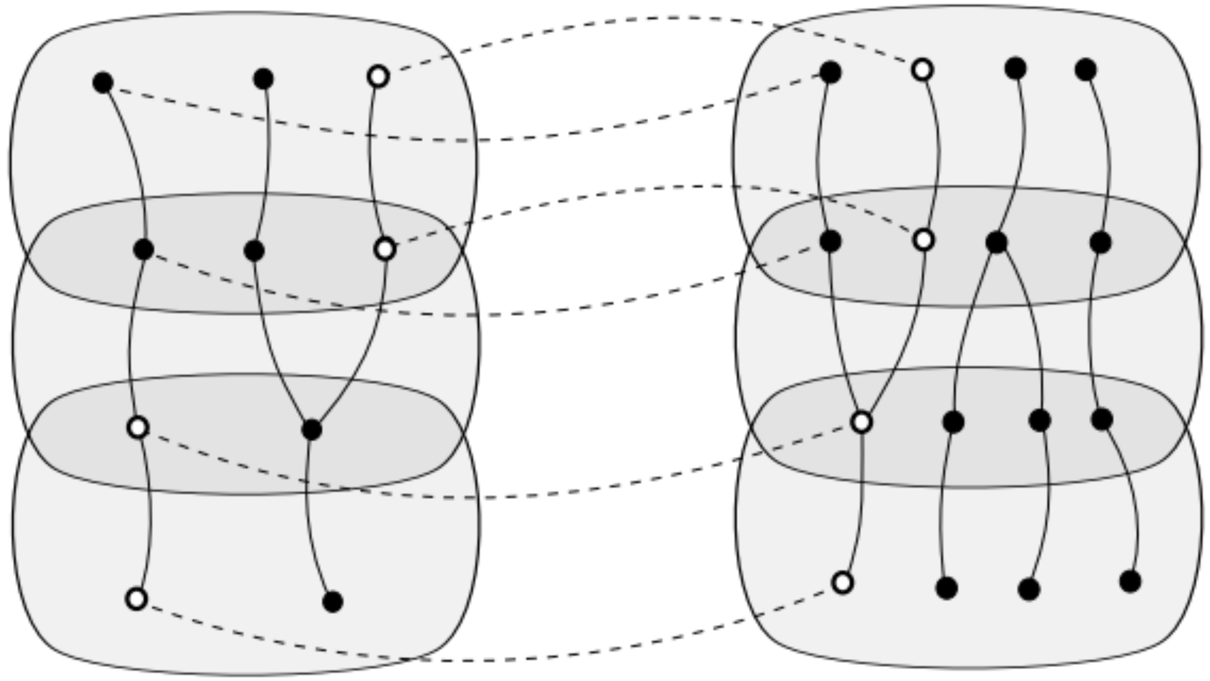
Coalgebraic fixpoint logic

Expressivity and completeness results



Fatemeh Seifan

Fatemeh Seifan
Coalgebraic fixpoint logic Expressivity and completeness results

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

Coalgebraic fixpoint logic
Expressivity and completeness results

Fatemeh Seifan

Coalgebraic fixpoint logic
Expressivity and completeness results

ILLC Dissertation Series DS-2024-05

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

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

The research for/publication of this doctoral thesis received financial assistance from Vrije Competitie grant 612.001.115 of the Netherlands Organisation for Scientifc Research (NWO).

Cover design by Fatemeh Seifan.
Printed and bound by US-AB.

Coalgebraic fixpoint logic: expressivity and completeness results

ACADEMISCH PROEFSCHRIFT

ter verkrijging van de graad van doctor
aan de Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. ir. P.P.C.C. Verbeek
ten overstaan van een door het College voor Promoties ingestelde commissie,
in het openbaar te verdedigen in de Aula der Universiteit
op woensdag 5 juni 2024, te 17.00 uur

door Fatemeh Seifan
geboren te Teheran

Promotiecommissie

Promotores:	prof. dr. Y. Venema	Universiteit van Amsterdam
	prof. dr. H.H. Hansen	Rijksuniversiteit Groningen
Copromotores:	dr. N.J.S. Enqvist	Stockholms universitet
Overige leden:	dr. A. Baltag	Universiteit van Amsterdam
	dr. N. Bezhanishvili	Universiteit van Amsterdam
	prof. dr. P. van Emde Boas	Universiteit van Amsterdam
	prof. dr. L. Schröder	Friedrich-Alexander University Erlangen-Nürnberg
	prof. dr. I. Walukiewicz	Université de Bordeaux

Faculteit der Natuurwetenschappen, Wiskunde en Informatica *to Afshin*

Contents

Acknowledgments **xi**

1 Introduction **1**

1.1 Modal logic, fixpoints and automata . . . 1
1.1.1 Modal logics . . . 1
1.1.2 The standard μ-calculus . . . 3
1.1.3 Automata for μ-calculus . . . 4
1.2 Coalgebras . . . 6
1.2.1 Systems as coalgebra . . . 6
1.2.2 Equivalence . . . 8
1.3 Coalgebraic logic and automata . . . 9
1.3.1 Coalgebraic modal logic . . . 10
1.3.2 Coalgebraic fixpoint logic . . . 12
1.3.3 Coalgebraic automata theory . . . 12
1.4 Contributions . . . 14
1.4.1 Chapter 3: Uniform Interpolation . . . 14
1.4.2 Chapter 4: Expressive Completeness . . . 16
1.4.3 Chapter 5: Axiomatic Completeness . . . 17

2 Preliminaries **21**

2.1 Modal logic and fixpoints . . . 21
2.1.1 Modal logics . . . 21
2.1.2 μ-calculus over Kripke models . . . 24
2.2 Parity games . . . 26
2.3 Category theory and coalgebras . . . 28
2.3.1 Sets and relations . . . 28
2.3.2 Coalgebras . . . 29
2.3.3 Properties of set functors . . . 33
2.3.4 Behavioural equivalence and Bisimulation . . . 36

2.3.5 Algebras vs coalgebras . . . 39
2.4 Coalgebraic fixpoint logic and automata . . . 40
2.4.1 Coalgebraic μ-calculus via the ∇-modality . . . 40
2.4.2 Coalgebraic μ-calculus via predicate liftings . . . 42
2.4.3 Coalgebraic automata theory . . . 44

3 Uniform Interpolation **47**
3.1 Special relation liftings . . . 48
3.2 Coalgebraic fixpoint logic . . . 54
3.3 Coalgebraic Automata . . . 60
3.4 Simulation . . . 62
3.5 Closure properties . . . 69
3.6 Logic and Automata . . . 74
3.7 Uniform Interpolation . . . 75
3.8 The monotone μ-calculus . . . 76
3.9 Conclusion . . . 77

4 Expressive Completeness **79**
4.1 The μ-calculus and monadic second-order logic . . . 82
4.2 Coalgebraic perspective . . . 83
4.2.1 Coalgebraic μ-calculus . . . 83
4.2.2 Coalgebraic $\mathtt{MSO}$ . . . 84
4.2.3 Coalgebraic $\mathtt{FOL}$ . . . 86
4.3 Automata and one-step languages . . . 88
4.3.1 Automata for $\mu\mathtt{ML}_\Lambda$ . . . 89
4.3.2 Automata for $\mathtt{MSO}_\Lambda$ . . . 90
4.3.3 Closure properties . . . 92
4.3.4 Second-order automata . . . 95
4.4 One-step expressive completeness . . . 103
4.5 Bisimulation invariance . . . 108
4.5.1 Adequate uniform constructions . . . 114
4.5.2 Weakly adequate uniform constructions . . . 115
4.5.3 Characterizing $\mu\mathtt{ML}_\Lambda$ inside $\mu\mathtt{ML}_{\Lambda\cup\{\Box^s\}}$ . . . 119
4.5.4 Applications . . . 126
4.6 Characterizing the monotone μ-calculus . . . 129
4.6.1 No adequate construction for $\mathcal{M}$ . . . 129
4.6.2 A weakly adequate uniform construction for $\mathcal{M}$ . . . 130
4.7 Conclusion . . . 137

5 Axiomatic Completeness **139**
5.1 Proof strategy . . . 140
5.2 Coalgebraic fixpoint logic . . . 142
5.2.1 Syntax . . . 142

5.2.2 Semantics . . . 143
5.2.3 Axiomatics . . . 143
5.3 Modal automata and one-step formulas . . . 145
5.3.1 One-step logic . . . 145
5.3.2 Modal automata . . . 153
5.3.3 Operations on modal automata . . . 154
5.3.4 Translating formulas to automata . . . 158
5.4 Games for Automata . . . 159
5.4.1 Traces . . . 160
5.4.2 The satisfiability game . . . 161
5.4.3 Consequence game . . . 163
5.5 Taming the traces - one step at a time . . . 166
5.5.1 Functional, clusterwise functional and thin relations . . . 167
5.6 Disjunctive and semi-disjunctive automata . . . 169
5.6.1 A key lemma . . . 175
5.7 A strong simulation theorem . . . 180
5.8 From automata to formulas . . . 194
5.9 Completeness . . . 201
5.10 Conclusion . . . 204

Bibliography **205**

Samenvatting **217**

Abstract **219**

Acknowledgments

It has taken me many years to finally complete this thesis, and now all that remains is to express my gratitude to the individuals who have been with me on this long journey.

On the top of my list are my promotor Yde Venema and co-promotors Sebastian Enqvist and Helle Havid Hansen. As a master's student in Iran, I had the opportunity to read many of Yde's articles. It wasn't until I had the opportunity to attend his lecture at ILLC that everything suddenly fell into place. He is an exceptional teacher who knows how to convey the intuition behind the most abstract concepts and their interconnections without sacrificing the precision. I have learned a lot from him both academically and personally and always enjoyed his interest in art and literature. While meeting his high standards as a researcher and supervisor was challenging, the experience has left me with cherished memories. I am deeply grateful for his care and patience over the years.

I am thankful to Sebastian for accepting to be my second supervisor. His support and generous willingness to share his knowledge were pivotal in completing this thesis. Our brainstorming sessions and his detailed feedback on various drafts of manuscripts and this thesis were invaluable.

I wish to express my gratitude to Helle, who joined the supervisory team later in the process. Without her, I can't imagine this thesis would have been completed. I appreciate Helle for her valuable feedback on numerous versions of this thesis, for her encouragement, and for all the late afternoon and weekend meetings. I am deeply thankful for all the support.

I extend my sincere gratitude to the members of my dissertation committee, Alexandru Baltag, Nick Bezhanishvili, Peter van Emde Boas, Lutz Schröder and Igor Walukiewicz, who have kindly agreed to read and evaluate my dissertation.

I am also thankful to Majid Alizadeh and Shahram Mohsenipoor for sparking my passion for logic.

To my family, especially my parents and sisters, I am grateful for their support. My mom's encouragement and belief in me during difficult times were instrumental in keeping me motivated on completing this thesis.

Last but foremost, with all my heart, I would like to thank Afshin, my best friend, who has been with me on this journey the longest.

Chapter 1

Introduction

The main topic of this thesis is coalgebraic fixpoint logic. Our general aim is to strengthen the link between the areas of logic, automata and coalgebra. More in particular, we provide a coalgebraic generalization of the automata-theoretic approach towards modal fixpoint logic, and address some questions regarding expressivity and completeness of coalgebraic fixpoint logic.

In this introduction we give an overview of how logic and automata meet coalgebras. Technical preliminaries are provided in Chapter 2. The structure of this chapter is as follows. In the next section we sketch the relation of modal logic, fixpoints and automata. Section 1.2 presents coalgebras as general models for transition systems. In Section 1.3 we explain how coalgebras encounter modal logic, fixpoints and automata and provide a suitable uniform framework to study various modal fixpoint logics. We finish with Section 1.4 where we discuss the contributions of this thesis.

1.1 Modal logic, fixpoints and automata

The connection between logic and automata has proven to be a fruitful area which provides useful insight on both topics. In this section we give a brief overview of the relation between logics featuring in this thesis and automata. We begin with modal logic as a key ingredient and building block for modal fixpoint logic. We refer the reader to [BdRV02] and [vB10] for thorough introductions to modal logic.

1.1.1 Modal logics

Modal logic is a branch of logic that extends classical propositional and predicate logic to include *modal* operators that formalise modalities such as “necessity” and “possibility”. Modal logic was originally developed by philosophers in the beginning of the 20th century and has been considered as a framework for philosophical logic. From the early 1930s, two types of mathematical semantics for modal logic developed: algebraic semantics and relational semantics. Algebraic semantics interprets modal operators on Boolean algebras and relational semantics, known as Kripke semantics, interprets modalities over relational structures called Kripke models. Kripke semantics for modal logic, which is of particular interest in this thesis, was introduced by Saul Kripke [Kri59] in late 1950s and had an irrefutable impact on the development of the theory of modal logic. The elements of Kripke models are thought of as possible worlds, moments of time or states of a computer program or a process. This view opened the door to connections with different disciplines. Modal logic crossed over to linguistics via the *Montague grammar* approach to natural language semantics [Mon74], and to computer science, where processes are modelled as relational structures [HM80, Mil81]. Indeed many of the logics that are used in computer science today for verification of hardware and software systems are variations of modal logics [BK08, HTK00].

Starting from the 1970s, an extensive theory has been developed for modal logics, including model theory and proof theory. One of the highlights of that theory was to take a perspective on modal logic, as a *fragment* of classical logics, such as first-order and second-order logics, rather than as an *extension* of classical propositional logic. A key result in this area was proved by Johan van Benthem, who showed that modal logic is the bisimulation-invariant fragment of first-order logic [vB76]. This theorem provides a characterization of the expressive power of modal logic. Compared to classical logics, modal logics have a good balance between *expressivity* and *complexity*; they tend to be *decidable* [Var96]. This balance is the foundation for the application of modal logics in automated program verification.

In the following, we present three examples of modal logics. More variations will be covered in Section 2.1.1.

Examples

1.1.1. EXAMPLE. *Basic modal logic* [BdRV02] is obtained by adding modal operators $\Diamond$ and $\Box$ to classical propositional logic. Formulas of this logic are evaluated over Kripke models, which are triples (W, R, V) where W is the set of states (worlds), $R \subseteq W \times W$ is an accessibility relation that relates a state to its successor, or "possible alternative worlds", and $V : \mathtt{X} \to \mathcal{P}W$ is a valuation of the set $\mathtt{X}$ of proposition letters of a given modal logic.

The semantics of modal formulas $\Diamond\varphi$ and $\Box\varphi$ over a Kripke model (W, R, V) is defined as follows:

$$\begin{aligned} w \Vdash \Diamond\varphi \quad &\text{iff} \quad v \Vdash \varphi \text{ for some } v \in W \text{ such that } wRv, \\ w \Vdash \Box\varphi \quad &\text{iff} \quad v \Vdash \varphi \text{ for all } v \in W \text{ such that } wRv. \end{aligned}$$

This semantics gives the reading of $\Diamond\varphi$ as "possibly φ" as φ holds in at least one world that is accessible from the current world. Similarly, $\Box\varphi$ reads as "necessarily φ" as φ holds in all worlds accessible from the current world.

1.1.2. EXAMPLE. As is clear from the semantics of $\Diamond$ and $\Box$ in basic modal logic, these modal operators just check the existence of successors with certain properties. To generalize these modalities, one may decide to *count* such successors of a current state in a Kripke model. This way Fine [Fin72] defined *graded* modalities $\Diamond^k$ and $\Box^k$ for *graded modal logic* with the following semantics over a Kripke model (W, R, V):

$$\begin{aligned} w \Vdash \Diamond^k\varphi \quad &\text{iff} \quad |\{v \in W \mid wRv \text{ and } v \Vdash \varphi\}| \geq k, \\ w \Vdash \Box^k\varphi \quad &\text{iff} \quad |\{v \in W \mid wRv \text{ and } v \nVdash \varphi\}| < k. \end{aligned}$$

This reads as $w \Vdash_V \Diamond^k\varphi$ iff "φ holds in at least k successors of w", and $w \Vdash_V \Box^k\varphi$ iff "φ fails in less than (!) k successors of w". It is clear that we obtain $\Diamond$ and $\Box$ as the special cases $\Diamond = \Diamond^1$ and $\Box = \Box^1$.

1.1.3. EXAMPLE. *Monotone modal logic* [Che80] is a generalization of basic modal logic in which the distribution of $\Box$ over conjunctions has been weakened to a monotonicity condition. The standard semantics for such logics is provided by so-called *monotone neighborhood models.* A monotone neighbourhood model is a structure (W, M, V) which has a set W of states and a valuation V, but instead of an accessibility relation, a function $M : W \to 2^{2^W}$ assigning to each state an upwards-closed set of subsets, called its neighbourhoods. The interpretation of $\Box\varphi$ in a monotone neighbourhood model (W, M, V) is then defined as follows:

$$w \Vdash \Box\varphi \quad \text{iff} \quad \{v \in W \mid v \Vdash \varphi\} \in M(w).$$

1.1.2 The standard μ-calculus

The standard μ-calculus is a fixpoint logic obtained by adding the least fixpoint operator μ and the greatest fixpoint operator ν to basic modal logic. It originates from seminal work of Dana Scott and Jaco de Bakker [SB69] on program verification, and is of interest to computer science because of its expressive power for describing properties of labeled transition systems. In the context of modal logic, works of Emerson, Clarke [CE80] and Park [Par80] on *correctness* and *fairness* properties of parallel programs, and work of Pratt [Pra81] on a decidable extension of modal logic with a minimization operator, prefigured the definition of μ-calculus. Their work on extending modal logics with fixpoints led Kozen to introduce and develop the version of μ-calculus used nowadays [Koz83]. This logic, which is presented in detail in Section 2.1.2, has a standard syntax and semantics over Kripke models (see Definitions 2.1.7 and 2.1.8), and yet the fixpoint operators, by capturing the full power of recursion, allow the logic to encode many dynamic and temporal logics including propositional dynamic logic PDL [FL79], branching time logic CTL* [EL86] and its widely used fragments – linear temporal logic LTL [Pnu77] and computation tree logic CTL [CE81]. We refer the reader to Chapter 3 in [BS01] for a more detailed discussion of the relation between fixpoints and recursion.

Where the μ-calculus subsumes several interesting modal logics for verification, it can itself be viewed as a fragment of monadic-second order logic MSO. MSO is the fragment of second-order logic SO, that only allows quantification over subsets of the domain. On binary trees, the μ-calculus is equivalent to MSO [Rab69, EJ91a, Niw88, Niw97]. On arbitrary structures, however, the μ-calculus is a proper fragment of MSO. A central result to this thesis is the theorem by Janin and Walukiewicz [JW96], who proved that over labelled transition systems, the μ-calculus is the bisimulation-invariant fragment of MSO. This result extends Johan van Benthem's characterisation theorem for modal logic, which states that basic modal logic is the bisimulation-invariant fragment of first-order logic.

From an algebraic perspective, the μ-calculus can be seen as an algebra of monotonic functions over a complete lattice. From this viewpoint, the semantics of μ-calculus formulas is based on the Knaster-Tarski fixpoint theorem [Tar57], which states that a monotone function over a complete lattice has a least fixpoint. The existence of a greatest fixpoint is then assured by duality. We refer the interested reader to [AN01], where the μ-calculus is studied mainly as an algebraic system rather than a logic.

Another important aspect of the μ-calculus is its tight connection to games. The game semantics for μ-calculus is defined via an infinite two-player game with perfect information; a *parity* game between players called $\exists$ and $\forall$. A priority map assigns to each position in the game a natural number called its *priority*. An infinite match is then winning for $\exists$ (losing for $\forall$) if and only if the minimum priority that occurs infinitely often in the match is even (see Section 2.2). The advantage of this parity game semantics is that it is more intuitive than the algebraic semantics, especially for complex formulas with nested fixpoint operators. As a framework, parity games can help to answer questions about the μ-calculus. A good example is the model checking problem for the μ-calculus which has been shown to be equivalent to the problem of solving parity games [Mos91, EJ91a]. Conversely, the μ-calculus provides a useful setting to study and describe properties of parity games. We refer the reader to Chapter 4 of [AN01] where properties of the μ-calculus are employed to prove results on parity games.

1.1.3 Automata for μ-calculus

Analogous to games, automata are effective tools in the theory of μ-calculus [GTW02]. In fact, most of the central results about the μ-calculus have been proved using automata as an alternative way of thinking about formulas.

An automaton consists of a set of *states* and a *transition map* defining how to get from one state to another state. Imposing extra structure on this basic definition, one obtains different types of automata operating on a variety of structures. There are two main dimensions along which we can classify automata. The first dimension is the type of the transition map which categorizes automata into the three classes of deterministic, non-deterministic, and alternating automata. The second dimension is the type of the acceptance condition, which describes when an automaton *accepts* or *rejects* a given structure. Here in this thesis we focus on the *parity acceptance condition.* Automata with a parity acceptance condition were introduced by Mostowski [Mos84] and Emerson and Jutla [EJ91a] independently. This parity condition is the same condition as in parity games, that is, an infinite run is accepting iff the minimum priority that occurs infinitely often is even.

The long tradition connecting logic and automata theory can be traced back to the seminal work of Büchi [Büc60] and Rabin [Rab69] showing that for infinite words and infinite binary trees, automata and MSO are expressively equivalent. In a more recent paper [Wal96] Walukiewicz introduced *alternating parity automata* for MSO on arbitrary trees, and generalised Rabin's result by proving that MSO and alternating parity automata have the same expressive power over arbitrary trees.

In the context of μ-calculus, automata provide a mathematical formalism to deal with the combinatorics of *traces* caused by the unravelling of fixpoint operators. The most frequently used automata for the μ-calculus are alternating parity automata for labeled transition systems [Wil01]. Two main results on the expressivity of μ-calculus that will be generalized to the level of coalgebras in this thesis using automata are the Janin-Walukiewicz [JW96] characterization theorem for the μ-calculus and MSO, and D'Agostino and Hollenberg's [DH00] result stating that the μ-calculus has uniform interpolation. We will provide more details about the automata-theoretic ideas and techniques used to prove these results in Section 1.4 and the respective chapters.

For now we focus on the axiomatisation of the μ-calculus to illustrate the efficiency of automata-theoretic techniques to establish important results in this area. Kozen, in [Koz83], aimed to prove a completeness result for the μ-calculus, but he discovered that the interaction between all connectives in μ-calculus led to difficulties in analysing fixpoint formulas. Hence he focused on a syntactic fragment he called *aconjunctive*, and proved a completeness result for this fragment. To address the difficulties that Kozen faced, Janin and Walukiewicz [JW95] viewed formulas as automata, and realised that while disjunctions are like nondeterministic choices, conjunctions cause "uncontrolled" branching of traces. This led them to define the class of *disjunctive* formulas in which conjunctions only occur in a restricted form by replacing the modal operators $\langle a \rangle$ and $[a]$ with a single connective.This connective is the labelled version of the ∇ modality that was independently introduced in coalgebraic modal logic for Kripke frames (see Section 1.3.1.1). Employing this connective they defined the so-called *μ-automata* (disjunctive automata) corresponding to disjunctive formulas. A key observation about this fragment of the μ-calculus is that many combinatorial difficulties encountered in the general case may be avoided for these formulas. For instance, the satisfaction problem is easy for disjunctive formulas. Janin and Walukiewicz utilized μ-automata to prove that every μ-formula is equivalent to a disjunctive one. A crucial part of their proof is based on a simulation theorem which shows that an alternating automaton can be transformed into an equivalent nondeterministic one [MS87].

Based on some of the ideas and results in [JW95], a few years later, Walukiewicz extended Kozen's completeness theorem and proved that Kozen's axiomatisation [Koz83] is sound and complete for the μ-calculus [Wal00].

1.2 Coalgebras

In this section, we aim to explain how coalgebra, the other main component of this thesis, enters into the picture of logic and automata we sketched in the previous section. Here we will give a brief introduction to basic coalgebraic concepts and leave further definitions for Section 2.3.2. For a detailed introduction to coalgebras in general we refer the reader to [Jac16] and [Rut00] where the theory of universal coalgebra and state-based evolving systems is developed.

The motivation to bring coalgebras to the theories of modal logic and automata is twofold. Mainly, coalgebras can be seen as generalisations of state-based systems such as streams, (infinite) trees, Kripke models, transition systems, and many others. Hence, coalgebra provides a general framework to study a variety of structurally different systems in a uniform way. On top of that, this framework comes with a general notion of equivalence and several reasoning principles. Secondly, coalgebras specify system behaviour in a *one-step* manner by listing the possible futures after one transition step of the system. This "one-step behaviour" is paralleled both on the level of modal (fixpoint) logic and coalgebra. In fact, many of the properties of modal (fixpoint) logic are already manifest at the one-step level, that is, at the level of formulas of modal depth one.

1.2.1 Systems as coalgebra

Informally, a coalgebra consists of a set A of states, or state space, together with a map $A \to \mathsf{T}A$ where T describes the type of transitions and observations that can be made in the coalgebra. In contrast to algebraic operations that are used to *construct* complex objects from simple ones, coalgebraic operations go *out of* the state space and *observe* the behaviour of states.

A simple example of a coalgebra is the function:

$$A \xrightarrow{a \mapsto (a,a)} A \times A$$

where the transition map $a \mapsto (a, a)$ is the diagonal morphism on a space or set A.

Another popular example of a coalgebra is the collection of streams (i.e., infinite words) A^ω over an alphabet set A [Rut00, Jac16]:

$$A^\omega \xrightarrow{(\mathsf{hd},\mathsf{tl})} A \times A^\omega$$

where the "head" function $\mathsf{hd} : A^\omega \to A$ maps a word $w = a.w'$ to its first letter a, and the "tail" function $\mathsf{tl} : A^\omega \to A^\omega$ maps $w = a.w'$ to w'. More in particular, the map $(\mathsf{hd}, \mathsf{tl})$ *de-constructs* elements of A^ω and tells us what is *observable* about a stream w.

Also labeled transition systems can be seen as coalgebras. A labeled transition system is a tuple (A, L, R), where A is a set of states, L is a set of labels and $R \subseteq A \times L \times A$ is a ternary relation. The labeled transition system (A, L, R) can be described as the following coalgebra:

$$A \xrightarrow{a \mapsto \{(l,a') \mid (a,l,a') \in R\}} \mathcal{P}(L \times A)$$

We now give the formal definition of a coalgebra over the category Set of sets and functions. The general definition is given in Definition 2.3.1, but all coalgebras considered in this thesis are coalgebras over Set, so we focus on this particular instance here. We assume the reader is familiar with the basic definitions of category and functor, and refer to Section 2.3 for more details.

Coalgebras Given a functor T on Set, a T-*coalgebra* is a pair $\mathbb{S} = (S, \sigma)$ where S is a *set of states* and $\sigma : S \to \mathsf{T}S$ is a *structure map*. A *coalgebra morphism*, or just morphism, from coalgebra $\mathbb{S} = (S, \sigma)$ to $\mathbb{S}' = (S', \sigma')$, written as $f : \mathbb{S} \to \mathbb{S}'$, is a map $f : S \to S'$ such that $\sigma' \circ f = (\mathsf{T}f) \circ \sigma$.

The structure map of a coalgebra describes the *one-step behaviour* for each state, and coalgebra morphisms are functions that respect these one-step behaviours. The "complete behaviour" or simply "behaviour" of a state can be obtained from one-step behaviour by repeated applications of the structure map. To formalise the notion of behaviour for coalgebras we apply *final coalgebras*, which can be thought of as a domain of complete behaviours. A T-coalgebra $\mathbb{S}'$ is final if for every T-coalgebra $\mathbb{S}$ there exists a unique morphism $\mathsf{beh} : \mathbb{S} \to \mathbb{S}'$. Given a final T-coalgebra, the behaviour of a state s in a T-coalgebra $\mathbb{S}$ can then be defined as $\mathsf{beh}(s)$.

One of the main motivations for studying coalgebras in the context of modal logic is that it creates unity in a landscape of widely varying modal logics. Here we show how Kripke structures (Example 1.1.1) and monotone neighbourhood structures (Example 1.1.3) can be seen as coalgebras. In Section 2.3.2 we give coalgebraic representations of the semantic structures of many other well-known modal logics including multiset frames for graded modal logic (Example 2.3.7) and probabilistic transition systems for probabilistic modal logic (Example 2.3.8).

1.2.1. EXAMPLE. A Kripke frame (W, R) is a Kripke model (W, R, V) without the valuation V (see Example 1.1.1). It is easy to check that Kripke frames can be represented as $\mathcal{P}$-coalgebras for the powerset functor $\mathcal{P} : \mathsf{Set} \to \mathsf{Set}$ which maps a set S to the set of all its subsets $\mathcal{P}S = \{V \mid V \subseteq S\}$. A function $f : S \to T$ is mapped to the direct image map $\mathcal{P}f : \mathcal{P}S \to \mathcal{P}T$, which is defined for any $V \subseteq S$ by $\mathcal{P}f(V) = f[V] = \{f(v) \mid v \in V\}$. A Kripke frame (W, R) can be represented as a $\mathcal{P}$-coalgebra $(W, R[-])$, where $R[-] : W \to \mathcal{P}W$ is defined as follows:

$$R[w] := \{w' \in W \mid w' \text{ is a } R\text{-successor of } w\}.$$

Given a Kripke model (W, R, V), a valuation $V : \mathtt{X} \to \mathcal{P}W$ can also be seen as a map $m_V : W \to \mathcal{P}\mathtt{X}$ by setting $m_V(w) := \{p \in \mathtt{X} \mid p \in V(w)\}$. We call m_V the *marking* associated with valuation V. Then it is easy to see that we can represent Kripke models as structures of the form (W, σ) where $\sigma : W \to \mathcal{P}W \times \mathcal{P}\mathtt{X}$ which are coalgebras for the functor $\mathcal{P} - \times \mathcal{P}\mathtt{X}$.
As it turns out, morphisms for these coalgebras correspond precisely to the so-called p-morphisms (or bounded morphisms) of modal logic [KKV03].

1.2.2. EXAMPLE. In [HK04] it was shown that monotone neighbourhood frames can be characterized as coalgebras for the set functor $\mathcal{M}$ that is defined on sets as:

$$\mathcal{M}(X) = \{Y \subseteq \mathcal{P}(X) \mid Y \text{ is upwards closed}, i.e. \text{ if } U \in Y \text{ and } U \subseteq V \text{ then } V \in Y\},$$

and $\mathcal{M}$ maps $f : S \to T$ to the double-inverse image map $\mathcal{M}f : \mathcal{M}S \to \mathcal{M}T$ defined by $\mathcal{M}f(\alpha) := \{U \subseteq T \mid f^{-1}[U] \in \alpha\}$. Moreover, it was shown that morphisms between $\mathcal{M}$-coalgebras correspond to bounded morphisms for monotone modal logic.

We will return to the functor $\mathcal{M}$ in Chapters 2 and 3.

1.2.2 Equivalence

One of the benefits of having a coalgebraic description of a system is that the theory of coalgebra provides general notions of behavioural equivalence and bisimulation that can be instantiated for concrete system types.

Behavioural equivalence Intuitively, if a functor T admits a final coalgebra, then behavioural equivalence between states of T-coalgebras can be formulated using the map beh: states s and s' are behaviourally equivalent if $\mathsf{beh}(s) = \mathsf{beh}(s')$. However, a final coalgebra is not required. Using that coalgebra morphisms respect behaviour, behavioural equivalence can be defined by identification via cospans of morphisms. States s and s' of coalgebras $\mathbb{S} = (S, \sigma)$ and $\mathbb{S}' = (S', \sigma')$ are behaviourally equivalent if they can be identified in a third coalgebra (Q,γ). That is, if there are coalgebra morphisms $f : (S,\sigma) \to (Q,\gamma)$ and $f' : (S',\sigma') \to (Q,\gamma)$ such that $f(s) = f'(s')$ (see the left part of Figure 1.1).

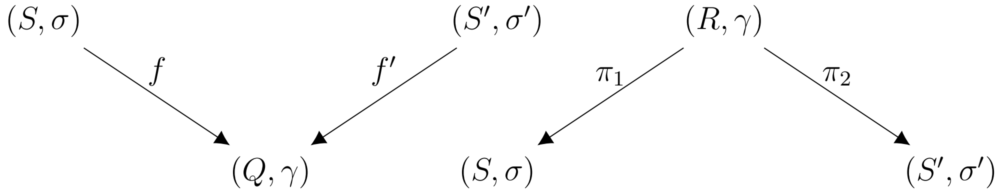


Figure 1.1: Behavioural Equivalence (left) and T-Bisimulation (right)

Bisimilarity Another path to take is to express equivalence by identification via spans of morphisms, which defines the fundamental equivalence notion of *bisimilarity*. Let $\mathbb{S} = (S,\sigma)$ and $\mathbb{S}' = (S',\sigma')$ be two T-coalgebras. A relation $R \subseteq S \times S'$ with projection maps $\pi_1 : R \to S$ and $\pi_2 : R \to S'$ is a T-bisimulation if there is a T-coalgebra structure $\gamma : R \to \mathsf{T}R$, such that π_1 and π_2 are coalgebra morphisms, see the right part of Figure 1.1. Then points $s \in \mathbb{S}$ and $s' \in \mathbb{S}'$, such that sRs' for a T-bisimulation R, are called T-bisimilar [Acz88, AM89]. This notion directly generalises bisimilarity of Kripke models.

Although there are two standard equivalence notions for coalgebras, it is well known that for many functors of interest, including functors representing Kripke frames, multiset frames and labelled transition systems, these two notions coincide. More precisely, if T preserves a certain categorical structure called weak pullback (see Definition 2.3.14), the notions of T-bisimilarity and behavioural equivalence for T coincide. This fails for the monotone neighbourhood functor $\mathcal{M}$ (see Example 2.3.16). In [HK04] Hansen shows that for $\mathcal{M}$ the coalgebraic bisimulation is strictly stronger than the logical notion of bisimulation for monotone neighbourhood models, whereas behavioural equivalence is equivalent with logical monotone bisimilarity. To resolve this situation Marti [Mar11] introduced another coalgebraic notion of bisimulation which generalizes T-bisimilarity and yields an adequate notion of bisimilarity for monotone neighborhood models. We will discuss this in more details in Section 2.3.4.1.

1.3 Coalgebraic logic and automata

We now launch into a coalgebraic generalisation of modal logics by introducing *coalgebraic modal logic* as a common platform to accommodate different modal logics, see [Mos99, KP11]. Next, following [Ven06, CKP09, FLV10] we extend coalgebraic modal logic with fixpoint operators to define *coalgebraic fixpoint modal logic*. Finally, *coalgebraic automata* are presented in order to generalise the link between logic and automata to the level of coalgebras. One of the main observations underlying this link is the *one-step* nature of modal operators and coalgebras; i.e. modal operators (only) specify properties of the (immediate) successor states. Similarly, coalgebras describe the one-step behaviour of systems: the structure map of a coalgebra gives access only to the immediate successor states. We will come back to this property of coalgebras and modal operators at the end of this section.

1.3.1 Coalgebraic modal logic

So far we advertised coalgebraic modal logic as a common platform for studying different modal logics. Yet they can also be viewed as specification languages for coalgebras. Therefore an ideal coalgebraic modal logic tries to be parametric in the functor T, which represents the type of coalgebras, and to keep the balance between uniformity and reflecting our intuition about specific variations of modal logics.

There are two main approaches to coalgebraic modal logic. In the following we describe both approaches without going into the technical details (we refer to 2.4.1 and 2.4.2 for detailed definitions).

1.3.1.1 Coalgebraic modal logic via ∇

In 1996, Barwise and Moss [BM96] introduced a non-standard modality for Kripke models, called the *cover modality* (here denoted by ∇ (" nabla") following [Ven06]) that operates on sets of formulas instead of on formulas. More precisely, for a finite set of formulas Φ, they defined $\nabla\Phi$ as a formula of the logic with the following semantics:

$\nabla\Phi$ holds at a state w of a Kripke model $\mathbb{W}$ iff *every formula* in Φ holds at *some successor* of w, while at the same time, *every successor* of w satisfies *some formula* in Φ.

Considering the standard syntax of basic modal logic, $\nabla\Phi$ is equivalent to the formula:

$$\Box \bigvee_{\varphi \in \Phi} \varphi \wedge \bigwedge \Diamond \Phi,$$

where $\Diamond\Phi = \{\Diamond\varphi \mid \varphi \in \Phi\}$. Conversely, the standard formulas $\Box\varphi$ and $\Diamond\varphi$ are respectively equivalent to $\nabla\emptyset \vee \nabla\{\varphi\}$ and $\nabla\{\top, \varphi\}$.

These equivalences show that the nabla modality and the standard modalities $\Diamond$ and $\Box$ are interdefinable, and hence the language based on ∇ is an alternative formulation of basic modal logic. Independently, Janin and Walukiewicz [JW95] made the same observation which led them to develop automata corresponding to formulas of the μ-calculus.

Following this path to define a coalgebraic modal logic, Moss generalised the result of [BM96] on Kripke models as coalgebras for the powerset functor. In

[Mos99], he initiated the idea of taking the coalgebra type functor itself to construct a modality. Given a set functor T, Moss assigned a modality ∇_{T} to T such that for each $\alpha \in \mathsf{TL}$ where $\mathtt{L}$ is the set of formulas, $\nabla_{\mathsf{T}}\alpha$ is a formula in $\mathtt{L}$. He then defined the semantics of this modality in terms of a *relation lifting* applied to the satisfaction relation (see Definition 2.3.26 and Equation 2.1). Fixing a functor T, we denote the arising modal language by $\mathtt{ML}^{\mathsf{T}}_{\nabla}$, and call it the ∇-based modal language for functor T.

While Moss' ∇-based formalism was recognised as a seminal contribution to the field, it had two main drawbacks. First of all, Moss defined the semantics of his modality via the so-called *Barr* lifting (see Example 2.3.27), and to make it well-behaved he needed to impose *weak-pullback preservation* on the functor (see Definition 2.3.14). So his formalism excluded some important coalgebras such as neighborhood and monotone neighborhood models and frames. And second, the syntax of Moss' modality is rather non-standard compared to $\Box$ and $\Diamond$.

Hence, a natural question to ask is how to obtain a more general coalgebraic modal language which uses standard modalities. The approach that we will discuss in the next section is an answer to this question.

1.3.1.2 Coalgebraic modal logic via predicate liftings

The predicate lifting approach was pioneered by Pattinson [Pat03b], and further developed by Cîrstea, Pattinson, Schröder and others, see e.g. [CP04, Sch08, SP09a]. In this setting, in order to obtain semantics for modalities for a functor T, one needs to choose a set Λ of *predicate liftings* for T (see Definition 2.4.11). Informally, a predicate lifting "lifts" a predicate over the state space S to a predicate over $\mathsf{T}S$. To give some intuition on the semantics of modal operators via predicate liftings, consider a formula φ as a description of a property of states of a Kripke model. Then the modalized formula $\Box\varphi$ asserts a property of one-step behaviours. Unlike the ∇-based language, modal languages using predicate liftings have a standard syntax and since relation liftings are not involved in the semantics, the constraint on weak-pullback preservation drops automatically.

Fixing a functor T and a set of predicate liftings Λ, we denote the arising coalgebraic modal logic by $\mathtt{ML}_{\Lambda}$. This logic is a direct generalisation of the basic modal logic, and indeed many other variations of modal logic arise as instances of the coalgebraic modal logic $\mathtt{ML}_{\Lambda}$ associated with a suitable functor T and modal signature Λ. The formal definition is given in Section 2.4.2 along with several examples including basic modal logic (Example 2.4.14) and monotone modal logic (Example 2.4.15).

One benefit of coalgebraic modal logic is that many results can be proved at the abstract level of coalgebras and these results can then be instantiated to concrete system types. Such results include a coalgebraic Hennessy-Milner theorem [Sch08], which ensures that coalgebraic modal logic is expressive enough to distinguish states of coalgebras that are not behaviourally equivalent, and a coalgebraic generalisation of the van Benthem characterisation theorem [SPL15]. Also axiomatization and completeness can often be proved at the coalgebraic level. We refer the reader to [SP09b] where a coalgebraic strong completeness result is proved using canonical models, and to [Pat03b] and [KKV12] and the references therein for completeness results for coalgebraic modal logic using predicate lifting and ∇ respectively. Finally we note that the balance between expressivity and complexity holds for coalgebraic modal logic as well. In particular, the satisfiability problem for coalgebraic modal logic is generally PSPACE-complete [SP08].

1.3.2 Coalgebraic fixpoint logic

In Section 1.1.2, we presented the standard μ-calculus and mentioned that adding fixpoint operators to modal logic results in a significant increase of the expressive power. Similar to what happens in modal logic, we can extend coalgebraic modal logic with fixpoint operators to obtain a coalgebraic μ-calculus. It is then natural to ask if coalgebraic μ-calculus provides a uniform framework to study different modal fixpoint logics, and whether results from standard μ-calculus can be generalised to the level of coalgebras.

Since we have two approaches towards coalgebraic modal logic, it is not surprising that coalgebraic μ-calculus comes in two variants: one based on the cover modality ∇, and one based on predicate liftings. For the detailed definition of the syntax in both settings, we refer to Definition 2.4.1 and Definition 2.4.12. In this thesis we will work with both approaches.

1.3.3 Coalgebraic automata theory

Given the success of the automata-theoretic approach towards the standard μ-calculus, a suitable uniform automata theory as a pillar of the theory of coalgebraic μ-calculus is desired. In this section we give a brief overview of some of the results regarding the development of such automata theory for coalgebraic μ-calculus. This thesis makes further contributions to this area, and as a main result, presented in Chapter 5, we show how to bring automata explicitly into the proof theory of the μ-calculus.

Coalgebraic automata are designed as abstract devices that operate on pointed coalgebras and either accept or reject them based on a parity condition. They generalise many well-known types of automata in a way that we can instantiate the coalgebra type to these structures and get automata such as word, tree and graph automata.

Automata for the ∇-based fixpoint logic were first introduced by Venema in [Ven06], where he also introduced ∇-based μ-calculus $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$. Focusing on functors that preserve weak pullbacks, Venema proposed three main variants of (alternating and non-deterministic) coalgebraic automata: T-*automata*, *chromatic* T-*automata* and *logical* T-*automata*. Although Venema proved that these three kinds of automata for colored coalgebras are just variants of one another, each can be more convenient for a specific purpose. For instance T-automata can be directly used to prove general and uniform results for automata operating on state-based systems (including relational structures). Chromatic T-automata are more convenient when results such as closure under projection need to be proved (see Chapter 3). And finally logical T-automata are more appropriate in case a syntactical connection to logical formulas is desired (e.g. Chapters 4 and 5).

A key result proved in [Ven06] states that there are effective translations between logical T-automata and $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$-formulas, and that logical T-automata and $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$-formulas are expressively equivalent over pointed coalgebras. The theory of T-automata was then further developed in [KV08], where several results including closure properties and a simulation theorem were established for such automata. Subsequently some of these results have been generalised from weak pull back-preserving functors to the class of functors with a quasi-functorial lax extension in [MSV15].

Coalgebraic automata for $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$ do not correspond directly to the coalgebraic μ-calculus with predicate liftings $\mu\mathtt{ML}_{\Lambda}$. In addition, all the variants of coalgebraic automata that have been so far mentioned are restricted to functors that preserve weak pullbacks. Therefore Fontaine et al. in [FLV10] introduced automata for coalgebras of arbitrary type. More precisely, for a set functor T and a set Λ of predicate liftings they defined a logical Λ-automaton which is an alternating parity automaton operating on T-coalgebras. In the same paper, Fontaine et al. also provided effective translations between Λ-automata and $\mu\mathtt{ML}_{\Lambda}$-formulas, and proved expressive equivalence of Λ-automata and $\mu\mathtt{ML}_{\Lambda}$-formulas over T-coalgebras.

1.3.3.1 One-step logic

One-step logic stems from the work on coalgebraic modal logic by Cîrstea, Pattinson, Schröder and others [Pat03b, CP04, CKP+08, SP09a, SP10] to provide the syntax and semantics for one-step formulas, that are formulas of modal depth one. This logic plays a pivotal role in strengthening the link between fixpoint logic, automata and coalgebras. Its importance is due to the observation that many of the properties of (coalgebraic) modal fixpoint logic are already manifest at the one-step level, that is, at the level of one-step formulas and one-step unfoldings of coalgebra states [FLV10, ESV16a, SSP17, ESV17].

By defining the transition map of logical automata with one-step formulas, and applying properties of one-step logic, we are able to separate two key aspects of the coalgebraic μ-calculus: the *one-step dynamics* encoded by modal operators and one-step unfolding of coalgebras, and the *combinatorics* related to nested fixpoints. In particular, we will see that the "trace theory" of an automaton is largely determined by the shape of the formulas of the one-step language. For example one of our key concepts, that of a *semi-disjunctive* automaton (see Definition 5.6.2), which is related to the notion of aconjunctive formulas in [Koz83], is defined in terms of a syntactic form of the one-step formulas, but is motivated by certain results about the acceptance game and the structure of traces for such automata.

The one-step perspective lets us go beyond expressive equivalence between formulas and automata, by establishing a syntactic link between automata and formulas, which puts automata to use in proof theory. This is one of our main contributions to the field and will be further discussed in Sections 1.4.3 and 5.8.

1.4 Contributions

In this thesis, we study some important properties of coalgebraic fixpoint logic, namely uniform interpolation, expressive completeness and axiomatic completeness. We devote a separate chapter to each of these properties and in this section we give an overview of our contribution in each chapter.

1.4.1 Chapter 3: Uniform Interpolation

In Chapter 3, which is based on [MSV15], we study *uniform interpolation* for the ∇-based coalgebraic fixpoint logic. Uniform interpolation is a stronger version of Craig Interpolation [Cra57]; the latter says that if a formula φ_1 implies a formula φ_2, then there exists a formula ψ, called an *interpolant*, such that every nonlogical symbol in ψ occurs both in φ_1 and φ_2, and φ_1 implies ψ and ψ implies φ_2. A logic then has uniform interpolation if the interpolant ψ for the formulas φ_1 and φ_2 depends only on φ_1 and the language that φ_2 shares with φ_1 (rather than on φ_2 itself). Although it is easy to show that classical propositional logic has uniform interpolation, not many logics have this property. For instance first-order logic does not enjoy uniform interpolation, even though it has Craig interpolation [Hen63, Pit92].

Generally, there are several motivations to consider interpolation properties in logics. As described in [D'A08] one may study interpolation for reasons that are *internal* to logic. For instance, in classical logic Craig interpolation can be used to prove definability and preservation results such as Beth and Lyndon theorems [Bet56] [Lyn59]. *External* motivations to study interpolation come from applications of this concept in computer science, specifically in software design and model checking [McM18].

In the theory of modal fixpoint logic, as we already mentioned in Section 1.1.3, D'Agostino and Hollenberg proved that the modal μ-calculus has uniform interpolation [DH00]. The main contribution of Chapter 3 is to take a coalgebraic perspective and generalise this result to a wider class of fixpoint logics. Our work mainly builds on results from [SV10] and [Mar11].

In [SV10], Santocanale and Venema took a coalgebraic perspective on monotone modal logic and reconstructed the syntax of this logic by replacing the box and diamond with a suitable ∇-modality. They defined the semantics of ∇ via a relation lifting $\widetilde{\mathcal{M}}$, which is different from the standard Barr lifting (see Example 2.3.27) and is appropriate for $\mathcal{M}$ which does not preserve weak pullbacks. Using this relation lifting they proved that monotone modal logic enjoys uniform interpolation. Generalizing the work in [SV10] for the monotone neighborhood functor, Marti [Mar11] proved that the ∇-based coalgebraic modal logic for T-coalgebras has uniform interpolation if T has a relation lifting (lax extension, see Definition 3.1.1) satisfying a certain property called *quasi-functoriality*, see Definition 3.1.5 and Example 3.1.7. The class of functors with a quasi-functorial lax extension includes all functors that preserve weak pullbacks (such as the Kripke functor), but also, the monotone neighbourhood functor $\mathcal{M}$.

Main result In Chapter 3, we prove that if a set functor T admits a quasi-functorial lax extension L, then the ∇-based coalgebraic fixpoint logic $\mu\mathtt{ML}^{L}_{\nabla}$ has the uniform interpolation property (Theorem 3.7.1). Using that the ∇-based *monotone μ-calculus* is expressively equivalent with the standard monotone μ-calculus, we also obtain uniform interpolation for the latter.

Generally, there are two paths to follow in order to prove a uniform interpolation theorem: A proof-theoretic approach and a semantic approach. A good example of the proof-theoretic approach is [Bíl07], which adopts Pitts' proof of uniform interpolation in intuitionistic propositional logic [Pit92]. The semantic approach is based on providing definability of bisimulation quantifiers, which implies the uniform interpolation property [Vis96a]. In Chapter 3 we follow the semantic approach and take an automata-theoretic perspective similar to [DH00] in order to show that bisimulation quantifiers are definable in $\mu\mathtt{ML}^{L}_{\nabla}$.

Proof strategy First, we generalise the class of alternating coalgebraic automata defined in [Ven06] and [KV08] for weak pullback-preserving set functors, to functors admitting a quasi-functorial lax extension. We then prove a simulation theorem (Theorem 3.4.5) stating that every alternating automaton can be replaced with an equivalent non-deterministic one. To be able to define inductive translations between formulas of $\mu\mathtt{ML}^{L}_{\nabla}$ and coalgebraic non-deterministic automata (Proposition 3.6.1), we prove closure properties of automata for boolean operators. Finally, as our main technical result, we prove that the class of automata we defined is closed under projection (Theorem 3.5.3). This result, which generalises Proposition 5.9 in [KV08], is the automata-theoretic counterpart of definability of bisimulation quantifiers, and our proof strategy for that is the same as in [KV08], but the construction here is more involved. We then apply this result and the translations between automata and formulas to show that bisimulation quantifiers are definable in our language (Proposition 3.7.2). Once we have bisimulation quantifiers in hand, it is routine to construct a uniform interpolant for any given pair of formulas (Theorem 3.7.1).

1.4.2 Chapter 4: Expressive Completeness

As we mentioned in Section 1.1.3, the Janin-Walukiewicz theorem can be seen as an expressive completeness result, stating that all *relevant* properties of Kripke models in monadic second-order logic can be expressed in the modal μ-calculus. More precisely, a monadic second-order formula is equivalent to a formula of μ-calculus iff it is invariant under bisimulation. In Chapter 4, which is based on [ESV15] and [ESV17], we address the question whether the Janin-Walukiewicz theorem can be generalized from Kripke structures to the setting of arbitrary coalgebras. The aim is to prove a characterisation theorem for the coalgebraic μ-calculus with predicate liftings as modalities. In order to formulate the theorem, we first employ a set Λ of predicate liftings for a set functor T to introduce a coalgebraic monadic second-order logic $\mathtt{MSO}_\Lambda$ interpreted over T-coalgebras, and we extend the link between logic, coalgebra and automata by defining automata for $\mathtt{MSO}_\Lambda$. We then take an automata-theoretic approach and prove our main result.

Main result In Chapter 4 we prove that if Λ consists of all monotone predicate liftings for T and T admits a so-called *adequate uniform construction* (see Definitions 4.5.3 and 4.5.8), then $\mu\mathtt{ML}_\Lambda$ is the fragment of $\mathtt{MSO}_\Lambda$ that is invariant under behavioural equivalence (Theorem 4.5.9). This theorem can be instantiated to obtain concrete results for particular logics. Examples include taking T to be the power set functor (standard Kripke structures), where the adequate uniform construction roughly consists of taking ω-fold products (see Example 4.5.32), the bag functor (Example 4.5.33), and all polynomial functors (Proposition 4.5.35).

However, it turns out that there are functors that provably do *not* admit an adequate uniform construction. A concrete example is the monotone neighborhood functor $\mathcal{M}$. As the final contribution of Chapter 4 we show how, with some extra work, a characterization result (Theorem 4.6.11) for the monotone μ-calculus can be derived using our main result (Theorem 4.5.9) with respect to a natural monadic second-order language for monotone neighborhood structures that we call "monotone $\mathtt{MSO}$" and denote by $\mathtt{MMSO}$.

What follows next is a sketch of our proof strategy.

Proof strategy We first define the coalgebraic monadic second-order language $\mathtt{MSO}_\Lambda$ for any set Λ of monotone predicate liftings (Definition 4.2.3), and let $\mathtt{MSO}_\mathsf{T}$ and $\mu\mathtt{ML}_\mathsf{T}$ denote the logics obtained by taking for Λ the set of *all* monotone predicate liftings T. Our main goal is then to answer the following question: For which T does the coalgebraic μ-calculus $\mu\mathtt{ML}_\mathsf{T}$ correspond to the fragment of $\mathtt{MSO}_\mathsf{T}$ that is invariant for behavioural equivalence. In other words, for which T do we have $\mu\mathtt{ML}_\mathsf{T} \equiv \mathtt{MSO}_\mathsf{T}/{\sim}$? To answer this question we take an automata-theoretic approach. Automata corresponding to $\mu\mathtt{ML}_\mathsf{T}$ already exist [FLV10], and are defined using the one-step modal language $\mathtt{ML}^1_\mathsf{T}$. Hence, we only need to define a class of coalgebraic parity automata corresponding to formulas in $\mathtt{MSO}_\mathsf{T}$. These automata are based on the one-step *second-order logic* $\mathtt{SO}^1_\mathsf{T}$ and are equivalent to $\mathtt{MSO}_\mathsf{T}$-formulas over coalgebras that are 'tree-like' in some sense (see Definition 4.3.8), similar to the case of standard MSO (Theorem 4.3.24). Denoting the class of automata for $\mathtt{MSO}_\mathsf{T}$ by $Aut(\mathtt{SO}^1_\mathsf{T})$, we can reformulate the earlier question on the level of automata instead of formulas: For which functors T do we have $Aut(\mathtt{ML}^1_\mathsf{T}) \equiv Aut(\mathtt{SO}^1_\mathsf{T})/\sim$? We approach this question *at the level of the one-step languages* $\mathtt{SO}^1_\mathsf{T}$ and $\mathtt{ML}^1_\mathsf{T}$, and identify that this is the case if T admits an *adequate uniform construction* (see Definitions 4.5.3 and 4.5.8). On the basis of this observation the proof of our generalisation of the Janin-Walukiewicz theorem (Theorem 4.5.9) is straightforward. Finally, we prove our second main characterization theorem (Theorem 4.5.13) for functors that do not admit an adequate uniform construction, but preserve finite sets and have a *weakly adequate uniform construction* (Definition 4.5.12), and as an application of this theorem we get a characterization result for the monotone μ-calculus with respect to monotone MSO (Theorem 4.6.11).

1.4.3 Chapter 5: Axiomatic Completeness

One of the main questions asked about any logic is whether it has a sound and complete axiomatization. In the seminal paper introducing the μ-calculus [Koz83], Kozen also proposed an axiom system for the logic, and proved a partial completeness result for a fragment of the μ-calculus, which he called the *aconjunctive* fragment. The completeness question of Kozen's axiomatization for the full language remained open for about a decade, until it was resolved by Walukiewicz [Wal00]. While the Walukiewicz completeness theorem is often cited and generally recognized as a landmark in the theory of the modal μ-calculus, it has remained something of an isolated point in the completeness theory of modal (fixpoint) logic. This is largely due to the complexity of Walukiewicz' proof. One source of this complexity lies in the general combinatorial issues involved in dealing with infinite *traces*, i.e., possible histories of formulas, recording unfoldings of fixpoint variables - see also the work of Niwinski and Walukiewicz [NW96] on tableau games for the modal μ-calculus. The other source of difficulty in Walukiewicz' proof seems to be an intricate mix of ideas from automata theory, game theory and logic (such as tableaux). It is the aim of Chapter 5, which is based on [ESV16b] and [ESV18], to clarify and generalise Walukiewicz' proof by applying ideas from automata theory and coalgebra. In particular, we set up a *framework* for dealing with the completeness problem, where we bring automata into the picture at an earlier stage and put traces and their combinatorics in the foreground.

Taking Walukiewicz' proof as starting point, and noting that the modality he worked with in his proof is in fact the nabla modality for the powerset functor, we first addressed the completeness question for the ∇-based coalgebraic fixpoint logic $\mu\mathtt{ML}_{\nabla}^{\overline{\mathsf{T}}}$ in [ESV16b, ESV18]. We later generalised this result to the predicate lifting setting in a separate paper [ESV19], which will be discussed in the follow up works section.

In order to handle the fixpoint operators of the ∇-based coalgebraic fixpoint logic $\mu\mathtt{ML}_{\nabla}$, we add Kozen's axiom and rule to the complete axiomatization of ∇-based coalgebraic modal logic $\mathtt{ML}_{\nabla}^{\overline{\mathsf{T}}}$ from [KKV12], and define an axiomatization $\mathbf{K}$ for $\mu\mathtt{ML}_{\nabla}^{\overline{\mathsf{T}}}$. Our main result is the following completeness theorem.

Main result In Chapter 5, we prove that if a set functor T preserves weak pullbacks and finite sets then the axiom system $\mathbf{K}$ for $\mu\mathtt{ML}_{\nabla}^{\overline{\mathsf{T}}}$ is sound and complete (Theorem 5.9.4).

To prove our main theorem, we follow the approach taken by Kozen and Walukiewicz, but with some differences. Before discussing the differences, let us first briefly review the main steps of Walukiewicz' proof which builds on Kozen's result for aconjunctive formulas. Walukiewicz' proof starts with the observation that the satisfaction problem is easy for so-called *disjunctive* formulas which correspond to *disjunctive* automata, i.e. non-deterministic parity automata operating on Kripke models. He then proves that every μ-calculus formula φ is semantically equivalent to a disjunctive formula φ', such that the implication $\varphi \to \varphi'$ is provable in Kozen's system. The first difference between our approach and Walukiewicz' proof strategy is that we work with a wider class of automata than disjunctive ones, and we formulate a precise connection between the proof theory and automata. This enables us to make automata the main building block of our proof strategy and reformulate Kozen's and Walukiewicz' arguments in an entirely automata-theoretic framework. As a second difference, we set up a framework where we may clearly distinguish *dynamics* (coalgebra) from *combinatorics* (trace management). And third, our approach is thoroughly game-theoretic in nature.

Proof strategy We start by defining coalgebraic modal automata using a one-step ∇-based coalgebraic modal language (Definition 5.3.17) and provide a translation which associates to every $\mu\mathtt{ML}_{\nabla}^{\overline{\mathsf{T}}}$-formula φ a semantically equivalent modal automaton $\mathbb{A}_{\varphi}$. We then provide a translation $\mathtt{tr}$ in the opposite direction, that is, from automata to formulas, and establish that for every formula $\varphi \in \mu\mathtt{ML}_{\nabla}^{\overline{\mathsf{T}}}$, we have φ is provably equivalent to $\mathtt{tr}(\mathbb{A}_{\varphi})$ (Proposition 5.8.1). Next we focus on two automata-related games: First, the *satisfiability game* (Definition 5.4.5) of [FLV10], which comprises the logical notion of tableau. Second, we introduce the *consequence game* (Definition 5.4.10) between two automata. This game resembles Walukiewicz' consequence game between tableaux, and can be considered of independent interest. We then isolate classes of automata which allow a relatively simple trace management in the satisfiability and consequence games. The first automata that naturally appear are the *disjunctive automata*. These automata admit a *trivial* trace theory, in the sense that the matches of the satisfiability game of a disjunctive automaton involves a single trace only. The second class of special automata consists of *semi-disjunctive* automata. These automata roughly correspond to the *weakly aconjunctive formulas* introduced by Walukiewicz. They are much less constrained than disjunctive automata, but their one-step formulas are still of a shape that guarantees the trace theory of an infinite match of the satisfiability and consequence games to be well-behaved. As a key step of our proof strategy we then establish the following coalgebraic generalisation of one of Walukiewicz' main lemmas (Theorem 5.1.2): For every formula $\varphi \in \mu\mathtt{ML}_{\nabla}^{\overline{\mathsf{T}}}$, there is a semantically equivalent disjunctive automaton $\mathbb{D}$ such that $\vdash_{\mathbf{K}} \varphi \to \mathtt{tr}(\mathbb{D})$. The completeness theorem (Theorem 5.9.4) is then a straightforward corollary of Theorem 5.1.2: If φ is an arbitrary consistent formula, then by Theorem 5.1.2 it is semantically equivalent to a consistent disjunctive automaton $\mathbb{D}$. But for disjunctive automata it is easy to prove that consistency implies satisfiability (applying Lemma 5.9.1 and Proposition 5.6.4), and so we are done.

Chapter 2

Preliminaries

In this chapter, we present a technical introduction to the material that will be covered throughout this thesis. Not surprisingly, most of this chapter concerns coalgebras and fixpoint logic. We refer the reader to [AN01] and [BS01] for fixpoint logic, and to [Rut00], [Jac16] and [Rut19] for detailed surveys on the theory of coalgebras. For introductions to coalgebraic logic one may consult [Pat03a] and [KP11].

2.1 Modal logic and fixpoints

We discussed the connection between modal logics, fixpoints and automata in Section 1.1. The goal of this section is to provide formal definitions of concepts that will be needed in this thesis.

2.1.1 Modal logics

We begin by introducing modal logics, and recall that these are logics obtained by extending classical propositional logic with operators called modalities, evaluated over transition systems. Here we cover some examples of modal logics and refer the reader to [BdRV02] for detailed definitions of syntax and semantics.

Examples

2.1.1. Example. We first recall *Basic modal logic* from Example 1.1.1. This logic, which we denote by $\mathtt{ML}$, is obtained by adding modal operators $\Diamond$ and $\Box$ to classical propositional logic [BdRV02]. Formulas of this logic are evaluated over Kripke models which are Kripke frames equipped with a valuation. More precisely, Kripke models are triples (W, R, V) with W as the set of states (worlds), the binary relation $R \subseteq W \times W$ as a successor (accessibility) relation, and a valuation $V : \mathtt{X} \to \mathcal{P}W$ where $\mathtt{X}$ is the set of proposition letters of a given modal logic. The semantics of modal formulas $\Diamond\varphi$ and $\Box\varphi$ over a Kripke model (W, R, V) is defined as follows:

$$w \Vdash_V \Diamond\varphi \quad \text{iff} \quad v \Vdash_V \varphi \text{ for some } v \in W \text{ such that } wRv,$$
$$w \Vdash_V \Box\varphi \quad \text{iff} \quad v \Vdash_V \varphi \text{ for all } v \in W \text{ such that } wRv.$$

This semantics gives the reading of $\Diamond\varphi$ as "possibly φ" as φ holds in at least one successor of the current state of the model. Similarly, $\Box\varphi$ reads as "necessarily φ" as φ holds in all successors to the current state.

2.1.2. Example. As is clear from the semantics of $\Diamond$ and $\Box$ in basic modal logic, these modal operators just check the existence of successors with certain properties. To generalize these modalities, one may decide to *count* such successors of a current state in a Kripke model. This way, Fine [Fin72] defined *graded* modalities $\Diamond^k$ and $\Box^k$ for *graded modal logic* [Fin72], with the following semantics over a Kripke model (W, R, V):

$$w \Vdash_V \Diamond^k\varphi \quad \text{iff} \quad |\{v \in W \mid wRv \text{ and } v \Vdash_V \varphi\}| \geq k,$$
$$w \Vdash_V \Box^k\varphi \quad \text{iff} \quad |\{v \in W \mid wRv \text{ and } v \nVdash_V \varphi\}| < k.$$

This reads as $w \Vdash_V \Diamond^k\varphi$ iff "φ holds in at least k successors of w", and $w \Vdash_V \Box^k\varphi$ iff "φ fails in at most (!) $k-1$ successors of w". It is clear that we obtain $\Diamond$ and $\Box$ as the special cases $\Diamond = \Diamond^1$ and $\Box = \Box^1$.

2.1.3. Example. The next idea to generalize $\Diamond$ and $\Box$ is to use rational numbers; e.g. *probabilities*, instead of natural numbers for graded modalities. The modalities of *probabilistic modal logic* [HM98] are written as $\Diamond_p$ and $\Box_p$ for a rational $p \in [0, 1]$. These modalities are interpreted over probabilistic transition systems (also called Markov chains). A probabilistic transition system is a pair (W, P) where W is a set of states and $P = (\mu_w)_{w \in W}$ is a family of probability distributions on W. The semantics of the modalities over a probabilistic model $\mathbb{W} = (W, P, V)$, i.e., a probabilistic transition system equipped with a valuation V is defined as follows:

$$w \Vdash_V \Diamond_p\varphi \quad \text{iff} \quad \mu_w([\![\varphi]\!]_{\mathbb{W}}) \geq p,$$
$$w \Vdash_V \Box_p\varphi \quad \text{iff} \quad \mu_w([\![\neg\varphi]\!]_{\mathbb{W}}) < p,$$

where $[\![\varphi]\!]_{\mathbb{W}} = \{v \in W \mid v \Vdash_V \varphi\}$ is the extension of φ over model $\mathbb{W}$. The reading of $\Diamond_p\varphi$ and $\Box_p\varphi$ is then respectively "φ holds with probability at least p in the next state" and "φ fails with probability less than p in the next state" .

2.1.4. Example. The Scott-Montague *neighbourhood semantics* (cf. [Sco70], [Mon70]) of modal logic generalizes Kripke semantics. The modality of neighbourhood modal logic is denoted by $\Box$, and is interpreted over neighbourhood structures. A neighbourhood frame is a pair $\mathbb{W} = (W, N)$ where W is a set of states and $N : W \to 2^{2^W}$ a neighbourhood function, which assigns to each state of W a collection of subsets of W that is called its neighbourhoods. Considering that the notion of neighbourhood here is very liberal - it is not even required that a point lies in all its neighbourhoods- it can be shown that neighbourhood models generalise Kripke models [Che80]. The semantics of $\Box$ over a neighbourhood model $\mathbb{W} = (W, N, V)$, i.e. a neighbourhood frame with a valuation, is defined as follows:

$$w \Vdash_V \Box\varphi \quad \text{iff} \quad [\![\varphi]\!]_{\mathbb{W}} \in N(w),$$

in words: w satisfies $\Box\varphi$ iff "the extension of φ is a neighbourhood of w".

2.1.5. Example. It is common to require a bit more structure on neighbourhood frames from Example 2.1.4, and define the *monotone neighbourhood* semantics [Che80, Han03]. A monotone neighbourhood model $\mathbb{W} = (W, N, V)$ is a neighbourhood model with the restriction that all neighbourhood collections are upsets (that is, if $U \in N(w)$ and $U \subseteq U'$, then $U' \in N(w)$). The semantics of $\Box\varphi$ is the same as over neighbourhood models:

$$w \Vdash_V \Box\varphi \quad \text{iff} \quad [\![\varphi]\!]_{\mathbb{W}} \in N(w).$$

2.1.6. Example. *Coalition logic* [Pau02] is a modal logic for reasoning about the coalitional power of agents in a strategic game. Given a finite set N of agents or players, subsets of N are called *coalitions*. In coalition logic for each coalition A we have a modal operator $[A]$, where $[A]\varphi$ informally reads as "coalition A has a joint strategy to ensure an outcome that satisfies φ". Formulas of coalition logic are interpreted over game models. A game model $\mathbb{W} = (W, G, V)$ consists of a set W of states, a valuation V, and a map G that at a state $w \in W$ assigns:

- a set S_i^w of available *moves* to each agent $i \in N$
- an *outcome function*

  $$f_w : (\textstyle\prod_{i \in N} S_i^w) \to W$$

  that determines the successor state of w, given a choice of a move by each agent.

For a coalition A and joint choices $\rho_A \in \prod_{i \in A} S_i^w$ and $\rho_{N-A} \in \prod_{i \in N-A} S_i^w$ of moves by the agents in A and the agents outside A, respectively, we let $\langle \rho_A, \rho_{N-A} \rangle$ represent the obvious induced element of $\prod_{i \in N} S_i^w$, i.e. the arising overall choice of moves. The semantics of the operators $[A]$ is then determined, for a state w with data S_i^w, f_w as above, by

$$w \Vdash_V [A]\varphi \;\text{ iff }\; \exists \rho_A \in \textstyle\prod_{i \in A} S_i^w . \forall \rho_{N-A} \in \prod_{i \in N-A} S_i^w . \, f_w\langle \rho_A, \rho_{N-A} \rangle \Vdash \varphi.$$

2.1.2 μ-calculus over Kripke models

Although we assume that the reader is familiar with the syntax and semantics of the μ-calculus, here we provide a quick review of the main notions that play a role in this thesis. We present the μ-calculus in two different syntactic formats. The first format, denoted by $\mu\mathtt{ML}$, is the standard μ-calculus which is obtained by adding fixpoint operators to the basic modal logic $\mathtt{ML}$. The second formulation is the ∇-based modal fixpoint logic denoted by $\mu\mathtt{ML}^{\mathcal{P}}_{\nabla}$, which we obtain by replacing the standard modalities of $\mu\mathtt{ML}$ by the modality "nabla" (∇) for the powerset functor.

To simplify the notation, we will focus on *monomodal logic* with only one modality. More detail on the general formalism can be found in [Koz83], [Wal00] and [AN01].

Standard μ-calculus

Throughout this section, we fix an infinite set of propositional variables.

2.1.7. Definition. The language $\mu\mathtt{ML}$ of the modal μ-calculus is given by the following grammar:

$$\varphi ::= \top \mid \bot \mid p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \Box\varphi \mid \neg\varphi \mid \mu x.\varphi \mid \nu x.\varphi,$$

where p and x are propositional variables, and the formation of the formulas $\mu x.\varphi$ and $\nu x.\varphi$ is subject to the constraint that the variable x is *positive* in φ, i.e., all occurrences of x in φ are in the scope of an even number of negations. Elements of $\mu\mathtt{ML}$ will be called *modal fixpoint formulas*, *μ-formulas*, or simply *formulas*.

The collection of *subformulas* of a formula is defined as usual. Syntactically, the fixpoint operators are similar to the quantifiers of first-order logic in the way they *bind* variables. They bind the variable that they occur with everywhere in the subformula to which they are applied. Given a formula φ, the free and bound variables are defined in the usual way. Fix a formula φ. The sets $FV(\varphi)$ and $BV(\varphi)$ of its *free* and *bound* variables are defined in the usual way. As a convention, the free variables of a formula φ are denoted by the symbols $p, q, r, \ldots$, and referred to as *proposition letters*, while we use the symbols $x, y, z, \ldots$ for the bound variables of a formula.

Semantics of $\mu\mathtt{ML}$

Formulas of $\mu\mathtt{ML}(\mathtt{X})$ are interpreted over Kripke models. A Kripke model $\mathbb{S} = (S, R, V)$ is a Kripke frame equipped with a valuation $V : \mathtt{X} \to \mathcal{P}S$.

