.

[LQ17] M Karoliina Lehtinen and Sandra Quickert. Σ_2^μ is decidable for Π_2^μ. *Accepted to Computability in Europe 2017*, 2017.

[Mar75] Donald A Martin. Borel determinacy. *Annals of Mathematics*, pages 363–371, 1975.

[Mat02] Radu Mateescu. Local model-checking of modal mu-calculus on acyclic labeled transition systems. In Joost-Pieter Katoen and Perdita Stevens, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 2280 of *Lecture Notes in Computer Science*, pages 281–295. Springer Berlin Heidelberg, 2002.

[MO98] Markus Mller-Olm. Derivation of characteristic formulae. *Electronic Notes in Theoretical Computer Science*, 18:159 – 170, 1998.

[MRR16] Matthias Mnich, Heiko Röglin, and Clemens Rösner. *New Deterministic Algorithms for Solving Parity Games*, pages 634–645. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

[MS95] David E. Muller and Paul E. Schupp. Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of rabin, mcnaughton and safra. *Theoretical Computer Science*, 141(1):69–107, 1995.

[Nai82] Mohan Nair. A new method in elementary prime number theory. *Journal of the London Mathematical Society*, s2-25(3):385–391, 1982.

[Niw86] Damian Niwiński. *On fixed-point clones*, pages 464–473. Springer Berlin Heidelberg, Berlin, Heidelberg, 1986.

[Niw97] Damian Niwiński. Fixed point characterization of infinite behavior of finite-state systems. *Theoretical Computer Science*, 189(1):1–69, 1997.

[NW05] Damian Niwiński and Igor Walukiewicz. Deciding nondeterministic hierarchy of deterministic tree automata. *Electron. Notes Theor. Comput. Sci.*, 123(C):195–208, March 2005.

[Ott99] Martin Otto. *Eliminating Recursion in the μ-Calculus*, pages 531–540. Springer Berlin Heidelberg, Berlin, Heidelberg, 1999.

[Pit06] Nir Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. In *In 21st Symposium on Logic in Computer Science (LICS'06)*, pages 255–264. IEEE Computer Society, 2006.

[Pnu77] Amir Pnueli. The temporal logic of programs. In *Proceedings of the 18th Annual Symposium on Foundations of Computer Science*, SFCS '77, pages 46–57, Washington, DC, USA, 1977. IEEE Computer Society.

[Pra76] Vaughan R. Pratt. Semantical Consideration on Floyd–Hoare Logic. In *Proceedings of the 17th Annual Symposium on Foundations of Computer Science*, SFCS '76, pages 109–121, Washington, DC, USA, 1976. IEEE Computer Society.

[Pra81] Vaughan R. Pratt. A decidable mu-calculus: Preliminary report. In *Proceedings of the 22Nd Annual Symposium on Foundations of Computer Science*, SFCS '81, pages 421–427, Washington, DC, USA, 1981. IEEE Computer Society.

[SA05] Luigi Santocanale and André Arnold. Ambiguous classes in μ-calculi hierarchies. *Theoretical Computer Science*, 333(1):265–296, 2005.

[Ste89] Bernhard Steffen. *Characteristic formulae*, pages 723–732. Springer Berlin Heidelberg, Berlin, Heidelberg, 1989.

[SW16] Michal Skrzypczak and Igor Walukiewicz. Deciding the Topological Complexity of Büchi Languages. In Yuval Rabani Ioannis Chatzigiannakis, Michael Mitzenmacher and Davide Sangiorgi, editors, *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, volume 55 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 99:1–99:13, Dagstuhl, Germany, 2016. Schloss Dagstuhl–Leibniz-Zentrum für Informatik.

[Var82] Moshe Y. Vardi. The complexity of relational query languages (extended abstract). In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing*, STOC '82, pages 137–146, New York, NY, USA, 1982. ACM.

[Var88] Moshe Y. Vardi. Decidability and undecidability results for boundedness of linear recursive queries. In *Proceedings of the Seventh ACM SIGACT-SIGMOD-SIGART Symposium on Principles of Database Systems*, PODS '88, pages 341–351, New York, NY, USA, 1988. ACM.

[Var14] Thomas Varghese. *Parity and Generalised Büchi Automata Determinisation and Complementation*. PhD thesis, 2014.

[Wal95a] Igor Walukiewicz. Completeness of Kozen's Axiomatisation of the Propositional Mu-Calculus. In *Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science*, LICS '95, pages 14–, Washington, DC, USA, 1995. IEEE Computer Society.

[Wal95b] Igor Walukiewicz. Notes on the propositional μ-calculus: Completeness and related results. *BRICS, NS-95-1, Tech. Rep*, 1995.

[Wal02] Igor Walukiewicz. Monadic second-order logic on tree-like structures. *Theoretical Computer Science*, 275(1):311–346, 2002.

[Wil01] Thomas Wilke. Alternating tree automata, parity games, and modal μ-calculus. *Bulletin of the Belgian Mathematical Society Simon Stevin*, 8(2):359, 2001.