2.1.8. Definition. By induction on the complexity of modal fixpoint formulas, we define a meaning function $[\![\cdot]\!]$, which assigns to a formula $\varphi \in \mu\mathtt{ML}$ its *extension*

$[\![\varphi]\!]^{\mathbb{S}} \subseteq S$ in any Kripke model $\mathbb{S} = (S, R, V)$. The clauses of this definition are standard:

$$\begin{array}{rcl} [\![\top]\!]^{\mathbb{S}} & := & S \\ [\![\bot]\!]^{\mathbb{S}} & := & \emptyset \\ [\![p]\!]^{\mathbb{S}} & := & V(p) \\ [\![\varphi \vee \psi]\!]^{\mathbb{S}} & := & [\![\varphi]\!]^{\mathbb{S}} \cup [\![\psi]\!]^{\mathbb{S}} \\ [\![\varphi \wedge \psi]\!]^{\mathbb{S}} & := & [\![\varphi]\!]^{\mathbb{S}} \cap [\![\psi]\!]^{\mathbb{S}} \end{array}$$

$$\begin{array}{rcl} [\![\Diamond\varphi]\!]^{\mathbb{S}} & := & \{s \in S \mid R[s] \cap [\![\varphi]\!]^{\mathbb{S}} \neq \varnothing\} \\ [\![\Box\varphi]\!]^{\mathbb{S}} & := & \{s \in S \mid R[s] \subseteq [\![\varphi]\!]^{\mathbb{S}}\} \\ [\![\neg\varphi]\!]^{\mathbb{S}} & := & S \setminus [\![\varphi]\!]^{\mathbb{S}} \\ [\![\mu x.\varphi]\!]^{\mathbb{S}} & := & \bigcap\{U \in \mathcal{P}S \mid [\![\varphi]\!]^{\mathbb{S}[x \mapsto U]} \subseteq U\} \\ [\![\nu x.\varphi]\!]^{\mathbb{S}} & := & \bigcup\{U \in \mathcal{P}S \mid U \subseteq [\![\varphi]\!]^{\mathbb{S}[x \mapsto U]}\}, \end{array}$$

where $\mathbb{S}[x \mapsto U]$ denotes the model $(S, R, V[x \mapsto U])$ obtained from $\mathbb{S}$ by replacing V with the valuation $V[x \mapsto U]$, which is exactly like V apart from mapping x to U.

Given a model $\mathbb{S} = (S, R, V)$ we also define a relation $\Vdash$ between S and $\mu\mathtt{ML}$ by saying that (s, φ) belongs to the relation $\Vdash$ if $s \in [\![\varphi]\!]^{\mathbb{S}}$. In the case that (s, φ) belongs to $\Vdash$ we say that φ is *true at* s or *holds* at s, or that s *satisfies* φ.

Representing standard μ-calculus using the ∇-modality

As we discussed in Section 1.3.1.1, Barwise and Moss [BM96] introduced a non-standard modality ∇ for Kripke models that operates on sets of formulas instead of on formulas. Using this modal operator we can define the ∇-based fixpoint language $\mu\mathtt{ML}_{\nabla}^{\mathcal{P}}$.

2.1.9. DEFINITION. The language $\mu\mathtt{ML}_{\nabla}^{\mathcal{P}}$ is given by the following grammar:

$$\varphi ::= \top \mid \bot \mid p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \nabla\Phi \mid \neg\varphi \mid \mu x.\varphi \mid \nu x.\varphi,$$

where p and x are propositional variables, $\Phi \subseteq_{\omega} \mu\mathtt{ML}_{\nabla}^{\mathcal{P}}$, and similar to the definition of $\mu\mathtt{ML}$, the formation of the formulas $\mu x.\varphi$ and $\nu x.\varphi$ is subject to the constraint that the variable x is *positive* in φ, i.e., all occurrences of x in φ are in the scope of an even number of negations.

To formulate the semantics of the ∇-modality, we use the standard definition from the literature (see [Mos99]), which applies the Egli-Milner lifting (see Example 2.3.28) of the satisfaction relation $\Vdash$:

$$\mathbb{S}, s \Vdash \nabla\Phi \text{ iff } R[s]\ \overline{\mathcal{P}}(\Vdash)\ \Phi.$$

Working out the details, this says that $\nabla\Phi$ holds at s iff every successor of s satisfies some formula in Φ and every formula in Φ holds in some successor of s. From this observation it is easy to derive that:

$$\nabla\Phi \equiv \bigwedge \Diamond\Phi \wedge \Box \bigvee \Phi,$$

where $\Diamond\Phi$ denotes the set $\{\Diamond\varphi \mid \varphi \in \Phi\}$. Conversely, the standard modal operators can be expressed in terms of the ∇-modality:

$$\begin{array}{lcl} \Diamond\varphi & \equiv & \nabla\{\varphi, \top\} \\ \Box\varphi & \equiv & \nabla\{\varphi\} \vee \nabla\varnothing. \end{array}$$

We note that $\nabla\varnothing$ holds at a point s iff s is a 'blind' world, that is, $R[s] = \varnothing$. Since ∇ and the standard modalities $\Diamond$ and $\Box$ are interdefinable, the languages $\mu\mathtt{ML}$ and $\mu\mathtt{ML}^{\mathcal{P}}_{\nabla}$ are expressively equivalent.

2.2 Parity games

Parity games [EJ91b, Mos91] are special graph games with a *parity winning condition*. These games are closely connected to the theory of fixpoint logic, and are used to define a game semantics for the standard μ-calculus and determine the acceptance condition for automata corresponding to this logic.

All the games that we consider in this thesis are graph games involving two players, standardly called *Éloise* ($\exists$) and *Abelard* ($\forall$).

Given a set A, let A^ω denote the collection of infinite words (streams) over A, respectively. For $\rho \in A^\omega$ we let:

$$\mathrm{Inf}(\rho) := \{a \in A \mid a \text{ occurs infinitely often in } \rho\}.$$

A *parity graph game* is a tuple $\mathcal{G} = (B_\exists, B_\forall, E, \Omega)$ where:

- $B_\exists, B_\forall$ are disjoint sets of positions for the players $\exists$ and $\forall$ respectively. The set $G := B_\exists \cup B_\forall$ is called the *board* of the game $\mathcal{G}$.
- $E \subseteq G \times G$ is a binary relation on G that defines *admissible moves*.
- $\Omega : G \to \omega$ is a *priority function*, i.e. a function from G to ω with finite range.

A *play* or *match* consists of an initial position $b_I \in G$ and a sequence of moves of the players according to the following rule:

- In position $b \in B_\exists$ ($b \in B_\forall$) player $\exists$ ($\forall$) has to move to some position $b' \in E[b]$, where $E[b] := \{b' \in G \mid (b, b') \in E\}$.

Therefore a (possibly infinite) match of $\mathcal{G}$ starting from initial position b_I is represented by a sequence of positions $b_0 b_1 b_2 \cdots$ where $b_0 = b_I$ and $b_{i+1} \in E[b_i]$. A match $\rho = b_0 b_1 \cdots$ from some position $b_0 = b_I$ is said to be *complete* if either $\rho = b_0 b_1 \cdots b_n$ is finite and $E[b_n] = \emptyset$ or $\rho = b_0 b_1 \cdots$ is infinite. We call an incomplete match $\rho = b_0 b_1 \cdots b_n$ a *partial* match.

Given a complete match ρ of $\mathcal{G}$, we say $\exists$ wins ρ if either

- $\rho = b_0 b_1 \cdots b_n$ and $b_n \in B_\forall$ or
- ρ is infinite and $max\{\Omega(b) \mid b \in \mathrm{Inf}(\rho)\}$ is even.

Otherwise $\forall$ wins.

An important property of parity graph games is their history-free determinacy, i.e. the fact that starting from any position of G either of the players has a history-free winning strategy [EJ91b, Mos91]. We will now formally define the notion of such a winning strategy and then state the theorem.

2.2.1. DEFINITION. Let $\mathcal{G} = (B_\exists, B_\forall, E, \Omega)$ be a parity graph game, $G = B_\exists \cup B_\forall$ the set of positions. A *strategy* for a player Π in $\{\exists, \forall\}$ is a function F mapping a partial match $b_0 b_1 \cdots b_n$ with $b_n \in B_\Pi$ to some position b. We call F an *admissible strategy* for Π from position b_0 if for all partial matches $\rho = b_0 b_1 \cdots b_n$ with $b_n \in B_\Pi$ we have $F(\rho) \in E[b_n]$. A strategy F is called *history-free* if F depends only on the actual position of the match and not on its history, that is $F(\rho) = F(\rho')$ for all matches ρ and ρ' that have the same last position.

Let F be a strategy for Π. Then a match ρ is called *F-guided* if for all initial parts $b_0 b_1 \cdots b_n$ of ρ ending at $b_n \in B_\Pi$ we have that $b_0 b_1 \cdots b_n F(b_0 b_1 \cdots b_n)$ is also an initial part of match ρ.

A strategy F for Π is called a *winning strategy* for Π in $\mathcal{G}$ from position $b \in G$ if F is an admissible strategy from b and all complete F-guided matches ρ starting from b are won by Π.

A position $b \in G$ is called a *winning position* for Π if there is a winning strategy for Π in $\mathcal{G}$ starting from b. We use the following notation to denote the set of winning positions of players $\exists$ and $\forall$ in $\mathcal{G}$:

$$\mathrm{Win}_\exists(\mathcal{G}) := \{b \in B_\exists \cup B_\forall \mid b \text{ is a winning position for } \exists\}$$
$$\mathrm{Win}_\forall(\mathcal{G}) := \{b \in B_\exists \cup B_\forall \mid b \text{ is a winning position for } \forall\}.$$

Now we are ready to state the main result about parity games:

2.2.2. THEOREM ([EJ91b, Mos91]). *Let $\mathcal{G} = (B_\exists, B_\forall, E, \Omega)$ be a parity graph game and $G = B_\exists \cup B_\forall$ the set of its positions. Then $\mathcal{G}$ is history-free determined, i.e. $G = \mathrm{Win}_\exists(\mathcal{G}) \cup \mathrm{Win}_\forall(\mathcal{G})$ and the player who has a winning strategy from a position b also has a history-free winning strategy.*

In addition to the original papers mentioned in the theorem, the reader can also check [Zie98] for the proof of this theorem.

2.3 Category theory and coalgebras

This section introduces some terminology and background on category theory and coalgebras. We presuppose that the reader has made contact with basic concepts from category theory before. For example, we assume familiarity with basic notions such as categories and functors. We refer the reader to [ML71] for detailed definitions.

2.3.1 Sets and relations

In this thesis, we work in the category Set, that has sets as objects and functions as arrows. It is assumed that the reader is familiar with the usual constructions on sets, so the following explanations are there to fix notation. The notation $f : X \to Y$ means that f is a function with domain X and codomain Y. The identity function for a set X is denoted by $id_X : X \to X$. The composition of two functions $f : X \to Y$ and $g : Y \to Z$ is the usual composition of functions written as $g \circ f : X \to Z$. For sets $X' \subseteq X$, the inclusion map from X' to X is denoted by $i_{X',X} : X' \hookrightarrow X$, $x \mapsto x$. For a function $f : X \to Y$, we define the set $\mathsf{ran}(f) = \{y \in Y \mid \exists x \in X, f(x) = y\} \subseteq Y$.

Another category that we will use frequently is the category Rel of sets and relations between sets. Its arrows from a set X to a set Y are all the relations between X and Y. We write $R : X \nrightarrow Y$ to indicate that R is a relation between X and Y. Note that a relation $R : X \nrightarrow Y$ as an arrow in the category Rel is not just a set of pairs, that is, a subset of $X \times Y$, but it also contains information about its domain and codomain. At some places, especially once we use relation liftings later, it matters what the domain and codomain of a relation are.

The graph of any function $f : X \to Y$ is a relation between X and Y for which we write again $f : X \nrightarrow Y$. It will be clear from the context, in which a symbol f occurs, whether it is meant as an arrow in Set or as an arrow in Rel.

The diagonal relations $\Delta_X : X \nrightarrow X$ are the identity arrows in the category Rel and defined by $(x, x') \in \Delta_X$ iff $x = x'$. The composition of two relations $R : X \nrightarrow Y$ and $S : Y \nrightarrow Z$ is written as $R \,;\, S : X \nrightarrow Z$ and defined by:

$$R \,;\, S := \{(x, z) \in X \times Z \mid (x, y) \in R \text{ and } (y, z) \in S \text{ for some } y \in Y\}.$$

The composition of relations is written the other way round than the composition of functions. So we have, using the identification of functions with the relation of its graph, that $g \circ f = f \,;\, g$ for functions $f : X \to Y$ and $g : Y \to Z$. The converse $R^\circ : Y \nrightarrow X$ of a relation $R : X \nrightarrow Y$ is given by $(y, x) \in R^\circ$ iff $(x, y) \in R$. The projections of a relation $R : X \nrightarrow Y$ are denoted by $\pi_1 : R \to X$ and $\pi_2 : R \to Y$. It holds that $R = \pi_1^\circ \,;\, \pi_2$.

For a relation $R : X \nrightarrow Y$ we define the following sets:

$$\mathsf{dom}(R) := \{x \in X \mid \exists y \in Y \; s.t. \; (x, y) \in R\} \subseteq X,$$

$$\mathsf{ran}(R) := \{y \in Y \mid \exists x \in X \ s.t.\ (x, y) \in R\} \subseteq Y.$$

We call a relation $R : X \nrightarrow Y$ *full* on X (respectively Y) if we have $\mathsf{dom}(R) = X$ (respectively $\mathsf{ran}(R) = Y$).

For any set X, let $\in_X : X \nrightarrow \mathcal{P}X$ be the *membership relation* between X and subsets of X.

2.3.2 Coalgebras

In this section, we formally define coalgebras. We assume, if not explicitly stated otherwise, that functors are covariant endofunctors on the category Set, and we show that all of the transition systems discussed in section 2.1.1 can be seen as coalgebras for suitable functors.

2.3.1. Definition. Given a setfunctor $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$, a T-*coalgebra* is a pair $\mathbb{S} = (S, \sigma)$ where S is a set of *states* and $\sigma : S \to \mathsf{T}S$ is the *structure map* of the coalgebra $\mathbb{S}$. A *coalgebra homomorphism* or just *morphism* from T-coalgebra $\mathbb{S} = (S, \sigma)$ to $\mathbb{S}' = (S', \sigma')$, written as $f : \mathbb{S} \to \mathbb{S}'$, is a map $f : S \to S'$ such that the following diagram commutes:

$$\begin{array}{ccc} S & \xrightarrow{f} & S' \\ \sigma \downarrow & & \downarrow \sigma' \\ \mathsf{T}S & \xrightarrow[\mathsf{T}f]{} & \mathsf{T}S' \end{array}$$

Intuitively, coalgebra morphisms are functions that preserve T-structure. The identity function on a T-coalgebra is always a coalgebra morphism, and the composition of two coalgebra morphisms is again a a coalgebra morphism. Thus the collection of all T-coalgebras together with T-coalgebra morphisms is a category, which we denote by Coalg(T).

2.3.2. Remark. In general, T-coalgebras can be defined for any endofunctor $\mathsf{T} : C \to C$, however since we only work with set functors in this thesis we restrict Definition 2.3.2 to endofunctors on Set.

2.3.3. Definition. Given a functor $\mathsf{T} : \mathsf{Set} \to \mathsf{Set}$, a *pointed* T-*coalgebra* is a pair $(\mathbb{S}, s)$ consisting of a T-coalgebra $\mathbb{S}$ together with a distinguished element s from the set of states of $\mathbb{S}$.

2.3.2.1 Examples

In the following, we consider some examples of coalgebras of set functors from [KP11]. Most of the examples concern modal logic, in the sense that they are coalgebraic representations of the semantic structures of modal logics discussed in Section 2.1.1.

2.3.4. Example. For our first example, we consider *deterministic transition systems with output*. These systems are such that on each state one can *observe* a value from a fixed set A, and the system moves on to a next state. Such a system can be represented as a coalgebra $(S, \sigma : S \to A \times S)$ for the functor T which maps a set S to $A \times S$. More precisely, the structure map σ for this coalgebra is defined in a way that $\pi_1(\sigma(s)) \in A$ denotes the current value of the system, and $\pi_2(\sigma(s)) \in S$ specifies the next state, with π_1 and π_2 denoting the projection maps on $A \times S$.
Considering the same functor $\mathsf{T} = (A \times -)$, it is easy to see that the system $(A^\omega, (\mathsf{hd}, \mathsf{tl}))$ of infinite words over A together with maps "head" and "tail" that we saw in Section 1.2.1, is a T-coalgebra. In fact, infinite words form a final deterministic transition system with output. We will say more about this in Section 2.3.2.2.

2.3.5. Example. [Rut00] Recall that *deterministic automata* are tuples $\mathbb{A} = (A, C, \Theta, F)$ such that A is a set of states, C is an alphabet, $\Theta : A \times C \to A$ is a transition map and $F \subseteq A$ is a collection of accepting states. Observe that we can represent F by the characteristic map $\chi_F : A \to 2$ (where 2 denotes the set $\{0, 1\}$) and represent Θ as a map $A \to A^C$ where A^C denotes the collection of maps from C to A. Hence we can model a deterministic automaton $\mathbb{A}$ as a T-coalgebra, where $\mathsf{T}A := 2 \times A^C$ for all A.

2.3.6. Example. Recall from Example 2.1.1 that a *Kripke frame* is a pair (W, R) with W as the set of states and the binary relation $R \subseteq W \times W$ as an accessibility relation. These structures are coalgebras for the *powerset* functor $\mathcal{P} : \mathsf{Set} \to \mathsf{Set}$, which maps a set S to the set of all its subsets $\mathcal{P}S = \{U \mid U \subseteq S\}$. A function $f : S \to T$ is mapped to $\mathcal{P}f : \mathcal{P}S \to \mathcal{P}T$, which is defined for any $U \subseteq S$ by $\mathcal{P}f(U) = f[U] = \{f(u) \mid u \in U\}$. A Kripke frame (W, R) is a $\mathcal{P}$-coalgebra $(W, R[-])$, where $R[-] : W \to \mathcal{P}W$ is defined as follows:

$$R[w] := \{w' \in W \mid w' \text{ is a } R\text{-successor of } w\}.$$

A Kripke model (W, R, V) is a Kripke frame together with a valuation $V : \mathtt{X} \to \mathcal{P}W$ where $\mathtt{X}$ is the set of proposition letters of a given modal logic. Note that V can also be seen as a map $m_V : W \to \mathcal{P}\mathtt{X}$ by setting $m_V(w) := \{p \in \mathtt{X} \mid p \in V(w)\}$. We call m_V the *marking* associated with valuation V. Then it is easy to see that Kripke models are structures of the form (W, σ) where $\sigma : W \to \mathcal{P}W \times \mathcal{P}\mathtt{X}$ which are coalgebras for the functor $\mathsf{T} = \mathcal{P} - \times \mathcal{P}\mathtt{X}$.

2.3.7. EXAMPLE. A *multiset frame* is a directed graph with $\mathbb{N}$-weighted edges, known also as a multigraph [DV02]. More precisely, a multiset frame has a (finite) number of successors, each of which comes with a weight or multiplicity. A multiset frame can be seen as a coalgebra for the *Bag (finitary multiset)* functor. The Bag (finitary multiset) functor $\mathcal{B} : \mathsf{Set} \to \mathsf{Set}$ sends a set S to the set of mappings $f : S \to \omega$ such that the set $\{u \in S \mid f(u) > 0\}$ is finite. The action on morphisms is given by letting, for $f \in \mathcal{B}X$ and $h : S \to Y$, the multiset $\mathcal{B}h(f) : Y \to \omega$ be defined by $w \mapsto \sum_{h(v)=w} f(v)$.

Multiset frames can be used to provide semantics for graded modal logic (see Example 2.1.2). Let us note that, originally, formulas of graded modal logic [Fin72] are interpreted over Kripke frames, not multiset frames. However, each Kripke frame can be seen as a multigraph with all edges with weight 1 and, conversely, to each multigraph we can associate a Kripke frame by adding enough copies of successor states.

2.3.8. EXAMPLE. Recall from Example 2.1.3 that a probabilistic transition system is a pair (W, P) where W is a set of states and $P = (\mu_w)_{w \in W}$ is a family of probability distributions on W. Probabilistic transition systems are coalgebras for the *finitary distribution functor* $\mathcal{D} : \mathsf{Set} \to \mathsf{Set}$, which maps a set X to the set of discrete probability distributions over X with finite support, i.e. $\mathcal{D}X$ is the set of all $\mu : X \to [0, 1]$ such that $\sum_{x \in X} \mu(x) = 1$, with finite support. $\mathcal{D}$ maps a function $f : X \to Y$ to the function $\mathcal{D}f : \mathcal{D}X \to \mathcal{D}Y$ mapping $\mu : X \to [0, 1]$ to

$$\begin{aligned} \mathcal{D}f(\mu) : Y &\longrightarrow [0, 1] \\ y &\mapsto \sum_{x \in f^{-1}(\{y\})} \mu(x). \end{aligned}$$

$\mathcal{D}$-coalgebras are generally referred to as Markov chains.

2.3.9. EXAMPLE. As we mentioned in Example 2.1.4, neighbourhood frames generalize Kripke frames and are used in the semantics of classical modal logic. Neighbourhood structures can be seen as coalgebras for the double *contravariant powerset* functor $\mathcal{N} := \mathcal{Q}\mathcal{Q}$ also known as the *neighbourhood* functor [HKP09]. The contravariant powerset functor $\mathcal{Q}$ maps a set S to $\mathcal{Q}S = \mathcal{P}S$. On functions $\mathcal{Q}$ is the inverse image map, that is, for a function $f : S \to T$ we have $\mathcal{Q}f : \mathcal{Q}T \to \mathcal{Q}S$, $V \mapsto f^{-1}[V]$. The neighborhood functor $\mathcal{N} : \mathsf{Set} \to \mathsf{Set}$ maps a set S to $\mathcal{N}S = \mathcal{Q}\mathcal{Q}S$ and a map $f : S \to T$ to a map $\mathcal{N}f : \mathcal{N}S \to \mathcal{N}T$ with $\mathcal{N}f(\alpha) = \{V \subseteq T \mid f^{-1}[V] \in \alpha\}$ for all $\alpha \in \mathcal{N}S$.

2.3.10. EXAMPLE. A monotone neighbourhood frame from example 2.1.5 can be characterized as a coalgebra for the *monotone neighbourhood* functor $\mathcal{M} : \mathsf{Set} \to \mathsf{Set}$. We need the following notion in order to define how $\mathcal{M}$ operates on sets and functions. Given a set S and an element $\alpha \in \mathcal{N}S$, we define

$$\alpha^{\uparrow} := \{X \in \mathcal{P}S \mid Y \subseteq X \text{ for some } Y \in \alpha\},$$

and we say that α is *upward closed* if $\alpha = \alpha^{\uparrow}$. The monotone neighborhood functor $\mathcal{M}$ is the restriction of the neighborhood functor to upward closed sets. More concretely, the functor $\mathcal{M}$ is given by $\mathcal{M}S := \{\beta \in \mathcal{N}S \mid \beta \text{ is upward closed}\}$, while for $f : S \to T$, we define $\mathcal{M}f : \mathcal{M}S \to \mathcal{M}T$ by $\mathcal{M}f(\beta) := \mathcal{N}f(\beta)$. It is straightforward to check $\mathcal{N}f(\beta)$ is upward closed whenever β is upward closed.

2.3.11. Example. Game frames (see Example 2.1.6) can also be represented as coalgebras. A game frame is a coalgebra for the *game* functor $\mathcal{G} : \mathsf{Set} \to \mathsf{Set}$ which is defined in the following. Let N be a set of agents and S_i be the set of strategies for the respective agent $i \in N$. The game functor $\mathcal{G}$ maps a set X to:

$$\mathcal{G}(X) = \{((S_i)_{i\in N}, f) \mid \forall i \in N(\emptyset \neq S_i \subseteq \omega) \text{ and } f : \prod_{i\in N} S_i \to X\}$$

where f is an outcome function that produces a new position of the game given the choice of individual strategies. $\mathcal{G}$ maps a function $h : X \to Y$ to the function $\mathcal{G}h : \mathcal{G}X \to \mathcal{G}Y$ by putting $\mathcal{G}h((S_i)_{i\in N}, f) := ((S_i)_{i\in N}, h \circ f)$.

2.3.2.2 Final Coalgebra

A *final* coalgebra for a functor T is a final or terminal object of the category Coalg(T), i.e., for every T-coalgebra $\mathbb{S} = (S, \sigma)$ there exists a unique morphism to the final coalgebra. This unique morphism, as demonstrated on many examples in [Rut00], expresses the "abstract behaviour of states of $\mathbb{S}$". Moreover, there are many well-known mathematical structures that are associated with the final coalgebra of some functor.

Below we give some concrete examples of final coalgebras from [Rut00].

2.3.12. Example. [Rut00] Consider the set functor $\mathsf{T} := (A\times -)$ of deterministic transition systems with output, and a T-coalgebra $\mathbb{S} = (S, \sigma : S \to A \times S)$. Define a map $\mathsf{beh} : S \to A^{\omega}$ by $\mathsf{beh}(s) = (a_0, a_1, a_2, \cdots)$ such that $\sigma(s) = (a_0, s_1)$, $\sigma(s_1) = (a_1, s_2)$ and $\sigma(s_2) = (a_2, s_3)$, etc. It is not difficult to check that the behaviour map $s \mapsto \mathsf{beh}(s)$ is the unique homomorphism from $\mathbb{S}$ to the coalgebra $(A^{\omega}, (\mathsf{hd}, \mathsf{tl}))$ (the collection of infinite words over A together with maps head and tail as mentioned in Section 1.2.1). This shows that $(A^{\omega}, (\mathsf{hd}, \mathsf{tl}))$ is the final object in the category Coalg$(A \times -)$.

2.3.13. Example. [Rut00] Given the set functor $\mathsf{T} := 2 \times (-)^C$, and recall that T-coalgebras are deterministic automata over the alphabet C. The collection $\mathcal{P}(C^*)$ of all languages over C provides the state space of the final coalgebra for this functor. Consider the following transition function $\sigma : \mathcal{P}(C^*) \to 2 \times \mathcal{P}(C^*)^C$ to turn $\mathcal{P}(C^*)$ into a T-coalgebra. Writing $\sigma(L) = (\sigma_1(L), \sigma_2(L))$ for $L \in \mathcal{P}(C^*)$, define:

$$\sigma_1(L) = \begin{cases} 1 & \text{if the empty word belongs to } L \\ 0 & \text{otherwise} \end{cases}$$

and

$$\sigma_2(L)(c) := \{w \in C^* \mid c.w \in L\}.$$

With this definition, the structure $(\mathcal{P}(C^*), \sigma)$ forms a final object in the category $\mathrm{Coalg}(2 \times (-)^C)$. Given a $(2 \times (-)^C)$-coalgebra $\mathbb{A}$, the unique homomorphism $\mathsf{lang} : \mathbb{A} \to (\mathcal{P}(C^*), \sigma)$ maps a state $a \in A$ to the language that is accepted by the deterministic automaton that is obtained by taking a as initial state of $\mathbb{A}$.

2.3.2.3 Coinduction

An important application of final coalgebras is provided by the notion of *co-induction*, both as a definition and proof principle. This principle is dual to that of induction. To see this, without going to technicalities, one can describe induction as "use of initiality for algebras". Dually, coinduction can be formulated by "use of finality for coalgebras" as mentioned in [JR97]. Finality of a coalgebra involves two concepts: *existence* which is used to *define* maps *into* the final coalgebra and *uniqueness* which corresponds to the coinduction proof principle.

Coinduction is an important concept in the theory of coalgebras. However it does not directly play a role in this thesis, thus we keep this section short and refer the interested reader to [JR97] for more information.

2.3.3 Properties of set functors

In this section, we gather some properties of set functors that are of interest in the setting of coalgebras.

Inclusion preservation A set functor T is i*nclusion preserving* if for all sets X' and X with $X' \subseteq X$, we have $\mathsf{T}i_{X',X} = i_{\mathsf{T}X',\mathsf{T}X}$, where i stands for the inclusion map.

Each set functor is "almost" inclusion-preserving. For each set functor T there exists an inclusion-preserving set functor T' such that the restriction of T and T' to all non-empty sets and non-empty maps are naturally isomorphic [AT90, Theorem on page 132].

Finite intersection preservation. A set functor T preserves finite intersections if for all sets X and Y we have: $\mathsf{T}(X \cap Y) = \mathsf{T}X \cap \mathsf{T}Y$.

Every set functor preserves finite non-empty intersections [Trn69] and [AGT09, Proposition 2.2].

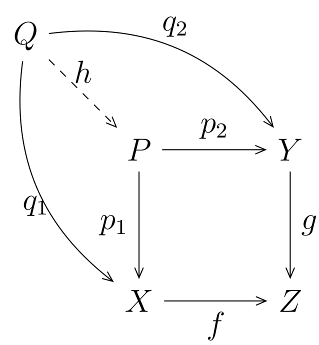
Preservation of surjective maps A set functor T preserves surjective maps if for all surjective maps $f : A \to B$ the map $\mathsf{T}f : \mathsf{T}A \to \mathsf{T}B$ is surjective too. All set functors preserve surjective maps [AT90, Proposition 4.2.(ii)].

Weak pullback preservation. A very important property of set functors for the theory of coalgebras is weak pullback preservation. Let us first define weak pullbacks.

2.3.14. DEFINITION. Given functions $f : X \to Z$ and $g : Y \to Z$, the *pullback* of f and g is the following set

$$P = \{(x, y) \in X \times Y \mid f(x) = g(y)\},$$

together with the projections $p_1 : P \to X$ and $p_2 : P \to Y$ such that $f \circ p_1 = g \circ p_2$. The pullback of f and g is determined up-to-isomorphism by the following universal property: For any set Q and functions q_1 and q_2 that satisfy $f \circ q_1 = g \circ q_2$, there is a *unique* function $h : Q \to P$ such that $q_1 = p_1 \circ h$ and $q_2 = p_2 \circ h$. This universal property is described in the following diagram.



If the function h is not necessarily unique, we call (P, p_1, p_2) a *weak pullback*.

2.3.15. DEFINITION. A functor T *preserves weak pullbacks* if it transforms every weak pullback (P, p_1, p_2) for f and g into a weak pullback $(\mathsf{T}P, \mathsf{T}p_1, \mathsf{T}p_2)$ for $\mathsf{T}f$ and $\mathsf{T}g$. An equivalent characterization is to require T to *weakly preserve pullbacks*, that is, to turn pullbacks into weak pullbacks [Rut00, GS05].

2.3.16. EXAMPLE. All the functors mentioned in the examples of section 2.3.2 preserve weak pullbacks, except for the neighborhood functor and its monotone variant.

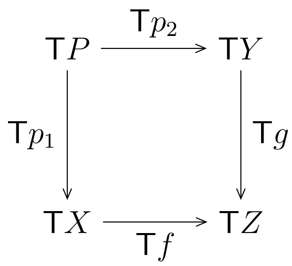
2.3.17. EXAMPLE. The set of *Kripke polynomial functors* (KPF) is inductively defined as follows:

$$\mathsf{T} ::= \mathsf{C}_A \mid \mathrm{Id} \mid \mathsf{T} \times \mathsf{T} \mid \mathsf{T} + \mathsf{T} \mid \mathsf{T}^D \mid \mathcal{P}\mathsf{T},$$

where for a finite set A, C_A denotes the *constant* functor mapping all sets to A, Id refers to the identity functor, and the $\times$ and $+$ denote binary product and disjoint union respectively. Furthermore given a finite set D, we write T^D for the functor mapping a set S to the D-fold product $(\mathsf{T}S)^D$. The class of *polynomial* *functors* (PF) consists of all functors $\mathsf{T} \in \text{KPF}$ that do not involve the power set functor. It can be shown that the property of preserving weak pullbacks is preserved under the operations $+$, $\times$, $(-)^D$ and $\mathcal{P}()$, so that all Kripke polynomial functors preserve weak pullbacks.

The next fact gives a characterisation for weak pullback-preserving set functors:

2.3.18. FACT. [Gum01] A set functor T preserves weak pullbacks if and only if for all maps $f : X \to Z$ and $g : Y \to Z$ we have: For all $u \in \mathsf{T}X, v \in \mathsf{T}Y$ with $\mathsf{T}f(u) = \mathsf{T}g(v)$ there is an element $w \in \mathsf{T}P$ such that $\mathsf{T}p_1(w) = u$ and $\mathsf{T}p_2(w) = v$, where $P = \{(x, y) \mid f(x) = g(y)\}$ with p_1 and p_2 as projections on X and Y.



Finite set preservation. A functor T *preserves finite sets* if $\mathsf{T}X$ is finite whenever X is.

2.3.19. DEFINITION. An inclusion preserving functor T is called *finitary* if for all sets X, we have:

$$\mathsf{T}X = \bigcup\{\mathsf{T}X' \mid X' \subseteq X \text{ and } X' \text{ is finite}\}.$$

The finitary version T_ω of an inclusion preserving functor T is defined such that it maps a set X to $\mathsf{T}_\omega X = \bigcup\{\mathsf{T}X' \mid X' \subseteq X \text{ and } X' \text{ is finite}\}$.

2.3.20. EXAMPLE. Given the power set functor $\mathcal{P}$, its finitary version $\mathcal{P}_\omega$ maps a set X to the set of all its finite subsets.

The reason that we are interested in finitary functors is that we want our language to be finitary, in the sense that a formula has only finitely many subformulas. The key property of finitary functors that will make this possible, is that every $\tau \in \mathsf{T}X$ is supported by a finite subset of X, and in fact, there will always be a *smallest* such set.

2.3.21. DEFINITION. Given a functor T, we define for every set X the map Base_X:

$$\begin{array}{rcl}\text{Base}^{\mathsf{T}}_X : \mathsf{T}_\omega X & \longrightarrow & \mathcal{P}_\omega X \\ \tau & \mapsto & \bigcap\{X' \subseteq X \mid \tau \in \mathsf{T}X'\}.\end{array}$$

The key point about this notion is that $\text{Base}(\tau)$ (we drop the subscript X and superscript T whenever possible) is the *least* set $U \in \mathcal{P}_\omega X$ such that $\tau \in \mathsf{T}U$.

2.3.22. EXAMPLE. The following examples are easy to check: $\text{Base}^{Id}_X : X \to \mathcal{P}X$ is the singleton map and $\text{Base}^{\mathcal{P}}_X : \mathcal{P}_\omega X \to \mathcal{P}_\omega X$ is the identity map on $\mathcal{P}_\omega X$.

Natural transformation

2.3.23. DEFINITION. A *natural transformation* $\lambda : \mathsf{F} \Rightarrow \mathsf{T}$ from a set functor F to a set functor T provides a function $\lambda_X : \mathsf{F}X \to \mathsf{T}X$ for every set X such that for all functions $f : X \to Y$ the following diagram commutes:

$$\begin{array}{ccc}\mathsf{F}X & \xrightarrow{\lambda_X} & \mathsf{T}X \\ {\scriptstyle \mathsf{F}f}\downarrow & & \downarrow{\scriptstyle \mathsf{T}f} \\ \mathsf{F}Y & \xrightarrow[\lambda_Y]{} & \mathsf{T}Y\end{array}$$

2.3.24. FACT. [KKV12] $\text{Base}^{\mathsf{T}} : \mathsf{T}_\omega \Rightarrow \mathcal{P}_\omega$ is a natural transformation if T preserves weak pullbacks.

2.3.4 Behavioural equivalence and Bisimulation

The coalgebraic formulation of systems enables us to define general notions of behavioural equivalence and bisimulation as the principal notions of equivalence between states of systems. As we discussed in Section 1.2.2, viewing coalgebra morphisms as behaviour preserving maps, we can define behavioural equivalence in terms of a pullback of a cospan of morphisms i.e., two states are behaviourally equivalent if and only if they can be identified by a pair of coalgebra morphisms (see Figure 1.1 on page 9).

2.3.25. DEFINITION. Given a set functor T and T-coalgebras (S, σ) and (S', σ'), we say that two states $s \in S$ and $s' \in S'$ are *behaviourally equivalent* if there is a T-coalgebra (Q, γ) and T-coalgebra morphisms $f : (S, \sigma) \to (Q, \gamma)$ and $f' : (S', \sigma') \to (Q, \gamma)$ such that $f(s) = f'(s')$.

In Section 1.2.2 we defined the notion of T-bisimulation between T-coalgebras as a span of coalgebra morphisms (see Figure 1.1), and mentioned that T-bisimilarity between T-coalgebras is closely related to behavioural equivalence [Rut00]. In particular, for all functors that preserve weak pullbacks the two notions coincide. In this section, we will define a more general notion of bisimulation for coalgebras, using *relation liftings*.

2.3.4.1 Relation liftings

2.3.26. Definition. A *relation lifting* L for a set functor T assigns to a relation $R : X \nrightarrow Y$ a relation $LR : \mathsf{T}X \nrightarrow \mathsf{T}Y$. We require relation liftings to preserve converse, meaning that $L(R^\circ) = (LR)^\circ$ for all relations R.

2.3.27. Example. Given a binary relation $R : X \nrightarrow Y$ with the projection maps $\pi_X : R \to X$ and $\pi_Y : R \to Y$, the so-called *Barr lifting* $\overline{\mathsf{T}}$ of a set functor T maps R to the relation $\overline{\mathsf{T}}R : \mathsf{T}X \nrightarrow \mathsf{T}Y$ defined as:

$$\overline{\mathsf{T}}R := \{(x, y) \mid \exists z \in \mathsf{T}R \text{ s.t. } \mathsf{T}\pi_X(z) = x \text{ and } \mathsf{T}\pi_Y(z) = y\}.$$

2.3.28. Example. The *Egli-Milner lifting* $\overline{\mathcal{P}}$ is a relation lifting for the covariant power set functor $\mathcal{P}$ that is defined for any $R : X \nrightarrow Y$ as $\overline{\mathcal{P}}R = \overrightarrow{\mathcal{P}}R \cap \overleftarrow{\mathcal{P}}R$, where:

$$\overrightarrow{\mathcal{P}}R := \{(U, V) \in \mathcal{P}X \times \mathcal{P}Y \mid \forall u \in U\ \exists v \in V \text{ s.t. } (u, v) \in R\},$$

$$\overleftarrow{\mathcal{P}}R := \{(U, V) \in \mathcal{P}X \times \mathcal{P}Y \mid \forall v \in V\ \exists u \in U \text{ s.t. } (u, v) \in R\}.$$

It is not difficult to check that $\overline{\mathcal{P}}$ is an instance of the Barr lifting.

The next result applies the Barr lifting to give another characterization of weak pullback-preserving functors:

2.3.29. Fact. [Rut98] A functor T weakly preserves pullbacks if and only if for all relations $R : X \nrightarrow Y$ and $Q : Y \nrightarrow Z$ we have:

$$\overline{\mathsf{T}}(R\,;Q) = \overline{\mathsf{T}}R\,;\overline{\mathsf{T}}Q.$$

2.3.30. Fact. [Mos99, Bal00]Let T be a set functor that preserves inclusions and weak pullbacks. Then the relation lifting $\overline{\mathsf{T}}$

(1) extends T to the category Rel: $\overline{\mathsf{T}}f = \mathsf{T}f$;

(2) is monotone: $R \subseteq Q$ implies $\overline{\mathsf{T}}R \subseteq \overline{\mathsf{T}}Q$;

(3) commutes with taking restrictions: $\overline{\mathsf{T}}(R{\restriction}_{X\times X'}) = (\overline{\mathsf{T}}R){\restriction}_{\mathsf{T}X\times\mathsf{T}X'}$.

The lifting of special relations, like the membership relation, is used to define notions that will be used in the next Chapters.

2.3.31. Definition. Given a functor T that preserves weak pullbacks, and a set X, we let $\in_X \subseteq X \times \mathcal{P}X$ denote the membership relation, restricted to X. We define the maps $\lambda^{\mathsf{T}}_X : \mathsf{T}\mathcal{P}X \to \mathcal{P}\mathsf{T}X$ by

$$\lambda^{\mathsf{T}}_X(\Phi) := \{\alpha \in \mathsf{T}X \mid \alpha \ \overline{\mathsf{T}}{\in_X}\ \Phi\},$$

and call members of $\lambda^{\mathsf{T}}_X(\Phi)$ *lifted members* of Φ. An object $\Phi \in \mathsf{T}\mathcal{P}X$ is a *redistribution of* $\Gamma \in \mathcal{P}\mathsf{T}X$ if $\Gamma \subseteq \lambda^{\mathsf{T}}_X(\Phi)$. In case $\Gamma \in \mathcal{P}_\omega\mathsf{T}_\omega X$, we call a redistribution Φ *slim* if $\Phi \in \mathsf{T}_\omega\mathcal{P}_\omega(\bigcup_{\alpha\in\Gamma} \mathrm{Base}(\alpha))$. The set of all slim redistributions of Γ is denoted as $SRD(\Gamma)$.

Slim redistributions will be later used in Chapter 3 to prove a simulation theorem for coalgebraic automata and in Chapter 5 in order to define an axiom system for coalgebraic fixpoint logic.

An important use of relation liftings is to yield a notion of bisimulation.

2.3.32. Definition. Let L be a relation lifting for T and let $\mathbb{S} = (S, \sigma)$ and $\mathbb{S}' = (S', \sigma')$ be two T-coalgebras. An L-*bisimulation* between $\mathbb{S}$ and $\mathbb{S}'$ is a relation $R : S \nrightarrow S'$ such that $(\sigma(s), \sigma'(s')) \in LR$, for all $(s, s') \in R$. Two states $s \in \mathbb{S}$ and $s' \in \mathbb{S}'$ are L-bisimilar if there is an L-bisimulation R between $\mathbb{S}$ and $\mathbb{S}'$ with $(s, s') \in R$.

To see the relation between L-bisimulation and the span definition of T-bisimulation in Section 1.2.2, one can check that if $L = \overline{\mathsf{T}}$ then the two notions coincide. More precisely, a relation $R : S \nrightarrow S'$ with projections $\pi_S : R \to S$ and $\pi'_S : R \to S'$ is a $\overline{\mathsf{T}}$-bisimulation between coalgebras $\mathbb{S} = (S, \sigma)$ and $\mathbb{S}' = (S', \sigma')$ iff there is a map $\gamma : R \to \mathsf{T}R$ such that the following diagram commutes:

$$\begin{array}{ccccc}
S & \xleftarrow{\ \pi_S\ } & R & \xrightarrow{\ \pi'_S\ } & S' \\
{\scriptstyle\sigma}\downarrow & & \downarrow{\scriptstyle\gamma} & & \downarrow{\scriptstyle\sigma'} \\
\mathsf{T}S & \xleftarrow[\ \mathsf{T}\pi_S\ ]{} & \mathsf{T}R & \xrightarrow[\ \mathsf{T}\pi'_S\ ]{} & \mathsf{T}S'
\end{array}$$

A motivation to define the notion of L-bisimulation is to get a simpler characterization of behavioral equivalence. Often it is easier to check whether there is a bisimulation between states of two coalgebras than to find two coalgebra morphisms into a third coalgebra that identify the states. Of course this only works for functors and relation liftings for which the notion of bisimilarity is the same as behavioral equivalence. If a set functor T preserves weak pullbacks, $\overline{\mathsf{T}}$-bisimilarity and behavioral equivalence coincide [Rut00]. In general, however, bisimilarity is a strictly stronger notion than behavioral equivalence. For instance one can give an example of monotone neighbourhood frames that are behaviorally equivalent while there is no $\overline{\mathcal{M}}$-bisimulation between them (see [HK04] for details).

In the next example, we will define a relation lifting $\widetilde{\mathcal{M}}$ for the monotone neighbourhood functor $\mathcal{M}$. This relation lifting is distinct from $\overline{\mathcal{M}}$ and captures the natural notion of bisimilarity associated with monotone neighborhood models. (check [AM89], [HK04] and [SV10] for more details).

2.3.33. Example. Based on the definition of $\overrightarrow{\mathcal{P}}R$ from Example 2.3.28, we can define a relation lifting $\widetilde{\mathcal{M}}$ for the monotone neighborhood functor $\mathcal{M}$ on a relation $R : X \nrightarrow Y$ as follows:

$$\widetilde{\mathcal{M}}R := \overrightarrow{\mathcal{P}}\overleftarrow{\mathcal{P}}R \cap \overleftarrow{\mathcal{P}}\overrightarrow{\mathcal{P}}R.$$

In words, $(\alpha, \beta) \in \widetilde{\mathcal{M}}R$ if for all $U \in \alpha$ there is some $V \in \beta$ such that for each $v \in V$ there exists a $u \in U$ with $(u, v) \in R$, and for all $V \in \beta$ there is some $U \in \alpha$ such that for all $u \in U$ there is a $v \in V$ satisfying $(u, v) \in R$.

We now recall the definition of bisimulation between $\mathcal{M}$-models [Pau99], and provide some intuition on the connection between $\widetilde{\mathcal{M}}$ and monotone neighbourhood bisimulation.

2.3.34. Definition. A *monotone neighbourhood bisimulation* between monotone neighborhood models $\mathbb{S}_1$ and $\mathbb{S}_2$ is a relation $R \subseteq S_1 \times S_2$ such that, if $s_1 R s_2$ then:

- $m_1(s_1) = m_2(s_2)$, with m_1 and m_2 denoting the colourings associated with V_1 and V_2;
- for all $Z_1 \in \sigma_1(s_1)$ there is $Z_2 \in \sigma_2(s_2)$ such that for all $t_2 \in Z_2$ there is $t_1 \in Z_1$ with $t_1 R t_2$;
- for all $Z_2 \in \sigma_2(s_2)$ there is $Z_1 \in \sigma_1(s_1)$ such that for all $t_1 \in Z_1$ there is $t_2 \in Z_2$ with $t_1 R t_2$.

2.3.5 Algebras vs coalgebras

We conclude this section with a brief note on the duality between algebra and coalgebra. Considering the duality between important concepts of the universal theories of algebra and coalgebra, one may be encouraged to study coalgebra directly as a dual notion of algebra. Here we want to emphasise that the duality between concepts such as initial algebra and final coalgebra, induction and coinduction does not mean that the category Alg(T) is formally dual of the category Coalg(T). We refer the reader to [Rut00] for a detailed discussion of what the duality between algebra and coalgebra means. In the following chart from [Rut00] we briefly present a conceptual comparison between notions related to algebras and coalgebras.

Algebra	Coalgebra
Algebra homomorphism	Coalgebra homomorphism
Congruence	Bisimulation
Initial algebra	Final coalgebra
Induction	Coinduction
Recursion: map out of a initial algebra	Corecursion: map into a final coalgebra

Table 2.1: Conceptual comparison between algebras and coalgebras [Rut00]

2.4 Coalgebraic fixpoint logic and automata

In this section, we introduce some terminology and background on coalgebraic logic and automata. We give definitions for the two approaches toward coalgebraic fixpoint logic introduced in Section 1.3, and continue by defining coalgebraic automata corresponding to formulas of coalgebraic fixpoint logic.

2.4.1 Coalgebraic μ-calculus via the ∇-modality

We start by defining the ∇-based fixpoint logic for a weak pullback-preserving set functor T. Generalizing the case of standard modal logic, to define the semantics of ∇-formulas, we involve the relation lifting $\overline{\mathsf{T}}$ of T.

2.4.1. Definition. Given a weak pullback-preserving functor T, the language $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$ of coalgebraic fixpoint formulas is defined by the following grammar:

$$\varphi ::= \top \mid \bot \mid p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \nabla\alpha \mid \neg\varphi \mid \mu x.\varphi \mid \nu x.\varphi,$$

where p belongs to the set of propositional variables, and $\alpha \in \mathsf{T}_\omega(\mu\mathtt{ML}^{\mathsf{T}}_{\nabla})$. As in Definition 2.1.9 there is a restriction on the occurrence of negation in formulas $\mu p.\varphi$ and $\nu p.\varphi$, namely: every occurrence of x in φ must be in the *scope* (see Remark 2.4.2) of an even number of negations. We denote by $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}(\mathtt{X})$ the set of formulas with free variables from set $\mathtt{X}$.

2.4.2. Remark. Strictly speaking the clause for nabla formulas in Definition 2.4.1 is not stated in a correct recursive way, since it makes use of the whole language $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$, that has yet to be defined. But the readers may observe that since T_ω is a *finitary* functor, what we are saying is simply that for any finite set X of formulas, any object $\alpha \in \mathsf{T}_\omega(X)$ corresponds to a formula. In fact, any $\alpha \in \mathsf{T}_\omega(\mu\mathtt{ML}^{\mathsf{T}}_{\nabla})$ belongs to the set $\mathsf{T}_\omega(\mathrm{Base}(\alpha))$, and we will call the formulas in $\mathrm{Base}(\alpha)$ the *immediate subformulas* of the formula $\nabla\alpha$. To define the notions of *scope* and *occurrence* for formulas in $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$, we inductively define the construction tree of a formula, where the children of a node labeled $\nabla\alpha$ are the formulas in $Base(\alpha)$. Free and bound variables are then standardly defined (see [Ven06]).

2.4.3. Definition. A formula φ is called *clean* if no variable occurs both free and bound in φ, and no two distinct occurrences of fixed point operators bind the same variable. A formula φ is called *guarded* if every subformula $\mu x.\psi$ of φ has the property that all occurrences of x inside ψ are within the scope of a ∇.

2.4.4. Definition. Fix a clean formula φ. The sets $FV(\varphi)$ and $BV(\varphi)$ of its *free* and *bound* variables are standardly defined by induction on the complexity of the formula φ. For the case that $\varphi = \eta x.\psi$ with $\eta \in \{\mu, \nu\}$ we define $FV(\varphi) := FV(\psi) \setminus \{x\}$ and $BV(\varphi) := BV(\varphi) \cup \{x\}$. We let $\mu\mathtt{ML}(\mathtt{X})$ denote the set of μ-formulas of which all free variables belongs to the set $\mathtt{X}$.

To introduce the semantics of $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$ we first generalise Kripke models and define the notion of T-models over a set $\mathtt{X}$ of propositional letters.

2.4.5. Definition. A T-model $\mathbb{S} = (S, \sigma, V)$ is a T-coalgebra (S, σ) together with a valuation $V : \mathtt{X} \to \mathcal{P}S$.

2.4.6. Remark. Similar to the coalgebraic description of Kripke models in Example 2.3.6, we can encode the valuation of a T-model into its coalgebra structure. Given a valuation $V : \mathtt{X} \to \mathcal{P}S$, we define a colouring (marking) $m_V : S \to \mathcal{P}\mathtt{X}$ by $m_V(s) := \{x \in \mathtt{X} \mid s \in V(x)\}$. So we can think of a T-model $\mathbb{S} = (S, \sigma, V)$ as a $\mathcal{P}\mathtt{X}$-colored T-coalgebra $\mathbb{S} = (S, \sigma, m_V)$. We can also think of a T-model as a coalgebra $\mathbb{S} = (S, \sigma_m)$ for the functor $\mathsf{T}_{\mathtt{X}}$ defined by $\mathsf{T}_{\mathtt{X}}S := \mathcal{P}\mathtt{X} \times \mathsf{T}S$ where $\sigma_m : S \to \mathsf{T}_{\mathtt{X}}S$ is given by $\sigma_m(s) := (m_V(s), \sigma(s))$. This way we encode the valuation into the functor itself.

2.4.7. Definition. Using the relation lifting $\overline{\mathsf{T}}$, we define the semantics for the language $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}(\mathtt{X})$ on a T-model $\mathbb{S} = (S, \sigma, V)$. Since apart from the ∇ modality, the definition of the satisfaction relation $\Vdash$ is exactly the same as it is for the μ-calculus (see Definition 2.1.8), here we only give the definition for the ∇ modality:

$$s \Vdash \nabla\alpha \;\text{ iff }\; (\sigma(s), \alpha) \in \overline{\mathsf{T}}(\Vdash). \tag{2.1}$$

Note that again the clause for ∇ is not stated in a correct recursive way. We can only suppose that $\Vdash\restriction_{S\times Base(\alpha)}$ is already defined. The actual recursive definition is that $s \Vdash \nabla\alpha$ iff $(\sigma(s), \alpha) \in \overline{\mathsf{T}}(\Vdash\restriction_{S\times Base(\alpha)})$. One can apply Fact 2.3.30 item (3) to prove that this definition is equal to the clause given above.

2.4.8. Remark. In the sequel, we will use $\mu\mathtt{ML}^{\overline{\mathsf{T}}}_{\nabla}$ to refer to the ∇-based logic obtained for a functor T with the semantics given by the relation lifting $\overline{\mathsf{T}}$, whereas $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$ is used to refer to the ∇-based language for T without specifying the semantics.

Considering T-models as $\mathsf{T}_{\mathtt{X}}$-coalgebras, notions such as behavioural equivalence, bisimulation and homomorphisms for T-models are instances of the general coalgebraic notions for $\mathsf{T}_{\mathtt{X}}$-coalgebras. In particular, we obtain the following definition of $\overline{\mathsf{T}}$-bisimilarity for T-models.

2.4.9. Definition. Two T-models $\mathbb{S}_1 = (S_1, \sigma_1, V_1)$ and $\mathbb{S}_2 = (S_2, \sigma_2, V_2)$ over $\mathtt{X}$ are called $\overline{\mathsf{T}}$-bisimilar if there exists a relation $R : S_1 \nrightarrow S_2$ such that for all $(s_1, s_2) \in R$ we have $(\sigma_1(s_1), \sigma_2(s_2)) \in \overline{\mathsf{T}}R$ and $m_{V_1}(s_1) = m_{V_2}(s_2)$.

In terms of Definition 2.3.32 this means that the T-models $\mathbb{S}_1 = (S_1, \sigma_1, V_1)$ and $\mathbb{S}_2 = (S_2, \sigma_2, V_2)$ are $\overline{\mathsf{T}}$-bisimilar if the $\mathsf{T}_{\mathtt{X}}$-coalgebras $\mathbb{S}_1 = (S_1, \sigma_{m_{V_1}})$ and $\mathbb{S}_2 = (S_2, \sigma_{m_{V_2}})$ are $\overline{\mathsf{T}}_{\mathtt{X}}$-bisimilar. It is clear that this definition generalises the notion of bisimilarity between Kripke models.

Using the appropriate notion of bisimulation for T-models, we obtain an important property of our coalgebraic fixed point logic: truth is bisimulation invariant.

2.4.10. Fact. [Mos99, Ven06] Let T-models $\mathbb{S}_1 = (S_1, \sigma_1, V_1)$ and $\mathbb{S}_2 = (S_2, \sigma_2, V_2)$ be two T-models over $\mathtt{X}$. Then for any $\overline{\mathsf{T}}$-bisimulation $R : S_1 \nrightarrow S_2$, any pair $(s_1, s_2) \in R$, and any $\mu\mathtt{ML}_{\nabla}^{\overline{\mathsf{T}}}$-formula φ it holds that:

$$(\mathbb{S}_1, s_1) \Vdash \varphi \text{ iff } (\mathbb{S}_2, s_2) \Vdash \varphi.$$

2.4.2 Coalgebraic μ-calculus via predicate liftings

As we discussed in Chapter 1, apart from a non-standard syntax, a drawback of the ∇-based logic is that, due to its dependence on relation liftings for T, it only works properly for weak pullback-preserving functors. As an alternative, Dirk Pattinson [Pat03b] and others developed a coalgebraic modal formalism, based on a standard syntax and semantics involving *predicate liftings*, that works for coalgebras of arbitrary type. In the following, we formally define this notion and recall from [FLV10] the syntax and semantics of the coalgebraic fixpoint logic $\mu\mathtt{ML}_\Lambda$ for a given set Λ of predicate liftings for T.

2.4.11. Definition. Given a set functor T, an *n-place predicate lifting* for T is a natural transformation

$$\lambda : \mathcal{Q}(-)^n \to \mathcal{Q} \circ \mathsf{T},$$

where $\mathcal{Q}(-)^n$ denotes the n-fold product of $\mathcal{Q}$ with itself. A predicate lifting λ is said to be *monotone* if

$$\lambda_X(Y_1, ..., Y_n) \subseteq \lambda_X(Z_1, ..., Z_n),$$

whenever $Y_i \subseteq Z_i$ for each i. The *Boolean dual* λ^d of λ is defined by

$$(Z_1, ..., Z_n) \mapsto \mathsf{T}X \setminus (\lambda_X(X \setminus Z_1, ..., X \setminus Z_n)).$$

2.4.12. Definition. Given a set functor T and a set of monotone predicate liftings Λ for T, the language $\mu\mathtt{ML}_{\Lambda}^{\mathsf{T}}$ of the coalgebraic μ-calculus based on Λ is defined as follows.

$$\varphi ::= p \mid \varphi \vee \varphi \mid [\lambda](\varphi_1, ..., \varphi_n) \mid \neg\varphi \mid \mu p.\varphi$$

where $p \in \mathtt{X}$, $[\lambda]$ is a modality associated with a monotone n-placed predicate lifting $\lambda \in \Lambda$, and, in $\mu p.\varphi$, no free occurrence of the variable p is in the scope of an odd number of negations. Note that having the negation in the syntax enables us to define the connectives $\wedge$, $\top$, $\bot$ and the greatest fixpoint operator ν.

The semantics of formulas on pointed T-models is defined by induction on the complexity of formulas. Here we only give the clause for the modalities, the other cases are standard.

2.4.13. Definition. Given a pointed T-model $(\mathbb{S}, s) = (S, \sigma, V, s)$ we set:

$$(\mathbb{S}, s) \Vdash [\lambda](\varphi_1, ..., \varphi_n) \quad \text{iff} \quad \sigma(s) \in \lambda_S([\![\varphi_1]\!], ..., [\![\varphi_n]\!]),$$

where $[\![\varphi_i]\!] = \{t \in S \mid (\mathbb{S}, t) \Vdash \varphi_i\}$.

As mentioned in Chapter 1 and earlier in this section, many well-known variations of fixpoint logics are instances of the definition of $\mu\mathtt{ML}_{\Lambda}$ for a properly chosen set of predicate liftings Λ.

2.4.14. Example. Recall from Example 2.3.6 that Kripke frames are coalgebras for the powerset functor $\mathcal{P}$. The semantics of the standard modality $\Diamond$ (see Example 1.1.1) is obtained from the unary predicate lifting $\Diamond^{\mathcal{P}}$ defined by setting for each $Z \subseteq X$:

$$\Diamond_X^{\mathcal{P}}(Z) := \{\alpha \in \mathcal{P}X \mid \alpha \cap Z \neq \emptyset\}.$$

The language and semantics $\mu\mathtt{ML}_{\{\Diamond\}}^{\mathcal{P}}$ is then equivalent to the standard modal μ-calculus $\mu\mathtt{ML}$. The dual of $\Diamond^{\mathcal{P}}$ is denoted by $\Box^{\mathcal{P}}$ as usual, and is defined for each $Z \subseteq X$ as follows:

$$\Box_X^{\mathcal{P}}(Z) := \{\alpha \in \mathcal{P}X \mid \alpha \subseteq Z\}.$$

2.4.15. Example. Recall from Example 2.3.10 that monotone frames are coalgebras for the monotone neighbourhood functor $\mathcal{M}$. The semantics of the usual box modality for monotone modal logic (see Example 1.1.3) is obtained by the unary predicate lifting, which we denote by the symbol $\Box^{\mathcal{M}}$ and define by setting, for $Z \subseteq X$:

$$\Box_X^{\mathcal{M}}(Z) := \{\alpha \in \mathcal{M}X \mid Z \in \alpha\}.$$

The language and semantics of $\mu\mathtt{ML}_{\{\Box\}}^{\mathcal{M}}$ is then equivalent to the *monotone μ-calculus*, denoted by $\mu\mathtt{MML}$.

2.4.16. Example. Recall from Example 2.3.7 that multiset frames are coalgebras for the bag functor $\mathcal{B}$. For each k the semantics of the graded modality $\Diamond^k$ (see Example 1.1.2) is obtained by the unary predicate lifting $\Diamond^k$ defined as follows: For all $k \in \omega$, all sets S and all subsets U of S we have:

$$\Diamond^k{}_S(U) := \{f : S \to \omega \mid k \geq \sum_{x \in U} f(x)\}.$$

The language and semantics of $\mu\mathtt{ML}^{\mathcal{B}}_{\{\Diamond^k \mid k \in \mathbb{N}\}}$ is then equivalent to the fixpoint extension of the graded modal logic we refer to as the *graded μ-calculus*.

2.4.3 Coalgebraic automata theory

In Chapter 1 we discussed how the μ-calculus benefits from its connection to areas like automata theory. Many interesting results about $\mu\mathtt{ML}$ rely on automata-theoretic proofs. For instance, the uniform interpolation theorem for $\mu\mathtt{ML}$ [DH00], the fact that $\mathtt{MSO}$ and the μ-calculus have the same expressive power on ranked trees [EJ91a], and the completeness theorem for $\mu\mathtt{ML}$ [Wal00] have been proved by applying automata-theoretic machinery. Inspired by these examples, one of the main tools to get the results of this thesis is the notion of a coalgebraic *modal automaton*. Before going through the definitions, we note that this section is included for the sake of completeness of the preliminaries and may seem a bit condensed. We will elaborate all notions and concepts discussed here in individual chapters.

We recall the notion of a coalgebraic modal automaton by first defining the *one-step* language which has been referred to as "rank-1" logic in the literature (see [Pat03b, CP04, SP10]) .

One-step language

∇-setting: Given a set A, we define the set $\mathtt{LF}(A)$ of lattice terms over A through the following grammar:

$$\pi ::= \bot \mid \top \mid a \mid \pi \wedge \pi \mid \pi \vee \pi,$$

where $a \in A$. Given two sets $\mathtt{X}$, A and a functor T, we define the set $\mathtt{1ML}_{\nabla}(\mathtt{X}, A)$ of *modal one-step formulas* over A with respect to $\mathtt{X}$ inductively by

$$\varphi ::= \bot \mid \top \mid p \mid \neg p \mid \nabla\beta \mid \varphi \wedge \varphi \mid \varphi \vee \varphi,$$

with $p \in \mathtt{X}$ and $\beta \in \mathsf{T}_\omega\mathtt{LF}(A)$.

One-step formulas are interpreted over one-step models.

2.4.17. Definition. Fix sets $\mathtt{X}$ and A and a functor T, and let $\mathsf{T}_{\mathtt{X}}$ denote the functor $\mathcal{P}\mathtt{X} \times \mathsf{T}-$ as in Remark 2.4.6. A *one-step* $\mathsf{T}_{\mathtt{X}}$*-model* over set A of variables is a triple (S, ξ, V) where S is any set, $\xi \in \mathsf{T}_{\mathtt{X}}S$ and $V : A \to \mathcal{P}S$ is a valuation on A.

Any valuation $V : A \to \mathcal{P}S$ can be extended to a meaning function $[\![-]\!]^0_V$: $\mathtt{LF}(A) \to \mathcal{P}S$ in a natural way defined as: $[\![a]\!]^0_V := \{s \in S \mid s \in V(a)\}$, $[\![\top]\!]^0_V := S$, $[\![\bot]\!]^0_V = \emptyset$ and standard clauses for $\wedge$ and $\vee$. We write $\mathbb{S}, s \Vdash^0_V \varphi$ to indicate $s \in [\![\varphi]\!]^0_V$.

The meaning function $[\![-]\!]^0_V$ induces a map $[\![-]\!]^1_V : \mathtt{1ML}_\nabla(\mathtt{X}, A) \to \mathcal{P}\mathsf{T}_\mathtt{X}S$ interpreting one-step formulas as subsets of $\mathsf{T}_\mathtt{X}S$. Before giving the definition of $[\![-]\!]^1_V$ we recall that every $\xi \in \mathsf{T}_\mathtt{X}S$ is of the form $(\mathtt{Y}, \tau) \in \mathcal{P}\mathtt{X} \times \mathsf{T}S$.

Going back to the map $[\![-]\!]^1_V$, it has the usual clauses for conjunction and disjunction, and the following clauses for the propositional letters and the modal operator:

- $\xi = (\mathtt{Y}, \tau) \in [\![p]\!]^1_V$ iff $p \in \mathtt{Y}$
- $\xi = (\mathtt{Y}, \tau) \in [\![\neg p]\!]^1_V$ iff $p \notin \mathtt{Y}$
- $\xi = (\mathtt{Y}, \tau) \in [\![\nabla\beta]\!]^1_V$ iff $(\tau, \beta) \in \overline{\mathsf{T}}(\Vdash^0_V)$

We write $\mathbb{S}, \xi \Vdash^1_V \varphi$ to indicate $\xi \in [\![\varphi]\!]^1_V$, and refer to this relation as the *one-step semantics*.

Predicate lifting setting: Given a set Λ of predicate liftings λ, and two disjoint sets A, $\mathtt{X}$ of variables, we define the set $\mathtt{1ML}_\Lambda(\mathtt{X}, A)$ of *modal one-step formulas* over a set A with respect to $\mathtt{X}$ by the following grammar:

$$\varphi ::= \bot \mid \top \mid p \mid [\lambda](\pi_1, ..., \pi_n) \mid [\lambda^d](\pi_1, ..., \pi_n) \mid \varphi \vee \varphi \mid \varphi \wedge \varphi,$$

where $p \in \mathtt{X}$, $\lambda \in \Lambda$ and $\pi_1, ..., \pi_n \in \mathtt{LF}(A)$ where $\mathtt{LF}(A)$ is the set of lattice formulas over A introduced above.

Given a one-step T-model (S, ξ, V), the meaning function $[\![-]\!]^1_V : \mathtt{1ML}_\Lambda(\mathtt{X}, A) \to \mathcal{P}\mathsf{T}_\mathtt{X}S$ is defined for modal formulas as follows. Semantics of the other connectives is defined standardly, similar to the ∇-setting.

- $\xi = (\mathtt{Y}, \tau) \in [\![[\lambda](\pi_1, ..., \pi_n)]\!]^1_V$ iff $\tau \in \lambda_S([\![\pi_1]\!]^0_V, \cdots [\![\pi_n]\!]^0_V)$
- $\xi = (\mathtt{Y}, \tau) \in [\![[\lambda^d](\pi_1, ..., \pi_n)]\!]^1_V$ iff $\tau \in \mathsf{T}S \setminus \lambda_S(S \setminus [\![\pi_1]\!]^0_V, \cdots S \setminus [\![\pi_n]\!]^0_V)$.

For example, in the case of the powerset functor $\mathcal{P}$, the semantics of the modality induced by the unary predicate lifting $\Diamond$ of Example 2.4.14 is given as follows:

$$\begin{aligned} \xi = (\mathtt{Y}, \tau) \in [\![\Diamond(\pi)]\!]^1_V \quad &\text{iff} \quad \tau \in \Diamond_S([\![\pi]\!]^0_V) \\ &\text{iff} \quad \tau \cap [\![\pi]\!]^0_V \neq \emptyset. \end{aligned}$$

We are now ready to define modal automata corresponding to the logics $\mu\mathtt{ML}^{\overline{\mathsf{T}}}_\nabla$ and $\mu\mathtt{ML}_\Lambda$:

2.4.18. DEFINITION. Let $\mathtt{X}$ be a finite set of variables. A *modal* $\mathtt{X}$*-automaton* over a one-step language $1\mathsf{L}(\mathtt{X}, A) \in \{1\mathtt{ML}_\nabla(\mathtt{X}, A), 1\mathtt{ML}_\Lambda(\mathtt{X}, A)\}$ is a quadruple $\mathbb{A} = (A, \Theta, \Omega, a_I)$ such that A is a finite set of *states*, $\Theta : A \to 1\mathsf{L}(\mathtt{X}, A)$ is the *transition map* of $\mathbb{A}$, $\Omega : A \to \omega$ is the *priority map* of $\mathbb{A}$, and a_I is the *initial state*.

Modal $\mathtt{X}$-automata run on T-models over the set $\mathtt{X}$, and acceptance is defined in terms of a two player game, the *acceptance game*.

2.4.19. DEFINITION. Let $\mathbb{A} = (A, \Theta, \Omega, a_I)$ be a modal $\mathtt{X}$-automaton and let $\mathbb{S} = (S, \sigma_m)$ be a T-model with $\sigma_m : S \to \mathcal{P}\mathtt{X} \times \mathsf{T}S$. The associated acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})$ is the parity game given by Table 2.2:

Position	Player	Admissible moves	Priority
$(a, s) \in A \times S$	$\exists$	$\{U : A \to \mathcal{P}S \mid \sigma_m(s) \in [\![\Theta(a)]\!]^1_U\}$	$\Omega(a)$
$U : A \to \mathcal{P}S$	$\forall$	$\{(b, t) \in A \times S \mid t \in U(b)\}$	0

Table 2.2: Acceptance Game (see [Ven06, FLV10])

The loser of a finite match is the player who got stuck. We declare the winner of an infinite match according to the parity condition, i.e. $\exists$ wins if the highest priority that appears infinitely often in the match is even, and $\forall$ is the winner otherwise. A pointed coalgebra $(\mathbb{S}, s_I)$ is *accepted* by the automaton $\mathbb{A}$ if (a_I, s_I) is a winning position for player $\exists$ in $\mathcal{A}(\mathbb{A}, \mathbb{S})$.

Given a weak pullback-preserving functor T, the following result from [Ven06] enables us to transfer between formulas of $\mu\mathtt{ML}^{\overline{\mathsf{T}}}_\nabla(\mathtt{X})$ and modal $\mathtt{X}$-automata.

2.4.20. FACT. There are effective procedures transforming a modal $\mathtt{X}$-automaton over $1\mathtt{ML}_\nabla(\mathtt{X}, A)$ into an equivalent $\mu\mathtt{ML}^{\overline{\mathsf{T}}}_\nabla(\mathtt{X})$-formula and vice versa.

The corresponding result for the predicate lifting setting which states that modal $\mathtt{X}$-automata can be used to present formulas of $\mu\mathtt{ML}^{\mathsf{T}}_\Lambda$ is proved in [FLV10]:

2.4.21. FACT. There are effective procedures transforming a modal $\mathtt{X}$-automaton over $1\mathtt{ML}_\Lambda(\mathtt{X}, A)$ into an equivalent $\mu\mathtt{ML}^{\mathsf{T}}_\Lambda(\mathtt{X})$-formula and vice versa.

Chapter 3

Uniform Interpolation

The main objective of this chapter is to study *Uniform Interpolation* for coalgebraic fixpoint logic $\mu\mathtt{ML}^{L}_{\nabla}$. Uniform interpolation is a stronger version of another type of interpolation that is known as *Craig Interpolation* [Cra57]. Roughly stated, Craig Interpolation says that if a formula φ_1 implies a formula φ_2, then there is a formula ψ, called an *interpolant*, which may only use propositional variables that appear both in φ_1 and in φ_2, such that φ_1 implies ψ, and ψ implies φ_2. A logic has uniform interpolation if the interpolant ψ for the formulas φ_1 and φ_2 depends only on φ_1 and the language that φ_2 shares with φ_1 (rather than on φ_2 itself). Although it is easy to show that classical propositional logic has uniform interpolation, not many logics have this property, for instance in the same paper where Henkin introduced this type of interpolation (the name has been coined many years later by Pitts [Pit92]) he proved that first-order logic has interpolation, but it does not enjoy the uniform version [Hen63].

Starting with the seminal work of Pitts [Pit92] who proved that intuitionistic logic has uniform interpolation, the study of this property for different logics has been actively pursued by various authors. In modal logic, Shavrukov [Sha94] proved that the Gödel-Löb logic **GL** has uniform interpolation. Subsequently, Ghilardi [Ghi95] and Visser [Vis96b] independently established the property for the modal logic **K**, while [GZ95] contains negative results for the modal logic **S4**. In the theory of modal fixpoint logic, as we already mentioned in Section 1.4.1, D'Agostino and Hollenberg proved that the modal μ-calculus has uniform interpolation [DH00].

In this chapter we generalise the result by D'Agostino and Hollenberg [DH00] to a wider class of fixpoint logics, including the *monotone μ-calculus*, which is the extension of monotone modal logic with fixpoint operators. More precisely, we work with ∇-based μ-calculus and restrict our attention to set functors that preserve finite sets and admit a certain type of relation lifting called a *quasi-functorial lax extension*. This class includes all functors that preserve weak pullbacks (such as the Kripke functor), but also the *monotone neighbourhood functor* $\mathcal{M}$ which is not weak pullback-preserving. Another nice property of this class of functors is that it is closed under various natural operations on functors.

The main result of this chapter is Theorem 3.7.1, where we prove that for a functor T with a quasi-functorial lax extension L, the coalgebraic fixpoint logic $\mu\mathtt{ML}^{L}_{\nabla}$ enjoys uniform interpolation. Our proof follows the semantic approach and is built on definability of bisimulation quantifiers. As usual in the setting of modal fixpoint logic, it is based on the link between logic and automata. We follow the automata-theoretic approach by D'Agostino and Hollenberg. That is we define a class of non-deterministic parity automata that closely correspond to our language. Our main technical result is Theorem 3.5.3 which shows that the class of coalgebraic automata associated with our logic are closed under projection. From this we can easily derive the definability of bisimulation quantifiers and consequently the proof of the Uniform Interpolation.

We start this chapter by formally introducing quasi-functorial lax extensions as special relation liftings and stating some of their properties.

3.1 Special relation liftings

In this section we define lax extensions, which are relation liftings (see Section 2.3.4.1) satisfying certain conditions that make them well-behaved in the context of coalgebra. Most of the results presented in this section are from [Mar11], and for the sake of completeness, we included some of the proofs from [Mar11] as well.

3.1.1. Definition. A relation lifting L for a functor T is called a *lax extension* if it satisfies, for all relations $R, R' : X \not\to Z$ and $S : Z \not\to Y$ and all functions $f : X \to Z$:
(L1) $R' \subseteq R$ implies $LR' \subseteq LR$;
(L2) $LR \,;\, LS \subseteq L(R \,;\, S)$;
(L3) $\mathsf{T}f \subseteq Lf$.
We say that a lax extension L *preserves diagonals* if it additionally satisfies:
(L4) $L\Delta_X \subseteq \Delta_{\mathsf{T}X}$.

Recall from Definition 2.3.26 that every relation lifting L for a functor T preserves inverse of relations, i.e., for all relations R we have $L(R^\circ) = (LR)^\circ$.

The following Proposition [Mar11, Proposition 3.13] summarises how the conditions (L1), (L2) and (L3) of a lax extension L directly entail useful properties of L-bisimulations.

3.1.2. Proposition. *For a lax extension L of T and T-coalgebras $\mathbb{S}$, $\mathbb{Q}$ and $\mathbb{Y}$ it holds that:*

(1) The graph of every coalgebra morphism f from $\mathbb{S}$ to $\mathbb{Q}$ is an L-bisimulation between $\mathbb{S}$ and $\mathbb{Q}$.

(2) *If $R : S \nrightarrow Q$ and $Z : Q \nrightarrow Y$ are L-bisimulations between $\mathbb{S}$ and $\mathbb{Q}$ respectively $\mathbb{Q}$ and $\mathbb{Y}$, then $(R\,;Z) : S \nrightarrow Y$ is an L-bisimulation between $\mathbb{S}$ and $\mathbb{Y}$.*

(3) *Every arbitrary union of L-bisimulations between $\mathbb{S}$ and $\mathbb{Q}$ is again an L-bisimulation between $\mathbb{S}$ and $\mathbb{Q}$.*

In Definition 3.1.1, we require only the inclusion (L4) for a lax extension to preserve diagonals. This is justified because condition (L3) together with the observation that $\Delta_{\mathsf{T}X} = \mathrm{id}_{\mathsf{T}X} = \mathsf{T}(\mathrm{id}_X)$ implies that $\Delta_{\mathsf{T}X} \subseteq L\Delta_X$. The proof of this is in the following Proposition which states some basic properties of lax extensions [Mar11, Proposition 3.10.].

3.1.3. PROPOSITION. *If L is a lax extension of T then for all functions $f : X \to Z$, $g : Y \to Z$ and all relations $R : X \nrightarrow Z$ and $S : Z \nrightarrow Y$ the following properties hold:*

(1) $\Delta_{\mathsf{T}X} \subseteq L\Delta_X$;

(2) $\mathsf{T}f\,;LS = L(f\,;S)$ *and* $LR\,;(\mathsf{T}g)^\circ = L(R\,;g^\circ)$;

and if L preserves diagonals then

(3) $L\Delta_X = \Delta_{\mathsf{T}X}$ *and* $Lf = \mathsf{T}f$;

(4) $\mathsf{T}f\,;(\mathsf{T}g)^\circ = L(f\,;g^\circ)$.

Proof:
For (1) recall that we identify a function with its graph. So we have that $\Delta_X = \mathrm{id}_X$ and we can calculate:

$$\begin{array}{rcll} \Delta_{\mathsf{T}X} = \mathrm{id}_{\mathsf{T}X} & = & \mathsf{T}\mathrm{id}_X & \mathsf{T} \text{ is a functor} \\ & \subseteq & L(\mathrm{id}_X) = L\Delta_X & \text{(L3)} \end{array}$$

The $\subseteq$-inclusion of $\mathsf{T}f\,;LS = L(f\,;S)$ in (2) holds because:

$$\mathsf{T}f\,;LS \subseteq Lf\,;LS \subseteq L(f\,;S),$$

where the first inclusion is given by (L3) and second one is given by (L2).

For the $\supseteq$-inclusion first note that $\Delta_{\mathsf{T}X} = \mathrm{id}_{\mathsf{T}X} \subseteq \mathsf{T}f\,;(\mathsf{T}f)^\circ$ for all maps $f : X \to Y$. Now consider the following argument:

$$\begin{array}{rcll} L(f\,;S) & \subseteq & \mathsf{T}f\,;(\mathsf{T}f)^\circ\,;L(f\,;S) & \Delta_{\mathsf{T}X} \subseteq \mathsf{T}f\,;(\mathsf{T}f)^\circ \\ & \subseteq & \mathsf{T}f\,;(Lf)^\circ\,;L(f\,;S) & \text{(L3)} \\ & = & \mathsf{T}f\,;Lf^\circ\,;L(f\,;S) & \text{preservation of converses} \\ & \subseteq & \mathsf{T}f\,;L(f^\circ\,;f\,;S) & \text{(L2)} \\ & \subseteq & \mathsf{T}f\,;LS. & \text{(L4) and } f^\circ\,;f \subseteq \Delta_Y \end{array}$$

For $LR\,;(\mathsf{T}g)^\circ = L(R\,;g^\circ)$ we can use the same argument and the fact that L preserves converses.

For (3) and (4) first notice that if L preserves diagonals then because of (L4) and (1) we have $L\Delta_X = \Delta_{\mathsf{T}X}$. The equation $\mathsf{T}f = Lf$ holds because of:

$$\begin{array}{rcll} Lf & = & L(f\,;\Delta_X) & \\ & = & \mathsf{T}f\,;L\Delta_X & (2) \\ & = & \mathsf{T}f. & L\Delta_X = \Delta_{\mathsf{T}X} \end{array}$$

Finally item (4) holds because of the following:

$$\begin{array}{rcll} \mathsf{T}f\,;(\mathsf{T}g)^\circ & = & \mathsf{T}f\,;L\Delta_X\,;(\mathsf{T}g)^\circ & \Delta_{\mathsf{T}X} = L\Delta_X \\ & = & L(f\,;\Delta_X\,;g^\circ) & (2) \text{ twice} \\ & = & L(f\,;g^\circ). & \end{array}$$

□

The next proposition [Mar11, Proposition 3.12.] shows that for a lax extension of an inclusion-preserving functor it does not really matter what the domain and codomain of a relation are.

3.1.4. Proposition. *For any lax extension L of an inclusion-preserving functor T we have that for all relations $R : X \nrightarrow Y$ and sets $X' \subseteq X$ and $Y' \subseteq Y$*

$$L(R{\restriction}_{X'\times Y'}) = (LR){\restriction}_{\mathsf{T}X'\times\mathsf{T}Y'}.$$

Proof:
We first rewrite the restriction of relation R as $R{\restriction}_{X'\times Y'} = (i_{X',X}\,;R\,;i^\circ_{Y',Y})$ where $i_{X',X} : X' \hookrightarrow X$ and $i_{Y',Y} : Y' \hookrightarrow Y$ are the inclusion maps. Then it follows that:

$$\begin{array}{rcll} L(R{\restriction}_{X'\times Y'}) & = & L(i_{X',X}\,;R\,;i^\circ_{Y',Y}) & \\ & = & \mathsf{T}i_{X',X}\,;LR\,;(\mathsf{T}i_{Y',Y})^\circ & \text{Proposition 3.1.3 (2)} \\ & = & i_{\mathsf{T}X',\mathsf{T}X}\,;LR\,;i^\circ_{\mathsf{T}Y',\mathsf{T}Y} & \mathsf{T} \text{ preserves inclusions} \\ & = & (LR){\restriction}_{\mathsf{T}X'\times\mathsf{T}Y'}. & \end{array}$$

□

Now we define the key concept of this chapter namely quasi-functorial lax extensions.

3.1.5. Definition. We call a lax extension L of a functor T *functorial*, if it distributes over composition, i.e., if $LR\,;LS = L(R\,;S)$, and *quasi-functorial*, if

$$LR\,;LS = L(R\,;S) \cap (\mathsf{dom}(LR) \times \mathsf{ran}(LS)), \tag{3.1}$$

for all relations $R : X \nrightarrow Z$ and $S : Z \nrightarrow Y$.

Recall from Section 2.3.1 and the definition of $\mathsf{dom}(LR) \subseteq \mathsf{T}X$ that $\tau \in \mathsf{dom}(LR)$ iff there is a $\gamma_R \in \mathsf{T}Z$ such that $(\tau, \gamma_R) \in LR$. Similarly $\nu \in \mathsf{ran}(LS) \subseteq \mathsf{T}Y$ iff there is a $\gamma_S \in \mathsf{T}Z$ such that $(\gamma_S, \nu) \in LS$. Hence the $\subseteq$-inclusion of (3.1) holds for any lax extension because of $(L2)$. So the $\supseteq$-inclusion is the actual substantial requirement. It is equivalent to the condition that for all $(\tau, \nu) \in L(R\,;\,S)$ if there is a $\gamma_R \in \mathsf{T}Z$ such that $(\tau, \gamma_R) \in LR$ and there is a $\gamma_S \in \mathsf{T}Z$ such that $(\gamma_S, \nu) \in LS$, then there is a $\gamma \in \mathsf{T}Z$ such that $(\tau, \gamma) \in LR$ and $(\gamma, \nu) \in LS$.

3.1.6. Example. The Barr extension $\overline{\mathsf{T}}$ (see Example 2.3.27) for a functor T that preserves weak pullbacks is functorial. In particular it satisfies $\overline{\mathsf{T}}R\,;\,\overline{\mathsf{T}}S = \overline{\mathsf{T}}(R\,;\,S)$ for all relations R and S. Clearly this implies that T is also quasi-functorial.

A key example which shows that the class of functors with a quasi-functorial lax extension is strictly bigger than the class of weak pullback-preserving functors is the monotone neighbourhood functor $\mathcal{M}$ [Mar11, Example 3.11.(ii)].

3.1.7. Example. The relation lifting $\widetilde{\mathcal{M}}$ as defined in Example 2.3.33 for the monotone neighbourhood functor $\mathcal{M}$, is quasi-functorial.

Proof:
It is easy to check that $\widetilde{\mathcal{M}}$ is a lax extension that preserves diagonals. So in the following we will just give the proof for the quasi-functoriality of $\widetilde{\mathcal{M}}$:

Take any two relations $R : X \nrightarrow Z$ and $S : Z \nrightarrow Y$. We need to show that for all $(\tau, \nu) \in \widetilde{\mathcal{M}}(R\,;\,S)$, if there are γ_R and γ_S in $\mathcal{M}Z$, with $(\tau, \gamma_R) \in \widetilde{\mathcal{M}}R$ and $(\gamma_S, \nu) \in \widetilde{\mathcal{M}}S$, then there is a $\gamma \in \mathcal{M}Z$ such that $(\tau, \gamma) \in \widetilde{\mathcal{M}}R$ and $(\gamma, \nu) \in \widetilde{\mathcal{M}}S$.

From the assumption that $(\tau, \gamma_R) \in \widetilde{\mathcal{M}}R \subseteq \overrightarrow{\mathcal{P}}\overleftarrow{\mathcal{P}}R$, we get that:

$$\forall A \in \tau, \exists U_A \in \gamma_R \text{ s.t. } (A, U_A) \in \overleftarrow{\mathcal{P}}R.$$

Similarly from $(\tau, \nu) \in \widetilde{\mathcal{M}}(R\,;\,S) \subseteq \overrightarrow{\mathcal{P}}\overleftarrow{\mathcal{P}}(R\,;\,S)$ we get that:

$$\forall A \in \tau, \exists V_A \in \nu \text{ s.t. } (A, V_A) \in \overleftarrow{\mathcal{P}}(R\,;\,S),$$

and from $(\gamma_S, \nu) \in \widetilde{\mathcal{M}}S \subseteq \overleftarrow{\mathcal{P}}\overrightarrow{\mathcal{P}}S$ and $(\tau, \nu) \in \widetilde{\mathcal{M}}(R\,;\,S) \subseteq \overleftarrow{\mathcal{P}}\overrightarrow{\mathcal{P}}(R\,;\,S)$ we have:

$$\forall B \in \nu, \exists U_B \in \gamma_S \text{ and } \exists V_B \in \tau \text{ s.t. } (U_B, B) \in \overrightarrow{\mathcal{P}}S \text{ and } (V_B, B) \in \overrightarrow{\mathcal{P}}(R\,;\,S).$$

From $(A, V_A) \in \overleftarrow{\mathcal{P}}(R\,;\,S)$ it follows that:

$$\forall v \in V_A, \exists a_v \in A \text{ s.t. } (a_v, v) \in R\,;\,S.$$

So there is a $z_v \in Z$ such that $(a_v, z_v) \in R$ and $(z_v, v) \in S$.

Now we define, for every $A \in \tau$, the following set $U'_A \subseteq Z$:

$$U'_A := U_A \cup \{z_v \in Z \mid v \in V_A\}.$$

We claim that $(A, U'_A) \in \overleftarrow{\mathcal{P}}R$. To prove this, take $u \in U'_A$, we need to show that there exists $t \in A$ such that $(t, u) \in R$. Since $u \in U'_A$, we have two cases:

(i) $u \in U_A$, then from $(A, U_A) \in \overleftarrow{\mathcal{P}}R$ we are done.

(ii) $u \in \{z_v \in Z \mid v \in V_A\}$. In this case from the definition of z_v we have that there exists $a_v \in A$ such that $(a_v, z_v) \in R$.

On the other hand because $\forall v \in V_A, (z_v, v) \in S$, we have that $(U'_A, V_A) \in \overleftarrow{\mathcal{P}}S$.

We can similarly define for every $B \in \nu$ a set $U'_B \subseteq Z$ such that:

$$(U'_B, B) \in \overrightarrow{\mathcal{P}}S \text{ and } (V_B, U'_B) \in \overrightarrow{\mathcal{P}}R.$$

Now we are ready to introduce $\gamma \in \mathcal{M}Z$:

$$\gamma := \{U \subseteq Z \mid \exists A \in \tau \text{ with } U'_A \subseteq U \text{ or } \exists B \in \nu \text{ with } U'_B \subseteq U\}.$$

It is clear that γ is upward closed, so $\gamma \in \mathcal{M}Z$. It is left to show that $(\tau, \gamma) \in \widetilde{\mathcal{M}}R$ and $(\gamma, \nu) \in \widetilde{\mathcal{M}}S$. We have that $(\tau, \gamma) \in \widetilde{\mathcal{M}}R$ iff $(\tau, \gamma) \in \overrightarrow{\mathcal{P}}\overleftarrow{\mathcal{P}}R$ and $(\tau, \gamma) \in \overleftarrow{\mathcal{P}}\overrightarrow{\mathcal{P}}R$. For the proof of $(\tau, \gamma) \in \overrightarrow{\mathcal{P}}\overleftarrow{\mathcal{P}}R$ note that for every $A \in \tau$ we have that $(A, U'_A) \in \overleftarrow{\mathcal{P}}R$. For $(\tau, \gamma) \in \overleftarrow{\mathcal{P}}\overrightarrow{\mathcal{P}}R$, pick a $U \in \gamma$. Then from the definition of γ it follows that $U'_A \subseteq U$ for some $A \in \tau$ or $U'_B \subseteq U$ for some $B \in \nu$. In the first case consider that $U_A \subseteq U'_A \subseteq U$ and by the assumption $(\tau, \gamma_R) \in \widetilde{\mathcal{M}}R \subseteq \overleftarrow{\mathcal{P}}\overrightarrow{\mathcal{P}}R$ we get that there exists $T \in \tau$ such that $(T, U_A) \in \overrightarrow{\mathcal{P}}R$, so $(T, U) \in \overrightarrow{\mathcal{P}}R$. The case for $(\gamma, \nu) \in \widetilde{\mathcal{M}}S$ can be checked using a similar argument. □

The following Proposition states some of the basic properties of quasi-functorial lax extensions concerning *fullness* of relations. Recall that a relation $R : X \nrightarrow Z$ is full on X(respectively Z) if $\mathsf{dom}(R) = X$ (respectively $\mathsf{ran}(R) = Z$).

3.1.8. Proposition. *Let* T *be a set functor and let* L *be a quasi-functorial lax extension for* T*. Then we have:*

(1) *L preserves fullness of relations :*
If $R : X \nrightarrow Z$ is full on both sides, then so is $LR : \mathsf{T}X \nrightarrow \mathsf{T}Z$.

(2) *If $R : X \nrightarrow Z$ is full on X and $i : Z \hookrightarrow Y$ is the inclusion map between Z and a given Y such that $Z \subseteq Y$, then $L(R\,;i)$ is full on $\mathsf{T}X$.*

Proof:

For the proof of (1) consider the following argument: Let $\pi_X : R \to X$ and

$\pi_Z : R \to Z$ denote the projection maps. Since $R = (\pi_X)^\circ ; \pi_Z$ and $LR = L((\pi_X)^\circ ; \pi_Z)$, from quasi-functoriality of L it follows that:

$$L(\pi_X)^\circ ; L\pi_Z = L((\pi_X)^\circ ; \pi_Z) \cap (\mathsf{dom}(L\pi_X)^\circ \times \mathsf{ran}(L\pi_Z)).$$

But since $R = (\pi_X)^\circ ; \pi_Z$ is full on both sides, the projection maps π_X and π_Z are surjective. It then follows that $\mathsf{T}\pi_X : \mathsf{T}R \to \mathsf{T}X$ and $\mathsf{T}\pi_Z : \mathsf{T}R \to \mathsf{T}Z$ are surjective, because set functors preserve surjectiveness. So $\mathsf{ran}(\mathsf{T}\pi_X) = \mathsf{dom}(\mathsf{T}\pi_X)^\circ = \mathsf{T}X$ and $\mathsf{ran}(\mathsf{T}\pi_Z) = \mathsf{T}Z$. Consequently we have:

$$L((\pi_X)^\circ ; \pi_Z) \cap \mathsf{T}X \times \mathsf{T}Z = L(\pi_X)^\circ ; L\pi_Z,$$

which implies $L((\pi_X)^\circ ; \pi_Z) = L(\pi_X)^\circ ; L\pi_Z$, since $L((\pi_X)^\circ ; \pi_Z) \subseteq \mathsf{T}X \times \mathsf{T}Z$.

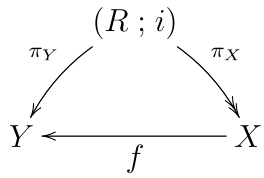
Hence in order to prove fullness of $LR = L((\pi_X)^\circ ; \pi_Z)$ on $\mathsf{T}X$ and $\mathsf{T}Z$ it is sufficient to prove that $L(\pi_X)^\circ ; L\pi_Z : \mathsf{T}X \nrightarrow \mathsf{T}Z$ is full on $\mathsf{T}X$ and $\mathsf{T}Z$. But we are done since:

$$\begin{array}{rcll} \mathsf{T}X & = & \mathsf{dom}((\mathsf{T}\pi_X)^\circ ; (\mathsf{T}\pi_Z)) & \mathsf{T}\pi_X \text{ is surjective} \\ & \subseteq & \mathsf{dom}(L(\pi_X)^\circ ; (L\pi_Z)) & (L3) \end{array}$$

and,

$$\begin{array}{rcll} \mathsf{T}Z & = & \mathsf{ran}((\mathsf{T}\pi_X)^\circ ; (\mathsf{T}\pi_Z)) & \mathsf{T}\pi_Z \text{ is surjective} \\ & \subseteq & \mathsf{ran}(L(\pi_X)^\circ ; (L\pi_Z)) & (L3) \end{array}$$

To prove (2) notice that $R ; i \subseteq X \times Y$ is full on X, so by the axiom of choice there exists a map $f : X \to Y$ such that $f \subseteq (R ; i)$. Hence we get $\mathsf{T}f \subseteq Lf \subseteq L(R ; i)$, and because $\mathsf{T}f$ is full on $\mathsf{T}X$, $L(R ; i)$ is also full on $\mathsf{T}X$.



□

We will finish this section with a remark on some of the closure properties of the class of functors with a quasi-functorial lax extension:

3.1.9. Fact. The collection FQL of functors with a quasi-functorial lax extension has the following properties:
(1) The identity functor $\mathcal{I} : \mathsf{Set} \to \mathsf{Set}$ is in FQL.
(2) For each set C, the constant functor $\mathsf{C} : \mathsf{Set} \to \mathsf{Set}$ is in FQL.
(3) The product $X \mapsto \mathsf{T}_1(X) \times \mathsf{T}_2(X)$ of two FQLs T_1 and T_2 is in FQL.
(4) The coproduct $X \mapsto \mathsf{T}_1(X) + \mathsf{T}_2(X)$ of two FQLs T_1 and T_2 is in FQL.

(5) The composition $X \mapsto (\mathsf{T}_1 \circ \mathsf{T}_2)(X)$ of a FQL functor T_1 and a functor T_2 which has a functorial lax extension, is in FQL.

Proof:
Items (1) to (4) are easy to check, so we will focus on the proof of item (5). Suppose that L_1 is a quasi-functorial lax extension for T_1 and L_2 is a functorial lax extension for T_2. We claim that $L_1 L_2$ is a quasi-functorial lax extension for $\mathsf{T}_1 \circ \mathsf{T}_2$. First observe that since L_1 and L_2 are lax extensions, $L_1 L_2$ is also a lax extension.

Take $(\alpha, \beta) \in L_1 L_2(R \,;\, S) \cap \mathsf{dom}(L_1 L_2 R) \times \mathsf{ran}(L_1 L_2 S)$. Hence we have:

$$(\alpha, \beta) \in L_1 L_2(R \,;\, S),$$

and

$$(\alpha, \beta) \in \mathsf{dom}(L_1(L_2 R)) \times \mathsf{ran}(L_1(L_2 S)).$$

From $(\alpha, \beta) \in L_1 L_2(R \,;\, S)$ and by functoriality of the relation lifting L_2 we get $(\alpha, \beta) \in L_1(L_2 R \,;\, L_2 S)$. Now from quasi-functoriality of L_1, together with $(\alpha, \beta) \in \mathsf{dom}(L_1 L_2 R) \times \mathsf{ran}(L_1 L_2 S)$, we have:

$$(\alpha, \beta) \in L_1(L_2 R) \,;\, L_1(L_2 S) = L_1 L_2 R \,;\, L_1 L_2 S.$$

□

3.2 Coalgebraic fixpoint logic

In this section, we recall the syntax and semantics of coalgebraic fixpoint logic, using the ∇-modality and a quasi-functorial lax extension L for a set functor T.

Syntax

3.2.1. Definition. Given a functor T and an infinite set of propositional variables $\mathtt{X}$, the language $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}(\mathtt{X})$ is defined by the following grammar:

$$\varphi ::= p \mid \neg\varphi \mid \bigwedge A \mid \bigvee A \mid \nabla\alpha \mid \mu p.\varphi$$

where p belongs to the set of propositional variables, $A \in \mathcal{P}_\omega(\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}(\mathtt{X}))$ and $\alpha \in \mathsf{T}_\omega(\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}(\mathtt{X}))$. There is a restriction on the formulation of the formula $\mu p.\varphi$, namely, every occurrence of p in φ may be in the scope (see Definition 2.4.2) of an even number of negations. Set $\top = \bigwedge \emptyset$ and $\bot = \bigvee \emptyset$, and note that having the negation in the syntax enables us to define the greatest fixpoint operator ν. We sometimes write $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$ as an abbreviation for $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}(\mathtt{X})$.

Note that later on in this chapter we may see negation and conjunction as maps $\neg : \mu\mathtt{ML}^{\mathsf{T}}_{\nabla} \to \mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$, $\varphi \mapsto \neg\varphi$ and $\bigwedge : \mathcal{P}_\omega(\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}) \to \mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$, $A \mapsto \bigwedge A$. Hence we can apply T to them and get maps $\mathsf{T}\neg$ and $\mathsf{T}\bigwedge$.

Given a set $\mathsf{Q} \subseteq \mathtt{X}$ we denote by $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}(\mathsf{Q})$ the restriction of $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}(\mathtt{X})$ to the set Q in the sense that it is the set of formulas with propositional variables from the set Q instead of $\mathtt{X}$.

3.2.2. Remark. For a given formula φ, we denote by P_φ the set of all propositional variables occurring in φ. Observe that for $\mathsf{Q}' \subseteq \mathsf{Q} \subseteq \mathtt{X}$, we have that $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}(\mathsf{Q}') \subseteq \mu\mathtt{ML}^{\mathsf{T}}_{\nabla}(\mathsf{Q})$. This can be proved by induction on the complexity of formulas in $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}(\mathsf{Q}')$.

Semantics

Given a quasi-functorial lax extension L for T, we define the semantics of formulas in $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$ over T-models (see Definition 2.4.5) as follows and refer to the logic by $\mu\mathtt{ML}^{L}_{\nabla}$:

3.2.3. Definition. Given a T-model $\mathbb{S} = (S, \sigma, m_V)$ we define the satisfaction relation $\Vdash_{\mathbb{S}}: S \nrightarrow \mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$ by induction:

$$\begin{array}{rll} s \Vdash_{\mathbb{S}} p & \text{iff} & p \in m_V(s) \\ s \Vdash_{\mathbb{S}} \neg\varphi & \text{iff} & \text{not } s \Vdash_{\mathbb{S}} \varphi \\ s \Vdash_{\mathbb{S}} \bigwedge A & \text{iff} & s \Vdash_{\mathbb{S}} \varphi \text{ for all } \varphi \in A \\ s \Vdash_{\mathbb{S}} \bigvee A & \text{iff} & s \Vdash_{\mathbb{S}} \varphi \text{ for some } \varphi \in A \\ s \Vdash_{\mathbb{S}} \nabla\alpha & \text{iff} & (\sigma(s), \alpha) \in L{\Vdash_{\mathbb{S}}} \\ s \Vdash_{\mathbb{S}} \mu p.\varphi & \text{iff} & s \in \bigcap\{X \subseteq S \mid [\![\varphi]\!]_{\mathbb{S}[p \mapsto X]} \subseteq X\}. \end{array}$$

3.2.4. Definition. Given logic $\mu\mathtt{ML}^{L}_{\nabla}$ we standardly define the relation of logical consequence $\vDash : \mu\mathtt{ML}^{L}_{\nabla} \nrightarrow \mu\mathtt{ML}^{L}_{\nabla}$ by $\varphi \vDash \varphi'$ iff:

$$s \Vdash_{\mathbb{S}} \varphi \text{ implies } s \Vdash_{\mathbb{S}} \varphi' \text{ for all states } s \text{ in any } \mathsf{T}\text{-model } \mathbb{S}.$$

The relation of logical equivalence $\equiv : \mu\mathtt{ML}^{L}_{\nabla} \nrightarrow \mu\mathtt{ML}^{L}_{\nabla}$ is defined by $\varphi \equiv \varphi'$ iff:

$$s \Vdash_{\mathbb{S}} \varphi \text{ iff } s \Vdash_{\mathbb{S}} \varphi' \text{ for all states } s \text{ in any } \mathsf{T}\text{-model } \mathbb{S}.$$

3.2.5. Example. By the following Proposition [Ven06, Proposition 5.15.], throughout this chapter we always assume that formulas of $\mu\mathtt{ML}^{L}_{\nabla}$ are clean and guarded (see Definition 2.4.3).

3.2.6. Proposition. *Every formula in $\mu\mathtt{ML}^{L}_{\nabla}$ is equivalent to some clean, guarded formula.*

3.2.7. Remark. Although the results on ∇-based μ-calculus in [Ven06] have been stated for weak pullback-preserving functors, some of the proofs do not really depend on preservation of weak pullbacks. Throughout this chapter, we will explicitly mention the cases where proofs need modification to be applied for functors with a quasi-functorial lax extension, and for other cases we simply refer to the results from [Ven06].

Bisimulation

In this section, we collect some of the bisimulation related notions that will be used later on in this Chapter.

3.2.8. Definition. The *projection* of a $\mathcal{P}\mathtt{X}$-coloured T-coalgebra (T-model) $\mathbb{S} = (S, \sigma, m_V)$ to a set $\mathsf{Q} \subseteq \mathtt{X}$ is defined as the $\mathcal{P}\mathsf{Q}$-coloured T-coalgebra $\mathbb{S}^{\mathsf{Q}} = (S, \sigma, m_V^{\mathsf{Q}})$ where $m_V^{\mathsf{Q}} : S \to \mathcal{P}\mathsf{Q}$ is given by $s \mapsto m_V(s) \cap \mathsf{Q}$.

3.2.9. Definition. Given two T-models $\mathbb{S}$ and $\mathbb{S}'$ and a set $\mathsf{Q} \subseteq \mathtt{X}$, a relation $R : S \nrightarrow S'$ is an *L_{Q}-bisimulation between $\mathbb{S}$ and $\mathbb{S}'$* if it is an L-bisimulation between $\mathbb{S}^{\mathsf{Q}} = (S, \sigma, m_V^{\mathsf{Q}})$ and $\mathbb{S}'^{\mathsf{Q}} = (S', \sigma', m_{V'}^{\mathsf{Q}})$. That is R is an L-bisimulation between the T-coalgebras (S, σ) and (S', σ') and it additionally preserves the colour of related states over Q. More precisely, for all $(s, s') \in R$ and $p \in \mathsf{Q}$ we have $p \in m_V(s)$ iff $p \in m_{V'}(s)$. Equivalently one can think of this condition as preservation of the truth of all propositional variables in Q, i.e., for all $(s, s') \in R$ and $p \in \mathsf{Q}$ we have

$$s \in V(p) \text{ iff } s' \in V'(p).$$

From this definition it is easy to see that for any $\mathsf{Q}' \subseteq \mathsf{Q} \subseteq \mathtt{X}$, if a relation R is an L_{Q}-bisimulation between T-models $\mathbb{S}$ and $\mathbb{S}'$, then it is also an $L_{\mathsf{Q}'}$-bisimulation between them.

3.2.10. Definition. Given a propositional variable $p \in \mathtt{X}$ and T-models $\mathbb{S}$ and $\mathbb{S}'$, a relation $R : S \nrightarrow S'$ is an *up-to-p $L_{\mathtt{X}}$-bisimulation* between $\mathbb{S}$ and $\mathbb{S}'$ if it is an $L_{\mathtt{X}\setminus\{p\}}$-bisimulation between them. Intuitively this means that the T-models $\mathbb{S}$ and $\mathbb{S}'$ are bisimilar if we disregard the propositional variable p. We write $\mathbb{S}, s \leftrightarroweq_p^L \mathbb{S}', s'$ to denote that s and s' are up-to-p $L_{\mathtt{X}}$-bisimilar.

Boolean dual

As we have already mentioned in the introduction of this chapter, our proof strategy for Uniform Interpolation is based on transforming formulas into automata and vice versa. The next Definition which can be seen as definability of the *Boolean dual* Δ of the modality ∇ is a crucial fact about $\mu\mathtt{ML}_{\nabla}^{L}$ which together with other properties of the logic, enables us to transform formulas to automata.

This fact will be explicitly used in the proofs of Proposition 3.5.2 and Proposition 3.6.1. A ‘boolean dual’ of a nabla formula $\nabla\alpha$ is a formula $\Delta\alpha$ such that:

$$\Delta\alpha \equiv \neg\nabla(\mathsf{T}\neg)\alpha.$$

Given a functor T, we focus on the definability of a Boolean dual for the nabla modality since the Boolean duals of other operators are standard.

It is easy to check [KV09] that in the case of the power set functor $\mathcal{P}$, defining $\Delta\Phi$ for a non-empty set $\Phi = \{\varphi_1, \varphi_2, \dots \varphi_n\}$ as:

$$\Delta\Phi := \nabla\emptyset \vee \bigvee\{\nabla\{\varphi\} \mid \varphi \in \Phi\} \vee \nabla\{\bigwedge\Phi, \top\},$$

we indeed obtain that:

$$\Delta\{\varphi_1, \varphi_2, \dots \varphi_n\} \equiv \neg\nabla\{\neg\varphi_1, \neg\varphi_2, \dots \neg\varphi_n\}.$$

In the general case, for each formula $\nabla\alpha$ we can find a set $Q(\alpha)$ such that the following definition:

$$\Delta(\alpha) := \bigvee\{\nabla\beta \mid \beta \in Q(\alpha)\},$$

provides a Boolean dual for ∇.

Note that in order to get a well-defined formula $\Delta\alpha$ from the above definition, we need the set $Q(\alpha)$ to be finite. This is the reason that in the next definition we require the functor T to preserve finite sets. See Definition 2.3.21 to recall the notion of Base for nabla formulas.

3.2.11. Definition. Let T be a functor that preserves finite sets and has a quasi-functorial lax extension L. Then for all $\alpha \in \mathsf{T}_\omega(\mu\mathtt{ML}^L_\nabla)$, we define $\Delta\alpha$ as the following formula:

$$\Delta\alpha := \bigvee\{\nabla(\mathsf{T}\bigwedge)(\Gamma) \mid \Gamma \in \mathsf{T}_\omega\mathcal{P}_\omega\text{Base}(\alpha), (\alpha, \Gamma) \notin L(\notin_{\text{Base}(\alpha)})\}.$$

The next Proposition states that $\Delta\alpha$ and $\nabla\alpha$ are actually Boolean duals. Moreover, it shows how we can push negations inside nablas and thereby decrease the modal rank at which negations occur in a formula.

3.2.12. Proposition. *For all $\alpha \in \mathsf{T}_\omega(\mu\mathtt{ML}^L_\nabla)$ we have:*

$$\neg\nabla(\mathsf{T}\neg)(\alpha) \equiv \bigvee\{\nabla(\mathsf{T}\bigwedge)(\Gamma) \mid \Gamma \in \mathsf{T}_\omega\mathcal{P}_\omega\text{Base}(\alpha), (\alpha, \Gamma) \notin L(\notin_{\text{Base}(\alpha)})\}.$$

Proof:
Fix a T-model $\mathbb{S} = (S, \sigma, V)$.
($\Longleftarrow$) For the direction from right to left: assume there is an $\Gamma \in \mathsf{T}_\omega\mathcal{P}_\omega\text{Base}(\alpha)$ such that $(\alpha, \Gamma) \notin L(\notin_{\text{Base}(\alpha)})$ and $s \Vdash_\mathbb{S} \nabla(\mathsf{T}\bigwedge)(\Gamma)$ for some $s \in S$. We need to show that $s \Vdash_\mathbb{S} \neg\nabla(\mathsf{T}\neg)(\alpha)$. Assume for contradiction that $s \Vdash_\mathbb{S} \nabla(\mathsf{T}\neg)(\alpha)$. This

means that $(\sigma(s), (\mathsf{T}\neg)(\alpha)) \in L\Vdash_{\mathbb{S}}$ and so $(\sigma(s), \alpha) \in L\Vdash_{\mathbb{S}} ; (\mathsf{T}\neg)^{\circ}$. By applying Proposition 3.1.3 (2) we rewrite this as:

$$(\alpha, \sigma(s)) \in (\mathsf{T}\neg) ; (L\Vdash_{\mathbb{S}}{}^{\circ}) = L(\neg ; \Vdash_{\mathbb{S}}^{\circ}). \tag{3.2}$$

On the other hand, from the assumption that $s \Vdash_{\mathbb{S}} \nabla(\mathsf{T}\bigwedge)(\Gamma)$ we get that:

$$(\sigma(s), (\mathsf{T}\bigwedge)(\Gamma)) \in L\Vdash_{\mathbb{S}}$$

and again by Proposition 3.1.3 we have that:

$$(\sigma(s), \Gamma) \in L(\Vdash_{\mathbb{S}} ; \bigwedge^{\circ}). \tag{3.3}$$

Equations (3.2) and (3.3) together with (L2) imply that:

$$(\alpha, \Gamma) \in L(\neg ; \Vdash_{\mathbb{S}}^{\circ}) ; L(\Vdash_{\mathbb{S}}; \bigwedge^{\circ}) \subseteq L(\neg ; \Vdash_{\mathbb{S}}^{\circ} ; \Vdash_{\mathbb{S}} ; \bigwedge^{\circ}). \tag{3.4}$$

Denote the relation $(\neg ; \Vdash_{\mathbb{S}}^{\circ} ; \Vdash_{\mathbb{S}} ; \bigwedge^{\circ}) : \mu\mathtt{ML}_{\nabla}^{L} \nrightarrow \mathcal{P}_{\omega}(\mu\mathtt{ML}_{\nabla}^{L})$ by R. Hence we have $(\alpha, \Gamma) \in LR$. Now recall that $\notin_{\mathrm{Base}(\alpha)}$: $\mathrm{Base}(\alpha) \nrightarrow \mathcal{P}_{\omega}\mathrm{Base}(\alpha)$ is the relation given as follows:

$$\varphi \notin_{\mathrm{Base}(\alpha)} A \text{ iff } \varphi \notin A.$$

From this and the definition of relation R we get that:

$$R\restriction_{\mathrm{Base}(\alpha)\times\mathcal{P}_{\omega}\mathrm{Base}(\alpha)} \subseteq \notin_{\mathrm{Base}(\alpha)},$$

because a formula whose negation is true at some state can not be a conjunct of a conjunction which is also true at that state. Moreover since $\alpha \in \mathsf{T}\mathrm{Base}(\alpha)$ and $\Gamma \in \mathsf{T}\mathcal{P}_{\omega}\mathrm{Base}(\alpha)$ we get:

$$(\alpha, \Gamma) \in LR\restriction_{\mathsf{T}\mathrm{Base}(\alpha)\times\mathsf{T}\mathcal{P}_{\omega}\mathrm{Base}(\alpha)}.$$

By Proposition 3.1.4 we have:

$$(\alpha, \Gamma) \in L(R\restriction_{\mathrm{Base}(\alpha)\times\mathcal{P}_{\omega}\mathrm{Base}(\alpha)}) \subseteq L(\notin_{\mathrm{Base}(\alpha)}),$$

which is a contradiction with the assumption $(\alpha, \Gamma) \notin L(\notin_{\mathrm{Base}(\alpha)})$.

($\Longrightarrow$) For the direction from left to right: assume that $s \Vdash_{\mathbb{S}} \neg\nabla(\mathsf{T}\neg)(\alpha)$ for some $s \in S$. Hence we have $(\sigma(s), (\mathsf{T}\neg)(\alpha)) \notin L\Vdash_{\mathbb{S}}$ and by Proposition 3.1.3 (2), we get that:

$$(\sigma(s), \alpha) \notin L\Vdash_{\mathbb{S}} ; (\mathsf{T}\neg)^{\circ} = L(\Vdash_{\mathbb{S}} ; \neg^{\circ}).$$

In order to find a suitable $\Gamma \in \mathsf{T}_{\omega}\mathcal{P}_{\omega}\mathrm{Base}(\alpha)$, consider the following map:

$$\begin{array}{rcl} f : S & \to & \mathcal{P}_{\omega}\mathrm{Base}(\alpha) \\ s & \mapsto & \{\varphi \in \mathrm{Base}(\alpha) \mid s \Vdash_{\mathbb{S}} \varphi\}, \end{array}$$

and set $\Gamma := \mathsf{T}f(\sigma(s))$. It is obvious that $\Gamma \in \mathsf{T}_\omega\mathcal{P}_\omega\mathrm{Base}(\alpha)$. We claim that Γ, which is defined in this way satisfies the following properties:

$$(i) \qquad (\alpha, \Gamma) = (\alpha, \mathsf{T}f(\sigma(s))) \notin L(\notin_{\mathrm{Base}(\alpha)}),$$

and

$$(ii) \qquad s \Vdash_{\mathbb{S}} \nabla(\mathsf{T}\bigwedge)(\Gamma) = \nabla(\mathsf{T}\bigwedge)(\mathsf{T}f(\sigma(s))).$$

To verify (i) we first show that:

$$f \,;\, \not\ni_{\mathrm{Base}(\alpha)} \subseteq \Vdash_{\mathbb{S}} \,;\, (\neg\restriction_{\mathrm{Base}(\alpha)})^{\circ}. \tag{3.5}$$

This inequality means that if a formula $\varphi \in \mathrm{Base}(\alpha)$ is not in $f(t)$ for some $t \in S$, then the negation of φ is true at t. It holds because if $\varphi \in f(t)$ for a $\varphi \in \mathrm{Base}(\alpha)$ then by definition of f it must be the case that $t \nVdash_{\mathbb{S}} \varphi$ and hence $t \Vdash_{\mathbb{S}} \neg\varphi$.

Now assume for contradiction that $(\alpha, \mathsf{T}f(\sigma(s))) \in L(\notin_{\mathrm{Base}(\alpha)})$. This entails that $(\alpha, \sigma(s)) \in L(\notin_{\mathrm{Base}(\alpha)}) \,;\, (\mathsf{T}f)^{\circ}$. So we have:

$$\begin{array}{rcll}
(\sigma(s), \alpha) \in \mathsf{T}f \,;\, L(\not\ni_{\mathrm{Base}(\alpha)}) & = & L(f \,;\, \not\ni_{\mathrm{Base}(\alpha)}) & \text{Proposition 3.1.3 (2)} \\
& \subseteq & L(\Vdash_{\mathbb{S}} \,;\, (\neg\restriction_{\mathrm{Base}(\alpha)})^{\circ}) & \text{(3.5) and (L1)} \\
& \subseteq & L((\Vdash_{\mathbb{S}} \,;\, \neg^{\circ})\restriction_{S\times\mathrm{Base}(\alpha)}) & \\
& = & (L(\Vdash_{\mathbb{S}} \,;\, \neg^{\circ}))\restriction_{\mathsf{T}S\times\mathsf{T}\mathrm{Base}(\alpha)} & \text{Proposition 3.1.4} \\
& = & L(\Vdash_{\mathbb{S}} \,;\, \neg^{\circ}). &
\end{array}$$

But this contradicts $(\sigma(s), \alpha) \notin L(\Vdash_{\mathbb{S}} \,;\, \neg^{\circ})$.

To prove (ii) we first observe that:

$$\Delta_S \subseteq \Vdash_{\mathbb{S}} \,;\, ((\bigwedge\restriction_{\mathrm{Base}(\alpha)})^{\circ} \,;\, f^{\circ}). \tag{3.6}$$

This holds because the conjunction of formulas that are true at a state is also true at this state.

Now consider:

$$\begin{array}{rcll}
(\sigma(s), \sigma(s)) \in \Delta_{\mathsf{T}S} & \subseteq & L\Delta_S & \text{Proposition 3.1.3 (1)} \\
& \subseteq & L(\Vdash_{\mathbb{S}} \,;\, (\bigwedge\restriction_{\mathrm{Base}(\alpha)})^{\circ} \,;\, f^{\circ}) & \text{(3.6) and (L1)} \\
& = & L \Vdash_{\mathbb{S}} \,;\, (\mathsf{T}\bigwedge\restriction_{\mathrm{Base}(\alpha)})^{\circ} \,;\, (\mathsf{T}f)^{\circ}. & \text{Proposition 3.1.3 (2)}
\end{array}$$

Hence if follows that:

$$(\sigma(s), \mathsf{T}(\bigwedge\restriction_{\mathrm{Base}(\alpha)})(\mathsf{T}f(\sigma(s)))) \in L\Vdash_{\mathbb{S}}.$$

This means that $(\sigma(s), \mathsf{T}\wedge(\Gamma)) \in L\Vdash_{\mathbb{S}}$ due to the following:

$$\begin{array}{rcl}
\mathsf{T}(\bigwedge\restriction_{\mathrm{Base}(\alpha)})(\mathsf{T}f(\sigma(s))) & = & \mathsf{T}(\bigwedge\restriction_{\mathrm{Base}(\alpha)})(\Gamma) \\
& = & \mathsf{T}\bigwedge(\Gamma).
\end{array}$$

Hence we get $s \Vdash_{\mathbb{S}} \nabla\mathsf{T}\bigwedge(\Gamma)$ which finishes the proof of (ii) and so the proof of Proposition 3.2.12.

□

In the case of weak pullback-preserving functors an analogous result has been proved in [SV10]. Shortly after, in [Mar11], it was proved that the result can be generalised to nabla formulas of functors that preserve finite sets and admit a quasi-functorial lax extension.

3.3 Coalgebraic Automata

In this section, we will work with a variant of modal $\mathtt{X}$-automata (see Definition 2.4.19) which are called *non-deterministic* T*-automata* for a given Set functor T. Similar to modal $\mathtt{X}$-automata, non-deterministic T-automata operate on T-models or coloured T-coalgebras. The difference with modal $\mathtt{X}$-automata that makes non-deterministic T-automata more convenient for the proof of our main technical result, i.e., *closure under projection* (Theorem 3.5.3), is the type of their transition map. The transition map of a non-deterministic T-automaton $\mathbb{A}$ over a colour set C doesn't directly involve modal formulas and is of the form $\Theta : A \times C \to \mathcal{P}\mathsf{T}A$. We now give the formal definition of this family of automata.

3.3.1. DEFINITION. Given a Set functor T and a lax extension L for T, a *non-deterministic* T*-automaton over a colour set* C or simply a T*-automaton*, is a tuple $\mathbb{A} = (A, \Theta, \Omega, a_I)$, with A some finite set of *states*, $\Theta : A \times C \to \mathcal{P}\mathsf{T}A$ the *transition function*, $\Omega : A \to \omega$ a *parity* map and $a_I \in A$ an *initial state*.

The acceptance condition for T-automata is formulated in terms of a parity game between Éloise ($\exists$) and Abélard ($\forall$). For an automaton $\mathbb{A}$ and a pointed C-coloured T-coalgebra $\mathbb{S} = (S, \sigma, \gamma, s_I)$, the acceptance game $\mathcal{A}(\mathbb{S}, \mathbb{A})$ is given by the rules of Table 3.1.

Position	Player	Admissible moves	Parity
$(s, a) \in S \times A$	$\exists$	$\{(\sigma(s), \vartheta)\ s.t.\ \vartheta \in \Theta(a, \gamma(s))\}$	$\Omega(a)$
$(\sigma(s), \vartheta) \in \mathsf{T}S \times \mathsf{T}A$	$\exists$	$\{Z : S \nrightarrow A \mid (\sigma(s), \vartheta) \in LZ\}$	0
$Z \subseteq S \times A$	$\forall$	Z	0

Table 3.1: Acceptance game for non-deterministic T-automaton

Positions of the form $(s, a) \in S \times A$ will be called *basic positions* of the game. A partial match of the game of the form $(s, a)(\sigma(s), \vartheta)Z(t, b)$ will be called a *round*. For the winning conditions, recall that finite matches are lost by the player who gets stuck. For infinite matches, consider an arbitrary such match:

$$\rho = (s_0, a_0)(\sigma(s_0), \vartheta_0)Z_0(s_1, a_1)(\sigma(s_1), \vartheta_1)Z_1(s_2, a_2)\ldots$$

Clearly, ρ induces an infinite sequence of states in A:

$$\rho_A := a_0a_1a_2...$$

Now according to the definition of parity games, $\exists$ is the winner of the match ρ if $max\{\Omega(a) \mid a \in \text{Inf}(\rho_A)\}$ is even. Otherwise $\forall$ wins ρ.

3.3.2. DEFINITION. A *(positional) strategy* for $\exists$ in the acceptance game $\mathcal{A}(\mathbb{S}, \mathbb{A})$ will be viewed as a pair of functions:

$$\Phi : S \times A \to \mathsf{T}A,$$

and

$$Z : S \times A \to \mathcal{P}(S \times A).$$

A position from which $\exists$ has a winning strategy is called a *winning position* for $\exists$. As usual the set of all winning positions for $\exists$ (resp. $\forall$) in $\mathcal{A}(\mathbb{S}, \mathbb{A})$ is denoted by $\text{Win}_\exists$ (resp. $\text{Win}_\forall$).

A strategy (Φ, Z) starting from a basic position $(s_I, b) \in S \times A$ is called *scattered* if the relation:

$$R = \{(s_I, b)\} \cup \bigcup\{Z_{s,a} \subseteq S \times A \mid (s, a) \in \text{Win}_\exists\},$$

with $Z_{s,a}$ the value of Z on (s, a), is the graph of some possibly partial function, i.e., for all $t \in S$ we have $|R[t]| \leq 1$. Finally, we say that T-automaton $\mathbb{A}$ *accepts* $(\mathbb{S}, s_I)$ if $\exists$ has a winning strategy in the game $\mathcal{A}(\mathbb{S}, \mathbb{A})@(s_I, a_I)$. If $\exists$ has a scattered winning strategy starting from (s_I, a_I), we will say $\mathbb{A}$ *strongly accepts* $(\mathbb{S}, s_I)$.

3.3.3. DEFINITION. A class of pointed coloured T-coalgebras will be called a T-language. A T-language L is *recognized* by some non-deterministic T-automaton $\mathbb{A}$ if any pointed T-coalgebra belongs to L iff it is accepted by $\mathbb{A}$. Given T-automata $\mathbb{A}$ and $\mathbb{B}$ over some colour set C, we call them *equivalent* if the language accepted by $\mathbb{A}$ is the same as the language accepted by $\mathbb{B}$, i.e., $\mathsf{L}(\mathbb{A}) = \mathsf{L}(\mathbb{B})$.

3.3.4. DEFINITION. Let $\mathbb{A} = (A, \Theta, \Omega, a_I)$ be a non-deterministic T-automaton over colour set C. We call a state $a \in A$ a *true state* of $\mathbb{A}$ if $\Omega(a)$ is even and $\Theta(a, c) = \mathsf{T}(\{a\})$. We will standardly use the notation $a_\top$ to refer to a true state. Given $(a, c) \in A \times C$ we call $\vartheta \in \Theta(a, c)$ a *satisfiable element* of $\mathbb{A}$ if there is a witnessing C-coloured T-coalgebra $(\mathbb{Q}_\vartheta, \rho, \gamma_Q)$, $\tau \in \mathsf{T}Q$ and a relation $Y_\vartheta : Q \nrightarrow A$ such that $(\tau, \vartheta) \in LY_\vartheta$ and $Y_\vartheta \subseteq \text{Win}_\exists(\mathcal{A}(\mathbb{Q}, \mathbb{A}))$. Finally, we call a T-automaton $\mathbb{A}$ *totally satisfiable* whenever ϑ is satisfiable for all $\vartheta \in \bigcup \mathsf{ran}(\Theta)$.

By the following proposition we can always assume without loss of generality that a T-automaton $\mathbb{A}$ over colour set C is totally satisfiable and has a true state. Furthermore, we may always assume that there exists a witnessing C-coloured T-coalgebra $\mathbb{Q}$ that works for all $(a, c) \in A \times C$ and $\vartheta \in \Theta(a, c)$.

3.3.5. PROPOSITION. *For any* T*-automaton* $\mathbb{A}$ *over a color set* C *we have that:*

(1) There is an equivalent T*-automaton* $\mathbb{A}'$ *such that* $\mathbb{A}'$ *has a true state.*
(2) There exists a totally satisfiable T*-automaton* $\mathbb{A}'$ *which is equivalent to* $\mathbb{A}$*.*
(3) If $\mathbb{A}$ *is totally satisfiable, then there is a* C*-colored witnessing coalgebra* $\mathbb{Q} = (Q, \rho, \gamma_Q)$ *and a relation* $Y : Q \nrightarrow A$ *with* $Y \subseteq \mathrm{Win}_{\exists}(\mathbb{Q}, \mathbb{A})$ *such that for all* $(a, c) \in A \times C$ *and* $\vartheta \in \Theta(a, c)$*, there is a* $\tau \in \mathsf{T}Q$ *such that* $(\tau, \vartheta) \in LY$*.*

Proof:

(1) Define $\mathbb{A}' := (A \cup \{a_\top\}, \Theta', \Omega', a'_I)$ such that $a'_I = a_I$ and for all $a \in A$, $\Theta(a) = \Theta'(a)$ and $\Omega(a) = \Omega'(a)$. For $(a_\top, c) \in A \times C$ define $\Theta'(a_\top, c) := \mathsf{T}(\{a_\top\})$ and $\Omega'(a_\top) := 0$. It is easy to check that $\mathbb{A}$ and $\mathbb{A}'$ are equivalent.

(2) We define $\mathbb{A}'$ over C by just removing the unsatisfiable elements of any $\Theta(a, c)$. $\mathbb{A}' = (A, \Theta', \Omega, a_I)$, where:

$$\Theta'(a, c) = \{\vartheta \in \Theta(a, c) \mid \vartheta \text{ is a satisfiable element}\}.$$

$\mathbb{A}'$ and $\mathbb{A}$ are equivalent since $\exists$ will never go through unsatisfiable elements in winning matches.

(3) Take the coproduct of all witnessing coalgebras $\mathbb{Q}_\vartheta$ for all $\vartheta \in \Theta(a, c)$ and all $(a, c) \in A \times C$. The relation Y is the union of all Y_ϑ.

□

3.4 Simulation

In this section, we will present the coalgebraic generalization of one of the well known results in automata theory: the *simulation theorem.* To state the coalgebraic version of this theorem, we first generalize our notion of a non-deterministic T-automaton to that of an *alternating* T-automaton, which has a transition map of the type $\Theta : A \times C \to \mathcal{P}\mathcal{P}\mathsf{T}A$.

3.4.1. Definition. Given a Set functor T and a lax extension L for T, an *alternating* T*-automaton over a colour set* C is a tuple $\mathbb{A} = (A, \Theta, \Omega, a_I)$, with A some finite set of *states*, $\Theta : A \times C \to \mathcal{P}\mathcal{P}\mathsf{T}A$ a *transition function*, $\Omega : A \to \omega$ a *parity* map and $a_I \in A$ an *initial state.* For an automaton $\mathbb{A}$ and a pointed C-coloured T-coalgebra $\mathbb{S} = (S, \sigma, \gamma, s_I)$, the acceptance game $\mathcal{A}(\mathbb{S}, \mathbb{A})$ is a parity game given by the rules of Table 3.2.

Position	Player	Admissible moves	Parity
$(s, a) \in S \times A$	$\exists$	$\{(\sigma(s), \Phi) \in S \times \mathcal{P}\mathsf{T}A \mid \Phi \in \Theta(a, \gamma(s))\}$	$\Omega(a)$
$(s, \Phi) \in S \times \mathcal{P}\mathsf{T}A$	$\forall$	$\{(s, \vartheta) \in S \times \mathsf{T}A \mid \vartheta \in \Phi\}$	0
$(\sigma(s), \vartheta) \in \mathsf{T}S \times \mathsf{T}A$	$\exists$	$\{Z : S \nrightarrow A \mid (\sigma(s), \vartheta) \in LZ\}$	0
$Z \subseteq S \times A$	$\forall$	Z	0

Table 3.2: Acceptance game for alternating T-automaton

As in the case of non-deterministic T-automata we say that $\mathbb{A}$ accepts $(\mathbb{S}, s_I)$ if $\exists$ has a winning strategy from basic position (s_I, a_I). A (positional) strategy for $\exists$ in the acceptance game $\mathcal{A}(\mathbb{S}, \mathbb{A})$ of an alternating T-automaton $\mathbb{A}$ is a pair of maps of the following type:

$$\Phi : S \times A \to \mathcal{P}\mathsf{T}A,$$

and

$$Z : S \times A \to \mathcal{P}(S \times A).$$

Note that any non-deterministic T-automaton can be identified with an alternating T-automaton by identifying members of $\Theta(a, c)$ with singleton sets. Then at each round of a match of the acceptance game for this alternating T-automaton, the first move by $\exists$ uniquely determines the choice of $\forall$ (the unique element of a singleton set).

3.4.2. Definition. We call a T-language *recognizable* if it is recognized by an alternating T-automaton. Similar to non-deterministic T-automata, we call alternating T-automata $\mathbb{A}$ and $\mathbb{B}$ equivalent if they recognize the same T-language, i.e, $\mathsf{L}(\mathbb{A}) = \mathsf{L}(\mathbb{B})$.

3.4.3. Remark. Note that although the definition of T-automata does not explicitly use the logic and ∇-modalities, the alternating T-automata can be seen as a notational variant of the modal $\mathtt{X}$-automata (see Definition 2.4.19). In a sense, the transition map Θ of an alternating T-automaton $\mathbb{A}$ can be seen as a map transforming a pair (a, c) to a disjunction of conjunctions of elements of $\mathsf{T}A$, and these correspond to ∇-formulas.

The following proposition [Ven06, Proposition 4.14.] shows that the alternating T-automata and modal $\mathtt{X}$-automata have the same recognizing power.

3.4.4. Proposition. *Alternating* T*-automata and modal* $\mathtt{X}$*-automata recognise the same classes of coalgebras.*

The next theorem is a coalgebraic version of the so called simulation theorem for T-automata. In case that the functor T preserves weak pullbacks the result has been proved in [KV08]. Here, with a very similar proof strategy, we generalise this result to the class of functors with a quasi-functorial lax extension.

3.4.5. Theorem (Simulation). *Let* T *be a functor with a quasi-functorial lax extension* L. *Every alternating* T*-automaton* $\mathbb{A} = (A, \Theta, \Omega, a_I)$ *is equivalent to a non-deterministic* T*-automaton. Hence a* T*-language is recognizable iff it is non-deterministically recognisable.*

Before going into the technical details of the construction, let us first provide some of the intuitions behind the proof. These intuitions ultimately go back to ideas of Muller and Schupp [MS95].

The main idea is to bring the players' interaction pattern $\exists\forall\exists\forall$ in one round of the acceptance games for automaton $\mathbb{A}$, into the pattern $\exists\forall$ (or more precisely: $\exists\exists\forall$). Concretely speaking, consider a basic position (s, a) in the acceptance game $\mathcal{A}(\mathbb{S}, \mathbb{A})$ for some T-coalgebra $\mathbb{S}$. From this position a round of a match of this game proceeds as follows:

- $\exists$ picks $\Phi \in \Theta(a, \gamma(s))$ moving to position $(\sigma(s), \Phi)$;
- $\forall$ picks $\vartheta \in \Phi$ moving to position $(\sigma(s), \vartheta)$;
- $\exists$ picks $Z_{s,\vartheta}$ such that $(\sigma(s), \vartheta) \in LZ_{s,\vartheta}$;
- $\forall$ picks $(t, b) \in Z_{s,\vartheta}$ as the next basic position.

This pattern may suggest that we modify the game in such a way that the relation $Z_\Phi : S \nrightarrow \mathcal{P}A$ which is the *gathering* of all relations $\{Z_{s,\vartheta} \mid \vartheta \in \Phi\}$ is an appropriate choice for $\exists$. This means that we can take (representations of) subsets of A as the states of the new automaton.

However, if we would simply take subsets of A to be states of the non-deterministic automaton we would get into trouble when defining the acceptance condition. The problem that occurs is similar to what one encounters when determinizing a stream automaton. The remedy is to define the non-deterministic automaton based on the set of *binary relations* over A, rather than subsets of A, and to link matches of the acceptance game of these automata via the notion of a *trace* through a sequence of binary relations.

Now we are ready for the technical details of proof of Theorem 3.4.5.

Proof:
As we already announced we will construct an equivalent non-deterministic automaton $\mathbb{A}^\sharp$ based on the set $A^\sharp$ of all binary relations over the set A. To give the definition of the transition map $\Theta^\sharp$ we need to define a *successor* map $m_a : A^\sharp \to \mathcal{P}A$ for each $a \in A$ given by:

$$m_a : R \mapsto R[a].$$

Thus $\mathsf{T}(m_a) : \mathsf{T}A^\sharp \to \mathsf{T}\mathcal{P}A$. The map $\mathsf{T}(m_a)$ enables us to link potential elements $\Gamma \in \Theta^\sharp(R, c)$ to T-redistributions (see Definition 2.3.31) in $\mathsf{T}\mathcal{P}A$ of objects $\Phi_a \in \Theta(a, c)$. The automaton $\mathbb{A}^\sharp$ is defined as follows:

$$\mathbb{A}^\sharp := (A^\sharp, \Theta^\sharp, NBT_\Omega, R_I),$$

where $A^\sharp = \mathcal{P}(A \times A)$, $R_I = \{(a_I, a_I)\}$, $\Theta^\sharp : A^\sharp \times C \to \mathcal{P}\mathsf{T}A^\sharp$ is given by:

$$\Theta^\sharp(R,c) := \{\Gamma \in \mathsf{T}A^\sharp \mid \forall a \in \mathsf{ran}(R)\ \exists \Phi_a \in \Theta(a,c)\ s.t. \\ \mathsf{T}(m_a)(\Gamma) \text{ is a } \mathsf{T}\text{-redistribution of } \Phi_a\},$$

and NBT_Ω is the set of all those infinite sequences of binary relations that do not contain any bad trace, i.e. a trace such that the highest priority occurring infinitely often is odd.

It is obvious that $\mathbb{A}^\sharp$ is non-deterministic, so it is left to prove that $\mathbb{A}^\sharp$ and $\mathbb{A}$ are equivalent.

Proof of equivalence: Fix a pointed C-colored T-coalgebra $(\mathbb{S}, s_I) = (S, \sigma, \gamma, s_I)$. We will prove the following:

$$\mathbb{A} \text{ accepts } (\mathbb{S}, s_I) \Longleftrightarrow \mathbb{A}^\sharp \text{ accepts } (\mathbb{S}, s_I). \tag{3.7}$$

We prove (3.7) via a comparison of the two acceptance games $\mathcal{A}(\mathbb{S}, \mathbb{A})$ and $\mathcal{A}(\mathbb{S}, \mathbb{A}^\sharp)$.

($\Longrightarrow$) For this direction assume that $\mathbb{A}$ accepts $(\mathbb{S}, s_I)$. So $\exists$ has a winning strategy (Φ, Z) starting from position $(\mathbb{S}, s)$. In the sequel we will define a winning strategy (Π, Q) for $\exists$ in the game $\mathcal{A}(\mathbb{S}, \mathbb{A}^\sharp) = \mathcal{A}^\sharp$. To do so we first define an auxiliary map $\zeta : S \to (S \to \mathcal{P}(A \times A))$ that is associated with Z and maps each $s \in S$ to a map $\zeta_s : S \to \mathcal{P}(A \times A)$ which is define as follows.

$$\begin{array}{rcl} \zeta_s : S & \to & \mathcal{P}(A \times A) \\ \zeta_s(t) & := & \{(a,b) \in A \times A \mid (t,b) \in Z_{s,a}\}. \end{array}$$

Now we define (Π, Q) as follows:

$$\begin{array}{rcl} \Pi : S \times A^\sharp & \to & \mathsf{T}A^\sharp \\ \Pi_{s,R} & := & (\mathsf{T}\zeta_s)\sigma(s), \end{array}$$

and

$$\begin{array}{rcl} Q : S \times \mathsf{T}A^\sharp & \to & \mathcal{P}(S \times A^\sharp) \\ Q_{s,\Sigma} & := & \mathrm{Gr}(\zeta_s). \end{array}$$

Since Q only depends on its first component, we simply drop Σ and write Q_s instead of $Q_{s,\Sigma}$.

The Claims 1 and 2 below state that playing this strategy, $\exists$ wins all matches starting from (s_I, R_I).

We call a position (s, R) in $\mathcal{A}^\sharp$ *safe* if $(s, a) \in \mathrm{Win}_\exists(\mathcal{A})$ for all $a \in \mathsf{ran}(R)$.

Claim 1. Given a safe position (s, R) we have:

(1) (Π, Q) provides legitimate moves at (s, R).

(2) Every $(t, R') \in Q_s$ is safe.

Proof of Claim 1: The main part of the proof consists of showing that $\Pi := \Pi_{s,R}$ is legitimate, i.e., $\Pi \in \Theta^\sharp(R, \gamma(s))$. Consider an arbitrary element $a \in \mathsf{ran}(R)$. By assumption $(s, a) \in \mathrm{Win}_\exists(\mathcal{A}(\mathbb{S}, \mathbb{A}))$. Recall that $\Phi_{s,a}$ and $Z_{s,a}$ are given by $\exists$'s winning strategy in $\mathcal{A}(\mathbb{S}, \mathbb{A})$. To prove the legitimacy of Π we show that $\mathsf{T}(m_a)(\Pi)$ is a T-redistribution of $\Phi_{s,a} \in \Theta(a, \gamma(s))$. It suffices to prove that for all $\vartheta \in \Phi_{s,a}$:

$$(\vartheta, \mathsf{T}m_a(\Pi)) \in L(\in_A).$$

To verify this, note that by legitimacy of $Z_{s,a}$ we have that:

$$(\sigma(s), \vartheta) \in LZ_{s,a}.$$

Now from $Z_{s,a} = \mathrm{Gr}(\zeta_s) \mathbin{;} m_a \mathbin{;} \ni_A$ we get:

$$(\sigma(s), \vartheta) \in L(\mathrm{Gr}(\zeta_s) \mathbin{;} \mathrm{Gr}(m_a) \mathbin{;} \ni_A).$$

Applying Proposition 3.1.3 (2) implies that:

$$(\sigma(s), \vartheta) \in \mathsf{T}(\zeta_s) \mathbin{;} \mathsf{T}(m_a) \mathbin{;} L(\ni_A).$$

Since $\Pi = \mathsf{T}\zeta_s(\sigma(s))$ is defined as the unique object such that $(\sigma(s), \Pi) \in \mathsf{T}(\zeta_s)$ (since ζ_s and so $\mathsf{T}\zeta_s$ are functions), it is immediate that $(\Pi, \vartheta) \in \mathsf{T}(m_a) \mathbin{;} L(\ni_A)$ and so:

$$(\vartheta, \mathsf{T}m_a(\Pi)) \in L(\in_A).$$

Legitimacy of Q_s is immediate by definition:

$$\begin{aligned} \Pi = \mathsf{T}\zeta_s(\sigma(s)) \Rightarrow (\sigma(s), \Pi) &\in \mathsf{T}(\zeta_s) \\ &= L\zeta_s \\ &= LQ_s. \end{aligned}$$

Part (2) of the Claim 1 is straightforward. Let $a, b \in A$, $t \in S$ and $R' \in A^\sharp$ be such that $(t, R') \in Q_s$ and $(a, b) \in R'$. We need to show that $(t, b) \in \mathrm{Win}_\exists(\mathbb{S}, \mathbb{A})$. Recall that by definition of Q_s, $(t, R') \in Q_s$ implies that $R' = \zeta_s(t)$ and by definition of ζ_s we get:

$$(a, b) \in R' \text{ iff } (t, b) \in Z_{s,a}.$$

We are done because $Z_{s,a}$ is given by $\exists$'s winning strategy in $\mathcal{A}(\mathbb{S}, \mathbb{A})$.

□

Claim 2. Playing strategy (Π, Q), $\exists$ wins all finite and infinite matches starting from (s_I, R_I).

Proof of Claim 2: We first prove this claim for finite matches. Applying Claim 1 it is straightforward to check that playing strategy (Π, Q) $\exists$ will never get stuck in finite partial matches.

For infinite matches consider a match:

$$(s_I, R_I)(s_1, R_I)\ldots$$

of $\mathcal{A}(\mathbb{S}, \mathbb{A}^\sharp)$ in which $\exists$ plays according to strategy (Π, Q). To show that this match is won by $\exists$, consider an arbitrary trace $a_I a_I a_1 a_2 \ldots$ on the sequence $R_I R_1 \ldots$. It suffices to show that this trace is not a bad trace. An inductive proof, using part (2) of Claim 1, shows that $(s_{i+1}, a_{i+1}) \in Z_{s_i, a_i}$ for every i. From this it is easy to find a (Φ, Z)-guided match:

$$(s_I, a_I)(s_1, a_1)\ldots$$

in $\mathcal{A}(\mathbb{S}, \mathbb{A})$ and hence $a_I a_I a_1 a_2 \ldots$ cannot be a bad trace, since the strategy (Φ, Z) was assumed to be a winning strategy for $\exists$. This finishes the proof of Claim 2 and so the proof of direction from left to right.

□

($\Longleftarrow$) For the direction from right to left of Theorem 3.4.5, assume that $\mathbb{A}^\sharp$ accepts $(\mathbb{S}, s_I)$. This means there exists a winning strategy χ for $\exists$ in the acceptance game $\mathcal{A}(\mathbb{S}, \mathbb{A}^\sharp)$ starting from position (s_I, R_I). We will use χ and equip $\exists$ with a strategy χ', in the game $\mathcal{A}(\mathbb{S}, \mathbb{A})$ initialized at (s_I, a_I), which has the following property:

For any (possibly finite) χ'-guided match $(s_I, a_I)(s_1, a_1)\ldots$ of $\mathcal{A}(\mathbb{S}, \mathbb{A})$, there is a χ-guided "shadow match" $(s_I, R_I)(s_1, R_1)\ldots$ of $\mathcal{A}(\mathbb{S}, \mathbb{A}^\sharp)$, satisfying the condition that:

$$a_{i+1} \in R_{i+1}[a_i] \text{ for all } i. \tag{3.8}$$

Hence, the sequence of A-states

$$a_I a_1 \ldots$$

of such a match is a trace of the $A^\sharp$-sequence

$$R_I R_1 \ldots$$

which we may associate with a χ-guided match. Since χ is by assumption winning for $\exists$ in $\mathcal{A}(\mathbb{S}, \mathbb{A}^\sharp)$, by definition of the winning condition NBT_Ω for $\mathbb{A}^\sharp$, the maximum parity occurring infinitely often on the trace must be even. This guarantees that $\exists$ wins all infinite matches of $\mathcal{A}(\mathbb{S}, \mathbb{A})$. Hence, it suffices to prove that at any finite stage of a χ'-guided match, she can maintain the above condition for one more round in $\mathcal{A}(\mathbb{S}, \mathbb{A})$.

Suppose then that $\exists$ has been able to keep this condition for k steps. That is, with the partial $\mathbb{A}$-match

$$(s_I, a_I)...(s_k, a_k)$$

we may associate a partial, χ-guided $A^\sharp$-match

$$(s_I, R_I)...(s_k, R_k)$$

such that:

$$a_{i+1} \in R_{i+1}[a_i] \text{ for all } i < k. \tag{3.9}$$

To simplify notation, we write $a = a_k$, $R = R_k$ and $s = s_k$, so we have $a \in \mathsf{ran}(R)$. Let $\Pi \in \mathsf{T}A^\sharp$ and $Q \subseteq S \times A^\sharp$, respectively, be the moves dictated by $\exists$'s winning strategy χ. So Π and Q are legitimate moves, that is, $\Pi \in \Theta^\sharp(R, \gamma(s))$ and $(\sigma(s), \Gamma) \in LQ$. Then by definition of $\Theta^\sharp$, and the fact that $a \in \mathsf{ran}(R)$, there is some $\Phi \in \Theta(a, \gamma(s))$ such that $\mathsf{T}m_a(\Phi)$ is a T-redistribution of Φ. This Φ is the next move of $\exists$ in the game $\mathcal{A}(\mathbb{S}, \mathbb{A})$.

So suppose that $\forall$ responds to $\exists$'s move with an object $\vartheta \in \Phi$. Then $\exists$ has to come up with a relation $Y : S \nrightarrow A$ such that

$$(\sigma(s), \vartheta) \in LY.$$

Our instruction for $\exists$ is to pick the following relation:

$$Y := Q \,;\, \mathrm{Gr}(m_a) \,;\, \ni_A,$$

or more precisely:

$$Y = \{(t, b) \in S \times A \mid b \in R'[a] \text{ for some } R' \in A^\sharp \text{ with } (t, R') \in Q\}.$$

We are done if we prove that this move is legitimate for $\exists$. To see why, distinguish the following cases. If $Y = \emptyset$ then $\forall$ gets stuck so $\exists$ wins immediately. But if $Y \neq \emptyset$, then to any $(s_{k+1}, a_{k+1}) \in Y$ that $\forall$ chooses as his next move, by definition we may associate a relation $R_{k+1} \in A^\sharp$ such that :

$$(a_k, a_{k+1}) \in R_{k+1} \text{ and } (s_{k+1}, R_{k+1}) \in Q.$$

In other words, we have shown that $\exists$ can indeed maintain the above mentioned condition (3.9) for one more round of the game. Thus it is left to show that Y is a legitimate move for $\exists$ in $\mathcal{A}(\mathbb{S}, \mathbb{A})$, which means we need to show that:

$$(\sigma(s), \vartheta) \in LY. \tag{3.10}$$

For this purpose, first observe that the definition of Y and the properties of L (Proposition 3.1.3 (2) and (L2)) imply that:

$$\begin{aligned} LQ\,;\mathsf{T}m_a\,;L(\ni_A) &= LQ\,;L(\mathrm{Gr}(m_a)\,;\ni_A) && (3.11)\\ &\subseteq L(Q\,;\mathrm{Gr}(m_a)\,;\ni_A) && (3.12)\\ &= LY. && (3.13)\end{aligned}$$

Now it follows from the legitimacy of Π in the game $\mathcal{A}(\mathbb{S},\mathbb{A}^\sharp)$, that $\mathsf{T}m_a(\Pi)$ is a T-redistribution of Φ, i.e.,

$$(\Pi,\vartheta)\in \mathsf{T}m_a\,;L(\ni_A). \quad (3.14)$$

From the legitimacy of Q we get that:

$$(\sigma(s),\Pi)\in LQ. \quad (3.15)$$

But then (3.10) immediately follows from (3.11), (3.14) and (3.15).

This finishes the proof of Theorem 3.4.5. □

3.5 Closure properties

We now come to the central part of this chapter - a discussion on closure properties of T-automata. When discussing closure properties, we say that a class $\mathcal{L}$ of T-languages is closed under some operation on T-languages if, whenever we apply this operation to a family of languages in $\mathcal{L}$, we obtain again a language in $\mathcal{L}$. As we will see in this section, the class of non-deterministically recognizable languages is closed under taking union and *projection*, whereas the class of recognizable languages is closed under union, intersection and complementation [Kup06] [KV09].

Closure under intersection and union

3.5.1. Proposition. *Let $\mathbb{A}$ and $\mathbb{A}'$ be two alternating T-automata over colour set C. Then there exist alternating T-automata $\mathbb{A}_\cap$ and $\mathbb{A}_\cup$ over colour set C such that for all pointed C-coloured T-coalgebras $(\mathbb{S},s_I)$ we have that:*

(1) $\mathbb{A}_\cap$ accepts $(\mathbb{S},s_I)$ iff both $\mathbb{A}$ and $\mathbb{A}'$ accept $(\mathbb{S},s_I)$, i.e, $\mathsf{L}(\mathbb{A}_\cap)=\mathsf{L}(\mathbb{A})\cap\mathsf{L}(\mathbb{A}')$.

(2) $\mathbb{A}_\cup$ accepts $(\mathbb{S},s_I)$ iff $\mathbb{A}$ or $\mathbb{A}'$ accepts $(\mathbb{S},s_I)$, i.e, $\mathsf{L}(\mathbb{A}_\cup)=\mathsf{L}(\mathbb{A})\cup\mathsf{L}(\mathbb{A}')$.

The proof of this proposition proceeds along the exact same lines as the proof of [KV08, Proposition 5.4], where the same result has been stated for T-automata when T is a weak pullback-preserving functor.

Closure under complementation

3.5.2. PROPOSITION. *Let T be a functor that preserves finite sets and admits a quasi-functorial lax extension L. Then for every alternating T-automaton $\mathbb{A}$ over colour set C, there exists another alternating T-automaton over C denoted by $\mathbb{A}^c$ (or $\neg\mathbb{A}$) that accepts exactly those pointed C-coloured T-coalgebras that are rejected by $\mathbb{A}$. More precisely, for all pointed C-coloured T-coalgebras $(\mathbb{S}, s_I)$ the following holds:*

$$(\mathbb{S}, s_I) \in \mathsf{L}(\mathbb{A}^c) \text{ iff } (\mathbb{S}, s_I) \notin \mathsf{L}(\mathbb{A}).$$

We call the automaton $\mathbb{A}^c$ the complement of automaton $\mathbb{A}$.

Proof:
Using that alternating T-automata correspond to modal $\mathtt{X}$-automata (cf. Remark 3.4.3), it is enough to prove that for any modal $\mathtt{X}$-automaton $\mathbb{A}$ its complement $\mathbb{A}^c$ exists. The proof proceeds along the exact same lines as the proof in [KV09] for weak pullback-preserving functors. The key idea of the proof is to define $\mathbb{A}^c$ by taking the Boolean duals of the formulas assigned by the transition map of $\mathbb{A}$, while the priority map is defined by simply raising all priorities by 1. To generalize the proof to the class of functors with a quasi-functorial lax extension it suffices to show that the Boolean duals of all formulas appearing in the range of transition map of a given automaton $\mathbb{A}$ are definable. But this is simply the case because of Proposition 3.2.12 and the fact that the Boolean duals of other connectives are standardly defined. □

Closure under projection

The main technical result of this chapter is Theorem 3.5.3, which is a generalization of [KV08, Proposition 5.9], where the same result is proved for the weak pullback-preserving functors. In the following theorem we will generalize this Proposition to the class of all functors with a quasi-functorial lax extension that preserves diagonals. The proof strategy is the same as in [KV08], but the construction here is more involved.

3.5.3. THEOREM (Closure under projection). *Given a non-deterministic T-automaton $\mathbb{A}$ over a colour set $\mathcal{P}\mathtt{X}$ and an element $p \in \mathtt{X}$, there exists a non-deterministic T-automaton $\exists_p.\mathbb{A}$ over the color set $\mathcal{P}(\mathtt{X} \setminus \{p\})$ such that:*

$$(\mathbb{S}, s_I) \in \mathsf{L}(\exists_p.\mathbb{A}) \text{ iff } (\overline{\mathbb{S}}, \overline{s}_I) \in \mathsf{L}(\mathbb{A}) \text{ for some } (\overline{\mathbb{S}}, \overline{s}_I) \text{ with } \mathbb{S}, s_I \leftrightarrows^L_p \overline{\mathbb{S}}, \overline{s}_I. \quad (3.16)$$

Proof:
Given a T-automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ over a color set $\mathcal{P}\mathtt{X}$, we define the T-automaton $\exists_p.\mathbb{A}$ over the color set $\mathcal{P}(\mathtt{X} \setminus \{p\})$ as the automaton $\exists_p.\mathbb{A} := (A, \Theta_p, \Omega, a_I)$, where Θ_p is given by:

$$\begin{aligned} \Theta_p : A \times \mathcal{P}(\mathtt{X} \setminus \{p\}) &\to \mathcal{P}\mathsf{T}A \\ (a, c) &\mapsto \Theta(a, c) \cup \Theta(a, c \cup \{p\}). \end{aligned}$$

We need to show that $\exists_p.\mathbb{A}$ satisfies (3.16).

($\Longleftarrow$) The right-to-left direction of (3.16) is straightforward: given a $\mathcal{P}\mathtt{X}$-coloured T-coalgebra $\overline{\mathbb{S}}$ we denote by $\overline{\mathbb{S}}_p$ the projection of $\overline{\mathbb{S}}$ to the set $\mathtt{X}\setminus\{p\}$, that is we omit p from the colouring and obtain a $\mathcal{P}(\mathtt{X}\setminus\{p\})$-coloured T-coalgebra. Now it is easy to check that all legitimate moves of $\exists$ in the game $\mathcal{A}(\overline{\mathbb{S}},\mathbb{A})$ are also legitimate in $\mathcal{A}(\overline{\mathbb{S}}_p,\exists_p.\mathbb{A})$.

($\Longrightarrow$) For the left-to-right direction of (3.16) assume that $\exists_p.\mathbb{A}$ accepts the $\mathcal{P}(\mathtt{X}\setminus\{p\})$-colored T-coalgebra $(\mathbb{S},s_I)=(S,\sigma,\gamma,s_I)$. We need to define a $\mathcal{P}\mathtt{X}$-colored coalgebra $(\overline{\mathbb{S}},\overline{s}_I)=(\overline{S},\overline{\sigma},\overline{\gamma},\overline{s}_I)$ such that $(\overline{\mathbb{S}},\overline{s}_I)$ belongs to $\mathsf{L}(\mathbb{A})$ and is up-to-p bisimilar with $(\mathbb{S},s_I)$.

By Proposition 3.3.5 we can without loss of generality assume that $\exists_p.\mathbb{A}$ has a true state and is totally satisfiable. Furthermore we can assume that there exists a $\mathcal{P}\mathtt{X}$-colored witnessing T-coalgebra $\mathbb{Q}=(Q,\rho,\gamma_Q)$. In the following, we will give the construction of $(\overline{\mathbb{S}},\overline{s}_I)$ using $(\mathbb{S},s_I)$ and $\mathbb{Q}$.

We put $\overline{S}:=(S\times A)\uplus Q$ and in order to define the coalgebra structure $\overline{\sigma}:\overline{S}\to\mathsf{T}\overline{S}$ we distinguish the following cases:

(1) If $q\in Q$, define $\overline{\sigma}(q):=\rho(q)$.

(2) If $(s,a)\in S\times A$ and $(s,a)\notin \mathrm{Win}_\exists(\mathbb{S},\exists_p.\mathbb{A})$, define $\overline{\sigma}(s,a):=\mathsf{T}\kappa_a(\sigma(s))$, where $\kappa_a:S\to S\times A$, $s\mapsto(s,a)$.

(3) In the case where $(s,a)\in S\times A$ and $(s,a)\in\mathrm{Win}_\exists(\mathbb{S},\exists_p.\mathbb{A})$, we define $\overline{\sigma}(s,a)$ as follows: from the winning strategy that witnesses $(s,a)\in\mathrm{Win}_\exists(\mathbb{S},\exists_p.\mathbb{A})$ we obtain a $\vartheta_{s,a}\in\Theta(a,\gamma(s))$ and a relation $Z_{s,a}:S\nrightarrow A$ such that $Z_{s,a}\subseteq\mathrm{Win}_\exists(\mathbb{S},\exists_p.\mathbb{A})$ and $(\sigma(s),\vartheta_{s,a})\in LZ_{s,a}$. Since $\mathbb{A}$ contains a true state, we can assume without loss of generality that $Z_{s,a}$ is full on S (we can add $(a_\top,t)$ to $Z_{s,a}$ for all $t\notin\mathsf{ran}(Z_{s,a})$). We have $Z_{s,a}=\pi_1^\circ\,;\,\pi_2$ where $\pi_1:Z_{s,a}\to S$ and $\pi_2:Z_{s,a}\to A$ are the projections of $Z_{s,a}$. These projections can be seen as relations from $(S\times A)\uplus Q$ to S and A respectively. It then follows that $Z_{s,a}\subseteq\pi_1^\circ\,;\,(\pi_2\uplus Y)$ where $Y:Q\nrightarrow A$ is given by Proposition 3.3.5 item (3). Since L is a lax extension one obtains that:

$$LZ_{s,a}\subseteq L(\pi_1^\circ;(\pi_2\uplus Y)),$$

and hence from $(\sigma(s),\vartheta_{s,a})\in LZ_{s,a}$ we get:

$$(\sigma(s),\vartheta_{s,a})\in L(\pi_1^\circ;(\pi_2\uplus Y)).$$

It also holds that:

$$\sigma(s)\in\mathsf{dom}(L(\pi_1^\circ)),$$

because $Z_{s,a}$ is full on S, so π_1° is full on S, and thus by Proposition 3.1.8 (2) $L(\pi_1^\circ)$ is full on $\mathsf{T}S$.

Moreover $\vartheta_{s,a}\in\mathsf{ran}(L(\pi_2\uplus Y))$ because $\vartheta_{s,a}\in\mathsf{ran}(LY)$ by the properties of Y and $LY\subseteq L(\pi_2\uplus Y)$.

With the quasi-functoriality of L it now follows that:

$$(\sigma(s), \vartheta_{s,a}) \in L(\pi_1^\circ); L(\pi_2 \uplus Y).$$

Hence it is possible to choose $\overline{\sigma}(s,a) \in \mathsf{T}((S \times A) \uplus Q)$ such that

$$(\sigma(s), \overline{\sigma}(s,a)) \in L(\pi_1^\circ) \text{ and } (\overline{\sigma}(s,a), \vartheta_{s,a}) \in L(\pi_2 \uplus Y). \tag{3.17}$$

To complete the definition of the $\mathcal{P}\mathtt{X}$-colored pointed coalgebra $(\overline{\mathbb{S}}, \overline{s}_I)$, we set $\overline{s}_I := (s_I, a_I)$ and define the coloring $\overline{\gamma} : \overline{S} \to \mathcal{P}\mathtt{X}$ by distinguishing the following cases:

(1) If $q \in Q$, define $\overline{\gamma}(q) := \gamma_Q(q)$.
(2) If $(s,a) \in S \times A$ and $(s,a) \notin \mathrm{Win}_\exists(\mathbb{S}, \exists_p.\mathbb{A})$, define $\overline{\gamma}(s,a) := \gamma(s)$.
(3) If $(s,a) \in S \times A$ and $(s,a) \in \mathrm{Win}_\exists(\mathbb{S}, \exists_p.\mathbb{A})$ we define $\overline{\gamma}(s,a)$ by considering the choice of $\exists$ at (s,a). Since (s,a) is a winning position for $\exists$, she picks an element $\vartheta_{s,a} \in \Theta_p(a, \gamma(s))$. The function Θ_p is defined such that $\Theta_p(a, \gamma(s)) = \Theta(a, \gamma(s)) \cup \Theta(a, \gamma(s) \cup \{p\})$. We set

$$\overline{\gamma}(s,a) := \begin{cases} \gamma(s) \cup \{p\} & \text{if } \vartheta_{s,a} \in \Theta(a, \gamma(s) \cup \{p\}), \\ \gamma(s) & \text{otherwise.} \end{cases}$$

We need to show that $\mathbb{S}, s_I \leftrightarroweq_p^L \overline{\mathbb{S}}, (s_I, a_I)$ and that $((s_I, a_I), a_I) \in \mathrm{Win}_\exists(\overline{\mathbb{S}}, \mathbb{A})$.

Claim 1. $\mathbb{S}, s_I \leftrightarroweq_p^L \overline{\mathbb{S}}, (s_I, a_I)$.

Proof of Claim 1: We show that the graph of the projection $\pi_S : S \times A \to S$ seen as a relation between $\overline{S}$ and S is an up-to-p bisimulation between $(\overline{\mathbb{S}}, \overline{s}_I)$ and $(\mathbb{S}, s_I)$. We need to prove that:

$$(\overline{\sigma}(s,a), \sigma(s)) \in L\pi_S \text{ and } \overline{\gamma}(s,a) \setminus \{p\} = \gamma(s) \text{ whenever } ((s,a), s) \in \pi_S.$$

That $\overline{\gamma}(s,a) \setminus \{p\} = \gamma(s)$ follows directly from the definition of $\overline{\gamma}$.

To prove $(\overline{\sigma}(s,a), \sigma(s)) \in L\pi_S$ we distinguish two cases:

(i) If $(s,a) \in S \times A$ and $(s,a) \notin \mathrm{Win}_\exists(\mathbb{S}, \exists_p.\mathbb{A})$ then the statement holds because, by definition $\overline{\sigma}(s,a) = \mathsf{T}\kappa_a(\sigma(s))$ and since L is a lax extension and $\kappa_a^\circ \subseteq \pi_S$, we have that

$$(\mathsf{T}\kappa_a(\sigma(s)), \sigma(s)) \in (\mathsf{T}\kappa_a)^\circ = L(\kappa_a^\circ) \subseteq L\pi_S.$$

(ii) If $(s,a) \in S \times A$ and $(s,a) \in \mathrm{Win}_\exists(\mathbb{S}, \exists_p.\mathbb{A})$ then we get by the definition of $\overline{\sigma}$ that $(\overline{\sigma}(s,a), \sigma(s)) \in L\pi_1$. It follows that $(\overline{\sigma}(s,a), \sigma(s)) \in L\pi_S$ because L is a lax extension and $\pi_1 \subseteq \pi_S$ since $\pi_1 : Z_{s,a} \to S$ is the projection of the relation $Z_{s,a} \subseteq S \times A$.

$\square$

Claim 2. $((s_I, a_I), a_I) \in \mathrm{Win}_\exists(\overline{\mathbb{S}}, \mathbb{A})$.

Proof of Claim 2: Given the relation $Y : Q \nrightarrow A$ from Proposition 3.3.5 item (3), let (Ψ, Y') be a winning strategy for $\exists$ witnessing that $Y \subseteq \mathrm{Win}_\exists(\mathbb{Q}, \exists_p.\mathbb{A})$, and (Φ, Z) be a winning strategy for $\exists$ in $\mathcal{A}(\mathbb{S}, \exists_p.\mathbb{A})$. We define $\exists$'s winning strategy $(\overline{\Phi}, \overline{Z})$ in $\mathcal{A}(\overline{\mathbb{S}}, \mathbb{A})$ as follows. The definitions are given for $(s, a) \in \mathrm{Win}_\exists(\mathbb{S}, \exists_p.\mathbb{A})$. For other elements of the set $\overline{S} \times A$ the following maps can be defined arbitrarily.

$$\begin{aligned} \overline{\Phi} : \overline{S} \times A &\to \mathsf{T}A \\ ((s,b),a) &\mapsto \vartheta_{s,a} \\ (q,a) &\mapsto \varepsilon_{q,a} \end{aligned} \qquad \begin{aligned} \overline{Z} : \overline{S} \times A &\to \mathcal{P}(\overline{S} \times A) \\ ((s,b),a) &\mapsto \pi_2 \uplus Y \\ (q,a) &\mapsto Y'_{q,a} \end{aligned}$$

where $\vartheta_{s,a}$ and $Z_{s,a}$ are given by the winning strategy (Φ, Z) at position (s, a), π_2 is the projection of $Z_{s,a} : S \nrightarrow A$ on S, and $\varepsilon_{q,a}$ and $Y'_{q,a}$ are given by the strategy (Ψ, Y') at position (q, a).

Claim 2A. For the following positions in $\mathcal{A}(\overline{\mathbb{S}}, \mathbb{A})$, the given strategy $(\overline{\Phi}, \overline{Z})$ provides legitimate moves for $\exists$:

(i) $\{(q, a) \in \overline{S} \times A \mid (q, a) \in \mathrm{Win}_\exists(\mathbb{Q}, \exists_p.\mathbb{A})\}$,

(ii) $\{((s, a), a) \in \overline{S} \times A \mid (s, a) \in \mathrm{Win}_\exists(\mathbb{S}, \exists_p.\mathbb{A})\}$.

Proof of Claim 2A:

(i) We need to show that $(\overline{\sigma}(q), \varepsilon_{q,a}) \in LY'_{q,a}$. This is the case since $\overline{\sigma}(q) = \rho(q)$ and the moves $\varepsilon_{q,a}$ and $Y'_{q,a}$ are given by $\exists$'s winning strategy (Ψ, Y') in $\mathcal{A}(\mathbb{Q}, \exists p.\mathbb{A})$.

(ii) We have to show that $(\overline{\sigma}(s, a), \vartheta_{s,a}) \in L(\pi_2 \uplus Y)$. By the definition of $\overline{\gamma}$ we have that $\vartheta_{s,a} \in \Theta(a, \overline{\gamma}(s, a))$. Now because $\vartheta_{s,a}$ and $\pi_2 \uplus Y$ are given by the winning strategy (Φ, Z), from the definition of $\overline{\sigma}$ and (3.17), we get that: $(\overline{\sigma}(s, a), \vartheta_{s,a}) \in L(\pi_2 \uplus Y)$.

□

Claim 2B. $(\overline{\Phi}, \overline{Z})$ guarantees $\exists$ to win any match of $\mathcal{A}(\overline{\mathbb{S}}, \mathbb{A})@((s_I, a_I), a_I)$.

Proof of Claim 2B: Note that according to the definition of $\overline{\mathbb{S}}$, at the end of each round of a match of $\mathcal{A}(\overline{\mathbb{S}}, \mathbb{A})$, there are two possibilities for $\forall$: he can either pick an element $(q, a) \in Q \times A$ or choose an element from $(S \times A) \times A$. The point is that if at some point he picks an element from the set $Q \times A$, because of the definitions of $\overline{\sigma}$ and $(\overline{\Phi}, \overline{Z})$, there is no way to go back to basic positions of the form $((s, b), a) \in (S \times A) \times A$. On the other hand, if he never picks an element from $Q \times A$, the match will never go through states of $\mathbb{Q}$. Hence to prove that the strategy $(\overline{\Phi}, \overline{Z})$ is winning, it suffices to distinguish the following two kinds of matches:

(i) At some stage $\forall$ chooses an element $(q, a) \in Y$. From this moment on, there is no way to go back to the states of $\mathbb{S}$ and since $Y \subseteq \mathrm{Win}_\exists(\mathbb{Q}, \exists_p.\mathbb{A})$, $\exists$ plays her winning strategy in $\mathcal{A}(\mathbb{Q}, \mathbb{A})$ and wins the match.

(ii) $\forall$ never picks an element of the form (q, a). In this case, any $(\overline{\Phi}, \overline{Z})$-guided match is of the form:

$$((s_I, a_I), a_I)((s_1, a_1), a_1)((s_2, a_2), a_2)\dots$$

This match corresponds to the (Φ, Z)-guided match:

$$(s_I, a_I)(s_1, a_1)(s_2, a_2)\dots$$

in the game $\mathcal{A}(\mathbb{S}, \mathbb{A})$. Since we assumed (Φ, Z) to be a winning strategy for $\exists$, $(\overline{\Phi}, \overline{Z})$ is also a winning strategy for her.

This finishes the proof of Claim 2.

□

And finally this proves Theorem 3.5.3. □

3.6 Logic and Automata

As we mentioned earlier, in this thesis we present the notion of automaton as an alternative way to think of a formula in fixpoint modal logic. In this section we will briefly justify how it is possible to safely transfer from automata to formulas and vice versa.

3.6.1. PROPOSITION. *There exists an effective procedure to transform a formula* $\varphi \in \mu\mathtt{ML}^{L}_{\nabla}(\mathtt{X})$ *to a non-deterministic* T*-automaton* $\mathbb{A}_\varphi$ *over colour set* $\mathcal{P}\mathtt{X}$*, such that for every pointed* $\mathcal{P}\mathtt{X}$*-colored* T*-coalgebra* $(\mathbb{S}, s)$ *we have:*

$$\mathbb{A}_\varphi \textit{ accepts } (\mathbb{S}, s) \textit{ iff } s \Vdash_{\mathbb{S}} \varphi.$$

Proof:
The construction proceeds along the exact same lines as the construction of a non-deterministic T-automaton from a given formula [Ven06, Theorem 2] in the case of weak pullback-preserving functors. The proof proceeds in the following stages:

(1) Using routine methods (see e.g. [Ven06, Theorem 2]), we can inductively show that every formula φ in our language can be effectively transformed into an equivalent alternating T-automaton. The proof presented in [Ven06] essentially works here. There is only one place in the proof of [Ven06, Theorem 2] where the functoriality of relation lifting $\overline{\mathsf{T}}$ of a weak pullback-preserving functor T has been used (line 17 on page 661 in the proof of [Ven06, Proposition 4.19]), and for this part of the proof, functoriality can be simply replaced by quasi-functoriality of relation lifting L, i.e., property 3.1 of Definition 3.1.5.

For the case of negation, i.e., formulas of the form $\neg\varphi$, by the induction hypothesis, we get an equivalent alternating T-automaton $\mathbb{B}_\varphi$ for φ, and then apply Proposition 3.5.2 to transform $\mathbb{B}_\varphi$ to its complement $\mathbb{B}^c_\varphi$. It is obvious that φ and $\mathbb{B}^c_\varphi$ are equivalent.

(2) At this stage, we apply the Simulation Theorem 3.4.5 and transform the alternating T-automaton we obtained at stage (1), to a non-deterministic T-automaton. □

3.6.2. PROPOSITION. *There exists an effective procedure transforming a non-deterministic T-automaton $\mathbb{A}$ over colour set $\mathcal{P}\mathtt{X}$ to an equivalent formula $\varphi_{\mathbb{A}}$ in $\mu\mathtt{ML}^L_\nabla(\mathtt{X})$.*

This result is rather standard, check [Ven06, Theorem 3] for a proof.

3.7 Uniform Interpolation

We can now prove the main theorem of this chapter, viz., uniform interpolation for $\mu\mathtt{ML}^L_\nabla$. As we announced in the introduction, our proof of Uniform Interpolation follows and generalises the proof in [DH00] which shows a similar result for μ-calculus.

3.7.1. THEOREM (Uniform interpolation). *Let T be a set functor that preserves finite sets, and let L be a quasi-functorial lax extension for T. For any formula $\varphi \in \mu\mathtt{ML}^L_\nabla$ and any set $\mathsf{Q} \subseteq \mathsf{P}_\varphi \subseteq \mathtt{X}$ of propositional variables, there is a formula $\varphi_\mathsf{Q} \in \mu\mathtt{ML}_\nabla(\mathsf{Q})$, effectively constructable from φ, such that for every formula $\psi \in \mu\mathtt{ML}^L_\nabla(\mathtt{X})$ with $\mathsf{P}_\varphi \cap \mathsf{P}_\psi \subseteq \mathsf{Q}$, we have that*

$$\varphi \models \psi \text{ iff } \varphi_\mathsf{Q} \models \psi.$$

In case that φ is fixpoint-free, so is φ_Q.

As mentioned earlier, our proof is based on the definability of the bisimulation quantifier in our language.

3.7.2. PROPOSITION. *Given any propositional variable p, there is a map:*

$$\exists p : \mu\mathtt{ML}^L_\nabla \longrightarrow \mu\mathtt{ML}^L_\nabla,$$

such that $\mathsf{P}_{\exists p.\varphi} = \mathsf{P}_\varphi \setminus \{p\}$ and for any pointed T-model $(\mathbb{S}, s)$ we have:

$$s \Vdash_\mathbb{S} \exists p.\varphi \text{ iff } s' \Vdash_{\mathbb{S}'} \varphi, \text{ for some } (\mathbb{S}', s') \text{ with } \mathbb{S}, s \leftrightarroweq^L_p \mathbb{S}', s' \tag{3.18}$$

for any formula $\varphi \in \mu\mathtt{ML}^L_\nabla(\mathtt{X})$.

Intuitively, (3.18) says that we can make the formula φ true by, indeed, changing the interpretation of p, although not necessarily here, but in an up-to-p bisimilar state. For a detailed study of bisimulation quantifiers in modal logic, see [Fre06].

Proof:
Take a formula $\varphi \in \mu\mathtt{ML}^{L}_{\nabla}(\mathtt{X})$. By Proposition 3.6.1 we can transform it to an equivalent T-automaton $\mathbb{A}_{\varphi}$. From Theorem 3.5.3 we have a T-automaton $\exists p.\mathbb{A}_{\varphi}$ such that:

$\exists p.\mathbb{A}_{\varphi}$ accepts $(\mathbb{S}, s)$ iff $\mathbb{A}_{\varphi}$ accepts $(\mathbb{S}', s')$ for some $(\mathbb{S}', s')$ with $\mathbb{S}, s \leftrightarrows^{L}_{p} \mathbb{S}', s'$.

Now by Proposition 3.6.2, we can transform the T-automaton $\exists p.\mathbb{A}_{\varphi}$ to an equivalent formula $\psi_{(\exists p.\mathbb{A}_{\varphi})}$ and put $\exists p.\varphi := \psi_{(\exists p.\mathbb{A}_{\varphi})}$. It is easy to check that:

$$s \Vdash_{\mathbb{S}} \psi_{(\exists p.\mathbb{A}_{\varphi})} \quad \text{iff} \quad s' \Vdash_{\mathbb{S}'} \varphi, \text{ for some } \mathbb{S}', s' \text{ with } \mathbb{S}, s \leftrightarrows^{L}_{p} \mathbb{S}', s'.$$

It is immediate that: $\mathsf{P}_{\exists p.\varphi} = \mathsf{P}_{\varphi} \setminus \{p\}$. □

We are now ready to prove the uniform interpolation theorem i.e. Theorem 3.7.1.

Proof of Theorem 3.7.1: Let $p_0, p_1, ..., p_{n-1}$ enumerate the propositional variables in $\mathsf{P}_{\varphi} \setminus \mathsf{Q}$, and set:

$$\varphi_{\mathsf{Q}} := \exists p_0 \exists p_1 \dots \exists p_{n-1}.\varphi.$$

It is not difficult to verify that φ_{Q} is fixpoint-free if φ is so.

In order to check that $\varphi \vDash \psi$ iff $\varphi_{\mathsf{Q}} \vDash \psi$, first assume that $\varphi \vDash \psi$. To prove that $\varphi_{\mathsf{Q}} \vDash \psi$, take a pointed T-model $(\mathbb{S}_0, s_0)$ with $s_0 \Vdash_{\mathbb{S}_0} \varphi_{\mathsf{Q}}$. By the semantics of the bisimulation quantifiers we get states s_i in T-models $\mathbb{S}_i$ for $i = 1, 2, \dots, n$ such that $s_i \leftrightarrows_{p_i} s_{i+1}$ for $i = 0, ..., n$ and $s_n \Vdash_{\mathbb{S}_n} \varphi$. From the latter fact it follows that $s_n \Vdash_{\mathbb{S}_n} \psi$ since we have assumed $\varphi \vDash \psi$. Because each of the witnessing up-to-p_i $L_{\mathtt{X}}$-bisimulations for $i = 0, 1, \dots, n-1$ is also an $L_{\mathtt{X}\setminus\{p_0,p_1,...,p_{n-1}\}}$-bisimulation, we can compose them and obtain an $L_{\mathtt{X}\setminus\{p_0,p_1,...,p_{n-1}\}}$-bisimulation between s_0 and s_n. Since $\mathsf{P}_{\psi} \subseteq \mathtt{X} \setminus \{p_0, p_1, \dots, p_{n-1}\}$ we get $s_0 \Vdash_{\mathbb{S}_0} \psi$.

For the other direction, we show that $\varphi \vDash \varphi_{\mathsf{Q}}$. Then $\varphi \vDash \psi$ follows by transitivity from $\varphi_{\mathsf{Q}} \vDash \psi$. Take any state s in a T-model $\mathbb{S} = (S, \sigma, V)$ with $s \Vdash_{\mathbb{S}} \varphi$. Then $s \Vdash_{\mathbb{S}} \varphi_{\mathsf{Q}}$ because s is up-to-p $L_{\mathtt{X}}$-bisimilar to itself for any $p \in \mathtt{X}$, since the identity on S is an $L_{\mathtt{X}}$-bisimulation.

□

3.8 The monotone μ-calculus

Monotone modal logic is a generalisation of normal modal logic in which the distribution of $\Box$ over conjunction is weakend to a monotonicity condition expressed as an axiom "$\Box(p \wedge q) \to \Box q$" or a rule "from $p \to q$ derive $\Box p \to \Box q$". The semantics of this logic is given over monotone neighbourhood models (see Example 2.1.5), which are $\mathcal{M}$-models for the monotone neighbourhood functor $\mathcal{M}$ (see Example 2.3.10). Then the interpretation of the modal formulas $\Box\varphi$ and $\Diamond\varphi$ in an $\mathcal{M}$-model $\mathbb{M} = (M, \sigma, V)$ is given by:

$$\mathbb{M}, m \Vdash \Box\varphi \iff \exists U \in \sigma(m) \text{ such that } \forall u \in U\ \mathbb{M}, u \Vdash \varphi,$$

$$\mathbb{M}, m \Vdash \Diamond\varphi \iff \forall U \in \sigma(m)\ \exists u \in U \text{ such that } \mathbb{M}, u \Vdash \varphi.$$

The *monotone μ-calculus* is the fixpoint extension of the monotone modal logic and has applications in the settings such as game logic [Par85] where the use of normal modalities is problematic.

This logic is expressively equivalent to the logic $\mu\mathtt{ML}_{\nabla}^{\widetilde{\mathcal{M}}}$ (see Example 2.3.33 and Definition 3.2.1), meaning that there are truth-preserving translations in both directions. To prove this equivalence, it is enough to show that modalities of the monotone μ-calculus and the language $\mu\mathtt{ML}_{\nabla}^{\widetilde{\mathcal{M}}}$ are interdefinable. In order to recall the logical structure of the satisfaction relation of the modal connectives of the monotone μ-calculus, we shall write $\langle\forall\exists\rangle$ in place of $\Diamond$, and $\langle\exists\forall\rangle$ in place of $\Box$. It is not difficult to check that the following equivalences over $\mathcal{M}$-models hold:

$$\begin{aligned}\langle\exists\forall\rangle\varphi &\equiv \nabla\{\{\varphi\},\{\top\}\} \vee \nabla\{\emptyset\}\\ \langle\forall\exists\rangle\varphi &\equiv \nabla\{\{\varphi,\top\}\} \vee \nabla\emptyset.\end{aligned}$$

Similarly, we can write ∇ in terms of $\langle\exists\forall\rangle$ and $\langle\forall\exists\rangle$ as follows:

$$\nabla\alpha \equiv \bigwedge_{A\in\alpha}\langle\exists\forall\rangle\bigvee A \wedge \bigwedge_{B\in\alpha^\bullet}\langle\forall\exists\rangle\bigvee B,$$

where $\alpha^\bullet$ is constructed from α and belongs to the set $\mathcal{Q}_\omega(\bigcup\alpha)$ (see [SV10] for more details).

Now, since logics of the monotone μ-calculus and $\mu\mathtt{ML}_{\nabla}^{\widetilde{\mathcal{M}}}$ are expressively equivalent, and because $\widetilde{\mathcal{M}}$ is a quasi-functorial relation lifting for $\mathcal{M}$ that preserves diagonals (see Example 3.1.7), the following result is an instance of Theorem 3.7.1.

3.8.1. Corollary. *The monotone μ-calculus enjoys uniform interpolation.*

3.9 Conclusion

In this chapter, we proved that the class of non-deterministic T-automata for functors that admit a quasi-functorial lax extension which preserves diagonals is closed under projection. This result implies that bisimulation quantifiers are definable in the coalgebraic fixpoint logic for this class of functors, hence the logic enjoys Uniform Interpolation. As an instance of our main theorem, we get that the monotone μ-calculus has uniform interpolation. An interesting goal for further research would be to study the class of functors possessing such a relation lifting in more detail. For instance, one might try to investigate and isolate properties of a lax extension which make it quasi-functorial and characterize this class of functors in categorical terms.

Proving uniform interpolation theorems for ∇-based coalgebraic modal logic [Mar11] and ∇-based coalgebraic fixpoint logic [MSV15] naturally suggests to study the case for the predicate lifting setting. Following up on the result of this chapter, in [SSP17], Schröder et alii do this for the non-fixpoint setting. They introduce a notion of one-step (uniform) interpolation and show that a coalgebraic modal logic has uniform interpolation if it has one-step interpolation. Additionally, they identify preservation of finite surjective weak pullbacks as a sufficient, and in the monotone case ($\mathcal{M}$) necessary, condition on the functor T for one-step interpolation. It is worth mentioning here that there is a connection between preservation of surjective weak pullbacks and having bisimulation products which were used to prove the Craig interpolation theorem for the monotone modal logic in [Han03]. More precisely, if a functor T preserves surjective weak pullbacks, then T has bisimulation products. We leave it as future work to investigate the general relationship between quasi-functoriality, preservation of surjective weak pullbacks and bisimulation products and refer the reader to [SSP17] and [Han03] for more details.

For the case of coalgebraic fixpoint logic with predicate liftings, Enqvist and Venema [EV17] proved uniform interpolation using the notion of *disjunctive basis*. More precisely, they used automata-theoretic techniques and proved that if the modal signature Λ for a set functor T has a disjunctive basis, then bisimulation quantifiers are definable in $\mu\mathtt{ML}_\Lambda$ and hence uniform interpolation holds for $\mu\mathtt{ML}_\Lambda$.

Chapter 4

Expressive Completeness

Due to the Janin-Walukiewicz theorem [JW96], which states that the modal μ-calculus captures exactly the bisimulation invariant fragment of monadic second-order logic, we get that any logic for labelled transition systems that is invariant for bisimulation, and that can be translated into monadic second order logic, can be seen as a fragment of the μ-calculus. This result is the μ-calculus counterpart of van Benthem's characterization theorem for basic modal logic [Ben76].

It is the aim of this chapter to generalise this result to a coalgebraic level. For that, we first introduce a notion of coalgebraic monadic second-order logic $\mathtt{MSO}_{\mathsf{T}}$ for coalgebras of type T. Our formalism combines two ideas from the literature. First of all, we looked for inspiration from the coalgebraic versions of first-order logic of Litak & alii [LPSS12]. These authors introduced Coalgebraic Predicate Logic as a common generalisation of first-order logic and coalgebraic modal logic, combining first-order quantification with coalgebraic syntax based on predicate liftings. Our formalism $\mathtt{MSO}_{\mathsf{T}}$ will combine a similar syntactic feature with second-order quantification. Second, following the tradition in automata-theoretic approaches towards monadic second-order logic, our formalism will be one-sorted. That is, we only allow second-order quantifcation in our language, relying on the fact that individual quantification, when called for, can be encoded as second-order quantification relativized to singleton sets. Since predicate liftings are defined as families of maps on powerset algebras, these two ideas fit together very well, to the effect that our second-order logic is in some sense simpler than the first-order formalism of [LPSS12].

The introduction of a monadic second-order logic $\mathtt{MSO}_{\mathsf{T}}$ for T-coalgebras naturally raises the question, for which T does the coalgebraic modal μ-calculus $\mu\mathtt{ML}_{\mathsf{T}}$, which denotes the logic $\mu\mathtt{ML}^{\mathsf{T}}_{\Lambda}$ from Definition 2.4.12 when Λ is the set of *all* monotone predicate liftings for T, correspond to the fragment of $\mathtt{MSO}_{\mathsf{T}}$ that is invariant under behavioural equivalence.

1. Question. *For which class of functors* T*, do we have* $\mu\mathtt{ML}_{\mathsf{T}} \equiv \mathtt{MSO}_{\mathsf{T}}/{\sim}$*?*

To answer this question we take an automata-theoretic perspective on the logics $\mu\mathtt{ML}_\mathsf{T}$ and $\mathtt{MSO}_\mathsf{T}$ via the one-step languages $\mathtt{SO}^1_\mathsf{T}$ and $\mathtt{ML}^1_\mathsf{T}$. We recall that the automata characterisation of $\mu\mathtt{ML}_\mathsf{T}$ is already available (see Fact 4.3.6). As a contribution of this chapter we define a class of parity automata $Aut(\mathtt{SO}^1_\mathsf{T})$ corresponding to formulas in $\mathtt{MSO}_\mathsf{T}$. Similar to the case of standard MSO, where the translation of formulas to automata is valid only over *trees* and not over Kripke frames in general [JW95], we need to introduce T-models that have *tree-like* structures which we call T-*tree models* (see Definition 4.3.8). We then show that there is an effective construction transforming any formula $\varphi \in \mathtt{MSO}_\mathsf{T}$ to an automaton $\mathbb{A}_\varphi \in Aut(\mathtt{SO}^1_\mathsf{T})$, which is equivalent to φ over T-tree models (Theorem 4.3.24). The proof of this result proceeds by induction on the complexity of $\mathtt{MSO}_\mathsf{T}$-formulas, and thus involves various closure properties of automata, such as closure under complementation, union and projection.

Having the automata-theoretic machinery for both $\mathtt{MSO}_\mathsf{T}$ and $\mu\mathtt{ML}_\mathsf{T}$ in place, we can reformulate Question 1 in terms of automata:

2. QUESTION. *Which functors* T *satisfy* $Aut(\mathtt{ML}^1_\mathsf{T}) \equiv Aut(\mathtt{SO}^1_\mathsf{T})/{\sim}$?

Continuing the program of [Ven14], we will approach this question *at the level of the one-step languages*, $\mathtt{SO}^1_\mathsf{T}$ and $\mathtt{ML}^1_\mathsf{T}$ and show that it suffices to establish a certain type of translation called a *uniform translation* between the corresponding one-step languages (see Definition 4.5.4). To start with, observe that any translation from one-step formulas in $\mathtt{SO}^1_\mathsf{T}$ to one-step formulas in $\mathtt{ML}^1_\mathsf{T}$ naturally induces a translation from $Aut(\mathtt{SO}^1_\mathsf{T})$ to $Aut(\mathtt{ML}^1_\mathsf{T})$. In Proposition 4.5.6 we show that if the second-order one-step language $\mathtt{SO}^1_\mathsf{T}$ admits a uniform translation then $Aut(\mathtt{ML}^1_\mathsf{T}) \equiv Aut(\mathtt{SO}^1_\mathsf{T})/{\sim}$ and hence we have the following auxiliary characterisation result (Theorem 4.5.7).

Theorem (Auxiliary Characterization Theorem I) Let Λ be an expressively complete set of monotone predicate liftings for a set functor T, and assume that the monotone fragment of the second-order one-step language $\mathtt{SO}^1_\Lambda$ admits uniform translations. Then:

$$\mu\mathtt{ML}_\Lambda \equiv \mathtt{MSO}_\Lambda/{\sim}.$$

The main observation we make here is that we can actually forget about the syntactic translation and the constraint on $\mathtt{SO}^1_\Lambda$, and focus entirely on the model theory of one-step models. More precisely, if we can find a suitable *uniform construction* (see Definitions 4.5.3) for the one-step models, satisfying a certain model-theoretic condition with respect to the second-order one-step language, the syntactic uniform translation $\mathtt{SO}^1_\mathsf{T} \to \mathtt{ML}^1_\mathsf{T}$ will come for free. In particular, we show that a uniform construction on the class of one-step models for the functor T that satisfies certain *adequacy* conditions, provides (1) a translation $(\cdot)^* : \mathtt{SO}^1_\mathsf{T} \to \mathtt{ML}^1_\mathsf{T}$, together with (2) a construction $(\cdot)_*$ transforming a pointed T-model $(\mathbb{S}, s)$ into a T-tree model $(\mathbb{S}_*, s_*)$ which is a coalgebraic pre-image of $(\mathbb{S}, s)$ such that for all

$\mathbb{A} \in Aut(\mathtt{SO}^1_{\mathsf{T}})$:

$$\mathbb{A} \text{ accepts } (\mathbb{S}_*, s_*) \text{ iff } \mathbb{A}^* \text{ accepts } (\mathbb{S}, s).$$

From this it follows that an automaton $\mathbb{A} \in Aut(\mathtt{SO}^1_{\mathsf{T}})$ is invariant for behavioural equivalence iff it is equivalent to the automaton $\mathbb{A}^* \in Aut(\mathtt{ML}^1_{\mathsf{T}})$.

On the basis of these observations we can prove the following generalisation of the Janin-Walukiewicz theorem (Theorem 4.5.9).

Theorem (Coalgebraic Bisimulation Invariance I) Let T be any set functor. If T admits an adequate uniform construction for every finite set Γ of second-order one-step formulas, then:

$$\mu\mathtt{ML}_{\mathsf{T}} \equiv \mathtt{MSO}_{\mathsf{T}}/{\sim}.$$

In our view, the proof of this theorem separates the 'clean', abstract part of bisimulation invariance results from the more functor-specific parts. As a consequence, Theorem 4.5.9 can be used to obtain immediate results in particular cases including the powerset functor, bag functor and all Kripke polynomial functors (Examples 4.5.32 - 4.5.34).

However, there are functors of interest that are not covered by the above version of the characterization theorem. A concrete example is the monotone neighbourhood functor which does not admit an adequate uniform construction. To resolve this situation, we prove a second version of the general characterization theorem (Theorem 4.5.13), where we work with a modified notion of adequate uniform construction called *weakly adequate* uniform constructions. We also require some additional constraints on T: it must preserve finite sets, and admits a *quasi-functorial lax extension* that preserves diagonal (see Definitions 3.1.5 and 3.1.1), which has also been used in Chapter 3.

Theorem (Coalgebraic Bisimulation Invariance II) Let T be any set functor that preserves finite sets and admits a quasi-functorial lax extension L that preserves diagonals. Let Λ be an expressively complete set of monotone predicate liftings for T. If T admits a weakly adequate uniform construction for every finite set of formulas of $\mathtt{SO}^1_{\Lambda}$, then

$$\mu\mathtt{ML}_{\Lambda} \equiv \mathtt{MSO}_{\Lambda}/{\sim}.$$

This version of the theorem covers the standard Janin-Walukiewicz theorem as an immediate application, just like the previous result. Additionally we prove that it yields an expressive completeness result for the monotone μ-calculus. In this respect, the result improves on one of our results in [ESV15], where we characterized an extension of the monotone μ-calculus with the global modality, but were unable to obtain a proper characterization theorem for the monotone μ-calculus itself. We will discuss the other interesting cases in Section 4.5.4.

We point out that since the second invariance theorem relies on the assumption that T preserves finite sets, some applications of the first result (for example the graded μ-calculus) cannot be captured by the second one. Hence neither of the two characterization results subsumes the other, so we have two distinct generalizations of the Janin-Walukiewicz theorem. Whether these results can be unified into a single result is left as an open problem.

4.1 The μ-calculus and monadic second-order logic

We start this chapter by briefly recalling the syntax and semantics of the μ-calculus and monadic second-order logic over Kripke models. Let $\mathtt{X}$ be a fixed infinite supply of variables.

μ-calculus. Here we present the formulas of the μ-calculus $\mu\mathtt{ML}$ in negation normal form, where negation only occurs at the propositional level:

$$\varphi ::= p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Box\varphi \mid \Diamond\varphi \mid \eta p.\varphi,$$

with $p \in \mathtt{X}$, $\eta \in \{\mu, \nu\}$, and in the formula $\eta p.\varphi$ no free occurrence of the variable p in φ may be in the scope of a negation. We define $\top := \nu p.p$ and $\bot := \mu p.p$.

The satisfaction relation between pointed Kripke models and formulas of $\mu\mathtt{ML}$ is defined as usual (see Definition 2.1.8).

Monadic second-order logic. The syntax of monadic second-order logic $\mathtt{MSO}$ is given by the following grammar:

$$\varphi ::= \mathsf{sr}(p) \mid p \subseteq q \mid R(p, q) \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists p.\varphi,$$

where $p, q \in \mathtt{X}$. Here sr stands for "source" and this connective is used to encode the *actual* world. It is needed since we aim to compare $\mathtt{MSO}$ with the μ-calculus, and formulas of the latter logic are evaluated relative to a point in the model. We define $\top := \forall p.p \subseteq p$ and $\bot := \neg\top$. Formulas are evaluated over pointed Kripke models by the following induction:

- $(S, R, V, u) \Vdash \mathsf{sr}(p)$ iff $V(p) = \{u\}$
- $(S, R, V, u) \Vdash p \subseteq q$ iff $V(p) \subseteq V(q)$
- $(S, R, V, u) \Vdash R(p, q)$ iff for all $v \in V(p)$ there is $w \in V(q)$ with vRw
- standard clauses for the boolean connectives
- $(S, R, V, u) \Vdash \exists p.\varphi$ iff $(S, R, V[p \mapsto Z], u) \Vdash \varphi$ for some $Z \subseteq S$.

4.1.1. Remark. In our version of monadic second-order logic we have only second-order variables and quantification. The idea is that first-order quantification, i.e., quantification over individual states, can be encoded as second-order quantification relativized to singleton sets.

Standardly we say that a formula $\varphi \in \mathtt{MSO}$ is *bisimulation invariant* if it has the same truth value over bisimilar pointed Kripke models.

4.1.2. Example. We call formulas φ and ψ semantically equivalent if they have the same truth value over all pointed Kripke models. We use the notation $\mathtt{L} \equiv \mathtt{L}'$ to say that every formula in $\mathtt{L}$ is semantically equivalent to a formula in $\mathtt{L}'$, and vice versa. The fragment of a language $\mathtt{L}$ that is invariant for behavioural equivalence is denoted by $\mathtt{L}/{\sim}$.

The Janin-Walukiewicz theorem [JW96] can be stated as follows.

4.1.3. Theorem (Janin-Walukiewicz Theorem). *Over Kripke models, a formula φ of $\mathtt{MSO}$ is equivalent to a formula of $\mu\mathtt{ML}$ iff φ is invariant for behavioural equivalence:*

$$\mu\mathtt{ML} \equiv \mathtt{MSO}/{\sim}.$$

4.2 Coalgebraic perspective

In this section we recall the syntax and semantics of coalgebraic fixpoint logic and then introduce coalgebraic monadic second-order logic. Additionally we will present a coalgebraic first-order logic (in Section 4.2.3.1) and prove that, as one would expect, the coalgebraic $\mathtt{MSO}$ we define here is an extension of coalgebraic first-order logic.

Throughout this chapter we let $\mathtt{X}$ be a fixed infinite supply of variables. We also require that set functors preserve all monics in Set, i.e. that $\mathsf{T}f$ is an injective map whenever f is. This is a very mild constraint, since it almost holds for all set functors already, the only possible exception being maps with empty domain [AT90]. So in the reminder of this chapter "set functor" will be taken to mean: "set functor that preserves all monics". All the functors we consider here will satisfy the constraint.

4.2.1 Coalgebraic μ-calculus

As we discussed in Chapter 1, there are two approaches to generalise the μ-calculus to a coalgebraic fixpoint logic: the ∇-setting and the predicate lifting setting. In this chapter we work in the predicate lifting setting. See Definition 2.4.11 to recall the notion of predicate lifting for a set functor T.

Syntax and semantics of $\mu\mathtt{ML}_\Lambda$

Given a set functor T and a set of monotone predicate liftings Λ for T, we present the language $\mu\mathtt{ML}_\Lambda$ of the coalgebraic fixpoint logic based on Λ in negation normal form; note that we make sure that the modal operators of the language are closed under Boolean duals.

$$\varphi ::= p \mid \neg p \mid \lambda(\varphi_1, ..., \varphi_n) \mid \lambda^d(\varphi_1, ..., \varphi_n) \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \eta p.\varphi,$$

where $p \in \mathtt{X}$, λ is any monotone n-place predicate lifting in Λ, $\eta \in \{\mu, \nu\}$, and, in $\eta p.\varphi$, no free occurrence of the variable p is in the scope of a negation. For the case when Λ consists of *all* monotone predicate liftings for T, we write $\mu\mathtt{ML}_\Lambda = \mu\mathtt{ML}_\mathsf{T}$.

The semantics of formulas over a pointed T-model (see Definition 2.4.5) is defined by induction on the complexity of formulas. Here we only recall the case of modal formulas. For $\lambda \in \Lambda$ and a T-model $\mathbb{S} = (S, \sigma, V)$, we set:

$$(\mathbb{S}, s) \Vdash \lambda(\varphi_1, ..., \varphi_n) \text{ iff } \sigma(s) \in \lambda_S([\![\varphi_1]\!], ..., [\![\varphi_n]\!]),$$

where $[\![\varphi_i]\!] = \{t \in S \mid (\mathbb{S}, t) \Vdash \varphi_i\}$.

As we discussed in Examples 2.4.14 and 2.4.2, the powerset functor $\mathcal{P}$ and the monotone neighbourhood functor $\mathcal{M}$ have unary predicate liftings denoted by $\Diamond$ and $\Box$ respectively. The logic $\mu\mathtt{ML}_{\{\Diamond\}}$ is precisely the standard μ-calculus, and the logic $\mu\mathtt{ML}_{\{\Box\}}$ is the monotone μ-calculus which we denote by $\mu\mathtt{MML}$. We recall that the notion of bisimulation arising from the quasi-functorial lax extension $\widetilde{\mathcal{M}}$ (see Definition 3.1.5 and Examples 2.3.33 and 3.1.7) captures neighborhood bisimilarity (see Definition 2.3.34) and this bisimilarity coincides with behavioural equivalence for $\mathcal{M}$.

It is well known that coalgebras and coalgebra morphisms for a set functor form a co-complete category, since its forgetful functor creates colimits [Bar93]. In particular this means that coproducts of arbitrary families of T-models exist, and they correspond to disjoint unions of models in the case of Kripke semantics. Concretely, we define disjoint unions of a family of T-models as follows:

4.2.1. Definition. Let $\{\mathbb{S}_i\}_{i\in I}$ be a family of T-models, where $\mathbb{S}_i = (S_i, \sigma_i, V_i)$. Then we define the *disjoint union* $\coprod_{i\in I} \mathbb{S}_i = (S', \sigma', V')$ by first setting $S' = \coprod_{i\in I} S_i$ to be the disjoint union of the sets S_i. Let f_i denote the inclusion map of S_i into S'. Then we define σ' to be the unique map with the property that $\sigma' \circ f_i = \mathsf{T} f_i(\sigma_i)$ for all $i \in I$. For $p \in \mathtt{X}$ we define $V'(p) = \bigcup_{i\in I} f_i[V_i(p)]$.

4.2.2. Fact. For each $j \in I$, the inclusion map $f_j : \mathbb{S}_j \to \coprod_{i\in I} \mathbb{S}_i$ is a T-model homomorphism.

4.2.2 Coalgebraic MSO

Now we will introduce the coalgebraic monadic second-order logic for a set Λ of monotone predicate liftings.

4.2.3. Definition. Given a set Λ of monotone [1] predicate liftings for T, we define the syntax of the monadic second-order logic $\mathtt{MSO}_\Lambda$ by the following grammar:

$$\varphi ::= \mathsf{sr}(p) \mid p \subseteq q \mid \lambda(p, q_1, .., q_n) \mid \varphi \vee \varphi \mid \neg\varphi \mid \exists p.\varphi,$$

where $p, q, q_1, ..., q_n \in \mathtt{X}$ and λ is any n-place monotone predicate lifting in Λ that is treated as an $n+1$-place predicate in the $\mathtt{MSO}_\Lambda$ language. For Λ equal to the set of all monotone predicate liftings for T, we write $\mathtt{MSO}_\Lambda = \mathtt{MSO}_\mathsf{T}$.

For the semantics, let $(\mathbb{S}, s)$ be a pointed T-model, where $\mathbb{S} = (S, \sigma, V)$. We define the satisfaction relation $\Vdash \subseteq S \times \mathtt{MSO}_\Lambda$ as follows:

- $(\mathbb{S}, s) \Vdash \mathsf{sr}(p)$ iff $V(p) = \{s\}$,
- $(\mathbb{S}, s) \Vdash p \subseteq q$ iff $V(p) \subseteq V(q)$,
- $(\mathbb{S}, s) \Vdash \lambda(p, q_1, ..., q_n)$ iff $\sigma(t) \in \lambda_S(V(q_1), .., V(q_n))$ for all $t \in V(p)$, this formula says that the condition $\lambda(q_1, ..., q_n)$ holds for the unfolding of each state that satisfies p.
- standard clauses for the Boolean connectives,
- $(\mathbb{S}, s) \Vdash \exists p.\varphi$ iff $(S, \sigma, V[p \mapsto Z], s) \Vdash \varphi$, some $Z \subseteq S$.

We introduce the following abbreviations:

- $p = q$ for $p \subseteq q \wedge q \subseteq p$,
- $\mathsf{Em}(p)$ for $\forall q.(p \subseteq q)$,
- $\mathsf{Sing}(p)$ for $\neg\mathsf{Em}(p) \wedge \forall q(q \subseteq p \to (\mathsf{Em}(q) \vee q = p))$,

expressing, respectively, that p and q are equal, that p denotes the empty set, and that p denotes a singleton.

Clearly, standard $\mathtt{MSO}$ is the language $\mathtt{MSO}_{\{\Diamond\}}$, where $\Diamond$ corresponds to the predicate lifting for the standard diamond modality over Kripke models (see Example 2.4.14). So we can reformulate the Janin-Walukiewicz theorem (Fact 4.1.3) as follows:

$$\mu\mathtt{ML}_{\{\Diamond\}} \equiv \mathtt{MSO}_{\{\Diamond\}}/{\sim} \tag{4.1}$$

Considering the lifting $\Box^\mathcal{M}$ for $\mathcal{M}$ from Definition 2.4.15 (which we may denote by $\Box$ for simplicity), we introduce the name *monotone monadic second-order logic* for the language $\mathtt{MSO}_{\{\Box\}}$, which we will henceforth denote by $\mathtt{MMSO}$. Note that the

[1] In the most general case, restricting to monotone predicate liftings is not needed. However, in the context of this chapter, where we take an automata-theoretic perspective on $\mathtt{MSO}_\mathsf{T}$, this restriction makes sense.

atomic $\mathtt{MSO}_{\{\Box\}}$-formula $\Box(p,q)$ encodes a pattern of quantifier alternation of the form $\forall\exists\forall$: it says that *every* state that satisfies p has *some* neighborhood Z such that *every* state in Z satisfies q.

As we mentioned earlier, the key question in this chapter will be to compare the expressive power of coalgebraic monadic second-order logic to that of the coalgebraic μ-calculus. The following observation, of which the (routine) proof is omitted, provides the easy part of the link.

4.2.4. Proposition. *Let Λ be a set of monotone predicate lfitings for a set functor T. There is an inductively defined translation $(\cdot)^{\diamond}$ mapping any formula $\varphi \in \mu\mathtt{ML}_{\Lambda}$ to a semantically equivalent formula $\varphi^{\diamond} \in \mathtt{MSO}_{\Lambda}$.*

4.2.3 Coalgebraic FOL

Since we call our language monadic second-order logic, one would expect it to be an extension of some *first-order* language. Fortunately, we have some background to build on here, as first-order logic for coalgebras was introduced in [LPSS12]. In this section we present coalgebraic FOL and verify that this language can be translated into our coalgebraic monadic second-order language.

4.2.3.1 Syntax and semantics of $\mathtt{FOL}_{\Lambda}$

Given a set Λ of monotone predicate liftings, the language $\mathtt{FOL}_{\Lambda}$ is introduced by the following grammar. Here, we let p range over propositional variables, which are now viewed as unary predicates. We also introduce a set Ind of individual variables and let $x, x_0, ..., x_k$ range over Ind.

$$\varphi ::= p(x) \mid \neg\varphi \mid \varphi \wedge \varphi \mid \exists x.\varphi \mid \lambda(x, \ulcorner x_0 : \psi_0 \urcorner, .., \ulcorner x_k : \psi_k \urcorner),$$

where we require that each x_i appears free in ψ_i. Free and bound variables are defined as usual, except that we count each occurrence of x_i in ψ_i as bound in the formula $\lambda(x, \ulcorner x_0 : \psi_0 \urcorner, .., \ulcorner x_k : \psi_k \urcorner)$. This is because the expression $\ulcorner x_i : \psi_i \urcorner$ should be thought of as a "comprehension" term, denoting the set of states that satisfy $\psi_i(x_i)$.

Given a T-model $\mathbb{S}$ and an assignment $a : \mathsf{Ind} \to S$, we define the satisfaction relation $\Vdash_a$ inductively by taking the usual clauses for atomic formulas $p(x)$, negations, conjunctions and the existential quantifier, and for predicate liftings we set:

$$\mathbb{S} \Vdash_a \lambda(x, \ulcorner x_0 : \psi_0 \urcorner, .., \ulcorner x_k : \psi_k \urcorner) \text{ iff } \sigma(a(x)) \in \lambda_S(Z_1, ..., Z_k)$$

where $Z_1, ..., Z_k$ are the unique sets such that

$$Z_i = \{s \in S \mid \mathbb{S} \Vdash_{a[x_i \mapsto s]} \psi_i\}$$

Here, $a[x_i \mapsto s]$ is the unique assignment which is like a except that x_i is mapped to s.

4.2.5. Proposition. *Let φ be a formula of $\mathtt{FOL}_\Lambda$ with free variables $x_1, ..., x_k$, and let $q_1, ..., q_k \in \mathsf{Var}$ be distinct from all predicates in φ. Then there is a formula $\tau[\varphi, q_1/x_1, ..., q_k/x_k]$ of $\mathtt{MSO}_\Lambda$ such that, for any model $\mathbb{S} = (S, \sigma, V)$ and any assignment $a : \mathsf{Ind} \to S$, we have*

$$\mathbb{S} \Vdash_a \varphi(x_1, .., x_k) \text{ iff } (S, \sigma, V^*, u) \Vdash \tau[\varphi, q_1/x_1, ..., q_k/x_k]$$

where $V^* : \mathsf{Var} \to \mathcal{P}(S)$ is like V except $V^* : q_i \mapsto \{a(x_i)\}$, for $i \in \{1, ..., k\}$, and u is an arbitrary state in S.

Proof:
This is proved by a simple induction on the complexity of formulas. For an atomic formula $p(x)$ put $\tau[p(x), q/x] = q \subseteq p$ for each $q \neq p$. We then take care of the inductive cases one by one: negations and conjunctions are trivial, and existential quantifiers are taken care of by setting:

$$\tau[\exists x.\varphi, q_1/y_1, ..., q_k/y_k] = \exists p.\mathsf{Sing}(p) \wedge \tau[\varphi, p/x, q_1/y_1, ..., q_k/y_k]$$

assuming that x appears free in φ (the other case is simpler).

Finally, for the "comprehension" formulas: consider the formula $\lambda(x, \ulcorner x_0 : \psi_0 \urcorner, .., \ulcorner x_k : \psi_k \urcorner)$. Assume that the free variables of this formula are listed as $x, y_0, ..., y_m$ (recalling that the variables $x_0, ..., x_k$ are bound). Let $p, q_0, ..., q_m$ be propositional variables distinct from all predicates appearing in this formula. We denote the obvious one-to-one correspondence between $x, y_0, ..., y_m$ and $p, q_0, ..., q_m$ by c, so that $c(x) = p$ and $c(y_j) = q_j$. For each $i \in \{1, ..., k\}$, pick a fresh second-order variable Z_i, and define the formula:

$$\theta_i(Z_i) := \forall X.\mathsf{Sing}(X) \to (X \subseteq Z_i \leftrightarrow \tau[\psi_i, Z_i/x_i, c(z_0)/z_0, ..., c(z_l)/z_l])$$

where the free variables of ψ_i are listed as $x_i, z_0, ..., z_l$. Intuitively, the formula θ_i says that the value of Z_i consists of exactly those states s that satisfy $\psi_i(x_i)$. We now set

$$\begin{aligned} \tau[\lambda(x, \ulcorner x_0 : \psi_0 \urcorner, .., \ulcorner x_k : \psi_k \urcorner), p/x, q_0/y_0, ..., q_m/y_m] &= \\ \exists Z_0 ... \exists Z_k.\theta_0(Z_0) \wedge ... \wedge \theta_k(Z_k) \wedge \lambda(p, Z_1, ..., Z_k) \end{aligned}$$

It is straightforward to check correctness of this translation. □

The next result is an immediate corollary of Proposition 4.2.5.

4.2.6. Corollary. *Let φ be any formula of $\mathtt{FOL}_\Lambda$ of a single free individual variable x. Then there is a sentence ψ of $\mathtt{MSO}_\Lambda$ such that, for any pointed T-model $(\mathbb{S}, s)$, and any assignment a mapping x to s, we have*

$$\mathbb{S} \Vdash_a \varphi \text{ iff } (\mathbb{S}, s) \Vdash \psi.$$

Proof:
Pick a fresh variable Z, and consider the formula $\tau[\varphi, Z/x]$ given by the previous proposition. We set:

$$\psi := \exists Z.(\mathsf{sr}(Z) \wedge \tau[\varphi, Z/x])$$

and we are done. □

4.3 Automata and one-step languages

In this section we first recall the definition of parity automata corresponding to the language $\mu\mathtt{ML}_\Lambda$ employing the *one-step language* $\mathtt{ML}^1_\Lambda(A)$ over a set A. We then introduce automata for the coalgebraic monadic second-order logic, and translate formulas of $\mathtt{MSO}_\Lambda$ into automata operating on tree-like coalgebras.

We start with the definition of one-step formulas $\mathtt{ML}^1_\Lambda(A)$ over a set A. Note that the one-step language that we define here is slightly different from the one in Section 2.4.3; here we only involve one set of variables in the grammar of $\mathtt{ML}^1_\Lambda(A)$. The reason is that in this chapter we work with *chromatic* automata, where the power set of $\mathtt{X}$ (the set of propositional variables) takes on the role of an alphabet of the automata. Moreover, unlike Section 2.4.3, here we use the same notation as the predicate liftings λ and its boolean dual λ^d (see Definition 2.4.11) to refer to the modalities arising from λ and λ^d.

4.3.1. Definition. Given a set Λ of monotone predicate liftings, the set $\mathtt{ML}^1_\Lambda(A)$ of *modal one-step formulas* over a set A of variables is given by the following grammar:

$$\varphi ::= \bot \mid \top \mid \lambda(\pi_1, ..., \pi_n) \mid \lambda^d(\pi_1, ..., \pi_n) \mid \varphi \vee \varphi \mid \varphi \wedge \varphi,$$

where $\pi_1, ..., \pi_n$ are formulas built up from variables in A using disjunctions and conjunctions. More formally, as in Section 2.4.3, we require that $\pi_1, ..., \pi_n \in \mathtt{LF}(A)$, the set of lattice formulas over A.

Formulas of $\mathtt{ML}^1_\Lambda(A)$ are interpreted over one-step T-models.

4.3.2. Definition. Given a functor T and a set of variables A, a *one-step* T-*model* over A is a triple (S, α, V) where S is any set, $\alpha \in \mathsf{T}S$ and $V : A \to \mathcal{Q}(S)$ is a valuation [2].

4.3.3. Definition. Given a one-step T-model (S, α, V), the semantics of formulas in $\mathtt{ML}^1_\Lambda(A)$ is given as follows:

- standard clauses for the boolean connectives,
- $(S, \alpha, V) \Vdash_1 \lambda(\pi_1, ..., \pi_n)$ iff $\alpha \in \lambda_X([\![\pi_1]\!]^0_V, ..., [\![\pi_n]\!]^0_V)$

where $[\![\pi_i]\!]^0_V \subseteq S$ is the (classical) truth set of the formula π_i under the valuation V (check Section 2.4.3 for precise definition of $[\![\ ]\!]^0_V$).

[2] Note that we have written the valuation V as having the type $A \to \mathcal{Q}(S)$ rather than $A \to \mathcal{P}(S)$ in this definition. This notation is equally correct since the covariant and contravariant powerset functors act the same on objects, and it is sometimes a more convenient notation since the naturality condition of predicate liftings is formulated in terms of $\mathcal{Q}$ and not $\mathcal{P}$.

4.3.1 Automata for $\mu\mathtt{ML}_\Lambda$

We can now define the class of automata used to characterize the coalgebraic μ-calculus.

4.3.4. DEFINITION. Let $\mathtt{X}$ be a finite set of propositional variables and Λ a set of monotone predicate liftings. Then a *($\mathtt{X}$-chromatic) modal Λ-automaton* is a tuple (A, Θ, Ω, a_I) where A is a finite set of states with $a_I \in A$,

$$\Theta : A \times \mathcal{P}(\mathtt{X}) \to \mathtt{ML}^1_\Lambda(A)$$

is the transition map of the automaton, and $\Omega : A \to \omega$ is the priority map. The class of these automata is denoted as $Aut(\mathtt{ML}^1_\Lambda)$.

Note that the automata we define here are the chromatic version of modal $\mathtt{X}$-automata defined in Definition 2.4.19. Here there are two distinct sets of "variables" involved in the automaton $\mathbb{A}$, and it is important to keep these apart since they have different roles: the variables in $\mathtt{X}$ are used to provide the alphabet of the automaton (and correspond to free variables of corresponding fixpoint formulas), while the variables in A are the states of the automaton (and correspond to bound variables of a corresponding fixpoint formula.)

The acceptance game for an automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ and a T-model (S, σ, V) is the parity game given by Table 4.1:

Position	Player	Admissible moves	Priority
$(a, s) \in A \times S$	$\exists$	$\{U \in (\mathcal{P}S)^A \mid (S, \sigma(s), U) \Vdash_1 \Theta(a, m_V(s))\}$	$\Omega(a)$
$U : A \to \mathcal{P}S$	$\forall$	$\{(b, t) \mid t \in U(b)\}$	0

Table 4.1: Acceptance game for modal Λ-automaton

The loser of a finite match is the player who got stuck, and the winner of an infinite match is $\exists$ if the highest priority that appears infinitely often in the match is even, and the winner is $\forall$ if this priority is odd. Note that the valuations U and V in Table 4.1 play fundamentally different roles: V is a fixed valuation given by the model on which the automaton is running, assigning values to the open variables of the automaton, while U is a "local" valuation assigning values to the states of the automaton, which correspond roughly to bound variables of a fixpoint formula.

Given a strategy χ for either player, a χ-guided match is standardly defined as in Definition 2.2.1. Winning strategies are also defined as usual.

4.3.5. DEFINITION. A modal Λ-automaton $\mathbb{A}$ *accepts* a pointed T-model $(\mathbb{S}, s)$, written $(\mathbb{S}, s) \Vdash \mathbb{A}$, if $\exists$ has a winning strategy in the acceptance game from the starting position (a_I, s). We say that an automaton $\mathbb{A}$ is *equivalent* to a formula

$\varphi \in \mu\mathtt{ML}_\Lambda$ if, for every pointed T-model $(\mathbb{S}, s)$, we have that $\mathbb{A}$ accepts $(\mathbb{S}, s)$ iff $(\mathbb{S}, s) \Vdash \varphi$.

4.3.6. FACT. [FLV10] Let T be a set functor, and Λ a set of monotone predicate liftings for T. Then:

$$\mu\mathtt{ML}_\Lambda \equiv Aut(\mathtt{ML}_\Lambda).$$

That is, there are effective transformations of formulas from $\mu\mathtt{ML}_\Lambda$ into equivalent automata in $Aut(\mathtt{ML}_\Lambda)$, and vice versa.

4.3.2 Automata for $\mathtt{MSO}_\Lambda$

In this section, we introduce automata for coalgebraic monadic second-order logic. As usual, we want to make sure that our generalisation recovers the standard cases. It is known that formulas of standard MSO are equivalent to parity automata over trees and this result can not be extended to arbitrary Kripke models. Hence we expect a similar constraint on the models that our modal Λ-automata run over them. This leads us to introduce the class of T-*tree models*.

4.3.7. DEFINITION. Given a set S and $\alpha \in \mathsf{T}S$, a subset $X \subseteq S$ is said to be a *support* for α if there is some $\beta \in \mathsf{T}X$ with $\mathsf{T}i_{X,S}(\beta) = \alpha$. A *supporting Kripke frame* for a T-coalgebra (S, σ) is a binary relation $R \subseteq S \times S$ such that, for all $u \in S$, $R(u) = \{v \mid (u, v) \in R\}$ is a support for $\sigma(u)$.

Whenever $X \subseteq S$ is a support for $\alpha \in \mathsf{T}S$, our assumption that T preserves inclusion maps, guarantees that there is a *unique* $\alpha' \in \mathsf{T}X$ such that $\mathsf{T}i_{X,S}(\alpha') = \alpha$. We shall denote this α' by $\alpha\restriction_X$.

4.3.8. DEFINITION. A T-*tree model* is a structure $(\mathbb{S}, R, u)$ where $\mathbb{S} = (S, \sigma, V)$ is a T-model, $u \in S$, and R is a supporting Kripke frame for the coalgebra (S, σ), and furthermore (S, R) is a tree rooted at u (i.e. there is a unique R-path from u to w for each $w \in S$).

To define automata for $\mathtt{MSO}_\Lambda$ we start by describing a very general class of automata first introduced in [Ven14]. The motivation for taking this general perspective is to emphasize that many automata-theoretic concepts and basic results apply already in this setting, and we believe the general automaton concept we introduce here has some independent theoretical interest. We start by the following remark: Note that for a finite set A of size n, an n-place predicate lifting λ can be seen as $\lambda : \mathcal{Q}^A \to \mathcal{Q} \circ \mathsf{T}$, when fixing an ordering $\{a_1, ..., a_n\}$ of the elements of A. In these cases we sometimes speak of "predicate liftings over A" rather that "n-place predicate liftings", but note that this is merely a notational variation rather than a substantial generalization of the notion of predicate liftings. A predicate lifting λ over A can be represented equivalently as a subset of $\mathsf{T}\mathcal{P}A$, via an application of the Yoneda lemma (this observation is due to Schröder [Sch08]). Following this observation we can view any predicate lifting λ over A as a one-step formula in $\mathtt{ML}^1_{\{\lambda\}}(A)$[3]. With this in mind we can write $(S, \alpha, V) \Vdash_1 \lambda$ instead of $\alpha \in \lambda_S(V)$.

In this notation the naturality constraint for a predicate lifting λ over A becomes, for given $\alpha \in \mathsf{T}S$, $f : S \to Y$ and $V : A \to \mathcal{Q}(Y)$:

$$(S, \alpha, \mathcal{Q}f \circ V) \Vdash_1 \lambda \Leftrightarrow (Y, \mathsf{T}f(\alpha), V) \Vdash_1 \lambda.$$

Following the notion of predicate liftings over a set A, we now define the notion of *abstract one-step formulas*. The basic idea behind the definition of these formulas is that, as a usual formula induces a truth set given a valuation, abstract one-step formulas induce a subset of $\mathsf{T}X$, given a valuation $V : A \to \mathcal{Q}X$.

4.3.9. DEFINITION. Given a finite set A, an *abstract one-step formula* over A assigns to each set X a map

$$\psi_X : (\mathcal{Q}X)^A \to \mathcal{Q}\mathsf{T}X.$$

4.3.10. DEFINITION. An *abstract one-step language* $\mathtt{L}$ consists of a collection $\mathtt{L}(A)$ of abstract one-step formulas for every finite set A.

Here we follow the tradition of abstract model theory where it is common to refer to classes of models satisfying some closure properties (such as closure under complement and intersection) as "abstract logics". Likewise, we can think of an abstract one-step formula over a set of A as a collection of one-step models over A (Definition 4.3.2) and call a set of abstract one-step formulas an "abstract one-step language", although there is no syntax involved.

Our automata will be indexed by a (finite) set of variables, corresponding to the set of free variables of the $\mathtt{MSO}_\mathsf{T}$-formula it represents.

4.3.11. DEFINITION. Let $\mathsf{P} \subseteq \mathtt{X}$ be a finite set of variables and let $\mathtt{L}$ be an abstract one-step language for the functor T. A (P-*chromatic*) $\mathtt{L}$-*automaton* is a structure (A, Θ, Ω, a_I) where

- A is a finite set, with $a_I \in A$,
- $\Omega : A \to \omega$ is a pariority map, and
- $\Theta : A \times \mathcal{P}(\mathsf{P}) \to \mathtt{L}(A)$ is the transition map of $\mathbb{A}$.

[3] More precisely, λ corresponds to a one-step formula in $\mathtt{ML}^1_{\{\lambda'\}}(A)$ where the lifting $\lambda' : \mathcal{Q}^n \to \mathcal{Q} \circ \mathsf{T}$ is obtained by composing λ with the natural isomorphism between $\mathcal{Q}^n$ and $\mathcal{Q}^A$ induced by some fixed bijection $f : n \to A$. If we write $A = \{a_1, \cdots, a_n\}$ with $a_k = f(k-1)$, this means that we have $\lambda_S(V) = \lambda'_S(V(a_1), \cdots, V(a_n))$. We shall permit some abuse of notation and simply identify λ with the associated lifting λ'.

The *acceptance game* of $\mathbb{A}$ with respect to a T-tree model (T, R, σ, V, u) is given by Table 4.2. We say that the automaton $\mathbb{A}$ accepts the pointed model (T, R, σ, V, u) if $\exists$ has a winning strategy in this game starting from position (a_I, u).

Position	Player	Admissible moves	Priority
$(a, s) \in A \times T$	$\exists$	$\{U : A \to \mathcal{P}(R(s)) \mid (R(s), \sigma(s){\restriction_{R(s)}}, U) \Vdash_1 \Theta(a, m_V(s))\}$	$\Omega(a)$
$U : A \to \mathcal{P}(T)$	$\forall$	$\{(b, t) \mid t \in U(b)\}$	0

Table 4.2: Acceptance game for $\mathtt{L}$-automaton

There are two main differences between these automata and the modal Λ-automata we have considered in Section 4.3.1; first, we have dropped the naturality constraint on the one-step language. Second, these automata will run on T-tree models rather than on T-models.

4.3.3 Closure properties

In this section we will present some closure properties for $\mathtt{L}$-automata. The notion of abstract one-step formulas is useful to obtain these closure properties. We start by closure under union and complementation. Closure under union is reduced to closer under disjunction for the abstract one-step language $\mathtt{L}(A)$. Here the same symbol $\vee$ to refer to disjunction for abstract one-step formulas, however the notion is defined via union of liftings.

4.3.12. Proposition. *If the abstract one-step language* $\mathtt{L}$ *is closed under disjunction, then the class of* $\mathtt{L}$*-automata is closed under union.*

Proof:
Suppose $\mathtt{L}$ is an abstract one-step language for T and $\mathbb{A} = (A, \Theta, \Omega, a_I)$ and $\mathbb{A}' = (A', \Theta', \Omega', a'_I)$ are two $\mathtt{L}$-automata. We can assume that A and A' are disjoint. Having closure under disjunction we define the automaton $\mathbb{A} \sqcup \mathbb{A}' = (A^*, \Theta^*, \Omega^*, a^*_I)$ as follows:

- $A^* = \{b\} \cup A \cup A'$ where b is any object not in $A \cup A'$
- $a^*_I = b$
- $\Theta^*(b, c) = \Theta(a_I, c) \vee \Theta(a'_I, c)$
- $\Theta^*(a, c) = \Theta(a, c)$ for $a \in A$, and $\Theta^*(a', c) = \Theta'(a', c)$ for $a' \in A'$
- $\Omega^*(b) = 0$
- $\Omega^*(a) = \Omega(a)$ for $a \in A$ and $\Omega^*(a') = \Omega'(a')$ for $a' \in A'$

It is routine to check that $\mathsf{L}(\mathbb{A} \sqcup \mathbb{A}') = \mathsf{L}(\mathbb{A}) \cup \mathsf{L}(\mathbb{A}')$, where $\mathsf{L}(\mathbb{A})$ denotes the language of the automaton $\mathbb{A}$. □

Next, we can show that L-automata are closed under complementation, provided that the one-step language L is closed under Boolean duals. The Boolean dual of an abstract one-step formulas and monotonicity of such formulas are defined exactly as in Definition 2.4.11. Additionally we call an L-automaton $\mathbb{A}$ *monotone* if, for all $a \in A$ and all colours $c \in \mathcal{P}(\mathtt{X})$, the generalized lifting $\Theta(a, c)$ is monotone in each variable in A.

4.3.13. Proposition. *If the monotone fragment of the abstract one-step language* L *is closed under boolean duals, then the class of monotone* L*-automata is closed under complementation.*

Proof:
Let $\mathbb{A} = (A, \Theta, \Omega, a_I)$ be a monotone L-automaton. We define the automaton $\mathbb{A}^c = (A^c, \Theta^c, \Omega^c, a^c_I)$ as follows:

- $A^c = A$
- $a^c_I = a_I$
- $\Theta^c(a, c) = \Theta(a, c)^d$
- $\Omega^c(a) = \Omega(a) + 1$

It can be checked that the language recognised by this automaton is precisely the complement of the language recognized by $\mathbb{A}$. □

The next property is *closure under existential projection* or simply *closure under projection*. We will prove this property for a certain class of L-automata, viz, the *non-deterministic automata*. Although the definition of these special automata is based on the idea of [JW96], our construction is different in the sense that we use semantical properties of one-step formulas instead of their syntactic shape.

4.3.14. Definition. An abstract one-step formula φ over A is said to be *special basic* if, for every one-step T-model (S, α, V) such that $(S, \alpha, V) \Vdash_1 \varphi$, there is a valuation $V^* : A \to \mathcal{Q}(S)$ such that

- $V^*(a) \subseteq V(a)$ for each $a \in A$,
- $V^*(a) \cap V^*(b) = \emptyset$ whenever $a \neq b$, and
- $(X, \alpha, V^*) \Vdash_1 \varphi$.

The second clause can also be stated as: $m_{V^*}(u)$ is either empty or a singleton, for all $u \in S$. Call an L-automaton *non-deterministic* if every generalized lifting $\Theta(a, c)$ is special basic.

It is easy to check that closure under disjunction for L implies that its special basic fragment is also closed under taking disjunctions. From this we obtain the following:

4.3.15. PROPOSITION. *If the abstract one-step language* L *is closed under disjunction, then the class of non-deterministic* L*-automata is closed under existential projection over* T*-tree models.*

Proof:
Suppose $\mathbb{A} = (A, \Theta, \Omega, a_I)$ is a non-deterministic P-chromatic L-automaton. Define the $\mathsf{P} \setminus \{p\}$-chromatic L-automaton $\exists p.\mathbb{A} = (A, \Theta_p, \Omega, a_I)$ by setting

$$\Theta_p(a, c) = \Theta(a, c) \vee \Theta(a, c \cup \{p\}).$$

It is easy to see that every T-tree model accepted by $\mathbb{A}$ is also accepted by $\exists p.\mathbb{A}$. Conversely, suppose $\exists p.\mathbb{A}$ accepts some T-tree model (S, R, σ, V, s_I). For each winning position (a, s) in the acceptance game, let $V_{(a,s)}$ be the valuation chosen by $\exists$ according to some given winning strategy χ. Note that we can assume that χ is a *positional* winning strategy, since $\exists p.\mathbb{A}$ is a parity automaton. Employing the fact that special basic liftings are closed under disjunction, it is not difficult to see that the automaton $\exists p.\mathbb{A}$ is a non-deterministic automaton, and so for each winning position (a, s) there is a valuation $V^*_{(a,s)} : A \to \mathcal{P}(R(s))$, which is an admissible move for $\exists$, such that $V^*_{(a,s)}(b) \subseteq V_{(a,s)}(b)$ and such that for all $b_1 \neq b_2 \in A$ we have $V^*_{(a,s)}(b_1) \cap V^*_{(a,s)}(b_2) = \emptyset$. Define the strategy χ^* by letting $\exists$ choose the valuation $V^*_{(a,s)}$ at each winning position (a, s) - this is a legitimate move since $\Theta_p(a, m_V(s))$ is special basic, and χ^* is still a winning strategy, since the valuations chosen by $\exists$ are smaller and so no new choices for $\forall$ are introduced. Furthermore, χ^* is clearly still a positional winning strategy.

From these facts it follows by a simple induction on the depth of the nodes in the supporting tree that the strategy χ^* is *scattered*, i.e. that for every $s \in S$ there is at most one automaton state a such that (a_s, s) appears in a χ^*-guided match of the acceptance game. So we can define a valuation V' like V except we evaluate q to be true at all and only the states s such that:

$$(R(s), \sigma(s), V^*_{(a_s,s)}) \Vdash_1 \Theta(a_s, c \cup \{q\}),$$

where a_s is a necessarily *unique* automaton state such that (a_s, s) appears in some χ^*-guided match, and c is the color consisting of the variables true under V at s. It is not hard to show that $\mathbb{A}$ accepts (S, R, σ, V', s_I). □

4.3.4 Second-order automata

We now introduce a concrete second-order one-step language $\mathtt{SO}^1_\Lambda$ for a given set Λ of monotone predicate liftings for a functor T and show that $\mathtt{MSO}_\Lambda$ can be translated into the class of automata corresponding to $\mathtt{SO}^1_\Lambda$. By the closure properties established in the previous section, the language $\mathtt{SO}^1_\Lambda$ needs to be closed under disjunction and boolean dual, and furthermore we need to establish that every automaton for this language is equivalent to a non-deterministic automaton.

4.3.16. Definition. Let Λ be a set of monotone predicate liftings for T. The set of *second-order one-step formulas* over any set of variables A and relative to the set of liftings Λ is defined by the grammar:

$$\varphi ::= a \subseteq b \mid \lambda(a_1, ..., a_n) \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists a.\varphi,$$

where $a, b, a_1, ..., a_n \in A$ and λ is any predicate lifting in Λ.

The semantics of a second-order one-step formula in a one-step model (S, α, V) (with $V : A \to \mathcal{P}(S)$) is defined by the following clauses:

- $(S, \alpha, V) \Vdash_1 p \subseteq q$ iff $V(p) \subseteq V(q)$,
- $(S, \alpha, V) \Vdash_1 \lambda(a_1, ..., a_n)$ iff $\alpha \in \lambda_S(V(a_1), ..., V(a_n))$,
- standard clauses for the Boolean connectives,
- $(S, \alpha, V) \Vdash_1 \exists a.\varphi$ iff $(S, \alpha, V[a \mapsto Z]) \Vdash_1 \varphi$ for some $Z \subseteq S$.

Fixing an infinite set of "one-step variables" Var_1, and given a finite set A, the set of *second-order one-step sentences* over A, denoted by $\mathtt{SO}^1_\Lambda(A)$, is the set of one-step formulas over $A \cup \mathsf{Var}_1$, with all free variables belonging to A. As before, for Λ equal to the set of all monotone predicate liftings for T, we write $\mathtt{SO}^1_\Lambda(A) = \mathtt{SO}^1_\mathsf{T}(A)$.

Note that unlike $\mathtt{ML}^1_\Lambda$, negations are allowed in the syntax of $\mathtt{SO}^1_\Lambda$. This means that second-order formulas are not monotone in general.

4.3.17. Remark. The difference between the one-step language $\mathtt{SO}^1_\Lambda$ and the full language $\mathtt{MSO}_\Lambda$ may seem rather subtle at first sight. It is important to note that an n-place predicate lifting now corresponds to an n-place predicate in the language, *not* an $n+1$-place predicate as in $\mathtt{MSO}_\Lambda$. While the formula $\lambda(a_1, ..., a_n)$ of $\mathtt{SO}^1_\Lambda$ expresses a property of one-step models, the formula $\lambda(p, q_1, ..., q_n)$ in $\mathtt{MSO}_\Lambda$ rather describes a property of the transition map of a coalgebra as a whole: it says that the condition $\lambda(q_1, ..., q_n)$ holds for the unfolding of each state s that satisfies p.

To see the relation between between $\mathtt{SO}^1_\Lambda(A)$ and the one-step language $\mathtt{L}(A)$ from Definition 4.3.10, note that any second-order one-step A-sentence φ can be regarded as a generalized predicate lifting over A, with

$$\varphi_X(V) = \{\alpha \in \mathsf{T}X \mid (X, \alpha, V) \Vdash_1 \varphi\}.$$

In particular, general concepts like monotonicity and closure under boolean duals apply to second-order one-step sentences. Also, note that second-order sentences are invariant under a natural notion of isomorphism:

4.3.18. Definition. An *isomorphism* between two one-step T-models (S_1, α_1, V_1) and (S_2, α_2, V_2) is a bijection $i : S_1 \to S_2$ such that $\mathsf{T}i(\alpha_1) = \alpha_2$ and $m_1(u) = m_2(i(u))$ for each $u \in S_1$, with m_1 and m_2 denoting as usual the markings associated with V_1 and V_2.

4.3.19. Observation. *Given any set of predicate liftings Λ and a set of variables A, any two isomorphic one-step T-models satisfy the same formulas in the one-step language $\mathtt{SO}^1_\Lambda(A)$.*

We can now introduce second-order automata. Recall that $\mathtt{SO}^1_\Lambda$ is the assignment of the one-step second-order A-sentences $\mathtt{SO}^1_\Lambda(A)$ to every set of variables A.

4.3.20. Definition. Let Λ be a set of monotone predicate liftings for T. A *second-order* Λ-*automaton* is an $\mathtt{SO}^1_\Lambda$-automaton. We write $Aut(\mathtt{SO}^1_\Lambda)$ to denote this class of automata, and $Aut(\mathtt{SO}^1_\mathsf{T})$ in case Λ is the set of *all* monotone predicate liftings for T.

We already mentioned that our aim in this section is to prove that every formula in $\mathtt{MSO}_\Lambda$ can be translated into an equivalent second-order Λ-automaton (over T-tree models). As one would expect, the proof is by induction on the complexity of formulas in $\mathtt{MSO}_\Lambda$. Closure properties of second-order automata enable us to cover all the induction steps. Since the language $\mathtt{SO}^1_\Lambda$ is closed under disjunction and Boolean duals of formulas we get closure under union and complementation immediately. To obtain the closure under projection we need a *simulation* theorem, because then the result follows from Proposition 4.3.15. A key observation in the proof of the simulation theorem is that we can always assume that the second-order one-step formulas that are involved in the construction of the second-order Λ-automata are monotone. The next Proposition justifies this assumption:

4.3.21. Proposition. *Let Λ be any set of monotone predicate liftings. Then every automaton $\mathbb{A} \in Aut(\mathtt{SO}^1_\Lambda)$ is equivalent to a monotone second-order automaton $\mathbb{A} \in Aut(\mathtt{SO}^1_\Lambda)$.*

Proof:
The idea comes from a useful trick by Walukiewicz [Wal02]. Enumerate A as $\{a_1, ..., a_k\}$, and replace each formula $\Theta(a, c)$ by

$$\exists Z_1 ... \exists Z_k . Z_1 \subseteq a_1 \wedge ... \wedge Z_k \subseteq a_k \wedge \Theta(a, c)[(Z_i/a_i)_{i \in \{1,...,k\}}],$$

where $\Theta(a, c)[(Z_i/a_i)_{i\in\{1,...,k\}}]$ is the result of substituting the variable Z_i for each free variable a_i in $\Theta(a, c)$. This new formula is monotone in the variables from A and the resulting automaton is equivalent to $\mathbb{A}$. □

4.3.22. THEOREM (Simulation). *Let Λ be a set of monotone predicate liftings for T. For any monotone automaton $\mathbb{A} \in Aut(\mathtt{SO}^1_\Lambda)$ there exists an equivalent non-deterministic $\mathbb{A}' \in Aut(\mathtt{SO}^1_\Lambda)$.*

The intuition behind the proof of this theorem is very similar to the idea of the proof of Theorem 3.4.5. Given an automaton $\mathbb{A} \in Aut(\mathtt{SO}_\Lambda)$, applying the *powerset construction*, we first define an auxiliary non-deterministic automaton $\mathbb{A}_n$ and then define $\mathbb{A}'$ based on the automaton $\mathbb{A}_n$. States of $\mathbb{A}_n$ are called *macro states* representing a collection of states of $\mathbb{A}$ simultaneously. More explicitly, $\mathbb{A}_n$ is based on the set of binary relations over A. As we explained in Chapter 3, we cannot simply take subsets of A as macro states, because we need to keep track of matches in the acceptance games of $\mathbb{A}$ and $\mathbb{A}_n$ to compare them and make sure that these automata are equivalent. Although we make sure that $\mathbb{A}_n$ is non-deterministic and is equivalent to $\mathbb{A}$, it is not the desired automaton, since it does not have a parity acceptance condition. Hence at the final stage of the proof of Theorem 4.3.22, we need to turn $\mathbb{A}_n$ into a parity automaton $\mathbb{A}'$.

Proof of Theorem 3.4.5: Fix an automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$. We consider the set $\mathcal{P}(A \times A)$ of binary relations over A as a set of variables. Let

$$\mathsf{Disj} := \bigwedge_{R \neq R' \subseteq A \times A} \forall X.(X \subseteq R \wedge X \subseteq R') \rightarrow \mathsf{Em}(X).$$

Pick a fresh variable Z_a for each $a \in A$. Given a one-step formula φ, let

$$\varphi[(Z_a/a)_{a\in A}]$$

be the result of substituting Z_a for each free variable $a \in A$ in φ. Enumerate the elements of A as $a_1, ..., a_k$, and define the formula $\mathsf{Sim}(\varphi, b)$ for $b \in A$ to be

$$\exists Z_{a_1} ... \exists Z_{a_k} . \bigwedge_{1 \leq i \leq k} \mathsf{Rep}(Z_{a_i}, \{R' \mid (b, a_i) \in R'\}) \wedge \varphi[(Z_a/a)_{a\in A}]),$$

where $\mathsf{Rep}(Z_{a_i}, \{R' \mid (b, a_i) \in R'\})$ is the formula:

$$\forall X.\mathsf{Sing}(X) \rightarrow (X \subseteq Z_{a_i} \leftrightarrow \bigvee \{X \subseteq R' \mid (b, a_i) \in R'\}).$$

In informal terms, the formula $\mathsf{Rep}(Z_{a_i}, \{R' \mid (b, a_i) \in R'\})$ says that the variable Z_{a_i} represents a disjunction of all the macro-states that contain a_i. The formula $\mathsf{Sim}(\varphi, b)$ can thus be thought of as reformulating the formula φ in terms of macro-states.

Let $\mathbb{A} = (A, \Theta, \Omega, a_I)$ be any monadic Λ-automaton. We can assume w.l.o.g. that $\mathbb{A}$ is monotone. We first construct the automaton $\mathbb{A}_n = (A_n, \Theta_n, F, a_I^*)$ with a non-parity acceptance condition $F \subseteq (A_n)^\omega$ as follows:

- $A_n = \mathcal{P}(A \times A)$
- $\Theta_n(R, c) = \mathsf{Disj} \wedge \bigwedge_{b \in \pi_2[R]} \mathsf{Sim}(\Theta(b, c), b)$
- $a_I^* = \{(a_I, a_I)\}$
- F is the set of streams over $\mathcal{P}(A \times A)$ with no bad traces.

Here, π_2 is the second projection of a relation R so that $\pi_2[R]$ denotes the range of R. We recall that a *trace* in a stream $R_1R_2R_3...$ over $\mathcal{P}(A \times A)$ is a stream $a_1a_2a_3...$ over A with $a_1 \in \pi_2[R_1]$ and $(a_{j-1}, a_j) \in R_j$ for $j > 1$. A trace is called *bad* if the maximum parity occurring infinitely many times in the stream $\Omega(a_1)\Omega(a_2)\cdots$ is odd.

Lemma 1. $\mathbb{A}_n$ accepts precisely the same T-tree models as $\mathbb{A}$.

Proof of Lemma 1: Fix a pointed T-tree model $(\mathbb{S}, R, s_I)$ where $\mathbb{S} = (S, \sigma, V)$. We want to show that $\mathbb{A}$ accepts $(\mathbb{S}, R, s_I)$ if and only if $\mathbb{A}_n$ does. That is, we want to show that the languages $\mathsf{L}(\mathbb{A})$ and $\mathsf{L}(\mathbb{A}_n)$ recognized by these two automata are the same.

$\mathsf{L}(\mathbb{A}) \subseteq \mathsf{L}(\mathbb{A}_n)$: Suppose first that $\mathbb{A}$ accepts $(\mathbb{S}, R, s_I)$. Let χ be a positional winning strategy for $\exists$ in the acceptance game, mapping each winning position (a, s) to a valuation $U : A \to \mathcal{Q}(R(s))$ such that:

$$(R(s), \sigma(s), U) \Vdash_1 \Theta(a, m_V(s)).$$

Such a strategy exists since $\mathbb{A}$ is a parity automaton, and so the acceptance game is a parity game. We define the winning strategy χ^* for $\exists$ in the acceptance game for $\mathbb{A}_n$ as follows:

Given a position (B, s), define the function $f_{B,s} : R(s) \to \mathcal{P}(A \times A)$ by setting

$$f_{B,s}(s') = \{(a, b) \mid a \in \pi_2[B] \text{ and } s' \in \chi(a, s)(b)\}$$

At the position (B, s), let $\exists$ choose the following valuation $\chi^*(B, s)$, defined by:

$$\chi^*(B, s)(B') = \{s' \in R(s) \mid f_{B,s}(s') = B'\}.$$

Our first claim is that, for each position of the form (B, s) where each (b, s) for $b \in \pi_2[B]$ appears in some χ-guided match of the acceptance game for $\mathbb{A}$ with start position (a_I, s_I), the move for $\exists$ given by the strategy χ^* is legitimate. To prove this claim we need to check that, for each position (B, s), we have

$$(R(s), \sigma(s), \chi^*(B, s)) \Vdash_1 \Theta_n(B, m_V(s)),$$

provided each (b, s) for $b \in \pi_2[B]$ appears in some χ-guided match. First, the formula Disj is true since the marking $\chi^*(B, s)$ is the inverse of a mapping from $R(s)$ to $\mathcal{P}(A \times A)$. We now have to check that, for each $a' \in \pi_2[B]$ we have

$$(R(s), \sigma(s), \chi^*(B, s)) \Vdash_1 \mathsf{Sim}(\Theta(a', m_V(s)), a').$$

We need to find sets $S_{a_1}, ..., S_{a_k} \subseteq R(s)$ such that $(R(s), \sigma(s), \chi^*(B, s))$ with the assignment $Z_{a_i} \mapsto S_{a_i}$ satisfies the formula

$$\bigwedge_{1 \leq i \leq k} \mathsf{Rep}(Z_{a_i}, \{B' \mid (a', a_i) \in B'\}) \wedge \Theta(a', m_V(s))[Z_a/a].$$

Since χ gives a legitimate move at the position (a', s) for each $a' \in \pi_2[B]$, each one-step model of the form $(R(s), \sigma(s), \chi(a', s))$ satisfies the formula $\Theta(a', m_V(s))$. Hence, if we assign to each variable Z_{a_i} the set $\chi(a', s)(a_i)$, then this variable assignment satisfies the formula

$$\Theta(a', m_V(s))[a \mapsto Z_a \mid a \in A].$$

Since the formula $\Theta(a', s)$ is monotone in all the variables A, the same is true for any larger assignment. So it now suffices to prove that:

$$\chi(a', s)(a_i) \subseteq \{\chi^*(B, s)(B') \mid (a', a_i) \in B'\},$$

since we can then safely take:

$$S_{a_i} = \{\chi^*(B, s)(B') \mid (a', a_i) \in B'\},$$

To prove this inclusion, suppose $s' \in \chi(a', s)(a_i)$. Let B' be the relation defined by:

$$(d, d') \in B' \Leftrightarrow d \in \pi_2[B] \text{ and } s' \in \chi(d, s)(d')$$

Clearly, $(a', a_i) \in B'$. Moreover, $f_{B,s}(s') = B'$ by definition, and so $s' \in \chi^*(B, s)(B')$ as required.

We now show that any χ^*-guided match with start position (a_I^*, s_I) is winning for $\exists$. We have to prove two things: first, that $\exists$ never gets stuck in a χ^*-guided match, and second, that $\exists$ wins every infinite χ^*-guided match, i.e. no infinite χ^*-guided match contains a bad trace.

First we show that $\exists$ never gets stuck. For this to be the case, all we need to show is that if (B, s) is the last position of some χ^*-guided partial match, then all the positions (a, s) for $a \in \pi_2[B]$ are winning positions for the strategy χ -

by our previous claim this guarantees the move $\chi^*(B, s)$ to be legal. We prove by induction on the length of a finite partial match that this holds for the last position of the match: it holds for $(\{(a_I, a_I)\}, s_I)$, clearly, since χ is a winning strategy at (a_I, s_I). Suppose that the induction hypothesis holds for a finite match with last position (B, s). Let (B', s') be any position such that $s' \in \chi^*(B, s)(B')$. Then

$$B' = f_{B,s}(s') = \{(a, b) \mid a \in \pi_2[B] \text{ and } s' \in \chi(a, s)(b)\}.$$

So suppose $b \in \pi_2[B']$. Then there is some a with $(a, b) \in B'$, and we must have $a \in \pi_2[B]$ and $s' \in \chi(a, s)(b)$. But since the position (a, s) is winning by the inductive hypothesis, this means that (b, s') is a winning position for χ, and we are done.

We now show that $\exists$ wins every infinite χ^*-guided match. For this, it suffices to show that every trace $a_1 a_2 a_3 \cdots$ in a χ^*-guided infinite match

$$(B_1, s_1), (B_2, s_2), (B_3, s_3), ...$$

corresponds to a χ-guided match

$$(a_I, s_I) = (a_1, s_1), (a_2, s_2), (a_3, s_3), ...$$

So fix a trace $a_1 a_2 a_3 \cdots$ meaning that for each a_i we have $(a_i, a_{i+1}) \in B_{i+1}$. We have to show that $s_{i+1} \in \chi(a_i, s_i)(a_{i+1})$ for each i. We have:

$$s^{i+1} \in \chi^*(B_i, s_i)(B_{i+1}),$$

meaning that B_{i+1} is equal to

$$f_{B_i, s_i}(s_{i+1}) = \{(a, b) \mid a \in \pi_2[B_i] \text{ and } s_{i+1} \in \chi(a, s_i)(b)\}.$$

In particular, since $(a_i, a_{i+1}) \in B_{i+1}$, this means we must have $s_{i+1} \in \chi(a_i, s_i)(a_{i+1})$ as required.

$\mathsf{L}(\mathbb{A}_n) \subseteq \mathsf{L}(\mathbb{A})$: Conversely, suppose $\mathbb{A}_n$ accepts $(\mathbb{S}, s_I)$ with winning strategy χ. We construct a winning strategy χ^* for $\exists$ w.r.t $\mathbb{A}$. Note that the strategy χ is not necessarily positional, since the acceptance game for $\mathbb{A}_n$ is not a parity game.

By induction on the length of a χ^*-guided partial match

$$\rho = (a_1, s_1), (a_2, s_2), (a_3, s_3)...(a_k, s_k)$$

with $(a_1, s_1) = (a_I, s_I)$, we are going to define a next legitimate move $\chi^*(\rho)$ for $\exists$, and by a simultaneous induction we construct a χ-guided partial match

$$\tau = (B_1, s_1), (B_2, s_2), (B_3, s_3)...(B_k, s_k)$$

with $(B_1, s_1) = (a_I^*, s_I)$, $a_j \in \pi_2[B_j]$ for each j and $(a_{j-1}, a_j) \in B_j$ for each $k \geq j > 1$. Furthermore we will make sure that whenever a χ^*-guided match ρ

is an initial segment of a match ρ', the χ-guided match associated with ρ is an initial segment of the χ-guided match associated with ρ'. It will follow at once that χ^* is a winning strategy, since $\exists$ never gets stuck in any χ^*-guided partial match and, furthermore, every infinite χ^*-guided match corresponds to a trace in some χ-guided infinite match.

The base case of the induction is the unique match of length 1 with the single position (a_I, s_I), and we take the corresponding position to be (a_I^*, s_I). Now, suppose χ^* has been defined on all matches of length $< k$, and let ρ be a χ^*-guided partial match of length k of the form:

$$(a_1, s_1), (a_2, s_2), (a_3, s_3)...(a_k, s_k)$$

By the inductive hypothesis we have a corresponding χ-guided match τ which we write as

$$(B_1, s_1), (B_2, s_2), (B_3, s_3)...(B_k, s_k)$$

with $a_k \in \pi_2[B_k]$. Now we define the next legal move $\chi^*(\rho)$ for $\exists$, and we show that for every position (a', s') such that $s' \in \chi^*(\rho)(a')$, we can find a relation B' such that $(a_k, a') \in B'$ and

$$(B_1, s_1), (B_2, s_2), (B_3, s_3)...(B_k, s_k), (B', s')$$

is a χ-guided match.

Since τ is a χ-guided partial match and χ is a winning strategy for $\exists$, we have that:

$$(R(s_k), \sigma(s_k), \chi(\rho)) \Vdash_1 \Theta_n(B_k, m_V(s_k)).$$

Since $a_k \in \pi_2[B_k]$, this means that there exist sets $S_b \subseteq R(s_k)$ for each $b \in A$ such that the 1-step model $(R(s_k), \sigma(s_k), \chi(N))$ satisfies the formula:

$$\bigwedge_{1 \leq i \leq k} \mathsf{Rep}(Z_b, \{B' \mid (a', a_i) \in B'\}) \wedge \Theta(a_k, m_V(s_k))[Z_b/b].$$

under the assignment $Z_b \mapsto S_b$. Hence, the valuation U defined by

$$U : b \mapsto S_b,$$

will be such that:

$$(R(s_k), \sigma(s_k), U) \Vdash_1 \Theta(a_k, m_V(s_k)).$$

So we set $\chi^*(\rho) = U$, a legal move. Note that we have

$$U(b) = \bigcup \{\chi(\tau)(B') \mid (a_k, b) \in B'\}.$$

Now, let (a', s') be such that $s' \in U(a')$. This means that there is some B' with $(a_k, a') \in B'$ and $s' \in \chi(\tau)(B')$. Hence, (B', s') satisfies the required conditions, and we are done. $\square$

The only thing left to do at this point is to transform $\mathbb{A}_n$ into an automaton that has its acceptance condition given by a parity map. The set of streams over $\mathcal{P}(A \times A)$ that contain no bad traces w.r.t. the parity map Ω is an ω-regular stream language, so let

$$\mathbb{Z} = (Z, \Theta_Z, \Omega_Z, z_I)$$

be a parity stream automaton that recognizes this language, with $\Theta_Z : Z \times \mathcal{P}(A \times A) \to Z$. We now construct the automaton

$$\mathbb{A}' = \mathbb{A}_n \odot \mathbb{Z} = (A'_n, \Theta'_n, \Omega'_n, a'_I)$$

as follows:

- $A'_n = A_n \times Z$
- $a'_I = (a^*_I, z_I)$
- $\Theta'_n((R, z), c) = \Theta_n(R, c)[(R', \Theta_Z(R, z))/R']_{R' \in \mathcal{P}(A \times A)}]$
- $\Omega'_n(R, z) = \Omega(z)$

It is not difficult to check that $\mathbb{A}_n \odot \mathbb{Z}$ is equivalent to $\mathbb{A}_n$. Since $\mathbb{A}_n \odot \mathbb{Z}$ is clearly still a non-deterministic automaton, this ends the proof of the simulation theorem.

Combining Proposition 4.3.15 with Theorem 4.3.22, we obtain closure under existential projection.

4.3.23. Proposition. *Let Λ be a set of monotone predicate liftings for a set functor T. Over T-tree models, the class of second-order Λ-automata is closed under existential projection.*

We are now ready to state the main result of this section:

4.3.24. Theorem. *For every formula $\varphi \in \mathtt{MSO}_\Lambda$ with free variables in P, there exists a monotone P-chromatic second-order Λ-automaton $\mathbb{A}_\varphi \in Aut(\mathtt{SO}^1_\Lambda)$ that is equivalent to φ over T-tree models.*

Proof:
We will prove this theorem by induction on the complexity of the formula $\varphi \in \mathtt{MSO}_\Lambda$. It is enough to cover the case for atomic formulas, since the other cases are immediate from the closure properties we have proved.

We present only the case of formulas of the form $\lambda(p, q_1, ..., q_n)$. First, note that by naturality of λ, we have $(\mathbb{S}, R, s) \Vdash \lambda(p, q_1, ..., q_n)$ if, and only if, for each $u \in V(p)$:

$$\sigma(s)\restriction_{R(s)} \in \lambda_{R(s)}(V(q_1) \cap R(s), ..., V(q_n) \cap R(s)).$$

With this in mind, it is not hard to check that the following automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ is equivalent to $\lambda(p, q_1, ..., q_n)$ over T-tree models, where:

- $A = (a_I, b_1, ..., b_n)$
- $\Theta(a_I, c) = \begin{cases} \forall Z.Z \subseteq a_I & \text{if } p \notin c \\ \lambda(b_1, ..., b_n) \wedge \forall Z.Z \subseteq a_I & \text{if } p \in c \end{cases}$
- $\Theta(b_i, c) = \begin{cases} \bot & \text{if } q_i \notin c \\ \top & \text{if } q_i \in c \end{cases}$
- $\Omega(a_I) = \Omega(b_1) = ... = \Omega(b_n) = 0$

□

Intuitively, this automaton goes on indefinitely searching the underlying tree of the model for states that satisfy p, and whenever it finds such a state u it checks whether $u \in \lambda_S(V(q_1), ..., V(q_n))$.

4.4 One-step expressive completeness

Our main result in this chapter is formulated in terms of the set of *all* monotone predicate liftings for a given functor, and we would argue that this is a rather natural choice. However, in some cases it is possible to choose some smaller set of liftings, in order to have a more concrete and manageable presentation of the syntax of the corresponding μ-calculus. It will then be important to choose an *expressively complete* set of predicate liftings, meaning that *any* monotone lifting for the functor can be defined by a formula in the modal one-step language. In particular, this will be required to recover the Janin-Walukiewicz theorem in its original form as a special case of our main results.

4.4.1. Definition. Let Λ be a set of monotone predicate liftings and ψ any abstract one-step formula over the finite set A. We say that ψ is Λ-*definable* if there is a formula $\varphi \in \mathtt{ML}^1_\Lambda(A)$ such that, for any one-step model (X, α, V) we have:

$$\alpha \in \psi_X(V) \text{ iff } (X, \alpha, V) \Vdash_1 \varphi.$$

We say that a set of monotone predicate liftings Λ is *expressively complete* if every monotone predicate lifting over any given finite set A is Λ-definable.

This raises the question: when is a set of predicate liftings expressively complete? In this section we give a partial answer to the question of when a set of liftings is expressively complete, which is quite useful in some special cases. Clearly the set of all monotone liftings for T is expressively complete, and arguably a rather natural choice. However, since all the one-step formulas in the language $\mathtt{ML}^1_\Lambda(A)$ are positive formulas, we cannot hope to express *every* predicate lifting for T in this language: we must make crucial use of monotonicity.

In the definition below, given a set A and a valuation $V : A \to \mathcal{Q}X$ we denote by $\Vdash_V$ the satisfaction relation between X and the set $\mathtt{LF}(A)$ of lattice formulas over A, corresponding to the valuation V.

4.4.2. Definition. Let L be a lax extension for T (see Definition 3.1.1), and let Λ be a set of predicate liftings. We say that Λ is *L-complete* if, for every finite set A and every $\alpha \in \mathsf{T}(\mathtt{LF}(A))$, there exists a formula $\nabla\alpha \in \mathtt{ML}^1_\Lambda(A)$ such that

$$(X, \beta, V) \Vdash_1 \nabla\alpha \text{ iff } (\beta, \alpha) \in L(\Vdash_V),$$

for every one-step model (X, β, V).

Essentially, L-completeness expresses the definability of a Moss-style nabla modality in terms of positive formulas built from liftings in Λ. For example, the standard nabla formula

$$\nabla\{\psi_1, ..., \psi_n\} := \Diamond\psi_1 \wedge ... \wedge \Diamond\psi_n \wedge \Box(\psi_1 \vee ... \vee \psi_n)$$

shows that the single lifting $\Box$ for $\mathcal{P}$ (denoted by $\Box^{\mathcal{P}}$ in Example 2.4.14) is $\overline{\mathcal{P}}$-complete. Simillarly the predicate liftings $\Box^{\mathcal{M}}$ and $\Diamond^{\mathcal{M}}$ (Example 2.4.15) and the relation lifting $\widetilde{\mathcal{M}}$ (Example 2.3.33) for the monotone neighbourhood functor $\mathcal{M}$, we have that $\Box^{\mathcal{M}}$ is $\widetilde{\mathcal{M}}$-complete:

$$\nabla\alpha := \bigwedge_{\Psi\in\alpha} \Box^{\mathcal{M}} \bigvee \Psi \wedge \bigwedge_{f\in\mathsf{ch}} \Diamond^{\mathcal{M}} \bigvee_{\Psi\in\alpha} f(\Psi),$$

where $\alpha \in \mathcal{M}(\mathtt{LF}(A))$, $\Diamond^{\mathcal{M}}$ is the dual of $\Box^{\mathcal{M}}$, and ch is the set of all choice functions for the family of sets α (see [SV10]).

We will now prove that in case T preserves finite sets and L is quasi-functorial (see Definition 3.1.5) and preserves diagonals, L-completeness implies expressive completeness. This immediately shows that $\Box^{\mathcal{P}}$ and $\Box^{\mathcal{M}}$ are expressively complete regarded as liftings for $\mathcal{P}$ and $\mathcal{M}$ respectively, since both functors preserve finite sets, and $\overline{\mathcal{P}}$ and $\widetilde{\mathcal{M}}$ are quasi-functorial lax extensions preserving diagonals [Mar11, Example 3.11.(ii)].

In the following we define the notion of a positive left-to-right one-step L-*bisimulation* between one-step models, and as the main lemma of this section, we show that if L is quasi-functorial and preserves diagonals then positive left-to-right L-bisimulations preserve satisfaction of all monotone predicate liftings.

4.4.3. Definition. A *one-step L-bisimulation* between one-step models (X, α, V) and (Y, β, U) is a relation $R : X \nrightarrow Y$ such that:

- If $(x, y) \in R$ then $m_V(x) = m_U(y)$.
- $(\alpha, \beta) \in LR$.

A relation R is said to be a *positive left-to-right one-step L-bisimulation* if:

- If $(x, y) \in R$ then $m_V(x) \subseteq m_U(y)$.
- $(\alpha, \beta) \in LR$.

4.4.4. LEMMA. *Let L be a quasi-functorial lax extension for T that preserves diagonals and let λ be any monotone predicate lifting. Suppose there is a positive left-to-right L-bisimulation from (X, α, V) to (Y, β, U), where U, V are defined on the finite set of variables A. Then $(X, \alpha, V) \Vdash_1 \lambda$ implies $(Y, \beta, U) \Vdash_1 \lambda$.*

Check Definitions 4.3.16 to recall the definition of $\Vdash_1$.

Proof:
Suppose that $(X, \alpha, V) \Vdash_1 \lambda$; we want to show that $(Y, \beta, U) \Vdash_1 \lambda$. First, let $X + Y$ denote the disjoint union of X and Y and let i_X and i_Y denote the insertion maps of X and Y respectively. Let W denote the valuation on $X + Y$ given by setting, for $a \in A$:

$$W(a) = i_X[V(a)] \cup i_Y[U(a)]$$

Using naturality of λ it is easy to show that $(X + Y, \mathsf{T}i_X(\alpha), W) \Vdash_1 \lambda$, and, to finish the proof it suffices to show that $(X + Y, \mathsf{T}i_Y(\beta), W) \Vdash_1 \lambda$ since we can then apply naturality again to get $(Y, \beta, U) \Vdash_1 \lambda$.

By our assumption, there is a positive left-to-right one-step bisimulation R from (X, α, V) to (Y, β, U). Let R' denote the relation on $X + Y$ defined by

$$R' = \{(i_X(x), i_Y(y)) \mid (x, y) \in R\}.$$

Claim 1. R' is a positive left-to-right one-step bisimulation from the one-step model $(X + Y, \mathsf{T}i_X(\alpha), W)$ to $(X + Y, \mathsf{T}i_Y(\beta), W)$.

Proof of Claim 1: The clause concerning the valuations is trivial, so we only prove that we have $(\mathsf{T}i_X(\alpha), \mathsf{T}i_Y(\beta)) \in LR'$. Note that

$$R' = \widehat{i_X}^{\circ} \,;\, R \,;\, \widehat{i_Y},$$

where $\widehat{f}$ denotes the graph of a function f. We have $(\alpha, \mathsf{T}i_X(\alpha)) \in L(\widehat{i_X})$ so $(\mathsf{T}i_X(\alpha), \alpha) \in L(\widehat{i_X})^{\circ} = L(\widehat{i_X}^{\circ})$. Similarly we have $(\beta, \mathsf{T}(i_Y)(\beta)) \in L(\widehat{i_Y})$. Since $(\alpha, \beta) \in LR$, we get

$$(\mathsf{T}i_X(\alpha), \mathsf{T}i_Y(\beta)) \in L(\widehat{i_X}^{\circ}) \,;\, L(R) \,;\, L(\widehat{i_Y}) \subseteq L(\widehat{i_X}^{\circ} \,;\, R \,;\, \widehat{i_Y}) = LR',$$

and so $(\mathsf{T}i_X(\alpha), \mathsf{T}i_Y(\beta)) \in LR'$ as required to conclude proof of Claim 1.

Now, let R'' be the relation $R' \cup \Delta_{X+Y}$ (where $\Delta_{X+Y} = \widehat{\Delta_{X+Y}}$). It is easy to see that R'' is still a positive left-to-right one-step bisimulation from $(X +$

$Y, \mathsf{T}i_X(\alpha), W)$ to $(X + Y, \mathsf{T}i_Y(\beta), W)$: the fact that $(\mathsf{T}i_X(\alpha), \mathsf{T}i_Y(\beta)) \in LR''$ follows from Claim 1 since $R' \subseteq R''$, and the clause for the valuation W continues to hold since we have only added pairs from the diagonal relation on $X + Y$. Let π_1 and π_2 denote the left and right projections of R'' into $X + Y$, respectively.

Claim 2. There is an element $\gamma \in \mathsf{T}R''$ such that $(\mathsf{T}i_X(\alpha), \gamma) \in L(\widehat{\pi_1})^\circ$ and $(\gamma, \mathsf{T}i_Y(\beta)) \in L(\widehat{\pi_2})$.

Proof of Claim 2: Since the relation R'' is clearly full on $X + Y$, i.e. we have

$$\mathsf{dom}(S) = \mathsf{ran}(S) = X + Y,$$

the projection maps π_1 and π_2 are surjective. Since all set functors preserve epimorphisms, there is $\gamma_1 \in \mathsf{T}R''$ with $\mathsf{T}\pi_1(\gamma_1) = \mathsf{T}i_X(\alpha)$ and $\gamma_2 \in \mathsf{T}R''$ with $\mathsf{T}\pi_2(\gamma_2) = \mathsf{T}i_Y(\beta)$. Hence, $(\mathsf{T}(i_X)(\alpha), \gamma_1) \in L(\widehat{\pi_1})^\circ$ and $(\gamma_2, \mathsf{T}(i_Y)(\beta)) \in L(\widehat{\pi_2})$. Hence, since $R'' = \widehat{\pi_1}^\circ \,;\, \widehat{\pi_2}$, we get:

$$(\mathsf{T}i_X(\alpha), \mathsf{T}i_Y(\beta)) \in L(\widehat{\pi_1}^\circ \,;\, \widehat{\pi_2}) \cap (\mathsf{dom}(L(\widehat{\pi_1}^\circ)) \times \mathsf{ran}(L(\widehat{\pi_2})))$$

Since we assumed that L was quasi-functorial, it follows that:

$$(\mathsf{T}i_X(\alpha), \mathsf{T}i_Y(\beta)) \in L(\widehat{\pi_1}^\circ) \,;\, L(\widehat{\pi_2}) = L(\widehat{\pi_1})^\circ \,;\, L(\widehat{\pi_2}),$$

which is just another way to state the desired conclusion of the claim.

So pick some $\gamma \in \mathsf{T}R''$ as described in Claim 2. Since L preserves diagonals we have $\mathsf{T}\pi_1(\gamma) = \mathsf{T}i_X(\alpha)$ and $\mathsf{T}\pi_2(\gamma) = \mathsf{T}i_Y(\beta)$. We define two valuations $P_1 : A \to \mathcal{Q}(R'')$ and $P_2 : A \to \mathcal{Q}(R'')$ as follows: for a variable $a \in A$, we set

$$P_1(a) = \{(x, y) \in R'' \mid x \in W(a)\} \text{ and } P_2(a) = \{(x, y) \in R'' \mid y \in W(a)\}.$$

In other words, $P_1 = \mathcal{Q}\pi_1 \circ W$, so by naturality together with the fact that we have $(X + Y, \mathsf{T}i_X(\alpha), W) \Vdash_1 \lambda$ and $\mathsf{T}\pi_1(\gamma) = \mathsf{T}i_X(\alpha)$ it follows that $(R'', \gamma, P_1) \Vdash_1 \lambda$. It is not difficult to check that from definition of W and R'', and the fact that R is a positive left-to-right L-bisimulation ($m_V(x) \subseteq m_U(y)$), we get that for each $a \in A$ $P_1(a) \subseteq P_2(a)$. So by monotonicity of λ we have $(R'', \gamma, P_2) \Vdash_1 \lambda$. But $P_2 = \mathcal{Q}\pi_2 \circ W$ so once again we can apply naturality to get $(X + Y, \gamma, W) \Vdash_1 \lambda$ as required. □

We can now state the main result of this section:

4.4.5. Theorem. *Let L be a quasi-functorial lax extension for T that preserves diagonals, and suppose that T restricts to finite sets. Then any L-complete set of monotone predicate liftings for T is expressively complete.*

Proof:
Suppose that λ is any monotone predicate lifting. Let (X, α, V) be any one-step model. We define the map $g : X \to \mathtt{LF}(A)$ by setting $g(x) = \bigwedge m_V(x)$. We set $ch(X, \alpha, V) = \nabla(\mathsf{T}g(\alpha))$. We think of this as the *characteristic formula* of the one-step model (X, α, V).

Claim. Let (Y, β, U) be any one-step model. Then $(Y, \beta, U) \Vdash_1 ch(X, \alpha, V)$ if, and only if, there is a positive left-to-right L-bisimulation from (X, α, V) to (Y, β, U).

Proof of Claim. It is easy to check that $(X, \alpha, V) \Vdash_1 ch(X, \alpha, V)$, so the "if" direction follows from the fact that $ch(X, \alpha, V)$ like all one-step formulas corresponds to a monotone predicate lifting and hence is preserved by positive left-to-right bisimulations. For the "only if" direction, suppose $(Y, \beta, U) \Vdash_1 ch(X, \alpha, V)$. Then $(\beta, \mathsf{T}g(\alpha)) \in L(\Vdash_U)$. Since $(\alpha, \mathsf{T}g(\alpha)) \in L(\widehat{g})$ we have

$$(\alpha, \beta) \in L(\widehat{g} \,;\, \Vdash_U^\circ)$$

so to show that the relation $\widehat{g} \,;\, \Vdash_U^\circ$ is a positive left-to-right one-step bisimulation, it suffices to prove that if $(x, y) \in (\widehat{g} \,;\, \Vdash_U^\circ)$ then we have $m_V(x) \subseteq m_U(y)$. But $(x, y) \in (\widehat{g} \,;\, \Vdash_U^\circ)$ simply means that $y \Vdash_U g(x)$ and the desired conclusion clearly follows and finishes the proof of claim.

Now, since T preserves finite sets, there are only finitely many one-step formulas over the set of variables A of the form $\nabla\alpha$, for $\alpha \in \mathsf{T}(\mathtt{LF}(A))$. Hence, we can safely take disjunctions of arbitrary sets of such formulas. Now, set:

$$\psi = \bigvee\{ch(X, \alpha, V) \mid (X, \alpha, V) \Vdash_1 \lambda\}.$$

We show that ψ defines λ: it is easy to see that λ implies ψ. Conversely, suppose that $(Y, \beta, U) \Vdash_1 \psi$. Then there is a one-step model (X, α, V) such that $(X, \alpha, V) \Vdash_1 \lambda$, and such that $(Y, \beta, V) \Vdash_1 ch(X, \alpha, V)$. Hence, there is a positive left-to-right one-step bisimulation from (X, α, V) to (Y, β, U). By Lemma 4.4.4, it follows that $(Y, \beta, U) \Vdash_1 \lambda$ as required. □

4.4.6. Corollary. *Suppose that* T *preserves finite sets, and let* L *be a quasi-functorial lax extension for* T *that preserves diagonals. Then a set of liftings* Λ *for* T *is expressively complete if, and only if, it is* L*-complete.*

Proof:
The "if" direction follows immediately from the previous theorem. For the "only if" direction, suppose (X, α, V) is a one-step model. It suffices to note that for any $\alpha \in \mathsf{T}\mathtt{LF}(A)$ we obtain a corresponding monotone predicate lifting λ^α over A defined by:

$$\beta \in \lambda^\alpha_X(V) \text{ iff } (\beta, \alpha) \in L(\Vdash_V).$$

Thus this predicate lifting must be Λ-definable if Λ is expressively complete. □

The following useful fact is fairly easy to check:

4.4.7. FACT. If Λ is a one-step expressively complete set of predicate liftings, then $\mathtt{MSO}_\Lambda \equiv \mathtt{MSO}_\mathsf{T}$ and $\mu\mathtt{ML}_\Lambda \equiv \mu\mathtt{ML}_\mathsf{T}$.

Now we can establish the following corollary of Theorem 4.3.24.

4.4.8. COROLLARY. *Suppose Λ is any expressively complete set of monotone predicate liftings for T. Then for every formula of $\mathtt{MSO}_\mathsf{T}$, there exists an equivalent second-order Λ-automaton over T-tree models.*

To conclude this section we recall that in Equation (4.1) we reformulated the Janin-Walukiewicz theorem for the standard μ-calculus and MSO as $\mu\mathtt{ML}_{\{\Diamond\}} \equiv \mathtt{MSO}_{\{\Diamond\}}/{\sim}$ which can be considered as an instance of the characterisation result corresponding to Question 1. Now since we know that the lifting $\Diamond$ for $\mathcal{P}$ (also denoted by $\Diamond^\mathcal{P}$) is expressively complete, by the above Fact we get that $\mathtt{MSO}_{\{\Diamond\}} \equiv \mathtt{MSO}_\mathcal{P}$ and $\mu\mathtt{ML}_{\{\Diamond\}} \equiv \mu\mathtt{ML}_\mathcal{P}$. Hence we can state the Janin-Walukiewicz theorem in yet another form, which corresponds to Question 2.

$$\mu\mathtt{ML}_\mathcal{P} \equiv \mathtt{MSO}_\mathcal{P}/{\sim}$$

4.5 Bisimulation invariance

This section continues the program of [Ven14], making use of the automata-theoretic translation of $\mathtt{MSO}_\mathsf{T}$ we have just established. The gist of our approach is that, in order to characterize a coalgebraic fixpoint logic $\mu\mathtt{ML}_\mathsf{T}$ as the bisimulation-invariant fragment of $\mathtt{MSO}_\mathsf{T}$, it suffices to establish a certain type of translation between the corresponding one-step languages. First we need some definitions.

4.5.1. DEFINITION. Given sets X, Y, a mapping $h : X \to Y$ and a valuation $V : A \to \mathcal{Q}(Y)$, we define the valuation $V_{[h]} : A \to \mathcal{Q}(X)$ by setting $V_{[h]}(b) = h^{-1}[V(b)]$ for each $b \in A$. In other words:

$$V_{[h]} := \mathcal{Q}h \circ V.$$

Note that for a pair of composable maps f, g, we have $V_{[f\circ g]} = (V_{[g]})_{[f]}$. The most important concept that we take from [Ven14] is that of a *uniform translation* (called *uniform correspondence* in [Ven14]).

4.5.2. DEFINITION. A *one-step frame* is a pair (X, α) with $\alpha \in \mathsf{T}X$. A *homomorphism* of one-step frames $h : (X', \alpha') \to (X, \alpha)$ is a map $h : X' \to X$ with $\mathsf{T}h(\alpha') = \alpha$. A one-step frame (X', α') together with a homomorphism $h : (X', \alpha') \to (X, \alpha)$ is called a *cover* of (X, α).

We can now define the notions of uniform translations and uniform constructions:

4.5.3. Definition. Given a functor T, a *uniform construction* for T is an assignment of a cover $h_\alpha : (X_*, \alpha_*) \to (X, \alpha)$ to every one-step frame (X, α).

We will usually denote the uniform construction consisting of an assignment of covers $h_\alpha : (X_*, \alpha_*) \to (X, \alpha)$ simply by $(-)_*$.

4.5.4. Definition. We say that an abstract one-step language $\mathtt{L}$ *admits uniform translations* with respect to Λ if, for any finite set A and any finite set Γ of abstract one-step formulas in $\mathtt{L}(A)$, there exists a uniform construction $(-)_*$ and a translation $(-)^* : \Gamma \to \mathtt{ML}^1_\Lambda(A)$ such that for any one-step model (X, α, V) and any $\varphi \in \Gamma$, we have

$$(X, \alpha, V) \Vdash_1 \varphi^* \text{ iff } (X_*, \alpha_*, V_{[h_\alpha]}) \Vdash_1 \varphi,$$

where h_α is the cover assigned to (X, α) by the construction. The pair consisting of the translation $(-)^*$ and the uniform construction $(-)_*$ will be referred to as a *uniform translation* for the set of formulas Γ.

As an example, consider the disjunctive formulas introduced by Walukiewicz.

4.5.5. Definition. For a P-chromatic second-order automaton $\mathbb{A} = (A, \Theta, a_I, \Omega)$ and a uniform translation for the finite set $\Theta[A \times \mathcal{P}(\mathsf{P})]$, we define a corresponding modal Λ-automaton $\mathbb{A}^* = (A, \Theta^*, a_I, \Omega)$, with Θ^* given by $\Theta^*(a, c) := (\Theta(a, c))^*$.

The proof of the next result closely follows that of the main result in [Ven14]. But since here we work with a more general class of models, we need an "unravelling"-like component to turn a T-model to a tree-like model.

4.5.6. Proposition. *Assume that $\mathtt{SO}^1_\Lambda$ admits uniform translations, and let $\mathbb{A}$ be a second-order Λ-automaton. Then for each pointed T-model $(\mathbb{S}, s)$ there is a T-tree model $(\mathbb{T}, R, t)$, with a T-model homomorphism f from $\mathbb{T}$ to $\mathbb{S}$, mapping t to s, and such that*

$$\mathbb{A} \textit{ accepts } (\mathbb{T}, R, t) \textit{ iff } \mathbb{A}^* \textit{ accepts } (\mathbb{S}, s).$$

Proof:
Consider any given pointed T-model $(\mathbb{S}_1, s_1)$ where $\mathbb{S}_1 = (S_1, \sigma_1, V_1)$. We are going to construct a T-tree model $(\mathbb{S}_2, R, s_2)$, $\mathbb{S}_2 = (S_2, \sigma_2, V_2)$, together with a model homomorphism from the underlying pointed T-model $\mathbb{S}_2$ to $\mathbb{S}_1$ mapping s_2 to s_1, and such that $\mathbb{A}$ accepts the T-tree model $(\mathbb{S}_2, s_2)$ if and only if $\mathbb{A}^*$ accepts the pointed T-model $(\mathbb{S}_1, s_1)$.

We construct this T-tree model as follows: for each $u \in S_1$, we define an associated pair (X_u, α_u) as follows: set $X_u = (S_1)_*$ and set $\alpha_u = \sigma_1(u)_*$. Observe that, by the construction of these one-step models, for each $u \in S_1$, there is a mapping $\xi_u : X_u \to S_1$ such that:

1. $\mathsf{T}(\xi_u)(\alpha_u) = \sigma_1(u)$,
2. For each valuation $U : A \to \mathcal{Q}(S_1)$, every $u \in S_1$ and every one-step formula $\Theta(a, c)$ appearing in $\mathbb{A}$, we have
$$(S_1, \sigma_1(u), U) \Vdash_1 \Theta^*(a, c) \text{ iff } (X_u, \alpha_u, U_{[\xi_u]}) \Vdash_1 \Theta(a, c).$$

The map ξ_u is given by $h_{\sigma_1(u)}$. We now construct the T-tree model $(S_2, R, \sigma_2, s_2, V_2)$ as follows: first, consider the set of all non-empty finite (non-empty) tuples $(v_1, ..., v_n)$ of elements in
$$\{s_1\} \cup \bigcup_{u \in S_1} X_u,$$
such that $v_1 = s_1$. We define, by induction, for each natural number $n > 0$ a subset M_n of this set, and a mapping $\gamma_n : M_n \to S_1$, as follows:

- Set $M_1 = \{(s_1)\}$, and define $\gamma_1(s_1) = s_1$.
- Set $M_{n+1} = \{\vec{v} \cdot w \mid \vec{v} \in M_n, w \in X_{\gamma_n(\vec{v})}\}$. Define $\gamma_{n+1}(\vec{v} \cdot w) = \xi_{\gamma_n(\vec{v})}(w)$.

Here, we write $\vec{v} \cdot w$ to denote the tuple $(v_1, ..., v_n, w)$ if $\vec{v} = (v_1, ..., v_n)$. Set $S_2 = \bigcup_{n>0} M_n$, and define $\gamma = \bigcup_{n>0} \gamma_n$. Define the relation $R \subseteq S_2 \times S_2$ to be
$$\{(\vec{v}, \vec{v} \cdot w) \mid \vec{v} \in S_2, w \in X_{\gamma(\vec{v})}\}$$
Note that there is, for every $\vec{v} \in S_2$, a bijection $i_{\vec{v}} : X_{\gamma(\vec{v})} \to R(\vec{v})$ given by $w \mapsto \vec{v} \cdot w$. Note also that, for each $\vec{v} \in S_2$, we have
$$\gamma \circ i_{R(\vec{v}), S_2} \circ i_{\vec{v}} = \xi_{\gamma(\vec{v})}$$

With this in mind, we define the coalgebra structure σ_2 by setting:
$$\sigma_2(\vec{v}) = \mathsf{T}(i_{R(\vec{v}), S_2} \circ i_{\vec{v}})(\alpha_{\gamma(\vec{v})})$$
Note that $\sigma_2(\vec{v})|_{R(\vec{v})} = \mathsf{T}i_{\vec{v}}(\alpha_\gamma(\vec{v}))$. Finally, set s_2 to be the unique singleton tuple with sole element s_1, and define the valuation V_2 by setting $m_2(\vec{v}) = m_1(\gamma(\vec{v}))$.

Clearly, $(S_2, R, \sigma_2, s_2, V_2)$ is a T-tree model. Denote the underlying T-model by $\mathbb{S}_2$. We can then prove the following two claims:

Claim 1. The map γ is a T-model homomorphism from $\mathbb{S}_2$ to $\mathbb{S}_1$.

Claim 2. $\mathbb{A}$ accepts $(S_2, R, \sigma_2, s_2, V_2)$ iff $\mathbb{A}^*$ accepts $\mathbb{S}_1$.

Proof of Claim 1: The map γ clearly respects the truth values of all propositional atoms, and $\gamma(s_2) = s_1$. It suffices to show that γ is a coalgebra morphism, i.e. that $\mathsf{T}\gamma(\sigma_2(\vec{v})) = \sigma_1(\gamma(\vec{v}))$ for all $\vec{v}$. Pick any $\vec{v} \in S_2$. We have:
$$\begin{aligned}
\mathsf{T}\gamma(\sigma_2(\vec{v})) &= \mathsf{T}\gamma \circ \mathsf{T}i_{R(\vec{v}), S_2} \circ \mathsf{T}(i_{\vec{v}})(\alpha_{\gamma(\vec{v})}) \\
&= \mathsf{T}(\gamma \circ i_{R(\vec{v}), S_2} \circ i_{\vec{v}})(\alpha_{\gamma(\vec{v})}) \\
&= \mathsf{T}(\xi_{\gamma(\vec{v})})(\alpha_{\gamma(\vec{v})}) \\
&= \sigma_1(\gamma(\vec{v}))
\end{aligned}$$

as required.

Proof of Claim 2: We have two parts that need to be proved here:

First part: $\mathbb{A}$ **accepts** $(S_2, R, \sigma_2, s_2, V_2)$ **implies** $\mathbb{A}^*$ **accepts** $(\mathbb{S}_1, s_1)$

Suppose that χ is a (positional) winning strategy in the acceptance game for $\mathbb{A}$ and $(S_2, R, \sigma_2, s_2, V_2)$. We are going to define a strategy χ^* for $\exists$ in the acceptance game for $\mathbb{A}^*$ and $\mathbb{S}_1$, with the property that for any χ^*-guided partial match

$$\rho = (a_1, s_1^1), ..., (a_n, s_n^1)$$

of length n with $s_i^i \in S_1$ and $s_1^1 = s_1$, there exists a χ-guided partial match

$$\rho_* = (a_1, s_1^2), ..., (a_n, s_n^2)$$

with $s_i^2 \in S_2$, $s_1^2 = s_2$ and $\gamma(s_i^2) = s_i^1$ for each index i, and chosen in such a way that if a χ^*-guided match ρ' is an extension of ρ, then the χ-guided match ρ'_* is an extension of the χ-guided match ρ_*.

We define the strategy χ^* by induction on the length of a partial match. For the partial match ρ consisting of the single position (a_I, s_1^1) we let ρ_* consist of the single position (a_I, s_2), and we define the valuation $\chi^*(\rho) : A \to \mathcal{Q}(S_1)$ by setting, for each $b \in A$,

$$\chi^*(\rho)(b) = \gamma[\chi(a_I, s_2)(b)].$$

Similarly, suppose that χ^* has been defined for all matches of length less than n, and let ρ be any match of length $n > 1$. If ρ is not χ^*-guided, then we can define $\chi^*(\rho)(b) = \emptyset$ for all $b \in A$. If ρ is χ^*-guided, then write

$$\rho = (a_1, s_1^1), ..., (a_n, s_n^1)$$

Since this match is χ^*-guided, by the induction hypothesis there is a χ-guided partial match

$$\rho_* = (a_1, s_1^2), ..., (a_n, s_n^2)$$

with $\gamma(s_i^2) = s_i^1$ for all i. We set, for each $b \in A$,

$$\chi^*(\rho)(b) = \gamma[\chi(a_n, s_n^2)(b)].$$

Clearly, with this definition, the induction hypothesis continues to hold for all χ^*-guided matches of length $n + 1$. Furthermore, we note that, whenever ρ is a χ^*-guided match of length n, the move $\chi^*(\rho)$ is legal: since the move $\chi(a_n, s_n^2)$ must be legal, we have

$$(R(s_n^2), \sigma_2(s_n^2) \restriction_{R(s_n^2)}, \chi(a_n, s_n^2)) \Vdash_1 \Theta(a_n, m_2(s_n^2)).$$

But since
$$\sigma_2(s_n^2)\restriction_{R(s_n^2)} = \mathsf{T}i_{s_n^2}(\alpha_{\gamma(s_n^2)}) = \mathsf{T}i_{s_n^2}(\alpha_{s_n^1}),$$
we see that $i_{s_n^2}$ is an isomorphism between the models $(R[s_n^2], \sigma_2(s_n^2)\restriction_{R(s_n^2)}, \chi(a_n, s_n^2))$ and $(X_{s_n^1}, \alpha_{s_n^1}, \chi(a_n, s_n^2)_{[i_{s_n^2}]})$, so we have:

$$(X_{s_n^1}, \alpha_{s_n^1}, \chi(a_n, s_n^2)_{[i_{s_n^2}]}) \Vdash_1 \Theta(a_n, m_2(s_n^2)).$$

For all $b \in A$, we have:

$$\begin{array}{rcl} \chi(a_n, s_n^2)_{[i_{s_n^2}]}(b) & \subseteq & \xi_{s_n^2}^{-1}[\xi_{s_n^2}[\chi(a_n, s_n^2)_{[i_{s_n^2}]}(b)]] \\ & = & \xi_{s_n^2}^{-1}[\gamma[\chi(a_n, s_n^2)(b)]] \\ & = & \xi_{s_n^2}^{-1}[\chi^*(\rho)(b)] \\ & = & \chi^*(\rho)_{[\xi_{(s_n^2)}]}(b) \end{array}$$

So by monotonicity we get:

$$(X_{s_n^1}, \alpha_{s_n^2}, \chi^*(\rho)_{[\xi_{s_n^2}]}) \Vdash_1 \Theta(a_n, m_2(s_n^2)).$$

Hence, since $\Theta^*(a_n, m_2(s_n^2))$ is the uniform translation of $\Theta(a_n, m_2(s_n^2))$, we get

$$(S_1, \sigma_1(s_n^2), \chi^*(\rho)) \Vdash_1 \Theta^*(a_n, m_2(s_n^2)),$$

as required. It is now easy to show that $\exists$ never gets stuck in a χ^*-guided partial match, and we also see that every infinite χ^*-guided match:

$$(a_I, s_1^1), (a_2, s_2^1), (a_3, s_3^1), ...$$

corresponds to an infinite χ-guided match:

$$(a_I, s_1^2), (a_2, s_2^2), (a_3, s_3^2), ...$$

and so since $\exists$ wins every infinite χ-guided match, she wins every infinite χ^*-guided match as well.

Second part: $\mathbb{A}^*$ accepts $(\mathbb{S}_1, s_1)$ implies $\mathbb{A}$ accepts $(S_2, R, \sigma_2, s_2, V_2)$

Let χ^* be a winning strategy for $\exists$ in the acceptance game for $\mathbb{A}^*$ and $\mathbb{S}_1$. We define the strategy χ as follows: given a partial match:

$$\rho = (a_1, s_1^2), ..., (a_n, s_n^2)$$

set $\chi(\rho) = \chi^*(\gamma(\rho))_{[\gamma]}$, where $\gamma(\rho)$ is the match:

$$(a_1, \gamma(s_1^2)), ..., (a_n, \gamma(s_n^2))$$

It is straightforward to check that, whenever ρ is a χ-guided match, $\gamma(\rho)$ is a χ^*-guided match, and that the move $\chi^*(\gamma(\rho))$ is legal if and only if the move $\chi(\rho)$ is legal. It follows that $\exists$ never gets stuck in a χ-guided partial match, and that she wins every infinite χ-guided match, since an infinite χ-guided match

$$(a_I, s_1^2), (a_2, s_2^2), (a_3, s_3^2), \ldots$$

corresponds to an infinite χ^*-guided match:

$$(a_I, \gamma(s_1^2)), (a_2, \gamma(s_2^2)), (a_3, \gamma(s_3^2)), \ldots$$

This concludes the proof.

The lemma now follows by combining the two claims. □

From this we get the following result, which is the characterisation theorem corresponding to Question 1. Here, we use the fact that every $\mathtt{MSO}_\Lambda$-formula is equivalent to a monotone second-order automaton over T-tree models, so that it suffices to find translations of monotone one-step formulas.

4.5.7. Theorem (Auxiliary Characterization Theorem I). *Let Λ be an expressively complete set of monotone predicate liftings for a set functor T, and assume that the monotone fragment of the second-order one-step language $\mathtt{SO}^1_\Lambda$ admits uniform translations. Then:*

$$\mu\mathtt{ML}_\Lambda \equiv \mathtt{MSO}_\Lambda/{\sim}.$$

Proof:
Given a formula φ of $\mathtt{MSO}_\Lambda$ which is invariant under behavioural equivalence, let $\mathbb{A}$ be an equivalent monotone second-order automaton, and let $(-)^*$ be a uniform translation of the monotone fragment of the one-step language $\mathtt{SO}^1_\Lambda$. Let ψ be a formula in $\mu\mathtt{ML}_\Lambda$ equivalent to the automaton $\mathbb{A}^*$ over T-tree models. Then for any pointed T-model $\mathbb{S}, s$, let $(\mathbb{T}, R, t)$ be the T-tree model provided by Proposition 4.5.6. Then we get:

$\mathbb{S}, s \Vdash \varphi$	$\Leftrightarrow$	$\mathbb{T}, t \Vdash \varphi$	φ is invariant under behavioural equivalence
	$\Leftrightarrow$	$\mathbb{T}, t \Vdash \mathbb{A}$	φ is equivalent to $\mathbb{A}$ over T-tree models
	$\Leftrightarrow$	$\mathbb{S}, s \Vdash \mathbb{A}^*$	Proposition 4.5.6
	$\Leftrightarrow$	$\mathbb{S}, s \Vdash \psi$	ψ is equivalent to $\mathbb{A}^*$

and the proof is done. □

4.5.1 Adequate uniform constructions

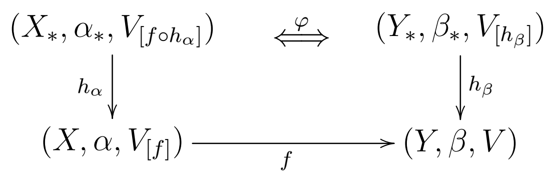
The formulation of Theorem 4.5.7 raises the question for which class of functors there always exists an expressively complete set of monotone predicate liftings such that $\mathtt{SO}^1_\Lambda$ admits a uniform translation? One may think that to answer this question we need to observe both syntax and semantics of formulas in $\mathtt{SO}^1_\Lambda$ closely, since the uniform translation involves a translation on the syntactic side and a uniform construction on the semantic side. However, we claim that the model-theoretic constraint on uniform constructions for one-step formulas will be enough and that the syntactic translation will come for free.

4.5.8. Definition. Let φ be any one-step formula in $\mathtt{SO}^1_\Lambda(A)$. Then a uniform construction $(-)_*$ for T is called *adequate* for φ if, for any pair of one-step frames (X, α) and (Y, β), any one-step frame homomorphism $f : (X, \alpha) \to (Y, \beta)$ and any valuation $V : A \to \mathcal{Q}(Y)$, we have the following:

$$(X_*, \alpha_*, V_{[f \circ h_\alpha]}) \Vdash_1 \varphi \text{ iff } (Y_*, \beta_*, V_{[h_\beta]}) \Vdash_1 \varphi \tag{4.2}$$

We say that the construction is adequate for a set of formulas Γ if it is adequate for every member of Γ.

The following diagram illustrates equation 4.2:



We are now ready to state our first main theorem:

4.5.9. Theorem (Coalgebraic Bisimulation Invariance I.). *Let T be any set functor. If T admits an adequate uniform construction for every finite set of second-order one-step formulas Γ, then:*

$$\mu\mathtt{ML}_\mathsf{T} \equiv \mathtt{MSO}_\mathsf{T}/{\sim}.$$

Proof:
By Theorem 4.5.7 it suffices to show that the monotone fragment of $\mathtt{SO}^1_\mathsf{T}$ admits uniform translations. Let Γ be a finite set of monotone formulas of $\mathtt{SO}^1_\mathsf{T}(A)$, for some finite set A, and suppose the uniform construction $(-)_*$ is adequate for Γ. Given $\varphi \in \Gamma$ we define a monotone predicate lifting λ over A by setting

$$\alpha \in \lambda_X(V) \text{ iff } (X_*, \alpha_*, V_{[h_\alpha]}) \Vdash_1 \varphi.$$

It is easy to check that this lifting is monotone, and naturality of λ is a direct consequence of the equation 4.2. If A has n elements and we list these as $(a_1, ..., a_n)$, then as we discussed at the bottom of page 91, we can view λ as a a formula in $\mathtt{ML}^1_\Lambda(A)$ of the form $\lambda'(a_1, ..., a_n)$, where λ' is the n-place predicate lifting corresponding to λ with this given ordering of A. We now get a uniform translation by mapping φ to the formula $\lambda'(a_1, ..., a_n)$. □

The following result for an expressively complete set Λ of predicate liftings follows from the fact that every formula in $\mathtt{SO}^1_\mathsf{T}(A)$ is semantically equivalent to a formula in $\mathtt{SO}^1_\Lambda(A)$. So it is enough to find an adequate uniform construction for every finite set of formulas in $\mathtt{SO}^1_\Lambda(A)$.

4.5.10. COROLLARY. *Let T be any set functor, and let Λ be an expressively complete set of monotone predicate liftings for T. If T admits an adequate uniform construction for every finite set of formulas of $\mathtt{SO}^1_\Lambda$, then*

$$\mu\mathtt{ML}_\Lambda \equiv \mathtt{MSO}_\Lambda/\!\sim.$$

If T preserves weak pullbacks (see Section 2.3.1), the Barr extension $\overline{\mathsf{T}}$ is a (functorial) lax extension and we can reformulate equation 4.2 as the following:

4.5.11. PROPOSITION. *Suppose that T preserves weak pullbacks, and let $(-)_*$ be a uniform construction for T. Then $(-)_*$ is adequate for φ if, and only if, for every pair of one-step $\overline{\mathsf{T}}$-bisimilar one-step models (X, α, V) and (Y, β, U) we have:*

$$(X_*, \alpha_*, V_{[h_\alpha]}) \Vdash_1 \varphi \Leftrightarrow (Y_*, \beta_*, U_{[h_\beta]}) \Vdash_1 \varphi.$$

4.5.2 Weakly adequate uniform constructions

As we shall see in Section 4.6 not all functors of interest admit adequate uniform constructions. The famous problematic (in the context of coalgebras) monotone neighbourhood functor $\mathcal{M}$ does not admit an adequate uniform construction. Hence we cannot compare the expressive power of the monotone μ-calculus and monotone monadic second order logic applying Theorem 4.5.9. Hence we require a second version of our main result, that makes use of what we will call *weakly adequate* uniform constructions. These are similar to adequate uniform constructions, except that the condition 4.2 is only required to hold in those cases where the bottom one-step homomorphism in the diagram is surjective.

4.5.12. DEFINITION. Let φ be any one-step formula in $\mathtt{SO}^1_\Lambda(A)$, and let $(-)_*$ be a uniform construction for T that assigns a *surjective* cover to each one-step frame. Then this construction is said to be *weakly adequate* for φ if, for any pair of one-step frames (X, α) and (Y, β), any *surjective* one-step frame homomorphism $f : (X, \alpha) \to (Y, \beta)$ and any valuation $V : A \to \mathcal{Q}(Y)$ equation 4.2 holds. A uniform construction is called weakly adequate for a finite set of formulas Γ if it is weakly adequate for every member of Γ.

Our goal is now to prove the following result. To recall the notion of quasi-functorial lax extension see Definitions 3.1.1 and 3.1.5

4.5.13. Theorem (Coalgebraic Bisimulation Invariance II). *Let* T *be a set functor that preserves finite sets and admits a quasi-functorial lax extension* L *that preserves diagonals. Let* Λ *be an expressively complete set of monotone predicate liftings for* T*. If* T *admits a weakly adequate uniform construction for every finite set of formulas of* $\mathtt{SO}^1_\Lambda$*, then*

$$\mu\mathtt{ML}_\Lambda \equiv \mathtt{MSO}_\Lambda/{\sim}$$

If T admits a quasi-functorial lax extension L, we can reformulate the condition 4.2 as follows:

4.5.14. Definition. Let L be a lax extension for T and let R be a one-step L-bisimulation between one-step models (X, α, V) and (Y, β, U). Then R is said to be a *global* one-step bisimulation if:

forth for every $u \in X$ there is a $v \in Y$ with uRv,
back for every $v \in Y$ there is a $u \in X$ with uRv.

4.5.15. Proposition. *Let* $(-)_*$ *be a uniform construction for* T *that assigns a surjective cover to every one-step frame, and let* L *be a quasi-functorial lax extension for* T*. Then* $(-)_*$ *is weakly adequate for the formula* φ *if and only if, for every pair of globally one-step* L*-bisimilar one-step models* (X, α, V) *and* (Y, β, U)*, we have:*

$$(X_*, \alpha_*, V_{[h_\alpha]}) \Vdash_1 \varphi \Leftrightarrow (Y_*, \beta_*, U_{[h_\beta]}) \Vdash_1 \varphi.$$

Proof:
The direction from right to left follows since the graph of a surjective one-step model homomorphism is clearly a global one-step L-bisimulation. For the converse, suppose that $(-)_*$ is weakly adequate, and let R be a global L-bisimulation between (X, α, V) and (Y, β, U). Since R is global its projection maps π_1, π_2 to X and Y respectively are surjective. Since set functors preserve surjective maps, and since $\widehat{\mathsf{T}f} \subseteq L(\widehat{f})$ for all maps f, we get that $\mathsf{dom}(L(\widehat{\pi_1}^\circ)) = \mathsf{T}X$ and $\mathsf{ran}(L(\widehat{\pi_2})) = \mathsf{T}Y$. Since L is quasi-functorial we now find $\gamma \in \mathsf{T}R$ such that $\mathsf{T}\pi_1(\gamma) = \alpha$ and $\mathsf{T}\pi_2(\gamma) = \beta$. Furthermore, we have $V_{[\pi_1]} = U_{[\pi_2]}$, since for all $(u, v) \in R$:

$$(u, v) \in V_{[\pi_1]}(a) \text{ iff } u \in V(a) \text{ iff } v \in U(a) \text{ iff } (u, v) \in U_{[\pi_2]}(a).$$

Hence $V_{[\pi_1 \circ h_\gamma]} = U_{[\pi_2 \circ h_\gamma]}$. Using the fact that $(-)_*$ is weakly uniform and the fact that the projection maps for R are surjective, we get

$$\begin{aligned}(X_*, \alpha_*, V_{[h_\alpha]}) \Vdash_1 \varphi \quad &\Leftrightarrow \quad (R_*, \gamma_*, V_{[\pi_1 \circ h_\gamma]}) \Vdash_1 \varphi \\ &\Leftrightarrow \quad (R_*, \gamma_*, U_{[\pi_2 \circ h_\gamma]}) \Vdash_1 \varphi \\ &\Leftrightarrow \quad (Y_*, \beta_*, U_{[h_\beta]}) \Vdash_1 \varphi\end{aligned}$$

as required.

The following diagram may be helpful to understand the above equations:

$$
\begin{array}{ccccccc}
(X_*, \alpha_*, V_{[h_\alpha]}) & \overset{\varphi}{\Longleftrightarrow} & (R_*, \gamma_*, V_{[\pi_1 \circ h_\gamma]}) & \overset{\varphi}{\Longleftrightarrow} & (R_*, \gamma_*, U_{[\pi_2 \circ h_\gamma]}) & \overset{\varphi}{\Longleftrightarrow} & (Y_*, \beta_*, U_{[h_\beta]}) \\
\downarrow{\scriptstyle h_\alpha} & & \downarrow{\scriptstyle h_\gamma} & & \downarrow{\scriptstyle h_\gamma} & & \downarrow{\scriptstyle h_\beta} \\
(X, \alpha, V) & \xleftarrow[\pi_1]{} & (R, \gamma, V_{[\pi_1]}) & & (R, \gamma, U_{[\pi_2]}) & \xrightarrow{\pi_2} & (Y, \beta, U)
\end{array}
$$

□

The key to prove our second main characterization theorem, Theorem 4.5.13, is to work with a modified version of the functor T instead of T itself. The idea is to define this modified functor T^{s} in such a way that one-step frame homomorphisms for T^{s} essentially correspond to *surjective* one-step frame homomorphisms for the original functor T, so that a weakly adequate uniform construction for T provides us with an *adequate* uniform construction for T^{s}. By Theorem 4.5.9 this gives us a characterization theorem for the language $\mu\mathtt{ML}_{\mathsf{T}^{\mathsf{s}}}$, and from this we want to deduce a characterization theorem for the original language $\mu\mathtt{ML}_{\mathsf{T}}$. But in order to do this we need to understand precisely how the two languages $\mu\mathtt{ML}_{\mathsf{T}}$ and $\mu\mathtt{ML}_{\mathsf{T}^{\mathsf{s}}}$ are related. It turns out that we can characterize $\mu\mathtt{ML}_{\mathsf{T}}$ neatly as a fragment of $\mu\mathtt{ML}_{\mathsf{T}^{\mathsf{s}}}$, and this will provide the link we need to obtain Theorem 4.5.13.

4.5.2.1 The supported companion of a functor

4.5.16. Definition. The *supported companion* T^{s} of T is the sub-functor of $\mathcal{P} \times \mathsf{T}$ defined by:

$$\mathsf{T}^{\mathsf{s}}(X) = \{(Z, \alpha) \in \mathcal{P}X \times \mathsf{T}X \mid Z \text{ supports } \alpha\}.$$

It is easy to check that this is indeed a well-defined subfunctor of $\mathcal{P} \times \mathsf{T}$, i.e.; for all sets X we have $\mathsf{T}^{\mathsf{s}}(X) \subseteq \mathcal{P}X \times \mathsf{T}X$, and for all maps $f : X \to Y$, the map $\mathsf{T}^{\mathsf{s}}(f)$ is the restriction of $(\mathcal{P} \times \mathsf{T})(f)$ to $\mathsf{T}^{\mathsf{s}}(Y)$. The first condition holds by definition. The second one holds since for any map $f : X \to Y$, any $\alpha \in \mathsf{T}X$ and any set $Z \subseteq X$, the image $f[Z] = \mathcal{P}f(Z)$ is a support for $\mathsf{T}f(\alpha)$ whenever Z is a support for α. The reader may note that what we have called "T-tree models" are actually special instances of T^{s}-models. We will show that this construction repairs the monotone neighborhood functor, so that the supported companion $\mathcal{M}^s$ of $\mathcal{M}$ admits an adequate uniform construction. Interestingly, the same construction happens to repair weak pullback preservation:

4.5.17. Proposition. *The functor $\mathcal{M}^s$ preserves weak pullbacks.*

We leave the verification of this to the reader; the argument is similar to the reasoning in [MV12] used to establish the existence of a well-behaved relation lifting for $\mathcal{M}$. In the next section we will prove a more general version of this result for T^s.

Note that the functor T^{s} comes equipped with a unary predicate lifting $\Box^{\mathsf{s}}$ defined by:

$$\Box^{\mathsf{s}}_X(Y) := \{(Z,\alpha) \in \mathsf{T}^{\mathsf{s}}X \mid Z \subseteq Y\}$$

The Boolean dual of this lifting will be denoted by $\Diamond^{\mathsf{s}}$.

4.5.18. Lemma. *Let* T *be any functor and* Λ *a set of predicate liftings for* T*. If* T *admits a weakly adequate uniform construction for every finite set of formulas in* $\mathtt{SO}^1_\Lambda$*, then its supported companion* T^{s} *admits a uniform construction for every finite set of formulas in* $\mathtt{SO}^1_{\Lambda\cup\{\Box^{\mathsf{s}}\}}(A)$*.*

Proof:
Fix a finite set of variables A. First, we note as a quite trivial observation that there is a translation

$$t : \mathtt{SO}^1_{\Lambda\cup\{\Box^{\mathsf{s}}\}}(A) \to \mathtt{SO}^1_\Lambda(A)$$

such that, for every one-step T-model (X,α,V), and every one-step formula $\varphi \in \mathtt{SO}^1_{\Lambda\cup\{\Box^{\mathsf{s}}\}}(A)$, we have the following condition:

$$(X,\alpha,V) \Vdash_1 t(\varphi) \text{ iff } (X,(X,\alpha),V) \Vdash_1 \varphi. \tag{4.3}$$

This translation can be defined by a straightforward induction on the complexity of formulas. The only interesting case is an atomic formula of the form $\Box^{\mathsf{s}}p$, for which we set

$$t(\Box^{\mathsf{s}}p) := \forall q.q \subseteq p.$$

With this in mind, let Γ be a finite set of formulas of $\mathtt{SO}^1_{\Lambda\cup\{\Box^{\mathsf{s}}\}}$, and let $(-)_*$ be a weakly adequate uniform construction for the set of formulas $t[\Gamma]$. Note that we can extend $(-)_*$ to a uniform construction for T^{s} in the following way: given a one-step frame $(X,(S,\alpha))$ for T^{s}, where $\alpha \in \mathsf{T}X$ and S is a support for α, the uniform construction $(-)_*$ gives a surjective cover $h_{(\alpha|_S)} : (S_*,(\alpha|_S)_*) \to (S,(\alpha|_S))$, and we let the new uniform construction assign to $(X,(S,\alpha))$ the cover

$$i_{S,X} \circ h_{(\alpha|_S)} : (S_*,(S_*,(\alpha|_S)_*)) \to (X,(S,\alpha))$$

The condition that $\mathcal{P}(i_{S,X} \circ h_{(\alpha|_S)})(S^*) = S$ follows by surjectivity of the map $h_{(\alpha|_S)} : S_* \to S$.

To check that this construction is adequate for each $\varphi \in \Gamma$, consider a one-step frame homomorphism $f : (X,(S,\alpha)) \to (X',(S',\alpha'))$ and let $V : A \to \mathcal{Q}(X')$ be a valuation. Let $U : A \to \mathcal{Q}(S')$ be the valuation defined by $U(a) = V(a) \cap S'$. To keep notation bearable we abbreviate $h_{(\alpha|_S)} = h$ and $h_{(\alpha'|_{S'})} = h'$. The map $f|_S$ is easily checked to be a surjective one-step homomorphism from $(S,\alpha|_S)$ to $(S',\alpha'|_{S'})$, so by weak adequacy of $(-)_*$ for $t(\varphi)$ we get:

$$(S_*,(\alpha|_S)_*,U_{[f|_S\circ h]}) \Vdash_1 t(\varphi) \Leftrightarrow (S'_*,(\alpha'|_{S'})_*,U_{[h']}) \Vdash_1 t(\varphi).$$

But it is easily checked that:

$$U_{[f|_S \circ h]} = V_{[f \circ i_{S,X} \circ h]} \text{ and } U_{[h']} = V_{[i_{S',X'} \circ h']}.$$

Together with the equation 4.3 for the translation t, this gives us:

$$(S_*, (S_*, (\alpha|_S)_*), V_{[f \circ i_{S,X} \circ h]}) \Vdash_1 \varphi \Leftrightarrow (S'_*, (S'_*, (\alpha'|_{S'})_*), V_{[i_{S',X'} \circ h']}) \Vdash_1 \varphi,$$

which shows that the uniform construction we defined for T^{s} is indeed adequate for φ, as required. □

4.5.3 Characterizing $\mu\mathtt{ML}_\Lambda$ inside $\mu\mathtt{ML}_{\Lambda \cup \{\Box^{\mathsf{s}}\}}$

A little issue that we need to address, before we can proceed to characterize the monotone μ-calculus, is just how the language $\mu\mathtt{ML}_\Lambda$ is related to $\mu\mathtt{ML}_{\Lambda \cup \{\Box^{\mathsf{s}}\}}$ for a given set of liftings Λ for T. The rest of this section will provide the answer, and give a characterization theorem for $\mu\mathtt{ML}_\Lambda$ as the fragment of $\mu\mathtt{ML}_{\Lambda \cup \{\Box^{\mathsf{s}}\}}$ that is invariant for behavioural equivalence. Formally we shall write $(\mathbb{S}, s) \sim (\mathbb{S}', s')$, for T^s-models $\mathbb{S}$ and $\mathbb{S}'$, to say that the respective underlying pointed T-models are behaviourally equivalent. To distinguish this from actual behavioural equivalence in the sense of the companion functor T^s, we write $(\mathbb{S}, s) \sim^s (\mathbb{S}', s')$ to say that these pointed models are behaviourally equivalent as T^s-models. We note that the behavioural equivalence relation $\sim^s$ between T^s-models amount to behavioural equivalence with respect to functors $\mathcal{P}$ and T for the underlying $\mathcal{P}$-models and T-models respectively.

We shall borrow a result from [FLV10]:

4.5.19. FACT. [FLV10] For any set of liftings Λ for any set functor T, the logic $\mu\mathtt{ML}_\Lambda$ has the finite model property.

4.5.20. DEFINITION. Given a lax extension L for functor T, we define a relation lifting L^{s} for T^{s} by setting:

$$((S, \alpha), (S', \alpha')) \in L^{\mathsf{s}}R \text{ iff } (S, S') \in \overline{\mathcal{P}}R \text{ and } (\alpha, \alpha') \in LR,$$

for a relation $R \subseteq X \times Y$.

We recall that a relation lifting for a functor T is called a T-relator if it provides a functor over the category **Rel** of sets and relations. The following proposition provides constraints on L that make L^s a relator.

4.5.21. PROPOSITION. *If L is quasi-functorial and preserves diagonals, then L^{s} is a T^{s}-relator.*

Proof:
It is clear that L^{s} preserves diagonals. We show that $L^{\mathsf{s}}(R\,;\,S) = L^{\mathsf{s}}R\,;\,L^{\mathsf{s}}S$ for arbitrary relations R, S.

Consider $R \subseteq X \times Y$ and $S \subseteq Y \times Z$, and let $((\alpha, X'), (\beta, Z')) \in L^{\mathsf{s}}(R; S)$. Then there is some $Y' \subseteq Y$ with $(X', Y') \in \overline{\mathcal{P}}R$ and $(Y', Z') \in \overline{\mathcal{P}}S$. Furthermore, X' supports α and Z' supports β, and we have:

$$(\alpha|_{X'}, \beta|_{Z'}) \in L(\widehat{i_{X',X}}; R; S; \widehat{i_{Z',Z}}^{\circ}).$$

But since we must have $\mathsf{dom}(\widehat{i_{X',X}}; R) = X'$, $\mathsf{ran}(S; \widehat{i_{Z',Z}}^{\circ}) = Z'$ and $\mathsf{ran}(\widehat{i_{X',X}}; R) = \mathsf{dom}(S; \widehat{i_{Z',Z}}^{\circ}) = Y'$, we can apply Proposition 3.1.8 together with quasi-functoriality of L to find $\gamma \in \mathsf{T}Y'$ with $(\alpha|_{X'}, \gamma) \in L(\widehat{i_{X',X}}; R)$ and $(\gamma, \beta|_{Z'}) \in L(S; \widehat{i_{Z',Z}}^{\circ})$. Clearly Y' supports $\mathsf{T}i_{Y',Y}(\gamma)$, so $(Y', \mathsf{T}i_{Y',Y}(\gamma)) \in \mathsf{T}^{\mathsf{s}}Y$, and a simple calculation will now show that

$$(\alpha, \mathsf{T}i_{Y',Y}(\gamma)) \in L^{\mathsf{s}}R \text{ and } (\mathsf{T}i_{Y',Y}(\gamma), \beta) \in LS$$

so $(\alpha, \beta) \in L^{\mathsf{s}}R; L^{\mathsf{s}}S$ as required. □

Proposition 4.5.21 is an important technical result for the rest of this chapter, because, by a theorem usually attributed to Carboni, Kelly and Wood, a set functor admits a relator if and only if it preserves weak pullbacks [CKW91]. Hence Proposition 4.5.21 transforms a functor T that admits a quasi-functorial lax extension, into a weak pullback preserving functor T^{s}. This might help to explain why the coalgebraic logics of functors admitting quasi-functorial lax extensions, like monotone modal logic (Proposition 4.5.17), have turned out to be generally well behaved.

We assume that we have at our disposal a fixed "universal" T^{s}-model $\mathbb{U} = (U, \gamma, V)$ which is a disjoint union of one isomorphic copy for every *finite* pointed T-model $(\mathbb{S}, s)$. It is not hard to see that such a model does exist: just take a disjoint union containing each T-model defined on some finite subset $n = \{0, ..., n-1\}$ of ω. Since the collection of all these models forms a set (of size at most $|\coprod_{n\in\omega} \mathsf{T}(n)|$) their disjoint union is a well-defined T^{s}-model.

4.5.22. Definition. Let $\mathbb{S}$ be any T^{s}-model. We define the T^{s}-model $\mathbb{S} \oplus \mathbb{U} = (S + U, \sigma_+, V_+)$ as a disjoint union, except we let the underlying Kripke accessibility relation also relate every point to every other point. More precisely, for each $s \in S$ we have

$$\sigma_+(s) = (S + U, \mathsf{T}(\iota_{S,S+U})(\alpha))$$

where $\sigma(s) = (X, \alpha)$, and similarly for each $u \in U$.[4]

The following lemmas are immediate from the definition.

[4] In this definition we have used "+" as the symbol for binary coproducts of sets.

4.5.23. LEMMA. *Let $\mathbb{S}$ be a finite T^{s}-model. Then for all $u \in U$, we have:*

$$(\mathbb{U}, u) \sim^s (\mathbb{S} \oplus \mathbb{U}, u).$$

Furthermore, for every point $s \in S$ there is some $u \in U$ with

$$(\mathbb{S} \oplus \mathbb{U}, s) \sim^s (\mathbb{S} \oplus \mathbb{U}, u).$$

Proof:
For every $s \in S$ there is some $u \in U$ and some L-bisimulation R_s between $\mathbb{S}$ and $\mathbb{U}$ such that $(s, u) \in R_s$. The relation

$$Z = \widehat{\iota_{U,S+U}} \cup \bigcup_{s \in S} R_s^{\circ}$$

is therefore an L-bisimulation from $\mathbb{U}$ to $\mathbb{S} \oplus \mathbb{U}$. In fact it is easily checked from Definition 4.5.20 that Z is an L^{s}-bisimulation, and since uZu for all $u \in U$ this takes care of the first part of the lemma.

For the second part of the lemma, if we set

$$Z' = \Delta_{S+U} \cup \bigcup_{s \in S} R_s,$$

where we recall that Δ_{S+U} is the diagonal relation on $S + U$, then this is an L-bisimulation from $\mathbb{S} \oplus \mathbb{U}$ to itself. Again, it is in fact an L^{s}-bisimulation, and since Z' relates every $s \in S$ to some $u \in U$ we are done - since L-bisimilarity is equivalent to behavioural equivalence when L is quasi-functorial and preserves diagonals [Mar11]. □

Clearly, it follows from this lemma that for every $s \in S$ there is some $u \in U$ with $(\mathbb{S} \oplus \mathbb{U}, s) \sim^s (\mathbb{U}, u)$.

4.5.24. LEMMA. $(\mathbb{S}, s) \sim (\mathbb{S} \oplus \mathbb{U}, s)$.

Proof:
Clearly, since the underlying T-model corresponding to $\mathbb{S} \oplus \mathbb{U}$ is formed as a disjoint union. □

Note that Lemma 4.5.24 is not guaranteed to hold if we replace $\sim$ by the finer equivalence relation $\sim^s$.

It will be convenient in this section to work with a second version of the acceptance game for a modal Λ-automaton, which we will call the *extended* acceptance game for $\mathbb{A}$ with respect to a T-model $\mathbb{S}$, denoted $\mathcal{E}(\mathbb{A}, \mathbb{S})$. Given a modal Λ-automaton $\mathbb{A} = (A, \Theta, a_I, \Omega)$ and a T-model $\mathbb{S} = (S, \sigma, V)$ this game has three types of positions: pairs of the form (ψ, s) with $\psi \in \mathtt{ML}^1_\Lambda(A)$, pairs of the form (ψ, s) with $\psi \in \mathtt{LF}(A)$ (lattice formulas over A), and maps $f : \mathtt{LF}(A) \to \mathcal{P}(S)$. Admissible moves are given in Table 4.3.

Position	Pl'r	Admissible moves	Priority
$(\psi_1 \vee \psi_2, s)$	$\exists$	$\{(\psi_1, s), (\psi_2, s)\}$	0
$(\psi_1 \wedge \psi_2, s)$	$\forall$	$\{(\psi_1, s), (\psi_2, s)\}$	0
$(a, s) \in A \times S$	$\exists$	$\{(\Theta(a, m_V(s)), s)\}$	$\Omega(a)$
$(\lambda(\psi_1, ..., \psi_n), s)$	$\exists$	$\{f : \mathtt{LF}(A) \to \mathcal{P}S \mid \sigma(s) \in \lambda(f(\psi_1), ..., f(\psi_n))\}$	0
$(\top, s)$	$\forall$	$\emptyset$	0
$(\bot, s)$	$\exists$	$\emptyset$	0
$f : \mathtt{LF}(A) \to \mathcal{P}S$	$\forall$	$\{(\psi, s) \mid s \in f(\psi)\}$	0

Table 4.3: The extended acceptance game $\mathcal{E}(\mathbb{A}, \mathbb{S})$

It is easy to see that the position $(\top, s)$ is always winning for $\exists$, and $(\bot, s)$ is always winning for $\forall$. The following result is entirely routine to prove:

4.5.25. THEOREM. *Let $\mathbb{A} = (A, \Theta, a_I, \Omega)$ be a modal Λ-automaton and $(\mathbb{S}, s)$ a pointed T-model. Then $\mathbb{A}$ accepts $(\mathbb{S}, s)$ if, and only if, (a_I, s) is a winning position in the extended acceptance game.*

We also have the following observation, in which we will be sloppy and denote the extended acceptance game between a modal Λ-automaton $\mathbb{A}$ and a T-model underlying a T^s-model $\mathbb{S}$ by $\mathcal{E}(\mathbb{A}, \mathbb{S})$.

4.5.26. OBSERVATION. *Let ψ be any one-step formula in $\mathtt{ML}^1_\Lambda(A)$ for a set of liftings Λ for some given set functor T, and let $\mathsf{P} \in \{\forall, \exists\}$. If $(\mathbb{S}, s) \sim^s (\mathbb{S}', s')$ then the position (ψ, s) is winning for P in $\mathcal{E}(\mathbb{A}, \mathbb{S})$ if and only if (ψ, s') is a winning position for P in $\mathcal{E}(\mathbb{A}, \mathbb{S}')$. In particular, this means that for a pair of T^s-models $\mathbb{S}$ and $\mathbb{S}'$ such that $(\mathbb{S}, s) \sim^s (\mathbb{S}', s')$, and a formula $\psi \in \mathtt{ML}^1_{\Lambda \cup \{\Box^\mathsf{s}\}}(A)$, the position (ψ, s) is winning for P in $\mathcal{E}(\mathbb{A}, \mathbb{S})$ if and only if (ψ, s') is a winning position for P in $\mathcal{E}(\mathbb{A}, \mathbb{S}')$.*

4.5.27. DEFINITION. Let $\mathbb{A} = (A, \Theta, a_I, \Omega)$ be a $\Lambda \cup \{\Box^\mathsf{s}\}$-automaton and let ψ be a formula in the range of Θ. We say that ψ is $\mathbb{A}$-*valid* if for all (finite) pointed T-models $(\mathbb{S}, s)$, the position (ψ, s) is winning in $\mathcal{E}(\mathbb{A}, \mathbb{S})$. We say that ψ is $\mathbb{A}$-*satisfiable* if there is *some* (finite) pointed T-model $(\mathbb{S}, s)$, such that the position (ψ, s) is winning in $\mathcal{E}(\mathbb{A}, \mathbb{S})$.

Let Λ be any set of predicate liftings for T and let $\mathbb{A}$ be any modal $\Lambda \cup \{\Box^\mathsf{s}\}$-automaton. Making use of our fixed universal T^s-model $\mathbb{U}$ (see the text above Definition 4.5.22), we shall define a translation $t_\mathbb{A} : \mathtt{ML}^1_{\Lambda \cup \{\Box^\mathsf{s}\}}(A) \to \mathtt{ML}^1_\Lambda(A)$ by induction as follows:

- For $\lambda \in \Lambda$, we set $t_{\mathbb{A}}(\lambda(\psi_1, ..., \psi_k)) = \lambda(\psi_1, ..., \psi_k)$, and similarly for the dual λ^d for every lifting $\lambda \in \Lambda$.
- $t_{\mathbb{A}}(\top) = \top$ and $t_{\mathbb{A}}(\bot) = \bot$
- $t_{\mathbb{A}}(\psi_1 \vee \psi_2) = t_{\mathbb{A}}(\psi_1) \vee t_{\mathbb{A}}(\psi_2)$ and $t_{\mathbb{A}}(\psi_1 \wedge \psi_2) = t_{\mathbb{A}}(\psi_1) \wedge t_{\mathbb{A}}(\psi_2)$.
- $t_{\mathbb{A}}(\Box^{\mathsf{s}}\psi) = \begin{cases} \top & \text{if } \psi \text{ is } \mathbb{A}\text{-valid} \\ \bot & \text{otherwise} \end{cases}$
- $t_{\mathbb{A}}(\Diamond^{\mathsf{s}}\psi) = \begin{cases} \top & \text{if } \psi \text{ is } \mathbb{A}\text{-satisfiable} \\ \bot & \text{otherwise} \end{cases}$

Note that this translation depends on the whole automaton $\mathbb{A}$, not just the set of variables A. So for any given set of variables A, we have one translation $t_{\mathbb{A}} : \mathtt{ML}^1_{\Lambda\cup\{\Box^{\mathsf{s}}\}}(A) \to \mathtt{ML}^1_{\Lambda}(A)$ for each automaton $\mathbb{A}$ with states A, and the translations will generally be different for different choices of $\mathbb{A}$. However, we will drop the index $\mathbb{A}$ from now on to simplify notation. Given a modal $\Lambda \cup \{\Box^{\mathsf{s}}\}$-automaton $\mathbb{A} = (A, \Theta, a_I, \Omega)$, we will write $t(\mathbb{A})$ for the modal Λ-automaton $(A, \Theta^t, a_I, \Omega)$ where Θ^t is defined by $\Theta^t(a, c) = t(\Theta(a, c))$. Clearly $t(\mathbb{A})$ is a modal Λ-automaton.

We shall view the translation t as a map defined on the domain $\mathtt{ML}^1_{\Lambda\cup\{\Box^{\mathsf{s}}\}}(A) \cup \mathtt{LF}(A)$ by setting $t(\psi) = \psi$ for $\psi \in \mathtt{LF}(A)$.

4.5.28. LEMMA. *For every finite pointed T^s-model $(\mathbb{S}, s)$, and for any modal $\Lambda \cup \{\Box^{\mathsf{s}}\}$-automaton $\mathbb{A}$, we have*

$$(\mathbb{S}, s) \Vdash t(\mathbb{A}) \textit{ iff } (\mathbb{S} \oplus \mathbb{U}, s) \Vdash \mathbb{A}$$

Proof:
For left to right, suppose that $(\mathbb{S}, s) \Vdash t(\mathbb{A})$. By Theorem 4.5.25 there is a strategy χ for $\exists$ in the extended acceptance game for $t(\mathbb{A})$ and the underlying T-model also denoted as $\mathbb{S}$, which is winning at (a_I, s). Without loss of generality we may assume that χ is positional, and a winning strategy at *every* winning position in $\mathcal{E}(t(\mathbb{A}), \mathbb{S})$. Our goal is to construct a positional strategy χ' for $\exists$ in the extended acceptance game for $\mathbb{A}$ and $\mathbb{S} \oplus \mathbb{U}$, which prescribes a move for $\exists$ at every position (ψ, v) belonging to $\exists$ and with $v \in S$, such that:

1. χ' assigns a legitimate move to every position belonging to $\exists$ of the form (ψ, v) with $v \in S$ such that $(t(\psi), v)$ is a winning position in $\mathcal{E}(t(\mathbb{A}), \mathbb{S})$,
2. every χ'-guided partial match ρ starting at (a_I, s) and ending with a position (ψ, v) satisfies one of the following two criteria:

   a $v \in S$ and $t[\rho]$ is a χ-guided match in $\mathcal{E}(t(\mathbb{A}), \mathbb{S})$ (hence consists only of winning positions for $\exists$).

   b (ψ, v) is a winning position in $\mathcal{E}(\mathbb{A}, \mathbb{S} \oplus \mathbb{U})$.

Here, given that $\rho = \pi_1....\pi_n$, we define $t[\rho] = t(\pi_1)....t(\pi_n)$ where $t(\psi, v) = (t(\psi), v)$ if $\psi \in \mathtt{1ML}_\Lambda(A) \cup \mathtt{LF}(A)$, and $t(f) = f$ for a position $f : \mathtt{LF}(A) \to \mathcal{P}(S)$. Clearly, we can build a winning strategy in $\mathcal{E}(\mathbb{A}, \mathbb{S} \oplus \mathbb{U})$ from such a strategy χ'.

We define the strategy χ' by a case distinction, given a position (ψ, v) belonging to $\exists$ with $v \in S$ and such that $(t(\psi), v)$ is a winning position in $\mathcal{E}(t(\mathbb{A}), \mathbb{S})$. If $\psi = \alpha_1 \vee \alpha_2$ then $t(\psi) = t(\alpha_1) \vee t(\alpha_2)$, so we set $\chi'(\psi, v) = \alpha_i$ where $\chi(t(\psi), v)$. If $\psi = \lambda(\varphi_1, ..., \varphi_n)$ then $t(\psi) = \lambda(\varphi_1, ..., \varphi_n)$ too, so we set $\chi'(\psi, v) = \chi(t(\psi), v)$. A simple naturality argument shows that this is still a legitimate move in $\mathcal{E}(\mathbb{A}, \mathbb{S} \oplus \mathbb{U})$.

Finally, the interesting case is the one involving the support modality: at a position $(\Diamond^{\mathsf{s}}\varphi, v)$, if $(t(\Diamond^{\mathsf{s}}\varphi, v))$ is winning for $\exists$ in $\mathcal{E}(\mathbb{A}, \mathbb{S})$ then we must have $t(\Diamond^{\mathsf{s}}\varphi) = \top$, because otherwise there is no admissible move for $\exists$ at this position, hence φ is $\mathbb{A}$-satisfiable. Hence, by the construction of $\mathbb{U}$, and by Observation 4.5.26 there is some $u \in U$ such that (φ, u) is winning for $\exists$ in $\mathcal{E}(\mathbb{A}, \mathbb{S} \oplus \mathbb{U})$. So we let the strategy χ' pick the mapping f given by $f(\varphi) = \{v\}$, and $f(\varphi') = \emptyset$ for all other lattice formulas. For the case involving the dual $\Box^{\mathsf{s}}$, if $t(\Box^{\mathsf{s}}\varphi) = \top$ then φ is valid, and we can let $\exists$ pick the map f sending φ to $S + U$ and $f(\varphi') = \emptyset$ for $\varphi \neq \varphi'$.

We also need to check that every χ'-guided partial match satisfies one of the conditions (a) or (b), and we prove this by an induction on the length of a partial match. The only interesting case is for the extension of a partial match ρ ending with a position $(\Box^{\mathsf{s}}\varphi, v)$. By the induction hypothesis on ρ, we have $v \in S$ and $(t(\Box^{\mathsf{s}}\varphi), v)$ is winning for $\exists$, hence we must have $t(\Box^{\mathsf{s}}\varphi) = \top$. This means that φ is $\mathbb{A}$-valid, so *any* move (φ, w) by $\forall$ answering the move $S + U$ by $\exists$ will satisfy the condition (b), i.e. (φ, w) is winning for $\exists$ in $\mathcal{E}(\mathbb{A}, \mathbb{S} \oplus \mathbb{U})$.

For right to left, suppose that $\exists$ has a winning strategy χ at the position (a_I, s) in the extended acceptance game for $\mathbb{A}$ with respect to $\mathbb{S} \oplus \mathbb{U}$. We shall give $\exists$ a winning strategy χ' at the same position in the game for $t(\mathbb{A})$ with respect to $\mathbb{S}$. We shall inductively associate with every χ'-guided partial match π of length k a χ-guided "shadow match" $(\psi_1, v_1), ..., (\psi_k, v_k)$ such that π is of the form

$$(t(\psi_1), v_1), ..., (t(\psi_k), v_k).$$

We shall also make sure that whenever π' is an extension of π, the shadow match associated with π' is an extension of the shadow match associated with π as well. It will clearly follow that $\exists$ wins every infinite χ'-guided match.

For the singleton match consisting of (a_I, s) we let (a_I, s) itself be the shadow match. (This is acceptable because, by convention, we have set $t(a_I) = a_I$.) For a match π of length k we define the move $\chi'(\pi)$ depending on the shape of the last position on the associated shadow match. Again we treat only the interesting cases.

If the last position on the shadow match is $(\lambda(\psi_1, ..., \psi_m), v)$ then χ provides a map $f : \mathtt{LF}(A) \to \mathcal{Q}(S + U)$ such that

$$\sigma_+(v) \in \lambda_{S+U}(f(\psi_1), ..., f(\psi_m)),$$

where σ_+ is given by Definition 4.5.22. We set $\chi'(\pi) = f'$, where $f' : \mathtt{LF}(A) \to \mathcal{Q}(S)$ is defined by $\theta \mapsto f(\theta) \cap S$ for each $\theta \in \mathtt{LF}(A)$. This move is legal since $v \in S$ and by naturality of λ. It is easy to see how to extend the shadow match for each response by $\forall$.

If the last position on the shadow match is $(\Box^{\mathsf{s}}\psi, v)$ then since this position is winning for $\exists$, it must be the case that every position (ψ, w) for $w \in S + U$ is winning for $\exists$, hence this holds for every $w \in U$. This can only be true if ψ is $\mathbb{A}$-valid and so we have $t(\Box^{\mathsf{s}}\psi) = \top$. This means we are done since $(\top, v)$ is a winning position for $\exists$. Similarly, if the last position on the shadow match is $(\Diamond^{\mathsf{s}}\psi, v)$, then there is some $w \in S + U$ such that (ψ, w) is winning for $\exists$, so ψ is $\mathbb{A}$-satisfiable. Hence $t(\Diamond^{\mathsf{s}}\psi) = \top$, and the conclusion follows as in the previous case. □

We can now prove that over T^s-models, $\mu\mathtt{ML}_\Lambda$ is equivalent to the fragment of $\mu\mathtt{ML}_{\Lambda\cup\{\Box^{\mathsf{s}}\}}$ that is invariant for behavioural equivalence:

4.5.29. Theorem. *Let T be any set functor equipped with a quasi-functorial lax extension L that preserves diagonals, and let Λ be any set of predicate liftings for T. Then over T^s-models:*

$$\mu\mathtt{ML}_\Lambda \equiv \mu\mathtt{ML}_{\Lambda\cup\{\Box^{\mathsf{s}}\}}/\!\sim$$

Proof:
Suppose a formula φ of $\mu\mathtt{ML}_{\Lambda\cup\{\Box^{\mathsf{s}}\}}$ is invariant for behavioural equivalence. By the finite model property for $\mu\mathtt{ML}_{\Lambda\cup\{\Box^{\mathsf{s}}\}}$ it suffices to show that φ is equivalent to a $\mu\mathtt{ML}_\Lambda$-formula over finite models. Let $\mathbb{A}$ be a modal $\Lambda \cup \{\Box^{\mathsf{s}}\}$-automaton equivalent to φ, and let ψ be a formula of $\mu\mathtt{ML}_\Lambda$ equivalent to the automaton $t(\mathbb{A})$. Consider an arbitrary finite pointed T^{s}-model $(\mathbb{S}, s)$. We have:

$$\begin{array}{rcll}
(\mathbb{S}, s) \Vdash \varphi & \Leftrightarrow & (\mathbb{S} \oplus \mathbb{U}, s) \Vdash \varphi & \text{(Lemma 4.5.24 + assumption on } \varphi) \\
& \Leftrightarrow & (\mathbb{S} \oplus \mathbb{U}, s) \Vdash \mathbb{A} & \\
& \Leftrightarrow & (\mathbb{S}, s) \Vdash t(\mathbb{A}) & \text{(Lemma 4.5.28)} \\
& \Leftrightarrow & (\mathbb{S}, s) \Vdash \psi &
\end{array}$$

as required.

□

4.5.30. Proposition. *Suppose Λ is an L-complete set of predicate liftings for T, where L is a quasi-functorial lax extension for T that preserves diagonals, and suppose that T preserves finite sets. Then $\Lambda \cup \{\Box^{\mathsf{s}}\}$ is expressively complete for T^{s}.*

Proof:
By Proposition 4.5.21, L^{s} is a quasi-functorial lax extension for T^{s} that preserves diagonals, and it is simple to check that $\Lambda \cup \{\Box^{\mathsf{s}}\}$ is L^{s}-complete. Since T^{s} preserves finite sets it follows from Corollary 4.4.6 that $\Lambda \cup \{\Box^{\mathsf{s}}\}$ is expressively complete. □

We can now combine Theorem 4.5.9, Lemma 4.5.18 and Proposition 4.5.30 to obtain the following result.

4.5.31. Theorem (Auxiliary Characterization Theorem II). *Let Λ be any expressively complete set of monotone predicate liftings for the set functor T, and suppose T preserves finite sets and admits a quasi-functorial lax extension that preserves diagonals. If there exists a weakly adequate uniform construction for every one-step formula in $\mathtt{SO}^1_\Lambda(A)$, for every finite set A, then over T^s-models we have:*

$$\mu\mathtt{ML}_{\Lambda\cup\{\Box^s\}} \equiv \mathtt{MSO}_{\Lambda\cup\{\Box^s\}}/{\sim^s}$$

Now we can finally prove Theorem 4.5.13, the second main characterization result:

Proof of Theorem 4.5.13:
Suppose that the formula φ of $\mathtt{MSO}_\Lambda$ is invariant for T-bisimilarity, where Λ is expressively complete, and suppose the lax extension L is quasi-functorial and preserves diagonals. It follows that φ is invariant for L-bisimilarity, since L-bisimilarity is equivalent to behavioural equivalence when L preserves diagonals [Mar11]. But then φ is clearly also L^{s}-bisimulation-invariant, regarded as a formula of $\mathtt{MSO}_{\Lambda\cup\{\Box^s\}}$. Hence, it is equivalent to a formula of $\mu\mathtt{ML}_{\Lambda\cup\{\Box^s\}}$ by Theorem 4.5.31. It then follows by Theorem 4.5.29 that φ is in fact equivalent to a formula of $\mu\mathtt{ML}_\Lambda$, as required. □

4.5.4 Applications

We will now cover some instances of our main characterisation results.

4.5.32. Example. As a first application, the standard Janin-Walukiewicz characterization of the modal μ-calculus can be seen as an instance of the first main characterization result (Theorem 4.5.9) by taking $\Lambda = \{\Diamond\}$ and $\mathsf{T} = \mathcal{P}$, recalling that $\mathtt{MSO} = \mathtt{MSO}_{\{\Diamond\}}$. The uniform construction for $\mathcal{P}$, which is adequate with respect to any set of one-step formulas, is given as follows: consider a one-step frame, i.e. a pair (X, α) with $\alpha \in \mathcal{P}(X)$. We take $X_* = \alpha_* = \alpha \times \omega$, and we let $h_\alpha : \alpha \times \omega \to X$ be the projection map.

We could also cover this application by the second main characterization result (Theorem 4.5.13), by taking $X_* = X \times \omega$, $\alpha_* = \alpha \times \omega$ and let $h_\alpha : X_* \to X$ be the projection map. This is a weakly adequate uniform construction.

4.5.33. Example. Consider the finitary multiset ("bags") functor $\mathcal{B}$ (see Example 2.3.7), Given a pair (X, α) where $\alpha : X \to \omega$ has finite support, we define

$$X_* = \bigcup\{\{u\} \times \alpha(u) \mid u \in X\}.$$

The mapping $\alpha_* : X_* \to \omega$ is defined by setting $\alpha_*(w) = 1$ for all $w \in X_*$. The map $h_\alpha : X_* \to X$ is defined by $(u, i) \mapsto u$. It is easy to check that whenever two one-step frames (X, α) and (Y, β) are related by some morphism $f : (X, \alpha) \to (Y, \beta)$, the models $(X_*, \alpha_*, V_{[f \circ h_\alpha]})$ and $(Y_*, \beta_*, V_{[h_\beta]})$ are isomorphic, for any valuation $V : A \to \mathcal{Q}(Y)$. It follows that the construction is adequate for any set of one-step formulas, hence we get $\mu\mathtt{ML}_\mathcal{B} \equiv \mathtt{MSO}_\mathcal{B}/{\sim}$.

4.5.34. Example. Consider the set of all *polynomial functors* defined by the "grammar":

$$\mathsf{T} ::= \mathsf{C} \mid \mathsf{Id} \mid \mathsf{T} \times \mathsf{T} \mid \coprod_{i \in I} \mathsf{T}_i \mid (-)^\mathsf{C}$$

where C is any constant functor for some set C, and Id is the identity functor on Set. These functors cover many important applications: streams, binary trees, deterministic finite automata and deterministic labelled transition systems are all examples of coalgebras for polynomial functors, as is the so called *game functor* whose coalgebras provide semantics for Coalition Logic. For this last instance, the "game functor" $\mathcal{G}$ for n agents (see Example 2.3.11) can be written in the form of a polynomial functor as follows:

$$\coprod_{\langle S_0, ..., S_{n-1} \rangle \in (\mathcal{P}(\omega) \setminus \{\emptyset\})^n} \{\langle S_0, ..., S_{n-1} \rangle\} \times \mathsf{Id}^{(S_0 \times ... \times S_{n-1})}$$

For a given set X, an element of $\mathcal{G}X$ will be a pair consisting of a vector $\langle S_0, ..., S_{n-1} \rangle$ of available strategies for each player, together with an "outcome map" f assigning an element of X to each strategy profile in $S_0 \times ... \times S_{n-1}$.

Every polynomial functor admits adequate uniform constructions for all sets of one-step formulas. The proof proceeds by a straighforward induction, which provides each polynomial functor T with a uniform construction such that for any one-step frame homomorphism $f : (X, \alpha) \to (Y, \beta)$ and any $V : A \to \mathcal{Q}(Y)$, the one-step models $(X_*, \alpha_*, V_{[f \circ h_\alpha]})$ and $(Y_*, \beta_*, V_{[h_\beta]})$ are *isomorphic*. Hence, we get:

4.5.35. Proposition. *For every polynomial functor* T*, we have:*

$$\mu\mathtt{ML}_\mathsf{T} \equiv \mathtt{MSO}_\mathsf{T}/{\sim}.$$

Proof:
We provide a sketch of the inductive construction of an adequate uniform construction.
Constant functor: For the constant functor C, a one-step frame is a pair (X, c) with $c \in C$. We set $X_* = \emptyset$, $c_* = c$ and $h_c : \emptyset \to X$ to be the unique inclusion of the empty set.
Identity functor: Given a one-step frame (X, u) for the identity functor, which consists of a set X and $u \in X$, we set $X_* = \{u\}$, $u_* = u$ and we set $h_u : \{u\} \to X$

to be the inclusion map sending u to itself.
Product: Suppose that T_1 and T_2 have associated adequate uniform constructions with the required property. Consider a one-step $\mathsf{T}_1 \times \mathsf{T}_2$-frame $(X, (\alpha, \beta))$ with $\alpha \in \mathsf{T}_1 X$ and $\beta \in \mathsf{T}_2 X$. Let $h_1 : (X_1, \alpha_1) \to (X, \alpha)$ be the cover assigned by the uniform construction for T_1 and let $h_2 : (X_2, \beta_2) \to (X, \beta)$ be the cover assigned by the uniform construction for T_2. Then we take X_* to be the disjoint union $X_1 + X_2$, and set

$$(\alpha, \beta)_* = (\mathsf{T}_1 i_1(\alpha_1), \mathsf{T}_2 i_2(\beta_2))$$

where $i_1 : X_1 \to X_1 + X_2$ and $i_2 : X_2 \to X_1 + X_2$ are the insertion maps for the co-product. Finally, we define the covering map $h_{(\alpha,\beta)} : X_1 + X_2 \to X$ be obtained by simply co-tupling the maps h_1, h_2, i.e. $h_{(\alpha,\beta)}$ is the map given by the universal property of the co-product applied to the diagram $X_1 \xrightarrow{h_1} X \xleftarrow{h_2} X_2$. We get:

$$\begin{aligned}
(\mathsf{T}_1 \times \mathsf{T}_2) h_{(\alpha,\beta)}((\alpha, \beta)_*) &= (\mathsf{T}_1 h_{(\alpha,\beta)}(\mathsf{T}_1 i_1(\alpha_1)), \mathsf{T}_2 h_{(\alpha,\beta)}(\mathsf{T}_2 i_2(\beta_2))) \\
&= (\mathsf{T}_1(h_{(\alpha,\beta)} \circ i_1)(\alpha_1), \mathsf{T}_2(h_{(\alpha,\beta)} \circ i_2)(\beta_2)) \\
&= (\mathsf{T}_1 h_1(\alpha_1), \mathsf{T}_2 h_2(\beta_2)) \\
&= (\alpha, \beta)
\end{aligned}$$

so $h_{(\alpha,\beta)}$ is indeed a covering map as required.
Exponentiation: The case of a functor T^{C} for some constant C is handled analogously with the case of binary products, so we leave it to the reader.
Co-product: This step of the construction is the easiest one. Suppose that each functor T_i for $i \in I$ is equipped with an adequate uniform construction. Let (X, α) be a one-step frame for the co-product $\coprod_{i \in I} \mathsf{T}_i$. Then since co-product is disjoint union in Set, there is a unique $i \in I$ with $\alpha \in \mathsf{T}_i X$, and so we define the cover $h_\alpha : (X_*, \alpha_*) \to (X, \alpha)$ merely by applying the uniform construction for each T_i. □

4.5.36. Remark. These uniform constructions were all designed in a case-by-case fashion, and at the present time we do not know whether there is any general recipe for producing an adequate uniform construction when it exists. What the constructions mentioned so far seem to have in common is that we want to produce enough equivalent (in some sense) copies of each state in a one-step model, but this is not always sufficient. In the next section we will see a somewhat more involved construction for the monotone μ-calculus, which aims to create sufficiently many copies of each state but also, crucially, sufficiently many *pairwise disjoint* copies of all the neighborhoods. In this case we are trying to neutralize not only the capability of the second-order one-step language to count states in one-step models, but also its capability to express how certain neighborhoods are related to each other, in particular, whether they overlap or not. For example, the second-order one-step language can express the property that any two neighborhoods intersect, or there is a smallest neighborhood contained in all others etc., and the uniform construction we provide needs to trivialize all such statements. In general, applying our main result as it stands may require a bit of creativity, and we regard it as an interesting (possibly quite hard) task for future research to come up with a result that makes the task entirely mechanical. We mention some related questions in our concluding section.

4.6 Characterizing the monotone μ-calculus

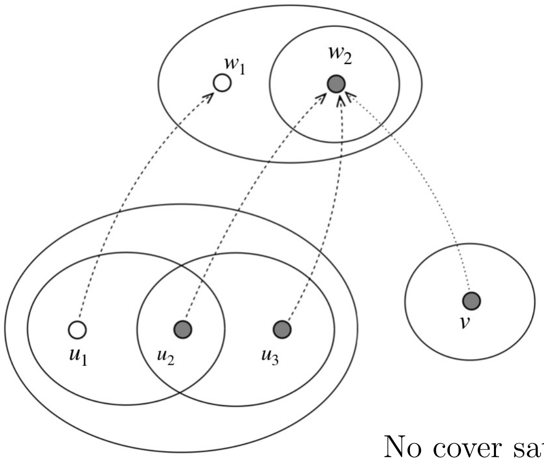
In this final section we will present the characterisation result for the monotone μ-calculus in detail. As we already mentioned, the first version of our characterization theorem cannot be applied to $\mathcal{M}$, since there is *no* adequate uniform construction for it.

In the following we will first explain why $\mathcal{M}$ does not admit an *adequate* uniform construction, and then define a uniform construction for it, which we show that is a *weakly adequate* uniform construction. We can then apply the second version of our main result.

4.6.1 No adequate construction for $\mathcal{M}$

4.6.1. Proposition. *The functor $\mathcal{M}$ does not admit an adequate uniform construction for the formula φ defined as: $\exists p.\neg(a \subseteq p)$, or equivalently $\neg\mathsf{Em}(a)$.*

Proof:
The formula φ just says that the value of a is non-empty. Suppose there existed an adequate uniform construction for this formula. Consider the situation depicted in the diagram below, which shows three one-step frames together with two one-step frame morphisms, one for each of the two bottom one-step frames. The top frame consists of two points $\{w_1, w_2\}$ and has neighborhoods $\{\{w_1, w_2\}, \{w_2\}\}$, the bottom left frame has points $\{u_1, u_2, u_3\}$ and neighborhoods $\{\{u_1, u_2\}, \{u_2, u_3\}, \{u_1, u_2, u_3\}\}$. Finally the bottom right frame has a single point v, and the associated singleton as its only neighborhood. In other words, the picture shows a co-span in the category of one-step frames and one-step frame morphisms. Furthermore, consider the valuation on the topmost one-step frame which makes a true at exactly the one state w_1, i.e. the one not belonging to the singleton neighborhood. This valuation is depicted in the diagram by representing the state where a is true by a blank circle, and the state where it is not true by a filled circle. The induced valuations on the bottom one-step frames via the frame morphisms are depicted in the same manner. With respect to these valuations, p will be true at u_1, but false at u_2, u_3 and v.



No cover satisfies φ

Every cover satisfies φ

Let us denote the top frame by (Y, β), its valuation by V, the bottom-left frame as (X, α), the bottom-right one as (X', α') and the corresponding frame morphisms as f and f' respectively. Now, the supposed adequate construction will assign a cover to each of the three frames, and we get valuations for each cover from the valuations depicted in the diagram. It follows from the defining condition 4.2 for adequacy that φ must have the same truth value in each of these covers. But this leads to a contradiction: it is not hard to check that the cover $h_\alpha : (X_*, \alpha_*) \to (X, \alpha)$ must be such that $(X_*, \alpha_*, V_{[f \circ h_\alpha]}) \Vdash \varphi$, because the image of the map h_α must support α and therefore include the one element of X coloured red. On the other hand, we clearly must have $(X'_*, \alpha'_*, V_{[f' \circ h_{\alpha'}]}) \nVdash \varphi$, which contradicts the condition 4.2. $\square$

4.6.2 A weakly adequate uniform construction for $\mathcal{M}$

We now define a uniform construction for the supported companion $\mathcal{M}^s$ of $\mathcal{M}$ w.r.t. a given finite set of formulas, and prove that it is weakly adequate. More precisely, we proceed as follows: Throughout the section we fix an arbitrary natural number k and a finite set A of variables, and define a uniform construction $(-)_*$ which will be shown to be weakly adequate for every formula in $\mathtt{SO}^1_{\{\Box\}}(A)$ of quantifier depth $\leq k$. It follows that there is a weakly adequate uniform construction for every finite set Γ of formulas in $\mathtt{SO}^1_{\{\Box\}}$, since we can apply the result to the maximum quantifier depth of any formula in Γ.

4.6.2. Definition. Given a set X, and an object $\alpha \in \mathcal{M}X$, put

$$X_* := X \times 2^k \times \mathcal{P}(X) \times \omega$$

and let π_X be the projection map from X_* to X. Define $\alpha_* \in \mathcal{M}(X_*)$ by setting $Z \in \alpha_*$ for $Z \subseteq X_*$ iff $\lceil Y, j \rceil \subseteq Z$ for some $Y \in \alpha$ and some $j < \omega$, where

$$\lceil Y, j \rceil := \{(u, i, Y, j) \mid u \in Y, i < 2^k\}.$$

The sets of the form $\lceil Z, j \rceil$ for $Z \in \alpha$ will be called the *basic members* of α_*. The set of all elements in X_* that do not belong to any basic member will be called the *residue* of the frame (X_*, α_*). Note that X_* is partitioned into each of the basic members along with the residue, as an extra partition cell.

The intuition behind the construction is that we want to create infinitely many *disjoint* copies of each neighborhood, and furthermore we want to create sufficiently many copies of each state *within* each copy of a neighborhood.

4.6.3. Proposition. *For every given one-step $\mathcal{M}$-frame (X, α), the projection map $\pi_X : (X_*, \alpha_*) \to (X, \alpha)$ is a surjective cover.*

Proof:
Clearly the projection map π_X is always surjective since $(u, 0, \emptyset, 0) \mapsto u$ for all $u \in X$. (Note that $0 < 2^n$ for all $n \in \omega$, so we always have $(u, 0, \emptyset, 0) \in X_*$.)

We need to check that $\mathcal{M}\pi_X(\alpha_*) = \alpha$. In other words, we have to check that for all $Z \subseteq X$, we have $Z \in \alpha$ iff $\pi_X^{-1}(Z) \in \alpha_*$. For left to right, if $Z \in \alpha$ then $\lceil Z, 0 \rceil$ is a basic member of α_*, and clearly $\lceil Z, 0 \rceil \subseteq \pi_X^{-1}(Z)$. Conversely, suppose $\pi_X^{-1}(Z) \in \alpha_*$. Then there is some basic member $\lceil Y, j \rceil \in \alpha_*$ with $\lceil Y, j \rceil \subseteq \pi_X^{-1}(Z)$. But then $Y \in \alpha$, and furthermore $Y \subseteq Z$: if $u \in Y$ then $(u, 0, Y, j) \in \lceil Y, j \rceil$, so $(u, 0, Y, j) \in \pi_X^{-1}(Z)$ meaning that $\pi_X(u, 0, Y, j) = u \in Z$. So $Z \in \alpha$ as required. □

The main goal of this section is to prove the following:

4.6.4. Proposition. *$(-)_*$ is weakly adequate for every formula in $\mathtt{SO}^1_{\{\Box\}}(A)$ of quantifier depth $\leq k$.*

Clearly it then follows that there is a weakly adequate uniform construction for every finite set Γ of formulas in $\mathtt{SO}^1_{\{\Box\}}$, since we can apply the result to the maximum quantifier depth of any formula in Γ.

To prove that $(-)_*$ is weakly adequate for all formulas of quantifier depth $\leq k$, it suffices to prove that for every pair of one-step $\widetilde{\mathcal{M}}$-bisimilar models (X, α, V) and (Y, β, U), the corresponding models $(X_*, \alpha_*, V_{[\pi_X]})$ and $(Y_*, \beta_*, U_{[\pi_Y]})$ satisfy the same formulas of quantifier depth $\leq k$, where $\widetilde{\mathcal{M}}$ is the lax extension for $\mathcal{M}$ defined in Example 2.3.33. It should not be too surprising that we can prove this, but the actual proof is not entirely trivial. From now on we keep the models (X, α, V) and (Y, β, U) fixed, as well as a global one-step bisimulation R relating these models. Throughout this section, we shall use the notation $(X, \alpha, V) \equiv_k (Y, \beta, U)$ to say that two one-step models satisfy the same formulas of $\mathtt{MSO}^1_{\{\Box\}}(A)$ with at most k nested quantifiers.

4.6.5. Definition. A propositional A-*type* τ is a subset of A. Given a set X and a valuation $V : A \to \mathcal{Q}(X)$, the propositional A-type of $v \in X$ is defined to be $m_V(v) = \{a \in A \mid v \in V(a)\}$.

4.6.6. Definition. Given a subset Z of X_* or Y_*, a valuation W such that $W : B \to \mathcal{Q}(X_*)$ or $W : B \to \mathcal{Q}(Y_*)$, and a natural number n, the n-*signature* of Z over B relative to the valuation W is the mapping $\sigma_Z : \mathcal{P}(B) \to \{0, ..., n\}$ defined by:

$$\sigma_Z(t) := \mathtt{min}(|\{x \in Z \mid m_W(x) = t\}|, n)$$

4.6.7. Definition. Let B be any superset of A, and let $V_1 : B \to \mathcal{Q}(X_*)$ and $V_2 : B \to \mathcal{Q}(Y_*)$. Then for any natural number n we write

$$(X_*, \alpha_*, V_1) \approx_n (Y_*, \beta_*, V_2),$$

and say that these one-step models *match up to depth* n, if:

1. For every n-signature σ over variables B, the number of basic elements of signature σ in α_* and β_* respectively are either both finite and equal, or both infinite.

2. The residues of the two one-step models have the same n-signature.

Using the assumption that the models (X, α, V) and (Y, β, U) are globally one-step $\widetilde{\mathcal{M}}$-bisimilar, we get the following lemma (which is the only point in the proof where we make use of the fact that R is a *global* L-bisimulation rather than an arbitrary one).

4.6.8. Lemma. $(X_*, \alpha_*, V_{[\pi_X]}) \approx_{2^k} (Y_*, \beta_*, U_{[\pi_Y]})$.

Proof:
To see that the residues of the two models have the same 2^k-signature, for one direction just note that if the residue of $(X_*, \alpha_*, V_{[\pi_X]})$ contains an element (u, i, Z, j) then it contains infinitely many elements of the same propositional type, namely one member (u, i, Z, p) for every $p \in \omega$. But then so will the residue of $(Y_*, \beta_*, U_{[\pi_Y]})$: just pick some v with uRv (again using the fact that R is a global one-step bisimulation). Since $v \notin \emptyset$, for every $p \in \omega$ the element $(v, 0, \emptyset, p)$ will be a member of the residue of $(Y_*, \beta_*, U_{[\pi_Y]})$ of the same propositional type as (u, i, Z, j).

Now for the basic elements. First note that, for any 2^k-signature σ, α_* either contains no basic elements of signature σ, or infinitely many: if there is some basic element $\lceil Z, j \rceil$ of signature σ, then for any $i \neq j$, the basic element $\lceil Z, i \rceil$ has the same 2^k-signature as $\lceil Z, j \rceil$ with respect to the valuation $V_{[\pi_X]}$. The same holds for β_* with respect to the valuation $U_{[\pi_Y]}$. Hence, it suffices to show that α_*

contains a basic element of signature σ w.r.t. $V_{[\pi_X]}$ iff β_* contains a basic element of signature σ w.r.t $U_{[\pi_Y]}$.

We consider only one direction: suppose that α_* contains a basic element $\lceil Z, j \rceil$ of signature σ, where $Z \in \alpha$. Then there must be $Z' \in \beta$ such that, for all $v \in Z'$, there is $u \in Z$ with uRv, since R was a one-step L-bisimulation. Furthermore, since R was a *global* one-step bisimulation, for every $w \in Z$ we can pick some $w' \in Y$ with wRw', and put

$$Z'' = Z' \cup \{w' \mid w \in Z\}$$

By monotonicity we have $Z'' \in \beta$. Furthermore, Z and Z'' are clearly related so that the following back-and-forth conditions hold: for all $u \in Z$ there is $v \in Z''$ with uRv, and for all $v \in Z''$ there is $u \in Z$ with uRv. Since any two states related by R have the same propositional type, it follows that the same propositional types appear in $\lceil Z, j \rceil$ and $\lceil Z'', j \rceil$. But since both these sets contain at least 2^k copies of every propositional type that appears in them, it follows that $\lceil Z, j \rceil$ and $\lceil Z'', j \rceil$ have the same 2^k-signature, as required. □

We are going to show, by induction on a natural number $m \leq k$, that if two one-step models of the form (X_*, α_*, V_1) and (Y_*, β_*, V_2) match up to depth 2^m, then they satisfy the same formulas of quantifier depth m. Together with the previous lemma, it then follows that the one-step models $(X_*, \alpha_*, V_{[\pi_X]})$ and $(Y_*, \beta_*, U_{[\pi_Y]})$ satisfy the same formulas of quantifier depth $\leq k$. For the basis case of $2^0 = 1$, we need the following result:

4.6.9. Lemma. *Let B be a set of variables containing A, and let $V_1 : B \to \mathcal{Q}(X_*)$ and $V_2 : B \to \mathcal{Q}(Y_*)$ be valuations such that:*

$$(X_*, \alpha_*, V_1) \approx_1 (Y_*, \beta_*, V_2)$$

Then these two one-step models satisfy the same atomic formulas of the one-step language $\mathtt{SO}^1_{\{\Box\}}(B)$.

Proof:
We only prove one direction for each case. Suppose first that:

$$(X_*, \alpha_*, V_1) \Vdash_1 p \subseteq q,$$

where $p, q \in B$. Suppose for a contradiction that $V_2(p) \nsubseteq V_2(q)$. Then there is some $(u, i, Z, j) \in Y_*$ such that $(u, i, Z, j) \in V_2(p) \setminus V_2(q)$. If (u, i, Z, j) comes from the residue of (Y_*, β_*) then since the residue of (X_*, V_α) has the same 1-signature, it must contain some element (u', i', Z', j') of the same 1-type, and so we cannot have $V_1(p) \subseteq V_1(q)$. The case where (u, i, Z, j) comes from a basic member is similar.

Now, suppose that:

$$(X_*, \alpha_*, V_1) \Vdash_1 \Box p$$

Then $V_1(p) \in \alpha_*$, so there is some basic element $\lceil Z, j \rceil \in \alpha_*$ with $\lceil Z, j \rceil \subseteq V_1(p)$. There must be some basic $\lceil Z', j' \rceil \in \beta_*$ of the same 1-signature over B as $\lceil Z, j \rceil$, and clearly it follows that $\lceil Z', j' \rceil \subseteq V_2(p)$ and so $V_2(p) \in \beta_*$ as required. $\Box$

We now only need the following lemma:

4.6.10. LEMMA. *Let B be a finite superset of A, let $0 < l \leq k$ and let $V_1 : B \to Q(X_*)$ and $V_2 : B \to \mathcal{Q}(Y_*)$ be valuations such that:*

$$(X_*, \alpha_*, V_1) \approx_{2^l} (Y_*, \beta_*, V_2)$$

Let q be any fresh variable. Then for any valuation V_1' over $B \cup \{q\}$ extending V_1 with some value for q, there exists a valuation V_2' over $B \cup \{q\}$ extending V_2, such that:

$$(X_*, \alpha_*, V_1') \approx_{2^{(l-1)}} (Y_*, \beta_*, V_2')$$

and vice versa.

Proof:
We only prove one direction since the other direction can be proved by a symmetric argument.

Let V_1' be given. By the hypothesis, for any 2^m-signature σ over the variables B, the number of basic elements of signature σ in α_* and β_* relative to V_1 and V_2 are either both finite and the same, or both infinite. Let $\sigma_1, ..., \sigma_k$ be a list of all the distinct 2^m-signatures over B such that the set of basic elements of α_* and β_* of signature σ_i, with $1 \leq i \leq k$, is non-empty but finite, and let $\sigma_{k+1}, ..., \sigma_l$ be a list of all the 2^m-signatures such that, for $k+1 \leq i \leq l$, there are infinitely many basic elements of α_* and of β_* of signature σ_i. Then, for each $i \in \{1, ..., l\}$, let $\alpha_*[\sigma_i]$ denote the set of basic elements in α_* of signature σ_i, and similarly let $\beta_*[\sigma_i]$ denote the set of basic elements of β_* of signature σ_i. Then $\alpha_*[\sigma_1], ..., \alpha_*[\sigma_l]$ is a partition of the set of basic elements of α_* into non-empty cells, and similarly $\beta_*[\sigma_1], ..., \beta_*[\sigma_l]$ is a partition of the set of basic elements of β_*.

Given the extended valuation V_1' on X_* defined on variables $B \cup \{q\}$, we similarly let $\tau_1, ..., \tau_{k'}$ be a list of all the 2^{l-1}-signatures over $B \cup \{q\}$ such that, for $1 \leq i \leq k'$, the set of basic elements of α_* of 2^{l-1}-signature τ_i is non-empty but finite. We let $\tau_{k'+1}, ..., \tau_{l'}$ be a list of all the 2^{l-1}-signatures over $B \cup \{q\}$ such that, for each i with $k'+1 \leq i \leq l'$, the set of basic elements of α_* of 2^{m-1}-signature τ_i is infinite. Let $\alpha_*[\tau_i]$ denote the set of basic elements of α_* of 2^{l-1}-signature τ_i, so that the collection $\alpha_*[\tau_1], ..., \alpha_*[\tau_{l'}]$ constitutes a second partition of the set of basic elements of α_*. It will be useful to introduce the abbreviation D_1 for the finite set $\alpha_*[\sigma_1] \cup ... \cup \alpha_*[\sigma_k]$, and the abbreviation D_2 for the finite set $\alpha_*[\tau_1] \cup ... \cup \alpha_*[\tau_{k'}]$.

For each i with $1 \leq i \leq k$, there is a bijection between the set $\alpha_*[\sigma_i]$ and $\beta_*[\sigma_i]$, and we can paste all these bijections together into a bijective map

$$f : \alpha_*[\sigma_1] \cup ... \cup \alpha_*[\sigma_k] \to \beta_*[\sigma_1] \cup ... \cup \beta_*[\sigma_k]$$

Since every basic element of α_* not in D_1 belongs to a 2^m-signature of which there are infinitely many basic elements in β_*, and since $D_1 \cup D_2$ is finite, it is easy to see that we can extend the map f to a map g which is an injection from the set $D_1 \cup D_2$ into the set of basic elements of β_*, such that for each basic element $\lceil Z, j \rceil$ in $D_1 \cup D_2$, $\lceil Z, j \rceil$ and $g(\lceil Z, j \rceil)$ have the same 2^m-signature over B, and such that $g \upharpoonright D_1 = f$. Each basic element of β_* not in the image of g must then be of one of the 2^m-signatures $\sigma_{k+1}, ..., \sigma_l$, and so we can partition the set of basic elements of β_* outside the image of g into the cells $\beta_*[\sigma_{k+1}] \setminus g[D_2], ..., \beta_*[\sigma_l] \setminus g[D_2]$. For each i with $k+1 \leq i \leq l$, let $\gamma^i_1, ..., \gamma^i_r$ list all infinite sets of the form $\alpha_*[\sigma_i] \cap \alpha_*[\tau_j]$ for $k'+1 \leq j \leq l'$. The list $\gamma^i_1, ..., \gamma^i_r$ must be non-empty, and so since the set $\beta_*[\sigma_i] \setminus g[D_2]$ is also infinite, we may partition it into r many infinite cells and list these as $\delta^i_1, ..., \delta^i_r$. Now, for each basic element $\lceil Z, j \rceil$ of β_*, we define a map $W_{\lceil Z,j \rceil}$ from $B \cup \{q\}$ to $\mathcal{P}(\lceil Z, j \rceil)$ by a case distinction as follows:

Case 1: $\lceil Z, j \rceil = g(\lceil Z', j' \rceil)$ for some $\lceil Z', j' \rceil \in D_1 \cup D_2$. Then $\lceil Z, j \rceil$ and $\lceil Z', j' \rceil$ have the same 2^m-signature over B. Using this fact we define the valuation $W_{\lceil Z,j \rceil}$ so that, for each $p \in B$, we have $W_{\lceil Z,j \rceil}(p) = V_2(p) \cap \lceil Z, j \rceil$, and so that $\lceil Z', j' \rceil$ and $\lceil Z, j \rceil$ have the same 2^{m-1}-signature over $B \cup \{q\}$ with respect to the valuations V'_1 and $W_{\lceil Z,j \rceil}$. We show how to assign the value of the variable q: for each propositional type t over $B \cup \{q\}$, there are three different possible cases to consider. If $\lceil Z', j' \rceil$ has $m' < 2^{m-1}$ elements of type $t \cup \{q\}$ over $B \cup \{q\}$, then pick m' many elements of $\lceil Z, j \rceil$ of type t and mark them by q. This is possible since $m' < 2^{l-1} \leq 2^m$ and $\lceil Z', j' \rceil$ and $\lceil Z, j \rceil$ have the same 2^l-signature. If there are $m' < 2^{l-1}$ elements of $\lceil Z', j' \rceil$ of type t over $B \cup \{q\}$, then pick m' elements of $\lceil Z, j \rceil$ of type t over B, and mark all the other elements of $\lceil Z, j \rceil$ of type t by q. Finally, if there are at least 2^{m-1} elements of $\lceil Z', j' \rceil$ of type $t \cup \{q\}$ over $B \cup \{q\}$ and at least 2^{l-1} elements of $\lceil Z', j' \rceil$ of type t over $B \cup \{q\}$, then all in all there must be at least 2^l elements of $\lceil Z', j' \rceil$ of type t over B, and so there must be at least 2^l elements of $\lceil Z, j \rceil$ of type t over B. Pick 2^{l-1} of these and mark them by q. Finally, let $W_{\lceil Z,j \rceil}(q)$ be the set of elements of $\lceil Z, j \rceil$ marked by q.

Case 2: $\lceil Z, j \rceil$ is not in the image of g. Then there must be some $i \in \{k'+1, ..., l'\}$ such that $\lceil Z, j \rceil \in \beta_*[\sigma_{k+1}] \setminus g[D_2]$, and this set is partitioned into $\delta^i_1, ..., \delta^i_r$. Let $\lceil Z, j \rceil \in \delta^i_j$, and pick some arbitary element $\lceil Z', j' \rceil$ of the set γ^i_j. Then $\lceil Z', j' \rceil$ and $\lceil Z, j \rceil$ have the same 2^m-signature over B and we can proceed as in Case 1.

We define the valuation V'_2 so that the intersection of $V'_2(q)$ with the union of all the basic members of β_* equals the union of the sets $W_{\lceil Z,j \rceil}(q)$ for $\lceil Z, j \rceil$ a basic element in β_*, and so that the residue of (Y_*, β_*) has the same $2^{(m-1)}$-signature as the residue of (X_*, α_*) with respect to the valuations V'_1 and V'_2. This can be done using the same reasoning as in the two previous cases. We now need to check that

$$(X_*, \alpha_*, V_1') \approx^{2^{(m-1)}} (Y_*, \beta_*, V_2')$$

First, suppose there are infinitely many basic elements of α_* of some 2^{m-1} signature τ_j, meaning that $k' \leq j \leq l'$. Then since the set $\alpha_*[\tau_j]$ is infinite, D_1 is finite and

$$\alpha_*[\tau_j] = (D_1 \cap \alpha_*[\tau_j]) \cup (\alpha_*[\sigma_{k+1}] \cap \alpha_*[\tau_j]) \cup ... \cup (\alpha_*[\sigma_l] \cap \alpha_*[\tau_j])$$

there must be some $i \in \{k+1, ..., l\}$ such that the set $\alpha_*[\sigma_i] \cap \alpha_*[\tau_j]$ is infinite. This means that $\alpha_*[\sigma_i] \cap \alpha_*[\tau_j]$ appears in the list $\gamma_1^i, ..., \gamma_r^i$, and so we see that all elements of some member of the list $\delta_1^i, ..., \delta_r^i$ will have the 2^{m-1}-signature τ_j. Since each member of this list is infinite, we see that there must be infinitely many basic elements of β_* of signature τ_j.

Conversely, suppose there are infinitely many basic elements of β_* of 2^{m-1}-signature τ_j over $B \cup \{q\}$. Then since the image of g is finite, some of these elements must be outside the image of g, which means that for some $i \in \{k+1, ..., l\}$, some member of the list $\delta_1^i, ..., \delta_r^i$ will consist of elements of signature τ_j. This means that some member of the list $\gamma_1^i, ..., \gamma_r^i$ will consist of elements of signature τ_j, and since each member of this list is infinite we see that α_* has infinitely many basic elements of 2^{m-1}-signature τ_j over $B \cup \{q\}$.

Finally, suppose that there are finitely many basic elements of α_* and β_* of 2^{m-1}-signature τ_j. We check that the mapping g restricts to a bijection between the basic elements of α_* and β_* of this signature. First, g is injective and maps basic elements of α_* of signature τ_j to basic elements of β_* of signature τ_j. It only remains to show that (the restriction of) g is surjective, i.e. each basic element $\lceil Z, r \rceil$ of signature τ_j is equal to $g(\lceil Z', r' \rceil)$ for some $\lceil Z', r' \rceil$. But suppose $\lceil Z, r \rceil$ is not in the image of g; then it is in one of the members of the list $\delta_1^i, ..., \delta_r^i$ for some i, and since each of these members is an infinite set of basic elements of the same signature, we see that there are infinitely many basic elements of β_* of signature τ_j, contrary to our assumption. Hence, the proof is done. □

We now get the following result as an application of Theorem 4.5.13 for the monotone neighbourhood functor. We recall that the monotone μ-calculus is the logic $\mu\mathtt{ML}_{\Box^{\mathcal{M}}}$ which we refer to by $\mu\mathtt{MML}$, and the monotone monadic second-order logic is the logic $\mathtt{MSO}_{\Box^{\mathcal{M}}}$ denoted by $\mathtt{MMSO}$ (see Definitions 2.4.15 and 4.2.3).

4.6.11. THEOREM. *The monotone μ-calculus is the fragment of monotone monadic second-order logic that is invariant for behavioural equivalence. In a formula:*

$$\mu\mathtt{MML} \equiv \mathtt{MMSO}/{\sim}.$$

Proof:
It suffices to prove that $\mathtt{MSO}_{\{\Box,\Box^s\}}/{\sim}$ is equal to $\mu\mathtt{ML}_{\{\Box,\Box^s\}}$, and then apply Theorem 4.5.29. For this, by Theorem 4.5.9 it suffices in turn to prove that the

construction $(-)_*$ is weakly adequate for all formulas in $\mathtt{SO}^1_{\{\Box,\Box^s\}}(A)$ of quantifier depth $\leq k$. We can prove this by combining the last three lemmas, using Ehrenfeucht-Fraïssé games for the one-step language. Lemmas 4.6.8 and 4.6.10 provide a recipe for how "Duplicator" can survive k steps of the game comparing the models $(X_*, \alpha_*, V_{[\pi_X]})$ and $(Y_*, \beta_*, U_{[\pi_Y]})$, and 4.6.9 guarantees that the valuations constructed at the end of the game will satisfy the same atomic formulas. Working out the full argument is entirely standard, so we leave the details to the reader. □

4.7 Conclusion

We conclude this chapter by mentioning some questions for future research:

1. Is there a good *categorical* and "logic free" characterization of those set functors T that admit an adequate (or weakly adequate) uniform construction, for instance, in terms of T preserving certain limits or colimits? Trying to answer this question would involve a deeper study of how the model theory of one-step languages is related to categorical properties of the type functor involved. Related to this, an anonymous referee of [ESV17] pointed out to us that the machinery of one-step frames, covers and uniform constructions are all taking place in the *category of elements* associated with the set functor T, and properties of the functor T are closely related to properties of the corresponding category of elements [BW95]. This could very well be a fruitful direction to investigate further.

2. Can we improve our work on the supported companion functor, to the effect that every set functor T has a companion T' that admits an adequate uniform construction? Relating this to the previous question, we would like to understand *why* the supported companion to $\mathcal{M}$ admits an adequate uniform construction, but not $\mathcal{M}$ itself. The construction achieves two things, in general: First, it ensures that every $\alpha \in \mathsf{T}^s X$ has a unique smallest support (even when X is infinite), often called its *base*. Second, and in our view more importantly, we get that the map $\mathsf{base}_X : \mathsf{T}^s X \to \mathcal{P}X$ defined by sending each α to its base is a natural transformation. Conditions for a functor under which this holds have been isolated by Gumm in [Gum05]. Are these conditions related to the existence of adequate uniform constructions?

3. It would be interesting to further explore the relation between $\mathtt{MSO}_\mathsf{T}$ and the first-order logic of Litak & alii [LPSS12] for T-coalgebras. For instance, an interesting question would be whether (on T-tree models) $\mathtt{MSO}_\mathsf{T}$ is equivalent to some extension of this first-order language with fixpoint operators.

4. Finally, there is the question of finding sufficient *and necessary* conditions for a Janin-Walukiewicz theorem to hold. Related to this question, we have not been able to produce an example of a functor for which the Janin-Walukiewicz theorem does *not* hold (the question of whether such an example can be found was raised to us by an anonymous referee). Given all that we can say for sure, it could be the case that a Janin-Walukiewicz theorem simply holds for *every* set functor, but we conjecture that at least some conditions on the functor are required.

Chapter 5

Axiomatic Completeness

In this chapter, we address the completeness question for the coalgebraic fixpoint logic $\mu\mathtt{ML}_{\nabla}^{\overline{T}}$. Our axiomatization $\mathbf{K}$ can be seen as a generalization of Kozen's proof system for the modal μ-calculus to the coalgebraic level of generality. It consists of an extension of the complete axiomatisation $\mathbf{M}$ for the ∇-based coalgebraic modal logic $\mathtt{ML}_{\nabla}^{\overline{T}}$ from [KKV12] with Kozen's axiom and rule for the fixpoint operators. Hence another way to view the result of this chapter is that it extends the completeness result of [KKV12] to the setting of coalgebraic fixpoint logic. We recall that the nabla approach that we take towards modal logic, deals with a coalgebraic generalization of the same cover modality that is the main operator featuring in Walukiewicz' completeness proof for the modal μ-calculus [Wal00], which as we mentioned in Section 1.3.1.1 was also used in [JW95] to define automata corresponding to formulas of the modal μ-calculus. Just as in Walukiewicz' completeness proof, we use translations between formulas of $\mu\mathtt{ML}_{\nabla}^{\overline{T}}$ and coalgebraic automata. The difference is that we will be more radical and bring automata into the picture at an earlier stage so that all our proofs involve automata in an essential way. The coalgebraic automata that we will employ here were developed by Venema [Ven06] as the automata-theoretic counterpart of $\mu\mathtt{ML}_{\nabla}^{\overline{T}}$.

Here, our approach will follow the same track as in Chapter 4: a pivotal role in our proofs will be played by the notion of a *one-step logic*. As we have seen in previous chapters, one-step logics are simple logical formalisms that feature as the codomain of the transition function of modal automata. Their importance lies in the observation that many results on modal fixpoint logics, often involving nontrivial automata-theoretic phenomena, can be understood at the basic level of one-step logic [FV18, ESV17]. In the following section we will review the main steps of our proof strategy.

5.1 Proof strategy

In this section, we outline our proof strategy. We split the ideas, concepts and technical results into two parts: "Automata, coalgebras and proof theory" and "Games and special automata".

Automata, coalgebras and proof theory Our automata-theoretic approach allows for an explicit study of the interaction between the two main parallel aspects of the completeness proof: the *combinatorics* involved in reasoning with fixpoints, and the *dynamics* encoded in the semantics of the modal operators. In this way, we can make the key concept of a *trace*, which is an essential but fairly informally discussed ingredient in Walukiewicz' proof, more explicit by developing a framework for "managing" traces. Thus our machinery separates *combinatorics* (trace management) from *dynamics* (coalgebra and one-step language), which allows us to deal with the combinatorial and the dynamic concepts in largely separate frameworks. On the other hand, the use of modal automata will allow us to combine these two features, to understand where and how the two perspectives interact, and how they connect to each other. In particular, we will see that the trace theory of an automaton is largely determined by the shape of the formulas of the one-step language.

As we discussed in Section 1.4.3, Walukiewicz' main goal in his proof strategy is to show that every formula of the μ-calculus *provably* implies a semantically equivalent disjunctive formula, that is, a formula in a normal form corresponding to a *disjunctive* automaton, i.e., a non-deterministic parity automaton operating on Kripke models. Technically, the way we achieve this is to work with the full class of coalgebraic modal automata instead of only with disjunctive ones, so we can link formulas and automata by much more elementary techniques: every formula is provably equivalent to a formula in a normal form, that is, the syntactic representation of some coalgebraic modal automaton. Formally, we define a recursive construction providing a modal automaton $\mathbb{A}_\varphi$ for each formula φ, and a translation in the converse direction providing a formula $\mathsf{tr}(\mathbb{A})$ for each modal automaton $\mathbb{A}$. We then prove the following proposition (with $\equiv_{\mathbf{K}}$ denoting provable equivalence with respect to system $\mathbf{K}$):

5.1.1. Theorem. *For every formula $\varphi \in \mu\mathtt{ML}_{\nabla}^{\overline{\top}}$, we have $\varphi \equiv_{\mathbf{K}} \mathsf{tr}(\mathbb{A}_\varphi)$.*

Theorem 5.1.1 takes us "half-way" towards Walukiewicz' result, since it enables us to apply proof-theoretic notions such as derivability and consistency to automata and hence the remainder of the distance can now be addressed by automata-theoretic methods.

Games and special automata The main tools that we employ in our automata-theoretic approach are two kinds of games for modal automata: the satisfiability game and the consequence game, and two special kinds of modal automata: in addition to the disjunctive automata, the class of semi-disjunctive automata.

The *satisfiability game* $\mathcal{S}(\mathbb{A})$ related to an automaton $\mathbb{A}$ was introduced in [FLV10]. It is an infinite two-player game, that can be seen as a streamlined, game-theoretic analog for automata to what tableaux are for formulas. In this game, the dynamics of the semantics appears in the moves of the player $\exists$ who has the role of "model builder", and attempts to construct a satisfying model one layer at a time, while constrained by the one-step transition structure of the automaton.

The combinatorics of the trace theory enters the picture through the winning condition for infinite matches. As we shall see, each infinite match naturally induces a trace graph, an intricate graph structure of which the finite and infinite paths correspond to $\mathbb{A}$-traces: finite and infinite sequences of states of the automaton $\mathbb{A}$. The winning condition of $\mathcal{S}(\mathbb{A})$ states that for $\exists$ to win the infinite match, all infinite traces, corresponding to full branches through this graph, need to satisfy the acceptance condition of $\mathbb{A}$. Intuitively then, the smaller and simpler the trace graph, the easier it is for her to win. In particular, it will be to her advantage if we restrict the use of conjunctions in the one-step language, since these correspond to branching in the trace graph.

As another contribution of this chapter, related to the satisfiability game, we identify a new class of automata, which we call *semi-disjunctive*. These automata can be viewed as an automata-theoretic counterpart to the "weakly aconjunctive formulas" introduced by Walukiewicz. The use of conjunctions in the one-step formulas of a semi-disjunctive automaton is restricted, and even though they are much less constrained than disjunctive automata, their one-step formulas are still of a shape that guarantees the trace theory of an infinite match of the satisfiability game to be well behaved, in the sense that we can guarantee that the collection of bad traces associated with a match of the satisfiability game for a semi-disjunctive automata is finite (modulo a natural equivalence relation of cofinal equality).

The *consequence game* $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ is an original contribution of our approach. It is an infinite two-player game which resembles Walukiewicz' consequence game between tableaux, and can be seen as a kind of implication game between the satisfiability games of two automata, concentrating on establishing *structural connections* between the automata. Its moves revolve around one of the players, prosaically named "player II", trying to establish some structural connection between the two automata to support the claim that $\mathbb{A}$ implies $\mathbb{A}'$. We write $\mathbb{A} \vDash_{\mathcal{C}} \mathbb{A}'$ in case he succeeds, in the sense of having a winning strategy in the game $\mathcal{C}(\mathbb{A}, \mathbb{A}')$.

Completeness proof Bringing all these ideas together, as a key step in our proof, we establish the following generalization of Walukiewicz' main technical result:

5.1.2. THEOREM. *For every formula $\varphi \in \mu\mathtt{ML}_{\nabla}^{\overline{\mathsf{T}}}$, there is a semantically equivalent disjunctive automaton $\mathbb{D}$ such that $\vdash_{\mathbf{K}} \varphi \to \mathsf{tr}(\mathbb{D})$.*

We prove this theorem by a formula induction, and it should not come as a surprise that the key inductive cases are those concerning the fixpoint operators. In particular, the case where φ is of the form $\varphi = \mu x.\alpha(x)$ requires all of the machinery developed earlier on. Finally, from Theorem 5.1.2, the completeness theorem is almost immediate. If φ is an arbitrary consistent formula, then by Theorem 5.1.2 it is semantically equivalent to a consistent disjunctive automaton $\mathbb{D}$. But for disjunctive automata it is easy to prove that consistency implies satisfiability (applying Lemma 5.9.1 and Proposition 5.6.4), and so we are done.

5.2 Coalgebraic fixpoint logic

In this chapter we will work with ∇-based coalgebraic fixpoint logic, as we did in Chapter 3. For the sake of completeness, we first recall the syntax and the semantics of coalgebraic fixpoint logic for a weak pullback-preserving functor T. Here we follow the notation discussed in Remark 2.4.8: we will use $\mu\mathtt{ML}_{\nabla}^{\overline{\mathsf{T}}}$ to refer to the ∇-based fixpoint logic obtained for a functor T with the semantics given by the relation lifting $\overline{\mathsf{T}}$, whereas $\mu\mathtt{ML}_{\nabla}^{\mathsf{T}}$ is used to refer to the ∇-based language for T without specifying the semantics.

5.2.1 Syntax

We fix an infinite set of propositional variables.

5.2.1. DEFINITION. The language $\mu\mathtt{ML}_{\nabla}^{\mathsf{T}}$ of coalgebraic fixpoint formulas is defined by the following grammar:

$$\varphi ::= \bot \mid \top \mid p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \nabla\alpha \mid \neg\varphi \mid \mu p.\varphi \mid \nu p.\varphi$$

where p belongs to the set of propositional variables, and $\alpha \in \mathsf{T}_{\omega}(\mu\mathtt{ML}_{\nabla}^{\mathsf{T}})$. There is a restriction on the formation of the formulas $\mu p.\varphi$ and $\nu p.\varphi$, namely, no occurrence of p in φ may be in the scope of an odd number of negations. We denote by $\mu\mathtt{ML}_{\nabla}^{\mathsf{T}}(\mathtt{X})$ the set of formulas with free variables from set $\mathtt{X}$, and as a convention we usually use letters $p, q, r, \ldots$ to denote bound variables and $x, y, z, \ldots$ for free variables of formulas (for the precise definition of the notions of scope and occurrence we refer the reader to Definition 2.4.2).

5.2.2. DEFINITION. Let φ and $\{\psi_z \mid z \in Z\}$ be modal fixpoint formulas, where Z is a set of variables that are free in φ. Then we let

$$\varphi[\psi_z/z \mid z \in Z]$$

denote the formula obtained from φ by simultaneously substituting each formula ψ_z for z in φ (with the usual understanding that no free variable in any of the ψ_z will get bound by doing so). In case Z is a singleton z, we will simply write $\varphi[\psi_z/z]$, or $\varphi[\psi]$ if z is clear from context. If $Z = Y_1 \uplus Y_2$, it will occasionally be convenient to write $\varphi[\psi_z/z \mid z \in Y_1, \psi_z/z \mid z \in Y_2]$ instead of $\varphi[\psi_z/z \mid z \in Z]$.

5.2.3. FACT. Let $\{\psi_y \mid y \in \mathtt{Y}\}$ and $\{\chi_z \mid z \in \mathtt{Z}\}$ be sets of formulas that are indexed by two disjoint sets of variables $\mathtt{Y}$ and $\mathtt{Z}$. Then for every formula φ we have

(1) $\varphi[\psi_y/y \mid y \in \mathtt{Y}][\chi_z/z \mid z \in \mathtt{Z}] = \varphi\Big[\psi_y[\chi_z/z \mid z \in \mathtt{Z}]/y \mid y \in \mathtt{Y}, \chi_z/z \mid z \in \mathtt{Z}\Big]$

(2) $\varphi[\psi_y/y \mid y \in \mathtt{Y}][\chi_z/z \mid z \in \mathtt{Z}] = \varphi\Big[\psi_y/y \mid y \in \mathtt{Y}, \chi_z/z \mid z \in \mathtt{Z}\Big]$, provided no $z \in \mathtt{Z}$ occurs freely in any ψ_y.

We will sometimes make the assumption (but always explicitly) that our formulas are in negation normal form.

5.2.2 Semantics

To introduce the semantics of $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$ we first define the notion of a T-model over a set $\mathtt{X}$ of propositional letters. In this chapter it will be more convenient to work with *markings* instead of *valuations*, so we will stick to the following definition of a T-model.

5.2.4. DEFINITION. A T-model $\mathbb{S} = (S, \sigma, m)$ is a T-coalgebra (S, σ) together with a marking $m : S \to \mathcal{P}\mathtt{X}$. It will be convenient to think of a T-model $\mathbb{S}$ as a coalgebra $\mathbb{S} = (S, \sigma_m)$ for the functor $\mathsf{T}_{\mathtt{X}}$ defined by $\mathsf{T}_{\mathtt{X}}S := \mathcal{P}\mathtt{X} \times \mathsf{T}S$ where $\sigma_m : S \to \mathsf{T}_{\mathtt{X}}S$ is given by the map (m, σ). It is obvious that any marking $m : S \to \mathcal{P}\mathtt{X}$ induces a valuation $V_m : \mathtt{X} \to \mathcal{P}S$ mapping p to the set $\{s \in S \mid p \in m(s)\}$.

Using the relation lifting $\overline{\mathsf{T}}$ from Example 2.3.27, we define the semantics for the language $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}(\mathtt{X})$ on T-models. Since apart from the nabla modality, the definition of the satisfaction relation $\Vdash_{V_m}$ is exactly the same as it is for the standard μ-calculus (see Definition 2.1.8), here we only recall the definition for the nabla modality:

$$s \Vdash_{V_m} \nabla\alpha \quad \text{iff} \quad (\sigma_m(s), \alpha) \in \overline{\mathsf{T}}(\Vdash_{V_m}).$$

The logic obtained by this particular semantics is denoted by $\mu\mathtt{ML}^{\overline{\mathsf{T}}}_{\nabla}$.

5.2.3 Axiomatics

Our derivation system $\mathbf{K}$ is the extension of the complete derivation system $\mathbf{M}$ for Moss' finitary logic [KKV12][KKV08] with rules and axioms for the fixpoint operators.

5.2.5. DEFINITION. The derivation system **K** which is uniformly parametric in the functor T is given by the following axioms and derivation rules in Table 5.1, together with any complete set of axioms and rules for classical propositional logic.

$(\Delta 1)$	$\dfrac{\{\varphi \to \psi \mid (\varphi, \psi) \in Z\}}{\nabla\alpha \to \nabla\beta}(\alpha, \beta) \in \overline{\mathsf{T}}Z$
$(\Delta 2)$	$\bigwedge\{\nabla\alpha \mid \alpha \in \Gamma\} \to \bigvee\{\nabla(\mathsf{T}\wedge)(\Phi) \mid \Phi \in SRD(\Gamma)\}$
$(\Delta 3)$	$\nabla(\mathsf{T}\vee)(\Phi) \to \bigvee\{\nabla\alpha \mid \alpha \mathrel{\overline{\mathsf{T}}{\in}} \Phi\}$
(A_f)	$\varphi(\mu x.\varphi(x)) \to \mu x.\varphi(x)$
(R_f)	$\dfrac{\varphi(\psi) \to \psi}{\mu x.\varphi \to \psi}$

Table 5.1: Rules and axioms of the system **K**

The axioms $(\Delta 2)$ and $(\Delta 3)$ are governing the interaction of ∇ with conjunctions and disjunctions respectively and can be seen as *modal distributive laws*. Here we see conjunction and disjunction ($\wedge$ and $\vee$) as maps from $\mathcal{P}_\omega(\mu\mathtt{ML}^{\mathsf{T}}_{\nabla})$ to $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$, so we can apply T to them and get maps $\mathsf{T}\wedge$ and $\mathsf{T}\vee$. In addition we denote the set of all slim redistributions of Γ by $SRD(\Gamma)$ as in Definition 2.3.31. The rule $(\Delta 1)$ can be read as a congruence and monotonicity rule in one. It has a side condition expressing that it may only be applied when the set of premisses is indexed by a relation Z such that (α, β) belongs to the lifted relation $\overline{\mathsf{T}}Z$. (A_f) and (R_f) are the standard axiom and rule for least fixpoints.

The notions of *derivability* with respect to this system is standard. If there is a derivation of the formula φ, we write $\vdash_{\mathbf{K}} \varphi$. Given formulas φ and ψ we say φ *provably implies* ψ, notation: $\varphi \leq_{\mathbf{K}} \psi$, if $\vdash_{\mathbf{K}} \varphi \to \psi$. We write $\varphi \equiv_{\mathbf{K}} \psi$ in the case that both $\varphi \leq_{\mathbf{K}} \psi$ and $\psi \leq_{\mathbf{K}} \varphi$ hold. A formula φ is **K**-*consistent* or simply *consistent* if $\varphi \to \bot$ is not derivable in **K**.

5.2.6. EXAMPLE. In the case of the power set functor ($\mathsf{T} = \mathcal{P}$) the axioms $(\Delta 2)$ and $(\Delta 3)$ look as follows respectively:

$$\bigwedge\{\nabla\alpha \mid \alpha \in \Gamma\} \to \bigvee\{\nabla\{\bigwedge\beta \mid \beta \in \Phi\} \mid \bigcup\Gamma = \bigcup\Phi \text{ and } \alpha \cap \beta \neq \emptyset \text{ for all } \alpha \in \Gamma, \beta \in \Phi\}$$

$$\nabla\{\bigvee\beta \mid \beta \in \Phi\} \to \bigvee\{\nabla\alpha \mid \alpha \subseteq \bigcup\Phi \text{ and } \alpha \cap \beta \neq \emptyset \text{ for all } \beta \in \Phi\}$$

5.3 Modal automata and one-step formulas

As mentioned in the introduction of this chapter, automata play a crucial role in the completeness proof presented by Walukiewicz [Wal00]. In the completeness proof we present in this chapter for $\mu\mathtt{ML}_{\nabla}^{\overline{\mathsf{T}}}$, we strengthen the role of automata and define a translation transforming a formula of $\mu\mathtt{ML}_{\nabla}^{\overline{\mathsf{T}}}$ into an equivalent *modal automaton*. Of course, there are already a few different methods available for this transformation, for instance, in the method introduced by Janin and Walukiewicz [JW95] first a tableau is built up from a given formula, and then the equivalent automaton is constructed from this tableau. The output automaton from this method is already a *non-deterministic* automaton (called *disjunctive* automaton in our approach) which is not suitable for our purpose, since we want to work with a wider class of modal automata introduced by Wilke [Wil01]. We will define the translation in one go by induction on the complexity of formulas, making use of certain closure properties of modal automata.

Here we first recall the notion of *one-step* logic. As we discussed in Section 2.4.3, one-step logic determines the transition map of modal automata. This notion originated from the theory of coalgebraic logic. Intuitively one-step logic provides the syntax and semantics to extract information about the one-step level of coalgebras. Notions like one-step syntax (a language consists of one-step formulas), one-step semantics, one-step derivation system and one-step model theory naturally come into the picture, see for instance[KKV12][SP09b].

5.3.1 One-step logic

Modal automata are based on the one-step language which consists of modal formulas of rank one. The basic definitions were already given in Section 2.4.3, but since one-step logic plays a crucial role in this chapter, we include the formal definitions here as well.

5.3.1. Definition. Given a set A, we define the set $\mathtt{LF}(A)$ of lattice terms over A through the following grammar:

$$\pi ::= \bot \mid \top \mid a \mid \pi \wedge \pi \mid \pi \vee \pi,$$

where $a \in A$. Given two sets $\mathtt{X}$ and A, we define the set $\mathtt{1ML}_{\nabla}(\mathtt{X}, A)$ of *modal one-step formulas* over A with respect to $\mathtt{X}$ inductively by

$$\alpha ::= \bot \mid \top \mid p \mid \neg p \mid \nabla\beta \mid \alpha \wedge \alpha \mid \alpha \vee \alpha,$$

with $p \in \mathtt{X}$ and $\beta \in \mathsf{T}_{\omega}\mathtt{LF}(A)$.

Note that elements from the two parameter sets, $\mathtt{X}$ and A, are treated quite differently in the syntax of one-step formulas: all occurrences of elements of $\mathtt{X}$, corresponding to the proposition letters, must be unguarded, whereas the elements of A, corresponding to bound variables of a formula and to states of our modal automata, may only occur in the scope of exactly one modality. Observe as well that all formulas in $\mathtt{1ML}_\nabla(\mathtt{X}, A)$ are positive in A.

One-step formulas will be interpreted in *one-step models* which consist of a one-step frame together with a marking.

5.3.2. DEFINITION. A *one-step* $\mathsf{T}_\mathtt{X}$*-frame* is a pair (S, ξ) with $\xi \in \mathsf{T}_\mathtt{X}S$. A *one-step* $\mathsf{T}_\mathtt{X}$*-model* over a set A of variables is a triple (S, ξ, m) consisting of a set S, a chosen object $\xi \in \mathsf{T}_\mathtt{X}S$ and a marking $m : S \to \mathcal{P}A$.

Using the definition of one-step model we can inductively define the semantics of the one-step language.

5.3.3. DEFINITION. The *one-step satisfaction relation* between one-step models and one-step formulas is defined as follows. Fix a one-step model (S, ξ, m) with $\xi = (\mathtt{Y}, \xi') \in \mathsf{T}_\mathtt{X}S$ and a modal one-step language $\mathtt{1ML}_\nabla(\mathtt{X}, A)$. First, we define the meaning function $[\![-]\!]^0_m : \mathtt{LF}(A) \to \mathcal{P}S$ for lattice formulas over A by induction, setting $[\![a]\!]^0_m = \{s \in S \mid a \in m(s)\}$ for $a \in A$, and treating conjunctions and disjunctions in the obvious manner. We write $S, s \Vdash^0_m \pi$ to indicate $s \in [\![\pi]\!]^0_m$.

The meaning function $[\![-]\!]^0_m$ induces a map $[\![-]\!]^1_m : \mathtt{1ML}_\nabla(\mathtt{X}, A) \to \mathcal{P}\mathsf{T}_\mathtt{X}S$ interpreting one-step formulas as subsets of $\mathsf{T}_\mathtt{X}S$. Before giving the definition of $[\![-]\!]^1_m$ we recall that every $\xi \in \mathsf{T}_\mathtt{X}S$ is of the form $(\mathtt{Y}, \xi') \in \mathcal{P}\mathtt{X} \times \mathsf{T}S$.

Going back to the map $[\![-]\!]^1_m$, it has the usual clauses for conjunction and disjunction, and the following clauses for the propositional letters and the modal operator:

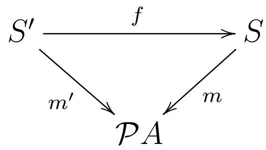
- $\xi = (\mathtt{Y}, \xi') \in [\![p]\!]^1_m$ iff $p \in \mathtt{Y}$
- $\xi = (\mathtt{Y}, \xi') \in [\![\neg p]\!]^1_m$ iff $p \notin \mathtt{Y}$
- $\xi = (\mathtt{Y}, \xi') \in [\![\nabla\beta]\!]^1_m$ iff $(\xi', \beta) \in \overline{\mathsf{T}}(\Vdash^0_m)$

We write $S, \xi \Vdash^1_m \varphi$, or $S, \xi, m \Vdash^1 \varphi$ to indicate $\xi \in [\![\varphi]\!]^1_m$, and refer to this relation as the *one-step semantics*. When it is clear from the context that we are dealing with on-step formulas and there is no risk for confusion, we may drop the superscript 1 from the notation of meaning function.

5.3.4. DEFINITION. Let α and α' be one-step formulas. The formula α is *one-step satisfiable* if there is a one-step model (S, ξ, m) such that $S, \xi, m \Vdash^1 \alpha$, and *one-step valid* if $S, \xi, m \Vdash^1 \alpha$ for all one-step models (S, ξ, m). We say that α' is a *one-step consequence* of α (written $\alpha \models^1 \alpha'$) if $S, \xi, m \Vdash^1 \alpha$ implies $S, \xi, m \Vdash^1 \alpha'$, for all one-step models (S, ξ, m), and that α and α' are *one-step equivalent*, notation: $\alpha \equiv^1 \alpha'$, if $\alpha \models^1 \alpha'$ and $\alpha' \models^1 \alpha$.

We also need morphisms between one-step frames and models.

5.3.5. Definition. A *one-step frame morphism* between two one-step frames (S', ξ') and (S, ξ) is a map $f : S' \to S$ such that $(\mathsf{T_X} f)\xi' = \xi$. In case such a map satisfies $m' = m \circ f$,



for some markings m and m' on S and S', respectively, we say that f is a *one-step model morphism* from (S', ξ', m') to (S, ξ, m).

One of the main results about the one-step language is bisimulation invariance, i.e., all one-step formulas are invariant for bisimulation between one-step models. An instance of the one-step bisimulation invariance that we consider below is called *naturality* of one-step formulas. In fact, naturality of one-step formulas is the property that ensures that the truth of one-step formulas is invariant under one-step morphisms, and so it enables us to safely apply maps to one-step models and get new one-step models that satisfy the same formulas. We call this property "naturality", since in the case of coalgebraic modal logic with predicate liftings, it boils down to the naturality of predicate liftings used in the syntax.

5.3.6. Proposition. *[Naturality of one-step formulas] Let $f : (S', \xi', m') \to (S, \xi, m)$ be a morphism of one-step models over A. Then for every one-step formula $\varphi \in \mathtt{1ML}_\nabla(\mathtt{X}, A)$ we have*

$$S', \xi', m' \Vdash^1 \varphi \iff S, \xi, m \Vdash^1 \varphi.$$

Formulating it differently, for any one-step frame (S', ξ'), any marking $m : S \to \mathcal{P}A$, and any map $f : S' \to S$, we have

$$S', \xi', m \circ f \Vdash^1 \varphi \iff S, (\mathsf{T_X} f)\xi', m \Vdash^1 \varphi.$$

As a specific instance of this invariance result we obtain the following corollary which we mention explicitly for future reference.

5.3.7. Corollary. *Let (S, ξ, m) be a one-step model over A, and let $T \subseteq S$ be a subset of S such that $\xi \in \mathsf{T_X} T$. Then for every formula $\varphi \in \mathtt{1ML}_\nabla(\mathtt{X}, A)$ we have*

$$S, \xi, m \Vdash^1 \varphi \iff T, \xi, m{\restriction}_T \Vdash^1 \varphi.$$

Proof:
Immediate from Proposition 5.3.6 by the observation that the inclusion map $i : T \hookrightarrow S$ is a one-step model morphism. □

The following proposition states that the meaning of a one-step formula only depends on the variables occurring in it.

5.3.8. PROPOSITION. *Let (S, ξ, m) be a one-step model over A, and let $\varphi \in \mathtt{1ML}_\nabla(\mathtt{X}, A)$ be a one-step formula which belongs to the set $\mathtt{1ML}_\nabla(\mathtt{X}, B)$, for some subset $B \subseteq A$. Then we have*

$$S, \xi, m \Vdash^1 \varphi \Longleftrightarrow S, \xi, m^B \Vdash^1 \varphi,$$

where m^B is the B-marking given by $m^B(s) := m(s) \cap B$.

5.3.9. DEFINITION. The *boolean* dual φ^δ of a formula $\varphi \in \mathtt{1ML}_\nabla(\mathtt{X}, A)$ is the formula we obtain from φ by simultaneously replacing all occurrences of $p \in \mathtt{X}$ with $\neg p$ and vice versa, $\vee$ with $\wedge$ and vice versa, and ∇ with Δ (see Definition 3.2.11).

Because all the formulas in $\mathtt{1ML}_\nabla(\mathtt{X}, A)$ are positive in every variable $a \in A$, we have the following monotonicity property.

5.3.10. PROPOSITION. *Let (S, α) be a one-step frame, and let $m, m' : S \to \mathcal{P}A$ be two markings such that $m(s) \subseteq m'(s)$, for all $s \in S$. Then we have*

$$S, \alpha, m \Vdash^1 \varphi \text{ implies } S, \alpha, m' \Vdash^1 \varphi,$$

for any formula $\varphi \in \mathtt{1ML}_\nabla(\mathtt{X}, A)$.

For technical reasons, we need the following *binary* version of the modal distributive law for one-step conjunctions. See [Ven19, Definition 5.25] for a more general version of this modal distributive law.

5.3.11. PROPOSITION. *Given $\alpha_1, \alpha_2 \in \mathsf{T}_\omega\mathtt{LF}(A)$ the following holds:*

$$\nabla\alpha_1 \wedge \nabla\alpha_2 \equiv_{\mathbf{K}} \bigvee\{\nabla(\mathsf{T}\wedge)\alpha \mid \alpha \in \mathsf{T}(\mathrm{Base}(\alpha_1) \times \mathrm{Base}(\alpha_2)) \\ \text{and } \mathsf{T}\pi_i(\alpha) = \alpha_i \text{ for } i \in \{1, 2\}\},$$

where the conjunction on the right hand side is the conjunction map between lattice terms over A, $\wedge : \mathtt{LF}(A) \times \mathtt{LF}(A) \to \mathtt{LF}(A)$.

To prove this result we can use properties of weak pullback-preserving functors to show that these formulas are semantically equivalent, and then from the one-step completeness result of [KKV12] derive that they are provably equivalent. We first define the notion of *canonical valuations* and then state the one-step completeness theorem from [KKV12], and finally prove Proposition 5.3.11.

5.3.12. Definition. Given a set A and let $A^\sharp := \mathcal{P}(A \times A)$ denote the set of binary relations over A. The *canonical* or *natural a-valuation* $V_a^A : A \to \mathcal{P}A^\sharp$ is given by

$$V_a^A : b \mapsto \{R \in A^\sharp \mid (a, b) \in R\}.$$

Its transpose, i.e., the corresponding *canonical* or *natural a-marking* on the set $A^\sharp$ is defined as the map $n_a^A : A^\sharp \to \mathcal{P}A$ given by

$$n_a^A : R \mapsto R[a] = \{b \in A \mid (a, b) \in R\}.$$

For $\alpha \in \mathsf{T}_{\mathtt{X}}A^\sharp$ and $\varphi \in \mathtt{1ML}_\nabla(\mathtt{X}, A)$, we write $\alpha \Vdash_a^1 \varphi$ to denote that $A^\sharp, n_a^A, \alpha \Vdash^1 \varphi$, and we define $[\![\varphi]\!]_a^1 := \{\alpha \in \mathsf{T}_{\mathtt{X}}A^\sharp \mid \alpha \Vdash_a^1 \varphi\}$. When there is no risk for confusion, we may drop the superscript A from the notation of natural valuation and marking.

5.3.13. Remark. The notation $[\![\varphi]\!]_a^1$ may seem to be somewhat ambiguous, since it does not refer to the ambient variable set A. However, by Proposition 5.3.6 and Corollary 5.3.7 it follows that, for any pair of sets A, B such that $\alpha \in \mathtt{1ML}_\nabla(\mathtt{X}, A) \cap \mathtt{1ML}_\nabla(\mathtt{X}, B)$ we have

$$\{\alpha \in \mathsf{T}_{\mathtt{X}}A^\sharp \mid A^\sharp, n_a^A, \alpha \Vdash^1 \varphi\} = \{\alpha \in \mathsf{T}_{\mathtt{X}}B^\sharp \mid B^\sharp, n_a^B, \alpha \Vdash^1 \varphi\}.$$

As another instance of Corollary 5.3.7, for any subset $\mathcal{R} \subseteq A^\sharp$ and for any object $\alpha \in \mathsf{T}_{\mathtt{X}}\mathcal{R}$ we have

$$A^\sharp, \alpha, n_a^A \Vdash^1 \varphi \iff \mathcal{R}, \alpha, n_a^A{\restriction_{\mathcal{R}}} \Vdash^1 \varphi,$$

where $n_a^A{\restriction_{\mathcal{R}}}$ is the natural a-marking on $A^\sharp$, restricted to $\mathcal{R}$. If no confusion is likely, we will often denote the marking $n_a^A{\restriction_{\mathcal{R}}}$ simply by n_a^A.

5.3.14. Remark. We may think of any object $\alpha \in \mathsf{T}_{\mathtt{X}}A^\sharp$ as a *family* $\{(A^\sharp, n_a^A, \alpha) \mid a \in A\}$ of one-step models *on the same one-step frame* $(A^\sharp, \alpha)$. It may occasionally be useful, however, to consider this 'family of one-step models' as one single model. To do so, we involve, for each $a \in A$, the substitution $\tau_a : A \to A \times A$ that *tags* each variable $b \in A$ with its 'origin' a, that is, $\tau_a : b \mapsto (a, b)$. One may verify, on the basis of a straightforward formula induction, that

$$A^\sharp, n_a^A, \alpha \Vdash^1 \varphi \iff A^\sharp, \mathsf{id}_{A^\sharp}, \alpha \Vdash^1 \varphi[\tau_a]$$

for each one-step formula $\varphi \in \mathtt{1ML}_\nabla(\mathtt{X}, A)$. In particular, it follows that

$$\alpha \in \bigcap_{a \in B} [\![\Theta(a)]\!]_a^1 \iff A^\sharp, \mathsf{id}_{A^\sharp}, \alpha \Vdash^1 \bigwedge_{a \in B} \varphi[\tau_a],$$

for any family $\{\Theta(a) \mid a \in B\}$ of formulas.

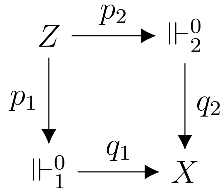
For the next proposition, recall the notion of Base from Definition 2.3.21.

5.3.15. Proposition (One-step completeness). *Let Θ be a map assigning a one-step formula over A to each $a \in A$, and let $\{\tau_a \mid a \in A\}$ be a collection of maps from A to formulas in $\mu\mathtt{ML}_{\nabla}^{\overline{\mathsf{T}}}$. If for some subset $B \subseteq A$ the conjunction $\bigwedge_{a \in B} \Theta(a)[\tau_a]$ is consistent, then there is some element $(\mathtt{Y}, \alpha) \in \mathsf{T}_{\mathtt{X}} A^\sharp$ such that $(\mathtt{Y}, \alpha) \in \bigcap_{a \in B} [\![\Theta(a)]\!]_a$ and, for each $Q \in$ Base(α), the conjunction $\bigwedge_{(a,b) \in Q} \tau_a(b)$ is consistent.*

This statement of the proposition is formulated in a way that fits our present setting and slightly deviates from how it is stated in [KKV12], but this version of the result follows from the one in [KKV12]. Intuitively, the map Θ will in practice be the transition map of some modal automaton, and the maps τ_a for $a \in A$ will be assignments of “strengthened” versions of formulas corresponding to states in A. Note that in the case when T is the powerset functor, an element of $\mathsf{T}_{\mathtt{X}} A^\sharp$ is just a pair $(\mathtt{Y}, \mathcal{R})$ where $\mathtt{Y}$ is a set of proposition letters, and Base is just the identity map. So in this case, the one-step completeness statement becomes:

5.3.16. Proposition (One-step completeness for $\mathsf{T} = \mathcal{P}$). *Let Θ be a map assigning a one-step formula over A to each $a \in A$, and let $\{\tau_a \mid a \in A\}$ be a collection of maps from A to formulas in $\mu\mathtt{ML}_{\nabla}^{\overline{\mathcal{P}}}$. If for some subset $B \subseteq A$ the conjunction $\bigwedge_{a \in B} \Theta(a)[\tau_a]$ is consistent, then there is some element $(\mathtt{Y}, \mathcal{R}) \in \mathcal{P}_{\mathtt{X}} A^\sharp$ such that $(\mathtt{Y}, \mathcal{R}) \in \bigcap_{a \in B} [\![\Theta(a)]\!]_a$ and, for each $Q \in \mathcal{R}$, the conjunction $\bigwedge_{(a,b) \in Q} \tau_a(b)$ is consistent.*

Proof of Proposition 5.3.11:
We prove the semantical equivalence over one-step models.
($\Longrightarrow$) Given a one-step model (X, ξ, m) such that $X, \xi \Vdash^1_m \nabla\alpha_1 \wedge \nabla\alpha_2$. In order to find a suitable $\alpha \in \mathsf{T}(\text{Base}(\alpha_1) \times \text{Base}(\alpha_2))$, we define the binary relation $Z : X \nrightarrow \text{Base}(\alpha_1) \times \text{Base}(\alpha_2)$ by $(x, (a, b)) \in Z$ iff $x \Vdash^0_m a$ and $x \Vdash^0_m b$. Here $\Vdash^0_m: X \nrightarrow \text{Base}(\alpha_1) \cup \text{Base}(\alpha_2)$ is the restriction of the satisfaction relation on X and $\text{Base}(\alpha_1) \cup \text{Base}(\alpha_2)$. Now consider the following diagram with $\Vdash^0_i$ denoting the relation $\Vdash^0_i: X \nrightarrow \text{Base}(\alpha_i)$ (the zero-step satisfaction relation according to the marking m). For $i \in \{1, 2\}$ the map q_i is the projection on the first component for the relation $\Vdash^0_i$, and $p_i : Z \to \Vdash^0_i$ is the map defined by $(x, (a_1, a_2)) \mapsto (x, a_i)$. It is easy to see that the following diagram is commutative, and (Z, p_1, p_2) is a pullback for (q_1, q_2).



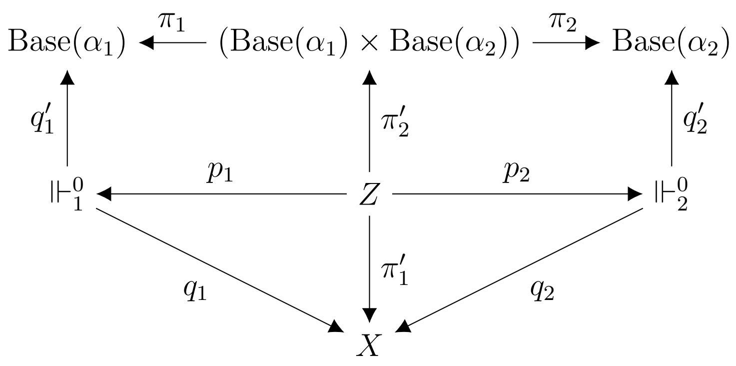


Figure 5.1: Extended pullback diagram

We extend this pullback diagram to the diagram in Figure 5.1, where π_i and π'_i are projection maps on $\text{Base}(\alpha_1) \times \text{Base}(\alpha_2)$ and Z respectively, and $q'_i : \Vdash^0_i \to \text{Base}(\alpha_i)$ is the second projection on $\Vdash^0_i$. It is easy to check that all the squares and triangles of the latter diagram commute.

It then follows by the functoriality of T that by applying T to the extended pullback diagram we still get commuting squares and triangles. Now because $X, \xi \Vdash^1_m \nabla\alpha_1 \wedge \nabla\alpha_2$ we get that $X, \xi \Vdash^1_m \nabla\alpha_i$ and so $(\xi, \alpha_i) \in \overline{\mathsf{T}} \Vdash^0_i$ for $i \in \{1, 2\}$.

By applying the definition of $\overline{\mathsf{T}}$ we obtain $\gamma_i \in \mathsf{T} \Vdash^0_i$ such that:

$$\mathsf{T}q_i(\gamma_i) = \xi \quad \text{and} \quad \mathsf{T}q'_i(\gamma_i) = \alpha_i. \tag{5.1}$$

From $\mathsf{T}q_1(\gamma_1) = \mathsf{T}q_2(\gamma_2) = \xi$ and the facts that (Z, p_1, p_2) is a pullback and T weakly preserves pullbacks, we get a $\gamma \in \mathsf{T}Z$ such that:

$$\mathsf{T}p_i(\gamma) = \gamma_i. \tag{5.2}$$

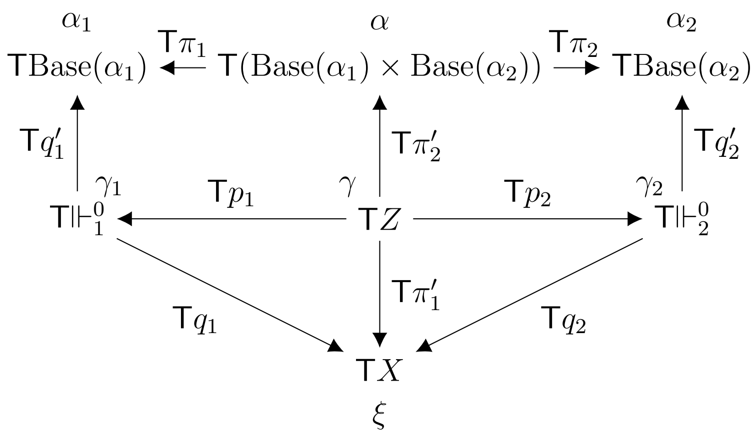
We claim that $\alpha := \mathsf{T}\pi'_2(\gamma) \in \mathsf{T}(\text{Base}(\alpha_1) \times \text{Base}(\alpha_2))$ does the job for us.

1. Claim. *For $\alpha := \mathsf{T}\pi'_2(\gamma)$ the following hold:*

(i) $X, \xi \Vdash^1_m \nabla(\mathsf{T}\wedge)\alpha$;

(ii) $\mathsf{T}\pi_i(\alpha) = \alpha_i$.

Proof of Claim 1: Consider the following diagram.



For $i \in \{1, 2\}$ we have

$$\begin{aligned}
\mathsf{T}\pi_i(\alpha) &= \mathsf{T}\pi_i((\mathsf{T}\pi_2')(\gamma)) \\
&= \mathsf{T}q_i'(\mathsf{T}p_i(\gamma)) \\
&\overset{(5.2)}{=} \mathsf{T}q_i'(\gamma_i) \\
&\overset{(5.1)}{=} \alpha_i
\end{aligned}$$

which proves item (*ii*) of the claim. For item (*i*) first observe that $Z \,;\, \wedge \subseteq \Vdash_m^0$ which implies that $\overline{\mathsf{T}}Z \,;\, \mathsf{T}\wedge \subseteq \overline{\mathsf{T}}\Vdash_m^0$. On the other hand we have:

$$\begin{aligned}
\mathsf{T}\pi_1'(\gamma) &= (\mathsf{T}q_2 \circ \mathsf{T}p_2)(\gamma) \\
&= \mathsf{T}q_2(\mathsf{T}p_2(\gamma)) \\
&\overset{(5.2)}{=} \mathsf{T}q_2(\gamma_2) \\
&\overset{(5.1)}{=} \xi
\end{aligned}$$

and $\mathsf{T}\pi_2'(\gamma) = \alpha$, hence $(\xi, \alpha) \in \overline{\mathsf{T}}Z$. Consequently $(\xi, (\mathsf{T}\wedge)\alpha) \in \overline{\mathsf{T}}\Vdash_m^0$ which means $X, \xi \Vdash_m^1 \nabla(\mathsf{T}\wedge)\alpha$. This finishes the proof of item (*i*) and so the proof of Claim.

□

($\Longleftarrow$) For this direction we assume:

$X, \xi \Vdash_m^1 \bigvee\{\nabla(\mathsf{T}\wedge)\alpha \mid \alpha \in \mathsf{T}(\mathrm{Base}(\alpha_1)\times\mathrm{Base}(\alpha_2)) \text{ and } \mathsf{T}\pi_i(\alpha) = \alpha_i \text{ for } i \in \{1, 2\}\}$.

This means there exists $\alpha \in \mathsf{T}(\mathrm{Base}(\alpha_1) \times \mathrm{Base}(\alpha_2))$ such that $X, \xi \Vdash_m^1 \nabla(\mathsf{T}\wedge)\alpha$ and $\mathsf{T}\pi_i(\alpha) = \alpha_i$ for $i \in \{1, 2\}$. So from this we get:

- $(\xi, (\mathsf{T}\wedge)\alpha) \in \overline{\mathsf{T}}\Vdash_m^0$

and by Fact 2.3.30 we have:

- $((\mathsf{T}\wedge)\alpha, \alpha) \in (\mathsf{T}\wedge)^\circ = \overline{\mathsf{T}}(\wedge^\circ)$
- $(\alpha, \alpha_i) \in \mathsf{T}\pi_i = \overline{\mathsf{T}}\pi_i$

From this it follows that $(\xi, \alpha_i) \in \mathsf{T}\Vdash^0_m \,;\, \overline{\mathsf{T}}\wedge^\circ \,;\, \overline{\mathsf{T}}\pi_i$, so that, using the fact that $\Vdash^0_m \,;\, \wedge^\circ \,;\, \pi_i \subseteq \Vdash^0_i$, and that T preserves weak pullbacks we consequently get $(\xi, \alpha_i) \in \overline{\mathsf{T}}\Vdash^0_i \subseteq \overline{\mathsf{T}}\Vdash^0_m$. This finishes the proof of Proposition 5.3.11. □

5.3.2 Modal automata

In this section we recall the definition of modal automata from Section 2.4.3 and present some automata-related notions that we will use in the remainder of this chapter.

5.3.17. Definition. A *(coalgebraic) modal* $\mathtt{X}$*-automaton* is a quadruple $\mathbb{A} = (A, \Theta, \Omega, a_I)$ such that A is a finite set of *states*, $\Theta : A \to \mathtt{1ML}_\nabla(\mathtt{X}, A)$ is the *transition map* of $\mathbb{A}$, $\Omega : A \to \omega$ is the *priority map* of $\mathbb{A}$, and a_I is the *initial state*.

The *underlying structure* of an automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ is the triple (A, Θ, Ω). With $b \in A$, we let $\mathbb{A}\langle b\rangle$ denote the variant of $\mathbb{A}$ that takes b as its starting state, i.e., $\mathbb{A}\langle b\rangle = (A, \Theta, \Omega, b)$.

Modal $\mathtt{X}$-automata run on T-models over $\mathtt{X}$, and acceptance is defined in terms of a two-player game, the *acceptance game*.

5.3.18. Definition. Let $\mathbb{A} = (A, \Theta, \Omega, a_I)$ be a modal $\mathtt{X}$-automaton and let $\mathbb{S} = (S, \sigma, m)$ be a T-model. The associated acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})$ is the parity game given by Table 5.2:

Position	Player	Admissible moves
$(a, s) \in A \times S$	$\exists$	$\{V : A \to \mathcal{P}S \mid \sigma_m(s) \in [\![\Theta(a)]\!]^1_V\}$
$V : A \to \mathcal{P}S$	$\forall$	$\{(b, t) \in A \times S \mid t \in V(b)\}$

Table 5.2: Acceptance Game

The loser of a finite match is the player who got stuck; the winner of an infinite match is $\exists$ if the greatest parity that appears infinitely often in the match is even, and it is $\forall$ if this parity is odd. A pointed T-model $(\mathbb{S}, s_I)$ is *accepted* by the automaton $\mathbb{A}$ if (a_I, s_I) is a winning position for player $\exists$ in $\mathcal{A}(\mathbb{A}, \mathbb{S})$. We refer to the class of pointed T-models that are accepted by $\mathbb{A}$ as the *language recognized by* $\mathbb{A}$, and denote it with $L(\mathbb{A})$.

5.3.19. Definition. Let $\mathbb{A}$ and $\mathbb{A}'$ be two modal automata. We say that $\mathbb{A}$ *(semantically) implies* $\mathbb{A}'$ (notation: $\mathbb{A} \leq \mathbb{A}'$), if $L(\mathbb{A}) \subseteq L(\mathbb{A}')$, and that $\mathbb{A}$ and $\mathbb{A}'$ are *equivalent* (notation: $\mathbb{A} \equiv \mathbb{A}'$), if they recognise the same language, i.e., $L(\mathbb{A}) = L(\mathbb{A}')$. An automaton $\mathbb{A}$ is equivalent to a formula φ in $\mu\mathtt{ML}^{\mathsf{T}}_{\nabla}$ if any pointed T-model $(\mathbb{S}, s)$ is accepted by $\mathbb{A}$ iff $(\mathbb{S}, s) \Vdash \varphi$.

In the sequel, we will need the following strong version of equivalence between automata.

5.3.20. Definition. Two modal automata $\mathbb{A} = (A, \Theta, \Omega, a_I)$ and $\mathbb{A}' = (A', \Theta', \Omega', a'_I)$ are *one-step equivalent*, (notation: $\mathbb{A} \equiv_1 \mathbb{A}'$), if $A = A'$, $\Omega = \Omega'$, $a_I = a'_i$ and $\Theta(a) \equiv_1 \Theta'(a)$ for all $a \in A$.

It is obvious that one-step equivalence implies equivalence.

In the remainder of this subsection we introduce various concepts and notations related to modal automata and automaton structures.

5.3.21. Definition. The *(directed) graph* of $\mathbb{A}$ is the structure $(A, E_{\mathbb{A}})$, where $aE_{\mathbb{A}}b$ if a occurs in $\Theta(b)$, and we let $\lhd_{\mathbb{A}}$ denote the transitive closure of $E_{\mathbb{A}}$. If $a \lhd_{\mathbb{A}} b$ we say that a is *active* in b. We write $a \bowtie_{\mathbb{A}} b$ if $a \lhd_{\mathbb{A}} b$ and $b \lhd_{\mathbb{A}} a$. A *cluster* of $\mathbb{A}$ is a cell of the equivalence relation generated by $\bowtie_{\mathbb{A}}$; a cluster C is *degenerate* if it is of the form $C = \{a\}$ with $a \not\bowtie_{\mathbb{A}} a$. For a state a we denote by C_a the unique cluster of $\mathbb{A}$ to which a belongs.

5.3.22. Definition. Fix an automaton structure $\mathbb{A} = (A, \Theta, \Omega)$. The *size* $|\mathbb{A}|$ of $\mathbb{A}$ is defined as the cardinality of its carrier A. We write $a \sqsubset_{\mathbb{A}} b$ if $\Omega(a) < \Omega(b)$, and $a \sqsubseteq_{\mathbb{A}} b$ if $\Omega(a) \leq \Omega(b)$. When clear from context we sometimes write $\sqsubset$ and $\sqsubseteq$ instead, dropping the reference to $\mathbb{A}$. We say that $\mathbb{A}$ is *positive* in a proposition letter $p \in \mathtt{X}$ if each occurrence of p in each formula $\Theta(a)$ is positive. A state $a \in A$ is called a *true* state of $\mathbb{A}$ if $\Theta(a) = \top$.

5.3.3 Operations on modal automata

We now introduce the logical operations on modal automata that will enable us to translate formulas to modal automata, and later to connect proof theoretic concepts with automata theory. Some of these operators like complementation and union are standard. Our definitions of least and greatest fixpoints of modal automata, and of substitution, are new as far as we know.

Conjunction and disjunction Suppose we are given modal $\mathtt{X}$-automata $\mathbb{A} = (A, \Theta_A, \Omega_A, a_I)$ and $\mathbb{B} = (B, \Theta_B, \Omega_B, b_I)$. We define the automaton $\mathbb{A} \wedge \mathbb{B} = (C, \Theta_C, \Omega_C, a_C)$ as follows:

- a_C is some arbitrarily chosen object not in $A \uplus B$, and C is defined to be $A \uplus B \uplus \{a_C\}$.

- $\Theta_C(a_C) := \Theta_A(a_I) \wedge \Theta_B(b_I)$ and $\Omega_C(a_C) := n + 1$ where n is the maximum priority of $\mathbb{A}$ and $\mathbb{B}$.
- For $a \in A$, $\Theta_C(a) := \Theta_A(a)$ and $\Omega_C(a) := \Omega_A(a)$.
- For $b \in B$, $\Theta_C(a) := \Theta_B(b)$ and $\Omega_C(b) := \Omega_B(b)$.

Disjunction is handled similarly by setting $\Theta_C(a_C) := \Theta_A(a_I) \vee \Theta_B(b_I)$ instead.

Negation Negation corresponds to *complementation* on the automata side [KV09], and for this we apply the notion of boolean dual φ^δ of a formula φ (see Definition 3.2.11 for boolean dual of ∇-formulas, other cases are defined standardly.

Given a modal automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ we define the automaton $\neg\mathbb{A} := (A, \Theta', \Omega', a_I)$ by setting, for each $a \in A$:

- $\Theta'(a) := \Theta(a)^\delta$
- $\Omega'(a) := \Omega(a) + 1$.

Nabla modality Given a finite set $\{\mathbb{A}_i\}_{i\in I}$ of automata with $\mathbb{A}_i = (A_i, \Theta_i, \Omega_i, a_i)$, and $\alpha \in \mathsf{T}\{\mathbb{A}_i\}_{i\in I}$, we define the automaton $\nabla\alpha := (A, \Theta, \Omega, a_I)$ as follows:

- $A = \biguplus_{i\in I}\{A_i\} \uplus \{a_I\}$ with $a_I \notin \biguplus_{i\in I}\{A_i\}$.

Θ' and Ω' agree with, respectively, Θ_i and Ω_i on A_i, whereas for the initial state a'_I we define:

- $\Theta(a_I) := \nabla\mathsf{T}f(\alpha)$, where $f : \{\mathbb{A}_i\}_{i\in I} \to \mathtt{LF}(A)$ is the map sending each automaton $\mathbb{A}_i$ to its initial state a_i.
- $\Omega'(a_I) := n + 1$,

where n is the maximum priority of $\{\mathbb{A}_i\}_{i\in I}$.

Next we define a substitution operation on automata.

Substitution Let $\mathbb{A} = (A, \Theta_A, \Omega_A, a_I)$ and $\mathbb{B} = (B, \Theta_B, \Omega_B, b_I)$ be modal automata over the languages $\mathtt{X} \uplus \{p\}$ and $\mathtt{X}$, respectively, and assume that $\mathbb{A}$ is positive in p. We define the modal $\mathtt{X}$-automaton $\mathbb{A}[\mathbb{B}/p]$ as the structure $(C, \Theta_C, \Omega_C, c_I)$, where:

- $C := A \uplus B$ and $c_I := a_I$.
- For $c \in A$, $\Theta_C(c) := \Theta_A(c)[\Theta_B(b_I)/p]$ and for $c \in B$, $\Theta_C(c) := \Theta_B(c)$.
- Finally, we set $\Omega_C(b) := \Omega_B(b)$ for $b \in B$ and $\Omega_C(a) := n + \Omega(a)$ for $a \in A$, where n is the least even number greater than any priority in $\mathbb{B}$.

Fixpoint operators We now turn to the difficult case of the definition of fixpoint operators on automata. The complexity mainly arises from two reasons. First, recall that in the one-step formulas appearing in the range of transition maps of modal automata, the proposition letters (corresponding to the free variables of a formula) are treated rather differently from the states of the automaton (which correspond to the bound variables of a formula). We have good reasons to do so, but when constructing the automaton $\eta x.\mathbb{A}$ from an automaton $\mathbb{A}$ there is a price to pay for this, related to the different status of the variable x in the two automata: while x is a free proposition letter in $\mathbb{A}$, and so appears only in unguarded positions in the one-step formulas, it is treated as a *state* of $\eta x.\mathbb{A}$ and must therefore appear only guarded in $\eta x.\mathbb{A}$. For this reason it will be necessary to *pre-process* the automaton $\mathbb{A}$ putting it in a shape $\mathbb{A}^x$ in which x is, in some sense, guarded.

Second, we have to be careful about how we go about this "pre-processing" of $\mathbb{A}$. This will become clearer once we consider the satisfiability game for modal automata in Section 5.4.

In order to turn to the construction of the auxiliary structure $\mathbb{A}^x$, we will need the following proposition.

5.3.23. PROPOSITION. *For every modal* $\mathtt{X}$*-automaton* $\mathbb{A}$ *positive in* $x \in \mathtt{X}$*, and any state* $a \in A$*, there are formulas* θ_0^a *and* θ_1^a *in which* x *does not appear, such that*

$$\Theta(a) \equiv_{\mathbf{K}} (x \wedge \theta_0^a) \vee \theta_1^a.$$

Proof:
First rewrite $\Theta(a)$ as a disjunction

$$(x \wedge \psi_0) \vee ... \vee (x \wedge \psi_n) \vee \psi'_0 \vee ... \vee \psi'_m$$

where each ψ_i and each ψ'_j is a conjunction consisting of literals distinct from x and formulas of the form $\nabla\beta$. This is then equivalent to

$$(x \wedge (\psi_0 \vee ... \vee \psi_n)) \vee (\psi'_0 \vee ... \vee \psi'_m)$$

and so we are done. □

5.3.24. EXAMPLE. Relying on the previous observation, we fix from now on for every automaton $\mathbb{A}$ and $a \in A$, one-step formulas θ_0^a, θ_1^a such that:

$$\Theta(a) \equiv_{\mathbf{K}} (x \wedge \theta_0^a) \vee \theta_1^a.$$

Now we are ready for the definitions of $\mathbb{A}^x$, $\mu x.\mathbb{A}$ and $\nu x.\mathbb{A}$.

5.3.25. DEFINITION. Let $\mathbb{A} = (A, \Theta_A, \Omega_A, a_I)$ be an $\mathtt{X} \uplus \{x\}$-automaton which is positive in x. By Proposition 5.3.23 for each $a \in A$ we may fix formulas $\theta_0^a, \theta_1^a \in \mathtt{1ML}_\nabla(\mathtt{X}, A)$ such that $\Theta(a) \equiv_{\mathbf{K}} (x \wedge \theta_0^a) \vee \theta_1^a$. We now define automata $\mathbb{A}^x$, $\mu x.\mathbb{A}$ and $\nu x.\mathbb{A}$. All three structures are based on the same carrier, viz., the set $(A \times \{0,1\}) \uplus \{\underline{x}\}$; we will denote states of the form (a, i) as a_i, if no confusion is likely. Of all these three automata, we specify their transition map Θ, priority map Ω and initial state i in Table 5.3. In this table, κ denotes the substitution

$$\kappa : a \mapsto (\underline{x} \wedge a_0) \vee a_1,$$

while n is the smallest even number that is greater than the maximum priority of $\mathbb{A}$.

Automaton	$\Theta(a_i)$	$\Theta(\underline{x})$	$\Omega(a_i)$	$\Omega(\underline{x})$	i
$\mathbb{A}^x$	$\theta_i^a[\kappa]$	x	$\Omega_A(a)$	0	$\underline{x}$
$\mu x.\mathbb{A}$	$\theta_i^a[\kappa]$	$\theta_1^{a_I}[\kappa]$	$\Omega_A(a)$	$n+1$	$\underline{x}$
$\nu x.\mathbb{A}$	$\theta_i^a[\kappa]$	$\theta_0^{a_I}[\kappa] \vee \theta_1^{a_I}[\kappa]$	$\Omega_A(a)$	$n+2$	$\underline{x}$

Table 5.3: The automata $\mathbb{A}^x$, $\mu x.\mathbb{A}$ and $\nu x.\mathbb{A}$

5.3.26. REMARK. The automaton $\mathbb{A}^x$ is *not equivalent* to $\mathbb{A}$, in the sense that it does not accept the same pointed pointed models as $\mathbb{A}$ does. On the other hand, it does contain all information that $\mathbb{A}$ does, and vice versa. The precise connection between $\mathbb{A}$ and $\mathbb{A}^x$ can best be expressed using the *translation map* that we will define in section 5.8. Running ahead of this, assume that we have defined, for each modal automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$, a map $\mathtt{tr}_{\mathbb{A}} : A \to \mu\mathtt{ML}_\nabla^{\overline{\mathtt{T}}}$ assigning to each state $a \in A$ an *equivalent* μ-calculus formula $\mathtt{tr}_{\mathbb{A}}(a)$ in the sense that $\mathbb{A}\langle a\rangle \equiv \mathtt{tr}_{\mathbb{A}}(a)$.

Phrased in terms of this translation map, the relation between $\mathbb{A}$ and $\mathbb{A}^x$ is given by the equivalences

$$\mathtt{tr}_{\mathbb{A}}(a) \equiv (x \wedge \mathtt{tr}_{\mathbb{A}^x}(a_0)) \vee \mathtt{tr}_{\mathbb{A}^x}(a_1)$$

and

$$\mathtt{tr}_{\mathbb{A}^x}(a_i) \equiv \theta_i^a[\mathtt{tr}_{\mathbb{A}}(b)/b \mid b \in A]$$

which hold for all $a \in A$ and $i \in \{0,1\}$.

The key to understand Definition 5.3.25, and to prove correctness of the constructions is the following proposition.

5.3.27. PROPOSITION. *Let φ_0, φ_1 be any formulas in which the variable x appears positively. Then:*

$$\mu x.(x \wedge \varphi_0) \vee \varphi_1 \equiv_{\mathbf{K}} \mu x.\varphi_1$$

and

$$\nu x.(x \wedge \varphi_0) \vee \varphi_1 \equiv_{\mathbf{K}} \nu x.\varphi_0 \vee \varphi_1$$

Proof:
We consider the case for μ first. One direction of the equivalence is immediate, since we have $\varphi_1 \leq_{\mathbf{K}} (x \wedge \varphi_0) \vee \varphi_1$. For the converse we have:

$$\begin{array}{rcll} ((x \wedge \varphi_0) \vee \varphi_1)[\mu x.\varphi_1/x] & = & ((\mu x.\varphi_1) \wedge \varphi_0[\mu x.\varphi_1/x]) \vee \varphi_1[\mu x.\varphi_1/x] & (1) \\ & \leq_{\mathbf{K}} & ((\mu x.\varphi_1) \wedge \varphi_0[\mu x.\varphi_1/x]) \vee \mu x.\varphi_1 & (2) \\ & \leq_{\mathbf{K}} & \mu x.\varphi_1 & (3) \end{array}$$

In more details, (1) is a syntactic equality by definition of substitution, (2) holds by $(A_f) : \varphi(\mu x.\varphi(x)) \to \mu x.\varphi(x)$ from $\mathbf{K}$ (see Definition 5.2.5) implying $\varphi_1[\mu x.\varphi_1/x] \to \mu x.\varphi_1$ and $(A) : (p \to q) \to ((r \vee p) \to (r \vee q))$ from propositional logic, and finally (3) is immediate by lattice law $(p \wedge q) \vee p = p$.

For the ν-case, again one direction is immediate since we have $(x \wedge \varphi_0) \vee \varphi_1 \leq_K \varphi_0 \vee \varphi_1$. For the other direction we reason as follows:

$$\begin{array}{rcll} \nu x.\varphi_0 \vee \varphi_1 & \equiv_{\mathbf{K}} & (\nu x.\varphi_0 \vee \varphi_1) \wedge (\nu x.\varphi_0 \vee \varphi_1) & \\ & \leq_{\mathbf{K}} & (\nu x.\varphi_0 \vee \varphi_1) \wedge (\varphi_0[\nu x.\varphi_0 \vee \varphi_1/x] \vee \varphi_1[\nu x.\varphi_0 \vee \varphi_1/x]) & (A_f, A) \\ & \leq_{\mathbf{K}} & \big((\nu x.\varphi_0 \vee \varphi_1) \wedge \varphi_0[\nu x.\varphi_0 \vee \varphi_1/x]\big) \vee \varphi_1[\nu x.\varphi_0 \vee \varphi_1/x] & \\ & = & ((x \wedge \varphi_0) \vee \varphi_1)[\nu x.\varphi_0 \vee \varphi_1/x] & \end{array}$$

and the proof is finished. □

5.3.28. Remark. We finish this subsection with noting that all the constructions defined above are semantically correct, in the sense that $L(\mathbb{A} \wedge \mathbb{B}) = L(\mathbb{A}) \cap L(\mathbb{B})$, etc. Since these statements follow from the results we shall prove in section 5.8 (in particular, from Proposition 5.8.16) we leave semantic proofs as exercises for the reader.

5.3.4 Translating formulas to automata

We finish this section by providing a translation associating a modal automaton to every $\mu\mathtt{ML}_{\nabla}^{\top}$-formula. As we mentioned earlier, we will define the translation by induction on the complexity of formulas and make use of operations we just defined.

5.3.29. Definition. By induction on the complexity of a formula $\varphi \in \mu\mathtt{ML}_{\nabla}^{\top}$ we define a modal automaton $\mathbb{A}_\varphi$.

First of all, we need to consider atomic formulas: given any propositional variable p, we take some arbitrary object a distinct from p to be the one and only state of $\mathbb{A}_p$, and define $\Theta_p(a) = p$, and $\Omega_p(a) = 0$.

With this in place, we can complete the translation as follows:

$$
\begin{array}{lcl}
\mathbb{A}_{\neg\varphi} & := & \neg\mathbb{A}_{\varphi} \\
\mathbb{A}_{\varphi\vee\psi} & := & \mathbb{A}_{\varphi} \vee \mathbb{A}_{\psi} \\
\mathbb{A}_{\nabla\alpha} & := & \nabla\mathbb{\alpha} \text{ where } \mathrm{Base}(\alpha) = \{\varphi_1, \varphi_2, ..., \varphi_n\} \text{ and } \mathbb{\alpha} \in \mathsf{T}\{\mathbb{A}_{\varphi_1}, \cdots, \mathbb{A}_{\varphi_n}\} \\
 & & \text{ is defined by } \mathbb{\alpha} = \mathsf{T}f(\alpha) \text{ with } f : \{\varphi_1, \varphi_2, ..., \varphi_n\} \to \{\mathbb{A}_1, \mathbb{A}_2, ..., \mathbb{A}_n\} \\
 & & \text{ in turn defined by } \varphi_i \mapsto \mathbb{A}_{\varphi_i} \\
\mathbb{A}_{\mu x.\varphi} & := & \mu x.\mathbb{A}_{\varphi}
\end{array}
$$

We finish by stating the semantic correctness of this definition. Since this proposition is not needed in the sequel, we leave the details of its proof, which proceeds by a routine induction on the complexity of formulas, as an exercise to the reader.

5.3.30. Proposition. *For every formula $\varphi \in \mu\mathtt{ML}^{\overline{\mathsf{T}}}_{\nabla}$ we have:*

$$\varphi \equiv \mathbb{A}_{\varphi}. \tag{5.3}$$

5.4 Games for Automata

In this section, we introduce two of our main tools: the satisfiability game $\mathcal{S}(\mathbb{A})$ associated with a modal automaton $\mathbb{A}$, and the consequence game $\mathcal{C}(\mathbb{A}, \mathbb{B})$ related to two modal automata $\mathbb{A}$ and $\mathbb{B}$.

Before we turn to the technicalities of the definitions, we start with an intuitive explanation of the satisfiability game, which is based on the infinite tableau game introduced by Niwiński & Walukiewicz [NW96]. Our variant, introduced in [FLV10] in the more general setting of the coalgebraic μ-calculus, can be seen as a streamlined, game-theoretic analog for automata to what tableaux are for formulas. To understand the game $\mathcal{S}(\mathbb{A})$, which is played by two players, $\forall$ and $\exists$, it helps to think of $\exists$ as defending the claim that the language $L(\mathbb{A})$ is nonempty. In fact, we may think of $\exists$'s winning strategies as blueprints for constructing (tree-based) structures that are to be accepted by $\mathbb{A}$. The role of $\forall$ in $\mathcal{S}(\mathbb{A})$ is rather different: he acts as a *path finder* in the (partial) structure constructed by $\exists$, his task being to challenge $\exists$ to come up with more evidence to her claims and to construct ever more detail of the structure. What distinguishes the satisfiability game from tableaux is that, because of the uniform internal structure of modal automata as compared to formulas, the interaction between the players can be shaped in a highly regulated pattern. The satisfiability game does not have separate rules dealing with specific connectives; in particular, all rules/moves dealing with Boolean connectives have been encapsulated in the streamlined interaction between $\exists$ and $\forall$.

For two reasons, it is also useful to relate the satisfiability game $\mathcal{S}(\mathbb{A})$ to the acceptance games associated with $\mathbb{A}$. First, similar to the acceptance games for

$\mathbb{A}$, the satisfiability game proceeds in *rounds*: one round of $\mathcal{S}(\mathbb{A})$ consists of first $\exists$ constructing (or aiming to construct) one more level of the tree structure for $\mathbb{A}$, and then $\forall$ picking one of the newly created nodes for further inspection. Second, and more in particular, every play of $\mathcal{S}(\mathbb{A})$ can be seen as a *bundle* of plays of the acceptance game played on exactly the structure that $\exists$ is constructing.

5.4.1 Traces

We first need some notation and terminology concerning streams of binary relations and the traces they carry. The key concept here is that of a *trace* running through an infinite sequence of binary relations. This concept appears in many papers dealing with decidability questions on fixpoint logics, going back to at least Streett & Emerson [SE89] (where it appears under the name 'derivation sequence'); we took our terminology from Niwiński & Walukiewicz [NW96].

5.4.1. Definition. Fix a set A. We let $A^\sharp$ denote the set of binary relations over A, that is, $A^\sharp := \mathcal{P}(A \times A)$. Given a finite word $\rho = R_0 R_1 R_2 \ldots R_k$ over the set $A^\sharp$, a *trace* through ρ over $A^\sharp$ is a finite A-word $a_0 a_1 a_2 ... a_{k+1}$ such that $a_i R_i a_{i+1}$ for all $i \leq k$. A trace through an $A^\sharp$-stream $R_0 R_1 R_2 R_3 ...$ is an A-stream $a_0 a_1 a_2 ...$ such that $a_i R_i a_{i+1}$ for all $i < \omega$.

Given a stream $\rho = R_0 R_1 R_2 ...$ over $A^\sharp$ we denote by $\rho{\restriction_k}$ the word $R_0 \cdots R_k$ and for a trace $\tau = a_0 a_1 a_2 ...$ on ρ we denote by $\tau{\restriction_k}$ the restricted trace $a_0 a_1 a_2 ... a_k$ on $\rho{\restriction_k}$. We use similar notation for restrictions of words over $A^\sharp$ of length $\geq k$.

It is often convenient to think of the set of finite traces providing a graph structure. Formally we define the trace graph of an $A^\sharp$-stream as follows.

5.4.2. Definition. Given an $A^\sharp$-stream $\rho = (R_n)_{n \geq 0}$, we define the *trace graph* $\mathbb{G}_\rho$ as the directed graph with vertices $\omega \times A$ and edges $E_{\mathbb{G}} := \{((i,a),(j,b)) \mid j = i+1 \text{ and } a R_i b\}$.

Observe that the infinite ρ-traces are in 1-1 correspondences with the maximal infinite paths through the trace graph.

5.4.3. Definition. Fix a finite set A and a priority map $\Omega : A \to \omega$. We let NBT_Ω denote the set of $A^\sharp$-streams that contain *no bad trace*, that is, no trace $\tau = a_0 a_1 \ldots$ such that $\max(\Omega[\mathit{Inf}(\tau)])$, the highest priority occurring infinitely often on τ, is odd.

It is not difficult to show that NBT_Ω is an ω-regular subset of $(A^\sharp)^\omega$.

5.4.4. Proposition. *Given a finite set A and a priority map $\Omega : A \to \omega$, there is a parity stream automaton recognizing the set NBT_Ω, seen as a stream language over $A^\sharp$.*

Proof:
It is easy to construct a nondeterministic parity stream automaton $\mathbb{A}$ recognizing the complement of NBT_Ω, that is, the set of $A^\sharp$-streams that do contain a bad trace. The Proposition is then immediate by the fact that the collection of ω-regular language is closed under taking complementation. □

5.4.2 The satisfiability game

In this section, we follow [Fon10] and [ESV18] to define the satisfiability game and give an indication of how it works.

As we already mentioned, every basic position of the satisfiability game for $\mathbb{A}$, i.e. $\mathcal{S}(\mathbb{A})$, can be represented as a subset B of A, which is called a macro-state of $\mathbb{A}$. Given such a macro-state B, $\exists$ claims that there is a pointed model $(\mathbb{S}, s)$ such that for all $b \in B$ the automaton $\mathbb{A} = (A, \Theta, \Omega, b)$ accepts $(\mathbb{S}, s)$. This means $\exists$ has to come up with a winning strategy from positions (b, s) in the acceptance game, for all $b \in B$. Clearly, if such a pointed structure $(\mathbb{S}, s)$ exists, then each position (b, s) is winning for her in the acceptance game for $(\mathbb{S}, s)$.

For each $t \in S$ and for each $b \in B$, we define the set A_t^b as the collection of states $b' \in A$ such that (b', t) is a possible basic position in the acceptance game following the basic position (b, s). Since B is a macro-state, we define $A_t := \bigcup\{A_t^b \mid b \in B\}$. Hence, each such a set is a potential next combination of states in A that $\exists$ has to be able to handle simultaneously. In this set-up $\exists$'s move would be based on the set $\{A_t \mid t \in S\}$. Now it is up to $\forall$ to choose a set from this collection, moving to the next macro-state.

With this definition of the game, a match of $\mathcal{S}(\mathbb{A})$ corresponds to a sequence $\rho := B_0B_1B_2...$ of basic positions, which are subsets of A. Now to clarify whether ρ is won by $\exists$ we could naively say that $\exists$ wins if there is no bad trace $b_0b_1b_2...$ in ρ. However, if there is such a bad trace $b_0b_1b_2...$, this would only be a problem if it actually corresponds to a match of the acceptance game. Up to now we know that each b_i occurs in *some* match of the acceptance game, but there is no way to know whether $b_0b_1b_2...$ is the projection of an actual match of the acceptance game. This shows that defining the game based on subsets of A doesn't work properly. A solution to this problem is to replace the subset B by a relation $R \in A^\sharp$. The range of R would play the same role as B. This helps us to remember which traces are relevant, when we define the winning condition.

We are now ready for the formal definition of the satisfiability game, where we use the notions of natural valuations introduced in Definition 5.3.12, and Base from Definition 2.3.21.

5.4.5. DEFINITION. The satisfiability game associated with a modal $\mathtt{X}$-automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ is denoted by $\mathcal{S}(\mathbb{A})$ and is given by Table 5.4:

Position	Player	Admissible moves
$R \in A^\sharp$	$\exists$	$\bigcap_{a \in \mathsf{ran}(R)} [\![\Theta(a)]\!]^1_a$
$(\mathtt{Y}, \alpha) \in \mathsf{T}_\mathtt{X}(A^\sharp)$	$\forall$	$\{R \mid R \subseteq R' \text{ for some } R' \in \text{Base}(\alpha)\}$

Table 5.4: Admissible moves in the satisfiability game $\mathcal{S}(\mathbb{A})$

The winner of an infinite play of the satisfiability game is given by the induced stream $\rho = R_0 R_1 \ldots \in (A^\sharp)^\omega$ of basic positions. This winner is $\exists$ if ρ belongs to the set NBT_Ω (also denoted by $NBT_\mathbb{A}$), that is, if ρ contains no bad traces, and it is $\forall$ otherwise. A winning strategy of $\forall$ in $\mathcal{S}(\mathbb{A})$ may be called a *refutation* of $\mathbb{A}$.

5.4.6. Example. Let $\mathbb{A} = (A, \Theta, \Omega, a_I)$ be a modal automaton. Since we will only consider matches of the satisfiability game $\mathcal{S}(\mathbb{A})$ that take the singleton $\{(a_I, a_I)\}$ as their starting position, we will often be sloppy and blur the difference between $\mathcal{S}(\mathbb{A})$ and the initialized game $\mathcal{S}(\mathbb{A})@\{(a_I, a_I)\}$.

5.4.7. Remark. An alternative and perhaps more natural version of $\mathcal{S}(\mathbb{A})$ would restrict the moves available to $\forall$ at position $(\mathtt{Y}, \alpha) \in \mathsf{T}_\mathtt{X} A^\sharp$ to the actual *elements* of Base(α), instead of allowing subsets of elements of Base(α). It is not so difficult to prove, however, that this version of the game is in fact equivalent to $\mathcal{S}(\mathbb{A})$ itself. Roughly, the reason for this is that in $\mathcal{S}(\mathbb{A})$ it never will be to $\forall$'s advantage at a position $(\mathtt{Y}, \alpha) \in \mathsf{T}_\mathtt{X} A^\sharp$ to pick a *strict* subset R of some relation $R' \in \text{Base}(\alpha)$: the bigger the relations that he picks, the more opportunities he has to obtain a bad trace.

Our motivation for taking $\mathcal{S}(\mathbb{A})$ as the standard version of our satisfiability game is simply that in some cases $\mathcal{S}(\mathbb{A})$ is technically more convenient to work with than its apparently simpler variant.

The following proposition from [FLV10] expresses the adequacy of the satisfiability game.

5.4.8. Proposition (Adequacy). *Let $\mathbb{A}$ be a modal automaton. Then $\exists$ has a winning strategy in $\mathcal{S}(\mathbb{A})$ iff the language recognized by $\mathbb{A}$ is non-empty.*

5.4.9. Remark. In general, $\mathcal{S}(\mathbb{A})$ is not a parity game, but we saw in Proposition 5.4.4 that the winning condition NBT_Ω is an ω-regular subset of $(A^\sharp)^\omega$. It follows from a result by Büchi & Landweber [BL69] that we may assume that winning strategies in $\mathcal{S}(\mathbb{A})$ only use finite memory. This observation is used in [FLV10] to prove the finite model property of coalgebraic modal automata, and hence, of the coalgebraic modal μ-calculus (predicate lifting approach).

5.4.3 Consequence game

As announced in our introduction, an important role in our approach is played by the *consequence game* $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ associated with two automata $\mathbb{A}$ and $\mathbb{A}'$, which is played by two players I and II. One may think of player II trying to show that automaton $\mathbb{A}$ *implies* $\mathbb{A}'$ by establishing a close structural connection between the two automata, and of player I trying to show this does not hold.

Matches of the consequence game $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ are tightly linked to the matches of the satisfiability games $\mathcal{S}(\mathbb{A})$ and $\mathcal{S}(\mathbb{A}')$, and this connection extends to the definition of the winning conditions of $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ in terms of winning conditions of $\mathcal{S}(\mathbb{A})$ and $\mathcal{S}(\mathbb{A}')$.

To describe the game, we consider a match of $\mathcal{C}(\mathbb{A}, \mathbb{A}')$. Each round of this match consists of three moves. At the start of the round, at a basic position $(R, R') \in A^\sharp \times A'^\sharp$, player I picks a local model $\alpha \in \mathsf{T}_\mathtt{X} A^\sharp$ for formulas given by the range of R, as if she was player $\exists$ in the satisfiability game $\mathcal{S}(\mathbb{A})$. Second, player II transforms this one-step model into a model for formulas given by the range of R', inducing a move for $\exists$ in the satisfiability game $\mathcal{S}(\mathbb{A}')$. More precisely, player II provides a map $f : A^\sharp \to A'^\sharp$ turning α into a model for R'. The admissibility of this move reveals the essentially coalgebraic nature of the game, using the fact that T is actually a functor, i.e., it operates on arrows (that are, functions) as well as on objects (sets). More specifically, player II's move f is admissible if the model α', that we obtain by applying the map $\mathsf{T}_\mathtt{X} f$ to the model α, is a model for R'. Player I then finishes the round by picking an element from the graph of the map f as the next basic position.

5.4.10. Definition. The *consequence game* $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ between modal automata $\mathbb{A} = (A, \Theta, \Omega, a_I)$ and $\mathbb{A}' = (A', \Theta', \Omega', a'_I)$ is given by the following table.

Position	Player	Admissible moves
$(R, R') \in A^\sharp \times A'^\sharp$	I	$\bigcap_{a \in \mathsf{ran}(R)} [\![\Theta(a)]\!]^1_a$
$(\alpha, R') \in \mathsf{T}_\mathtt{X} A^\sharp \times A'^\sharp$	II	$\{f : A^\sharp \to A'^\sharp \mid \mathsf{T}_\mathtt{X} f(\alpha) \in \bigcap_{b \in \mathsf{ran}(R')} [\![\Theta'(b)]\!]^1_b\}$
$f : A^\sharp \to A'^\sharp$	I	$\{(R, R') \mid f(R) = R'\}$

Table 5.5: Consequence Game

As we already mentioned, a pair of the form (R, R') in the above definition will be called a *basic position* of the consequence game. Similar to the satisfiability game, our standard assumption is that $(\{(a_I, a_I)\}, \{(a'_I, a'_I)\})$ is the starting position of $\mathcal{C}(\mathbb{A}, \mathbb{A}')$. We declare player I to be the winner of an infinite match $(R_0, R'_0)(R_1, R'_1)(R_2, R'_2)...$ if there exists a bad trace on the $\mathbb{A}'$-side, i.e. through $R'_0 R'_1 R'_2...$ but no bad trace on the $\mathbb{A}$-side i.e. through $R_0 R_1 R_1....$ In all other cases player II is the winner. Whenever II has a winning strategy in $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ we say that $\mathbb{A}'$ is a *game consequence* of $\mathbb{A}$ and denote this fact with $\mathbb{A} \models_{\mathcal{C}} \mathbb{A}'$.

5.4.11. Remark. The consequence game can be seen as a kind of communication or implication game between the satisfiability games of the two automata involved. As such, the construction of $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ from $\mathcal{S}(\mathbb{A})$ and $\mathcal{S}(\mathbb{A}')$ is vaguely reminiscent of the operation $\langle -, - \rangle$ on games, defined by Santocanale [San02], where Santocanale's construction in its turn is the result of enriching fixpoint theory with ideas from the game semantics of linear logic (see, e.g., Blass [Bla92] or Joyal [Joy95]). Note however that the actual moves of our game crucially involve modal one-step logic, in a way that makes $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ rather different from the game $\langle \mathcal{S}(\mathbb{A}), \mathcal{S}(\mathbb{A}') \rangle$ one would obtain by applying Santocanale's construction.

The following result can be seen as a soundness result for the consequence game.

5.4.12. Proposition. *Given modal* $\mathtt{X}$*-automata* $\mathbb{A}$ *and* $\mathbb{A}'$ *we have that* $\mathbb{A} \vDash_{\mathcal{C}} \mathbb{A}'$ *implies* $L(\mathbb{A}) \subseteq L(\mathbb{A}')$.

Proof:
Fix a pointed $\mathsf{T}_{\mathtt{X}}$-coalgebra $\mathbb{S} = (S, \sigma, s_I)$, a winning strategy χ for $\exists$ in the initialized acceptance game $\mathcal{A}(\mathbb{A}, \mathbb{S})@(a_I, s_I)$ and a winning strategy f for Player II in $\mathcal{C}(\mathbb{A}, \mathbb{A}')$. For simplicity we assume without loss of generality that the strategy χ is positional (recalling that $\mathcal{A}(\mathbb{A}, \mathbb{S})$ is a parity game). Our goal is to provide a winning strategy χ' for $\exists$ in the game $\mathcal{A}(\mathbb{A}', \mathbb{S})@(a'_I, s_I)$. By induction on the length of a χ'-guided match with basic positions $(a'_I, s_I), (a'_1, s_1)...(a'_{n+1}, s_{n+1})$, we shall define an f-guided shadow match $(R_0, R'_0)(R_1, R'_1)...(R_n, R'_n)$ such that the following conditions hold:

(1) $a_I a_1 ... a_{n+1}$ is a trace through $R_0 R_1 ... R_n$ iff $(a_I, s_I)(a_1, s_1)...(a_{n+1}, s_{n+1})$ is a χ-guided match; furthermore, each $b \in \mathsf{ran}(R_n)$ is the last element of some trace through $R_0 R_1 ... R_n$.

(2) $a'_I a'_1 ... a'_{n+1}$ is a trace through $R'_0 R'_1 ... R'_n$.

Furthermore, we shall associate these shadow matches in a uniform manner, so that the shadow match of an initial segment of a partial match Σ is an initial segment of the shadow match associated with Σ. First, note that this means that $\exists$ wins all χ'-guided infinite matches: if $(a'_I, s_I)(a'_1, s_1)(a'_2, s_2)...$ is a loss for $\exists$ then $a'_I a'_1 a'_2 ...$ is a bad trace through $R'_0 R'_1 R'_2 ...$ in the shadow match $(R_0, R'_0)(R_1, R'_1)(R_2, R'_2)...$ by condition (1). Since this shadow match was f-guided and f is a winning strategy, this means there must be some bad trace $a_I a_1 a_2 ...$ through $R_0 R_1 R_2 ...$, and by condition (2) we get that $(a_I, s_I)(a_1, s_1)(a_2, s_2)...$ is an infinite χ-guided match, which furthermore is a loss for $\exists$. This is a contradiction since χ was a winning strategy by assumption.

We now show how $\exists$ can respond to any move by $\forall$ while maintaining the induction hypothesis. Suppose we are given a χ'-guided partial match Σ consisting of positions $(a'_I, s_I), (a'_1, s_1)...(a'_{n+1}, s_{n+1})$ with a shadow match:

$$(R_0, R'_0)(R_1, R'_1)...(R_n, R'_n)$$

satisfying the conditions (1) and (2). For each $a \in \mathsf{ran} R_n$, by (1) there is a χ-guided partial match with last position (a, s_n). So we see that the move $\chi(a, s_n) : A \to \mathcal{P}S$ prescribed by the winning strategy χ is legitimate for each such a. Define the map $h : S \to A^\sharp$ by

$$h(v) = \{(b, b') \mid b \in \mathsf{ran} R_n \;\&\; v \in \chi(b', s_n)\}$$

Then it follows by naturality (see Proposition 5.3.6) that $\mathsf{T}_\mathtt{X} h(\sigma(s_n))$ is a legitimate move for Player I in the consequence game at the position (R_n, R'_n), i.e. $\mathsf{T}_\mathtt{X} h(\sigma(s_n)) \in \bigcap_{a \in \mathsf{ran} R_n} [\![\Theta(a)]\!]_a$. So the winning strategy f provides a map $g : A^\sharp \to A'^\sharp$ such that $\mathsf{T}_\mathtt{X}(g \circ h)(\sigma(s_n)) \in \bigcap_{b' \in \mathsf{ran} R'_n} [\![\Theta'(b')]\!]_{b'}$. In particular, we get $\mathsf{T}_\mathtt{X}(g \circ h)(\sigma(s_n)), V_{a'_n} \Vdash^1 \Theta'(a'_n)$, where $V_{a'_n}$ is the natural valuation at a'_n. It follows by naturality that $\sigma(s_n), V_{m \circ g \circ h} \Vdash^1 \Theta'(a'_n)$, where we let m denote the marking associated with the natural valuation $V_{a'_n}$ and $V_{m \circ g \circ h}$ denote the valuation corresponding to the marking $m \circ g \circ h$. So we set $\chi'(\Sigma) = V_{m \circ g \circ h}$, and this is a legitimate move in the game $\mathcal{A}(\mathbb{A}', \mathbb{S})@(a'_I, s_I)$. Furthermore, if $v \in V_{m \circ g \circ h}(b')$, then $b' \in m(g(h(v)))$, so $(a'_n, b') \in g(h(v))$, and we can continue the shadow match with the pair $(h(v), g(h(v)))$. Then the extended shadow match

$$(R_0, R'_0)(R_1, R'_1)...(R_n, R'_n)(h(v), g(h(v)))$$

satisfies both conditions (1) and (2), so the proof is finished. □

It should be stressed that the converse direction of Proposition 5.4.12 does *not* hold in general. If $\mathbb{A}'$ is a game consequence of $\mathbb{A}$, the existence of a winning strategy for player II in the consequence game indicates a close *structural* relation between $\mathbb{A}$ and $\mathbb{A}'$, far tighter than what is required for $\mathbb{A}$ being a semantic consequence of $\mathbb{A}'$. In the following we will present a counter-example, but we first give an example of two automata that do satisfy the game consequence relation. Note that this example is closely linked to the fixpoint rule of Kozen's axiom system.

5.4.13. Proposition. *For all modal automata $\mathbb{A}$ that are positive in x, we have $\mathbb{A}^x[\mu x.\mathbb{A}/x] \vDash_{\mathcal{C}} \mu x.\mathbb{A}$.*

Proof:
Recall from Definition 5.3.25 that the automaton $\mu x.\mathbb{A}$ has the same carrier as the automaton $\mathbb{A}^x$, and that the automaton $\mathbb{A}^x[\mu x.\mathbb{A}]$ is built from $\mu x.\mathbb{A}$ together with one disjoint copy of $\mathbb{A}^x$, so $\mathbb{A}^x[\mu x.\mathbb{A}]$ will contain for each state a in $\mu x.\mathbb{A}$ an extra state a' corresponding to a belonging to the disjoint copy of $\mathbb{A}^x$. With this in mind, we define a map f from states of $\mathbb{A}^x[\mu x.\mathbb{A}]$ to states of $\mu x.\mathbb{A}$ by mapping a' to a, and a to itself, for each state a in $\mu x.\mathbb{A}$. This map induces a map F from relations over the states of $\mathbb{A}^x[\mu x.\mathbb{A}]$ to relations over the states of $\mu x.\mathbb{A}$ by the assignment:

$$F : R \mapsto \{(f(a), f(b)) \mid (a, b) \in R\}$$

Thus we get a strategy for Player II in the game $\mathcal{C}(\mathbb{A}^x[\mu x.\mathbb{A}/x], \mu x.\mathbb{A})$ defined by choosing the map $F\restriction_{\text{Base}(\alpha)}$ as a response to every move $(\mathtt{Y}, \alpha)$ made by Player I. It can be checked that this is a winning strategy for Player II. $\square$

It is not too hard to convince oneself that the automata $\mathbb{A}[\mu x.\mathbb{A}]$ and $\mathbb{A}^x[\mu x.\mathbb{A}]$ are *semantically* equivalent, but the consequence game is a stronger concept and while Proposition 5.4.13 shows that $\mathbb{A}^x[\mu x.\mathbb{A}/x] \vDash_{\mathcal{C}} \mu x.\mathbb{A}$, in general it is somewhat surprisingly not true that $\mathbb{A}[\mu x.\mathbb{A}/x] \vDash_{\mathcal{C}} \mu x.\mathbb{A}$. An explicit example is given in [ESV18, Example 5.20]. Intuitively, what is driving this example is that the construction $\mathbb{A}^x$ splits states of $\mathbb{A}$ into disjunctions, which gives Player I some extra power in the game $\mathcal{C}(\mathbb{A}[\mu x.\mathbb{A}], \mu x.\mathbb{A})$ since she can choose which disjunct of a one-step formula to make true on the left-hand side of the game, while the choice may be already made for Player II on the right-hand side. This illustrates our earlier point that $\mathbb{A} \models_{\mathcal{C}} \mathbb{A}'$ indicates a rather tight structural relation between the two automata.

To finish this subsection we mention the following fact showing that the game consequence relation between modal automata is reflexive and transitive.

5.4.14. Proposition. *Let $\mathbb{A}$, $\mathbb{A}'$ and $\mathbb{A}''$ be modal automata.*

(1) $\mathbb{A} \vDash_{\mathcal{C}} \mathbb{A}$;

(2) if $\mathbb{A} \vDash_{\mathcal{C}} \mathbb{A}'$ *and* $\mathbb{A}' \vDash_{\mathcal{C}} \mathbb{A}''$ *then* $\mathbb{A} \vDash_{\mathcal{C}} \mathbb{A}''$.

Proof:
The proof of the first item is trivial. Item (2) follows from a routine check that the composition of any two winning strategies for player II in games $\mathcal{C}(\mathbb{A}, \mathbb{A}')$ and $\mathcal{C}(\mathbb{A}', \mathbb{A}'')$ is a winning strategy for II in game $\mathcal{C}(\mathbb{A}, \mathbb{A}'')$. $\square$

5.4.15. Remark. Applying this Proposition one can define the category of modal $\mathtt{X}$-automata by setting $\mathtt{X}$-automata as objects and defining arrows as follows:

$$\mathbb{A} \to \mathbb{B} \text{ if and only if } \mathbb{A} \vDash_{\mathcal{C}} \mathbb{B}.$$

It is immediate from Proposition 5.4.14 that this collection forms a category.

5.5 Taming the traces - one step at a time

As we have seen in the satisfiability and consequence games, the winner of the matches of these games is determined by the shape of traces through streams of binary relation. So a good understanding of trace graphs of such streams will be crucial for us. But sometimes the combinatorics of these trace graphs gets rather involved. For this reason, we will be interested in classes of modal automata for which the trace graphs of matches of the satisfiability and consequence games have a simpler structure. We start by isolating certain classes of binary relations over A (for a fixed modal automaton $\mathbb{A}$), such that the trace graphs of streams of such relations have a simple structure. We then introduce corresponding one-step languages that produce modal automata for which infinite matches of satisfiability and consequence games can indeed be assumed, without loss of generality, to have relations belonging to these restricted classes of relations in $A^\sharp$.

5.5.1 Functional, clusterwise functional and thin relations

The simplest class of relations that we shall consider are the *functional* ones:

5.5.1. Definition. Let $\mathbb{A} = (A, \Theta, \Omega, a_I)$ be a modal automaton, and let $R \in A^\sharp$ be a relation. We call R *functional*, if each $a \in A$ has at most one R-successor. This element, if it exists, is denoted as a^+_R, or as a^+ in case R is clear from context. The set of functional relations in $A^\sharp$ will be denoted by ${A^\sharp}_f$.

The trace combinatorics of streams of functional relations is *trivial*, as described by the following proposition. Although the result is obvious, we state it explicitly for emphasis:

5.5.2. Proposition. *Fix a modal automaton $\mathbb{A}$ and let $R_0R_1R_2\ldots$ be any stream over ${A^\sharp}_f$. Then for each $a \in \mathsf{dom}(R_0)$ there is at most one infinite trace on this stream beginning with a, and if each R_i is total then the correspondence is one-to-one.*

In particular, there are at most $|A|$ many infinite traces on any stream on ${A^\sharp}_f$. Since there is in total only a bounded number of traces to consider, it is easy to check whether there is any bad trace on such a stream.

Recalling the notion of a cluster for an automaton $\mathbb{A}$ from Definition 5.3.21, we can define a wider class of relations that maintains some of this simplicity.

5.5.3. Definition. Given a fixed modal automaton $\mathbb{A}$, a relation $R \in A^\sharp$ is said to be *clusterwise functional* if:

(1) for all $a, b \in A$ with aRb, we have $b \lhd_{\mathbb{A}} a$;

(2) for all $a \in A$, there is at most one $b \in C_a$ such that aRb.

The set of clusterwise functional relations in $A^\sharp$ will be denoted by ${A^\sharp}_{cf}$.

Generally, a stream over the alphabet ${A^\sharp}_{cf}$ will have infinitely many traces. However, the trace combinatorics of these streams is still much simpler than in the general case. The next Proposition clarifies this. We recall that two streams σ, τ over any alphabet are said to be eventually equal if there is a $k \in \omega$ such that $\sigma(j) = \tau(j)$ for all $j \geq k$.

5.5.4. PROPOSITION. *Given a modal automaton $\mathbb{A}$, let $\rho = R_0R_1R_2\dots$ be any stream over ${A^\sharp}_{cf}$. Then there exists a collection F of at most $|A|$ many infinite traces over $\rho = R_0R_1R_2\dots$, such that every infinite trace on this stream is eventually equal to some trace in F.*

Proof:
By the first condition on clusterwise functionality, every infinite trace τ through ρ eventually ends up in a cluster C, in the sense that $\exists n.\forall k \geq n.\tau(k) \in C$. It thus suffices to prove that the relation of eventual equality, taken over the set of traces that eventually end up in an arbitrary but fixed cluster C, is an equivalence relation of size at most $|C|$.

Suppose for contradiction that this is not the case, i.e., there are traces $\{\tau_i \mid 0 \leq i \leq |C|\}$, all ending up in C, and such that τ_i and τ_j are eventually equal only if $i = j$. Then we can find an $n \in \omega$ such that for all $k \geq n$ *each* $\tau_i(k)$ belongs to C. By the pigeon hole principle then there must be distinct indices i and j such that $\tau_i(n) = \tau_j(n)$. But by clusterwise functionality this implies that $\tau_i(k) = \tau_j(k)$ for all $k \geq n$, so that τ_i and τ_j are eventually equal after all. □

In particular this means that we only have to examine the $|A|$ many traces in F to decide whether there is a bad trace on $R_0R_1R_2\dots$, since two eventually equal traces are clearly either both bad or both not bad.

Cluster-wise functional relations are *almost* the key concept that we need, but it turns out that we are going to require a little bit of extra generality. While the number of infinite traces of a stream over ${A^\sharp}_{cf}$ is essentially finite in the sense of Proposition 5.5.4, we shall finally consider a wider class of relations for which the corresponding streams have the property that there are essentially only finitely many *bad traces*.

5.5.5. DEFINITION. Let $\mathbb{A}$ be a modal automaton and let C be a cluster of $\mathbb{A}$. An element $a \in C$ is called a *maximal even element* of C if it has the maximal priority in C, and this priority is even. A relation $R \in A^\sharp$ is *thin with respect to* $\mathbb{A}$ and a if:

(1) for all $b \in A$ with aRb we have $b \lhd_{\mathbb{A}} a$;

(2) for all $b_1, b_2 \in A$ with $b_1, b_2 \in R[a] \cap C_a$, either $b_1 = b_2$ or one of b_1 and b_2 is a maximal even element of C_a.

We call R $\mathbb{A}$-thin or simply thin, if it is thin with respect $\mathbb{A}$ and all $a \in A$. We denote the collection of thin relations in $A^\sharp$ by ${A^\sharp}_{thin}$.

A motivating observation about thin relations is the following.

5.5.6. PROPOSITION. *Given a modal automaton $\mathbb{A}$, let $\rho = R_0R_1R_2\dots$ be any stream of thin relations over ${A^\sharp}_{thin}$. Then there exists a collection F of at most*

$|A|$ many infinite traces on ρ, such that any given bad *trace on ρ is eventually equal to some trace F.*

Proof:
We prove this proposition by a similar argument as used for Proposition 5.5.4, the differences being that we restrict attention to bad traces, and, in the reductio argument, let $n \in \omega$ satisfy the additional requirement that for all $k \geq n$, and all i, $\tau_i(k)$ is *not* the maximal even element of C. □

Again, this combinatorial property of thin relations significantly simplifies the problem of checking whether an infinite stream has a bad trace, since we only have to check whether the bounded collection F contains a bad trace or not. To exploit this nice property of these relations, we will introduce the second version of the satisfiability game:

5.5.7. Definition. Given a modal automaton $\mathbb{A}$, the *thin satisfiability game* for $\mathbb{A}$, denoted by $\mathcal{S}_{thin}(\mathbb{A})$ is defined as the satisfiability game $\mathcal{S}(\mathbb{A})$ except that the moves of $\forall$ are constrained so that $\forall$ may only choose $\mathbb{A}$-thin relations. That is, R is a legitimate move by $\forall$ at some position in $\mathcal{S}_{thin}(\mathbb{A})$ iff R is a legitimate move at the same position in $\mathcal{S}(\mathbb{A})$, and R is an $\mathbb{A}$-thin relation.

In general, the game $\mathcal{S}_{thin}(\mathbb{A})$ is not equivalent to $\mathcal{S}(\mathbb{A})$ in the sense that there is always a winning strategy for the same player in both games: since the moves of $\forall$ are restricted in $\mathcal{S}_{thin}(\mathbb{A})$, it may be that $\exists$ has a winning strategy in $\mathcal{S}_{thin}(\mathbb{A})$ but not in $\mathcal{S}(\mathbb{A})$. In the following subsection, we shall arrive at a class of modal automata for which the equivalence does hold.

5.6 Disjunctive and semi-disjunctive automata

As mentioned in the introduction to this chapter, an important role in our proof of completeness is played by two kinds of special automata that correspond to classes of functional and thin binary relations and hence allow somewhat simpler trace graphs: *disjunctive* and *semi-disjunctive* automata.

The conditions on these automata can be nicely expressed in terms of restrictions on the one-step language.

5.6.1. Definition. Given sets $\mathtt{X}$ and A we define the sets $\mathtt{LitC}(\mathtt{X})$ and $\mathtt{1ML}^d_\nabla(\mathtt{X}, A)$ by respectively:

$$\pi \in \mathtt{LitC}(\mathtt{X}) ::= \bot \mid \top \mid p \wedge \pi \mid \neg p \wedge \pi$$

where $p \in \mathtt{X}$, and

$$\alpha \in \mathtt{1ML}^d_\nabla(\mathtt{X}, A) ::= \bot \mid \pi \wedge \nabla\beta \mid \alpha \vee \alpha,$$

where $\pi \in \mathtt{LitC}(\mathtt{X})$ and $\beta \in \mathsf{T}A$. Elements of $\mathtt{1ML}^d_\nabla(\mathtt{X}, A)$ are called *one-step disjunctive formulas* and a modal automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ is *disjunctive* if $\Theta(a)$ belongs to $\mathtt{1ML}^d_\nabla(\mathtt{X}, A)$ for all $a \in A$.

5.6.2. Definition. Let $\mathbb{A}$ be a modal automaton and let C be a cluster of $\mathbb{A}$. The set of *(zero-step) C-safe conjunctions*, denoted by $\mathtt{Conj}^C_0(A)$ consists of formulas of the form $\bigwedge B$ with $B \subseteq A$, such that for all $b_1 \neq b_2 \in B \cap C$, either b_1 or b_2 is a maximal even element of C. The grammar

$$\alpha \in \mathtt{1ML}^{s(C)}_\nabla(\mathtt{X}, A) ::= \bot \mid \pi \wedge \nabla\gamma \mid \alpha \vee \alpha,$$

where $\pi \in \mathtt{LitC}(\mathtt{X})$ and $\gamma \in \mathsf{T}\mathtt{Conj}^C_0(A)$, defines the set $\mathtt{1ML}^{s(C)}_\nabla(\mathtt{X}, A)$ of *(one-step) C-safe formulas*. We call a one-step formula α *semi-disjunctive with respect to* $a \in A$ if α is a C_a-safe formula. A modal automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ is *semi-disjunctive* if $\Theta(a)$ is semi-disjunctive with respect to a for all $a \in A$.

5.6.3. Remark. Disjunctive automata are a subclass of semi-disjunctive ones since disjunctive formulas are semi-disjunctive. In more details, given a disjunctive automaton $\mathbb{A}$ and let C be a cluster of $\mathbb{A}$. It is easy to check that each element $a \in A$ can be seen as a C_a-safe conjunction. This implies that $A \subseteq \mathtt{Conj}^C_0(A)$, and by the assumption that T preserves inclusions, we get that this $\mathsf{T}A \subseteq \mathsf{T}\mathtt{Conj}^C_0(A)$.

Note that as we already mentioned, it is not always the case that for an arbitrary automaton $\mathbb{A}$ the satisfiability game $\mathcal{S}(\mathbb{A})$ and the thin satisfiability $\mathcal{S}_{thin}(\mathbb{A})$ are equivalent, but for semi-disjunctive automata it holds:

5.6.4. Proposition. *Let $\mathbb{A}$ be a semi-disjunctive automaton. Then each player $\Pi \in \{\exists, \forall\}$ has a winning strategy in $\mathcal{S}(\mathbb{A})$ if and only if she/he has a winning strategy in $\mathcal{S}_{thin}(\mathbb{A})$.*

Before going to the details of the proof we state the following remark which will be used in the proof of Proposition 5.6.4

5.6.5. Remark. Given a modal automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ and a state $b \in A$, we denote the marking we get from the natural valuation $V_b : A \to \mathcal{P}A^\sharp$ by $n_b : A^\sharp \to \mathcal{P}A$. Then from naturality of one-step formulas (see Proposition 5.3.6) we find that for every $\alpha \in \mathsf{T}_\mathtt{X} A^\sharp$:

$$A^\sharp, \alpha \Vdash^1_b \Theta(b) \Longleftrightarrow \mathcal{P}A, \mathsf{T}_\mathtt{X} n_b(\alpha) \Vdash^1_{V_\mathtt{Id}} \Theta(b),$$

where $V_\mathtt{Id}$ is the valuation corresponding to the identity marking $\mathtt{Id} : \mathcal{P}A \to \mathcal{P}A$.

Proof of Proposition 5.6.4: It is clear that any winning strategy for $\exists$ in $\mathcal{S}(\mathbb{A})$ is still a winning strategy for her in $\mathcal{S}_{thin}(\mathbb{A})$. Conversely, we prove that any winning strategy χ of $\exists$ in $\mathcal{S}_{thin}(\mathbb{A})$ can be transformed into a winning strategy χ' for her in the full game. This strategy χ' will be such that all witnesses α picked by $\exists$ in a χ'-guided match are such that $\mathrm{Base}(\alpha)$ contains only thin relations.

To prove our claim, first of all note that the moves by $\exists$ in the satisfiability game are elements of $\mathsf{T}_{\mathtt{X}} A^\sharp$, so they are pairs containing a subset of set $\mathtt{X}$ as the first component and an element of the set $\mathsf{T}A^\sharp$ as the second component. In the following proof, to simplify the notation, and because everything on the propositional side will be kept the same for both strategy χ and χ', we focus on the second component and think of α and α' as elements of $\mathsf{T}A^\sharp$.

Take a partial χ-guided match $R_0\alpha_0R_1\alpha_1...R_n\alpha_n$, and suppose that

$$\alpha_n \in \bigcap_{a\in \mathsf{ran} R_n} [\![\Theta(a)]\!]_a$$

is the first place such that $\mathrm{Base}(\alpha)$ contains a non-thin relation. We will provide an alternative move α'_n satisfying the following properties:

(i) α'_n is a legitimate move for $\exists$ at position R_n;

(ii) $\mathrm{Base}(\alpha'_n)$ contains only thin relations;

(iii) $(\mathrm{Base}(\alpha'_n), \mathrm{Base}(\alpha_n)) \in \overrightarrow{\mathcal{P}}\subseteq$.

These conditions are there to make sure that we have trace reflection, i.e., for all χ'-guided matches ρ there exists a corresponding χ-guided *shadow-match* such that traces through ρ corresponds to the traces of the shadow match. More particularly, since we ensure that $(\mathrm{Base}(\alpha'_n), \mathrm{Base}(\alpha_n)) \in \overrightarrow{\mathcal{P}}\subseteq$, and that every element of $\mathrm{Base}(\alpha'_n)$ is thin, it follows that every response by $\forall$ to the move α'_n in the satisfiability game $\mathcal{S}(\mathbb{A})$ is also an admissible response to the original move α_n (which was part of the winning strategy χ) in the thin satisfiability game $\mathcal{S}_{thin}(\mathbb{A})$. Hence every χ'-guided play is winning, since it has the same traces as some χ-guided shadow play.

Suppose that $Q \in \mathrm{Base}(\alpha_n)$ is not thin. Then there is an $a \in A$, and $b_1 \neq b_2 \in C_a \cap Q[a]$ such that neither of them is a maximal even element of C_a. In case there are more Q, obtain α'_n by iterating the procedure we are about to describe.

Since $\Theta(a)$ is semi-disjunctive, it is equivalent to $\bigvee\Gamma$ for some Γ where each $\gamma \in \Gamma$ is of the form $\gamma = \pi \wedge \nabla\beta$ for some $\beta \in \mathsf{TConj}_0^{C_a}(A)$ and $\pi \in \mathtt{LitC(X)}$. Now from $\alpha_n \in \bigcap_{a\in\mathsf{ran}R_n}[\![\Theta(a)]\!]_a$ we get for all $a \in \mathsf{ran}R_n$ a disjunct $\pi \wedge \nabla\beta$ such that $\alpha_n \in [\![\pi \wedge \nabla\beta]\!]_a$. Hence $\alpha_n \in [\![\nabla\beta]\!]_a$ which implies that $(\alpha_n, \beta) \in \overline{\mathsf{T}} \Vdash_a^0$.

Define $\mathcal{R} \in \mathcal{P}A^\sharp$ to be the following collection:

$$\mathcal{R} := (\mathrm{Base}(\alpha_n) \setminus \{Q\}) \cup \{Q_1\} \cup \{Q_2\},$$

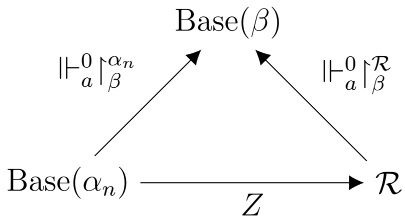
where for $i = 1, 2$ we define $Q_i := Q \setminus \{(a, b_i)\}$. In words, we obtain $\mathcal{R}$ from $\mathrm{Base}(\alpha_n)$ by replacing Q with Q_1 and Q_2. We now introduce relation Z :

$\mathrm{Base}(\alpha_n) \nrightarrow \mathcal{R}$ by:

$$Z := \{(R, R) \in \mathrm{Base}(\alpha_n) \times \mathcal{R} \mid R \neq Q\} \cup \{(Q, Q_1), (Q, Q_2)\}.$$

Note that Z is the converse of the graph of a surjective map from $\mathcal{R}$ to $\mathrm{Base}(\alpha_n)$.

Claim. The following diagram in the category Rel, in which the relation $\Vdash^0_a$ is the zero-step satisfaction relation given by the valuation V_a, the relation $\Vdash^0_a{\restriction}^{\alpha_n}_{\beta}$ is the restriction of $\Vdash^0_a$ to $\mathrm{Base}(\alpha_n) \times \mathrm{Base}(\beta)$, and the relation $\Vdash^0_a{\restriction}^{\mathcal{R}}_{\beta}$ is the restriction of $\Vdash^0_a$ to $\mathcal{R} \times \mathrm{Base}(\beta)$, is commutative.



Proof of Claim. The commutativity of the above diagram boils down to the following identity:

$$\Vdash^0_a{\restriction}^{\alpha_n}_{\beta} = Z \,;\, \Vdash^0_a{\restriction}^{\mathcal{R}}_{\beta}.$$

($\subseteq$) Since $\beta \in \mathsf{T}\mathtt{Conj}^{C_a}_0(A)$, elements of $\mathrm{Base}(\beta) \in \mathcal{P}\mathtt{Conj}^{C_a}_0(A)$ are zero-step C_a-safe conjunctions of the form $\bigwedge B$ for some $B \subseteq A$. Now given $(Y, \bigwedge B) \in (\Vdash^0_a{\restriction}^{\alpha_n}_{\beta})$, from the definition of $\Vdash^0_a$ we get that for all $b \in B$ it holds that $(a, b) \in Y$. In case that $Y \neq Q$ we have $(Y, Y) \in Z$ and $(Y, \bigwedge B) \in (\Vdash^0_a{\restriction}^{\mathcal{R}}_{\beta})$. In case $Y = Q$, because $\bigwedge B$ is a zero-step C_a-safe conjunction, for all $b \neq b' \in B$ either b or b' is a maximal even element of C_a. Recall that b_1 and b_2 obtained from the assumption that Q is not thin, and since they were both assumed not to be maximal even elements, at most one of b_1 and b_2 belongs to B. Depending on whether $b_1 \in B$ or $b_2 \in B$ we get a $Y' = Q \setminus \{(a, b_i)\} \in \mathcal{R}$ with $(Y, Y') \in Z$ such that for all $b \in B$ we have $(a, b) \in Y'$. Hence $(Y', \bigwedge B) \in (\Vdash^0_a{\restriction}^{\mathcal{R}}_{\beta})$ and so we find that $(Y, \bigwedge B) \in (Z \,;\, \Vdash^0_a{\restriction}^{\mathcal{R}}_{\beta})$ indeed.

($\supseteq$) Conversely assume that $(Y, \bigwedge B) \in (Z \,;\, \Vdash^0_a{\restriction}^{\mathcal{R}}_{\beta})$. Then there is an $Y' \in \mathcal{R}$ such that $(Y, Y') \in Z$ and $(Y', \bigwedge B) \in (\Vdash^0_a{\restriction}^{\mathcal{R}}_{\beta})$. Now since $\bigwedge B \in \mathtt{Conj}^{C_a}_0(A)$ and $Y' \subseteq Y$, because for each $(Y, Y') \in Z$ either $Y \neq Q$ and $Y = Y'$ or $Y = Q$ and $Y' = Q_i$, we get that $(Y, \bigwedge B) \in (\Vdash^0_a{\restriction}^{\alpha_n}_{\beta})$. This finishes the proof of the Claim.

Returning to the proof of Proposition 5.6.4, we first observe that by the commutativity of the diagram it follows that:

$$\overline{\mathsf{T}}(\Vdash^0_a{\restriction}^{\alpha_n}_{\beta}) = \overline{\mathsf{T}}(Z) \,;\, \overline{\mathsf{T}}(\Vdash^0_a{\restriction}^{\mathcal{R}}_{\beta}).$$

But then from $(\alpha_n, \beta) \in \overline{\mathsf{T}}(\Vdash^0_a{\restriction}^{\alpha_n}_{\beta})$ we may derive the existence of an $\alpha'_n \in \mathsf{T}\mathcal{R} \subseteq \mathsf{T}A^{\sharp}$ such that $(\alpha_n, \alpha'_n) \in \overline{\mathsf{T}}Z$ and $(\alpha'_n, \beta) \in \overline{\mathsf{T}}(\Vdash^0_a{\restriction}^{\mathcal{R}}_{\beta})$. We now prove that α'_n satisfies the conditions (i)-(iii).

For (i) we show that $\alpha'_n \in \bigcap_{a\in \mathsf{ran} R_n} [\![\Theta(a)]\!]_a$. Take an arbitrary $b \in \mathsf{ran} R_n$ and consider the natural b-marking $n_b : A^\sharp \to \mathcal{P}A$ defined by $R \mapsto R[b] := \{c \in A \mid (b,c) \in R\}$. It is clear from the definition of the relation Z that for $b \neq a$ we have that for all $z = (R, R') \in Z$, $n_b \circ \pi_1(z) = n_b \circ \pi_2(z)$ $(\star)$, because either $R \neq Q$ and $R = R'$ or $R = Q$ and $R' = Q_i$. So in particular for $(\alpha_n, \alpha'_n) \in \overline{\mathsf{T}}Z$, from the definition of $\overline{\mathsf{T}}Z$ (see Definition 2.3.27) and clause $(\star)$ above, we have that $\mathsf{T}n_b(\alpha_n) = \mathsf{T}n_b(\alpha'_n)$. Now since α_n satisfies $\Theta(b)$ and because $\mathsf{T}n_b(\alpha_n) = \mathsf{T}n_b(\alpha'_n)$, by Remark 5.6.5, α'_n also satisfies $\Theta(b)$ (for $b \neq a$). For $b = a$ from $(\alpha'_n, \beta) \in \overline{\mathsf{T}}(\Vdash^0_a\restriction^{\mathcal{R}}_\beta)$ we get that α'_n satisfies $\Theta(a)$.

Clause (ii) is immediate from the definition of $\mathcal{R}$, and the fact that $\alpha'_n \in \mathsf{T}\mathcal{R}$ implies that $\mathrm{Base}(\alpha'_n) \subseteq \mathcal{R}$.

For (iii) we need to prove that $(\mathrm{Base}(\alpha'_n), \mathrm{Base}(\alpha_n)) \in \overrightarrow{\mathcal{P}}\subseteq$. To show this, observe that from $(\alpha'_n, \alpha_n) \in \overline{\mathsf{T}}Z^\circ$ and the fact that Base is natural transformation for weak pullback-preserving functors, we immediately get that:

$$(\mathrm{Base}(\alpha'_n), \mathrm{Base}(\alpha_n)) \in \overline{\mathcal{P}}Z^\circ \subseteq \overline{\mathcal{P}}(\subseteq) \subseteq \overrightarrow{\mathcal{P}}(\subseteq).$$

To finish the proof of Proposition 5.6.4, note that by repeating the same process we can define χ' inductively for matches of arbitrary length and get a winning strategy for $\exists$ such that all the witnesses picked by $\exists$ have a *thin* Base, i.e, a Base which contains only thin relations. And finally, the results for $\forall$ directly follow from the results for $\exists$ and the determinacy of the satisfiability game. □

The final proposition of this section summarizes some of the closure properties of (semi)-disjunctive automata. Here we say an automaton is (semi-)disjunctive *modulo one-step equivalence* if is it one-step equivalent to a (semi-)disjunctive automaton.

5.6.6. PROPOSITION. *Let $\mathbb{A}$ and $\mathbb{B}$ be modal automata. Then we have:*

(1) If $\mathbb{A}$ is disjunctive, then it is also semi-disjunctive;

(2) If $\mathbb{A}$ and $\mathbb{B}$ are disjunctive, then so is $\mathbb{A} \vee \mathbb{B}$;

(3) If $\mathbb{A}$ and $\mathbb{B}$ are semi-disjunctive, then so is $\mathbb{A} \vee \mathbb{B}$;

(4) If $\mathbb{A}$ and $\mathbb{B}$ are semi-disjunctive, then so is $\mathbb{A} \wedge \mathbb{B}$, modulo one-step equivalence;

(5) If $\mathbb{A}$ and $\mathbb{B}$ are semi-disjunctive then so is $\mathbb{A}[\mathbb{B}/x]$, modulo one-step equivalence;

(6) If $\mathbb{A}$ is disjunctive and positive in x, then $\nu x.\mathbb{A}$ and $\mathbb{A}^x$ are semi-disjunctive, modulo one-step equivalence.

Proof:
The first three statements are immediate consequences of the definition. We skip the proof of the fourth statement: it is similar to but simpler than that of (5), since in the case of $\mathbb{A} \wedge \mathbb{B}$ there is only one state, where we have to replace a conjunction of ∇-formulas by a disjunction of appropriate ∇-formulas, viz., the initial one.

Clause (5): Let's denote the automaton $\mathbb{A}[\mathbb{B}/x]$ by $\mathbb{D}$. We need to prove that for all $d \in D := A \uplus B$ the formula $\Theta_D(d)$ is one-step equivalent to a formula which is semi-disjunctive with respect to d. First observe that the states from $\mathbb{B}$ cause no problem whatsoever: for $d \in B$ we have that $\Theta_D(d) = \Theta_B(d)$, so the one-step formulas assigned to any $d \in B$ are d-safe since $\mathbb{B}$ is semi-disjunctive. So it is enough to prove the semi-disjunctivity (modulo one-step equivalence) of formulas belonging to the $\mathbb{A}$-part of the automaton $\mathbb{A}[\mathbb{B}/x]$. In the following we will show that for all $d \in A$ the formula $\Theta_D(d) = \Theta_A(d)[\Theta_B(b_I)/x]$ is semi-disjunctive with respect to d.

Pick $d \in A$. Because $\Theta_A(d)$ is a semi-disjunctive formula, it is of the form $\bigvee_{i \in I} \pi_i \wedge \nabla\gamma_i$ with $\gamma_i \in \mathtt{TConj}_0^{C_d}(A)$ and since $\mathbb{B}$ is semi-disjunctive too, $\Theta_B(b_I)$ is of the form $\bigvee_{k \in K} \pi'_k \wedge \nabla\beta_k$ with $\beta_k \in \mathtt{TConj}_0^{C_{b_I}}(B)$. Note that x can only occur in lattice terms π_i, hence we have the following (if $x \notin \pi$ then $\Theta_A(d)[\Theta_B(b_I)/x] = \Theta_A(d)$ and so is semi-disjunctive):

$$\begin{array}{rcl}\Theta_A(d)[\Theta_B(b_I)/x] & = & \bigvee_{i \in I} \pi_i[\Theta_B(b_I)/x] \wedge \nabla\gamma_i \\ & = & \bigvee_{i \in I} (\pi_i \setminus x) \wedge (\bigvee_{k \in K} \pi'_k \wedge \nabla\beta_k) \wedge \nabla\gamma_i\end{array}$$

It is not difficult to check that by the rules of propositional logic, we can rewrite $\Theta_A(d)[\Theta_B(b_I)/x]$ as a disjunction of conjunctions of the form $(\pi \setminus x) \wedge \pi' \wedge \nabla\beta \wedge \nabla\gamma$, and consider each of these conjunctions separately. In this proof, for simplicity and without loss of generality, we assume that there is only one such conjunction and $\Theta_A(d)[\Theta_B(b_I)/x]$ is of the following form:

$$\Theta_A(d)[\Theta_B(b_I)/x] = (\pi \setminus x) \wedge \pi' \wedge \nabla\beta \wedge \nabla\gamma.$$

To show that $\Theta_A(d)[\Theta_B(b_I)/x] = \pi'' \wedge \nabla\beta \wedge \nabla\gamma$ with $\pi'' = (\pi \setminus x) \wedge \pi'$ is equivalent to a semi-disjunctive formula with respect to d, it is enough to prove that $\nabla\beta \wedge \nabla\gamma$ is equivalent to a disjunction $\bigvee_{j \in J} \nabla\delta_j$ such that each disjunct $\nabla\delta_i$ is a one-step C_d-safe formula, i.e., $\delta_j \in \mathtt{TConj}_0^{C_d}(D)$. Because then we get $\Theta_A(d)[\Theta_B(b_I)/x] \equiv_{\mathbf{K}} \bigvee_{i \in I} \pi''_i \wedge \nabla\delta_i$ and so is a semi-disjunctive formula with respect to d. But this is the case by the modal distribution law from Proposition 5.3.11:

$$\begin{array}{r}\nabla\beta \wedge \nabla\gamma \equiv_K \bigvee\{\nabla(\mathsf{T}\wedge)\alpha \mid \alpha \in \mathsf{T}(\mathrm{Base}(\beta) \times \mathrm{Base}(\gamma)) \text{ s.t.} \\ \mathsf{T}\pi_1(\alpha) = \beta \text{ and } \mathsf{T}\pi_2(\alpha) = \gamma\},\end{array}$$

and since there is no cluster of automaton $\mathbb{A}[\mathbb{B}/x]$ that contains states from both A and B, the image of the restriction of the conjunction map $\wedge$ to $\mathrm{Base}(\gamma) \times \mathrm{Base}(\beta)$ is contained in the set of zero-step C_d-safe conjunctions and so we have that $(\mathsf{T}\wedge)\alpha$ for $\alpha \in \mathsf{T}(\mathrm{Base}(\gamma) \times \mathrm{Base}(\beta))$ belongs to$\mathsf{T}\mathtt{Conj}_0^{C_d}(A)$.

Clause (6): Recall from Proposition 5.3.23 that for every one-step formula $\Theta(a)$ there exist formulas θ_i^a in which x does not appear and such that $\Theta(a) \equiv_{\mathbf{K}} (x \wedge \theta_0^a) \vee \theta_1^a$. Assume that the automaton $\mathbb{A}$ is disjunctive. Then it is not difficult to check that without loss of generality we can assume that formulas θ_i^a themselves are disjunctive. The problem is that when we define the transition maps for the automata $\mathbb{A}^x$ and $\nu x.\mathbb{A}$, the substitution $\kappa : a \mapsto (\underline{x} \wedge a_0) \vee a_1$ introduces conjunctions in the scope of modality ∇. We claim however, that:

if $\alpha \in \mathtt{1ML}^d_\nabla(\mathtt{X}, A)$, then $\alpha[\kappa]$ is equivalent to a one-step C_a-safe formula for every a.

Clearly we may restrict our attention to the case of nabla formulas $\alpha = \nabla\beta$ with $\beta \in \mathsf{T}A$. To prove that $\nabla\beta[\kappa]$ is a one-step C_a-safe formula for every a, we use the following one-step equivalence [KKV08] to distribute ∇ over disjunctions (cf. Axiom $\Delta 3$)

$$\nabla(\mathsf{T}\vee)\Phi \equiv_1 \bigvee\{\nabla\gamma \mid \gamma\,(\overline{\mathsf{T}}\!\in)\,\Phi\}.$$

Now given the substitution κ, we can think of it as $\bigvee \circ g$ where $g : a \mapsto \{\underline{x} \wedge a_0, a_1\}$. So we get that $\nabla\beta[\kappa] = \nabla(\mathsf{T}\vee)(\mathsf{T}g)(\beta)$ is one-step equivalent to a disjunction of $\nabla\gamma$ with $\gamma\,(\overline{\mathsf{T}}\!\in)\,\mathsf{T}g(\beta)$. To finish the proof it is enough to show that in the case of both $\mathbb{A}^x$ and $\nu x.\mathbb{A}$ we have $\gamma \in \mathsf{T}\mathtt{Conj}_0^{C_a}$ for all a.

From $(\gamma, \mathsf{T}g(\beta)) \in \overline{\mathsf{T}}\!\in$ and definitions of Base (Definition 2.3.21) and Barr lifting (Example 2.3.27), we get that $(\mathrm{Base}(\gamma), \mathrm{Base}(\mathsf{T}g(\beta))) \in \overline{\mathcal{P}}\!\in$. Now from naturality of Base we have: $(\mathrm{Base}(\gamma), g[\mathrm{Base}(\beta)]) \in \overline{\mathcal{P}}\!\in$. We claim that for both $\mathbb{A}^x$ and $\nu x.\mathbb{A}$ it holds that $\mathrm{Base}(\gamma) \subseteq \mathtt{Conj}_0^{C_a}$ so that $\gamma \in \mathsf{T}\mathtt{Conj}_0^{C_a}$ for all a. To prove the claim take some $\tau \in \mathrm{Base}(\gamma)$. We have that there is $a \in \mathrm{Base}(\beta)$ such that $\tau \in g(a) = \{\underline{x} \wedge a_0, a_1\}$. It is easy to see that τ is a zero-step C_a-safe conjunction because in the case of $\mathbb{A}^x$ the claim follows from the fact that $\underline{x}$ forms a degenerate cluster on its own, so that $\underline{x}$ and a_0 belong to different clusters. In the case of $\nu x.\mathbb{A}$ the state $\underline{x}$ is by construction the maximal even element of its cluster, so that again the $\underline{x} \wedge a_0$ is a safe zero-step conjunction. Hence we have $\mathrm{Base}(\gamma) \subseteq \mathtt{Conj}_0^{C_a}$. □

5.6.1 A key lemma

We now come to one of the key results of this chapter. The role of this Theorem in the overall completeness proof is to establish a link between the two games we have introduced for modal automata, the satisfiability game and the consequence game.

5.6.7. THEOREM. *Let $\mathbb{A}$ be a semi-disjunctive automaton and $\mathbb{D}$ be an arbitrary modal automaton such that $\mathbb{A} \vDash_{\mathcal{C}} \mathbb{D}$. Then $\mathbb{A} \wedge \neg\mathbb{D}$ has a thin refutation.*

This theorem is an automata-theoretic version of Lemma 36 from [Wal00], one of the key lemmas of Walukiewicz' completeness proof, and at the same time it generalizes that result in two ways. First, our coalgebraic approach extends the result from the power set functor $\mathcal{P}$ to a set functor T. Second, we prove the result for an arbitrary automaton $\mathbb{D}$ instead for a disjunctive one.

Before we prove this theorem, we formulate an auxiliary lemma. Recall that the transition map of the automaton $\neg\mathbb{D}$ is defined by taking boolean duals of the formulas assigned by the transition map of $\mathbb{D}$, and the priority map is defined by simply raising all priorities by 1. We shall need the following fact on boolean duals, which is a straightforward consequence of the definitions.

5.6.8. PROPOSITION. *Let (S, σ) be a one-step $\mathsf{T}_{\mathtt{X}}$-frame, let φ be a one-step formula in $\mathtt{1ML}_{\nabla}(\mathtt{X}, A)$ and let $m, m' : S \to \mathcal{P}(A)$ be two markings such that $S, m, \sigma \Vdash^1 \varphi$ and $S, m', \sigma \Vdash^1 \varphi^{\delta}$. Then for some $a \in A$ and some $s \in S$ we have $a \in m(s) \cap m'(s)$.*

Proof of Theorem 5.6.7: We first fix the notation, let $\mathbb{A} = (A, \Theta_A, \Omega_A, a_I)$ and $\mathbb{D} = (D, \Theta_D, \Omega_D, d_I)$ and denote the automaton $\mathbb{A} \wedge \neg\mathbb{D}$, denoted by $\mathbb{B} = (B, \Theta_B, \Omega_B, b_I)$ with $B = A \cup D \cup \{b_I\}$. Let σ be the winning strategy for player II in $\mathcal{C}(\mathbb{A}, \mathbb{D})$ which exists by the assumption that $\mathbb{A} \vDash_{\mathcal{C}} \mathbb{D}$. We will use σ to construct a winning strategy χ for $\forall$ in $\mathcal{S}_{thin}(\mathbb{B})$. Given a χ-guided partial match ρ in $\mathcal{S}(\mathbb{B})$ with basic positions

$$R_0 R_1 R_2 ... R_n$$

our aim is to introduce a response R_{n+1} for $\forall$ to every possible move $\gamma = (\mathtt{Y}, \Gamma)$ by $\exists$, such that:
(i) R_{n+1} is a legitimate move for $\forall$, i.e., $R_{n+1} \subseteq R'$ for some $R' \in \mathrm{Base}(\Gamma)$;
(ii) $\mathsf{ran}(R_{n+1}) \cap D$ is a singleton;
(iii) There is a σ-guided partial $\mathcal{C}(\mathbb{A}, \mathbb{D})$-match $(S_0, S'_0)(S_1, S'_1)...(S_n, S'_n)$, where
(a) $S_0 = \{(a_I, a_I)\}$ and $S'_0 = \{(d_I, d_I)\}$;
(b) $S_1 = \{(a_I, a) \in A^2 \mid (a_I, a) \in R_1\}$ and $\{(d_I, d) \in D^2 \mid (a_I, d) \in R_1\} \subseteq S'_1$;
(c) for each $i > 1$ we have $R_i \cap A^2 = S_i$ and $R_i \cap D^2$ is a singleton $\{(d, d')\}$ with $d \in \mathsf{ran}(R_{i-1}) \cap D$ and $(d, d') \in S'_i$.

By definition we have $R_0 = \{(b_I, b_I)\}$, $S_0 = \{(a_I, a_I)\}$ and $S'_0 = \{(d_I, d_I)\}$. The definition of R_1 will be given after we show how to define R_{n+1} for $n \geq 1$, as it will be a variation on this general case.

Suppose the inductive hypothesis has been maintained for the partial match ρ with basic positions $R_0R_1R_2...R_n$, with shadow match $(S_0, S'_0)(S_1, S'_1)...(S_n, S'_n)$, and let $\gamma = (\Upsilon, \Gamma) \in \mathsf{T}_{\mathsf{X}}(B)^\sharp$ be an arbitrary move chosen by $\exists$ at the position R_n. Let d be the unique element of $\mathsf{ran}(R_n) \cap D$ and let $\mathsf{ran}(R_n) \cap A = \{a_1, ..., a_k\}$. Then we have by the admissibility of γ that:

$$\gamma \in \bigcap_{1 \le i \le k} [\![\Theta_A(a_i)]\!]_{a_i} \cap [\![\Theta_D(d)^\delta]\!]_d.$$

In terms of natural valuations (Definition 5.3.12), this means:

$$B^\sharp, \gamma, n^B_{a_i} \Vdash^1 \Theta_A(a_i) \ \text{ for all } \ a_1, ..., a_k$$

and

$$B^\sharp, \gamma, n^B_d \Vdash^1 \Theta_D(d)^\delta.$$

Then by Corollary 5.3.7 and Proposition 5.3.8, we obtain that

$$\mathrm{Base}(\Gamma), \gamma, n^D_d \Vdash^1 \Theta_D(d)^\delta, \tag{5.4}$$

and

$$A^\sharp, \gamma, n^A_a \Vdash^1 \Theta_A(a) \text{ for all } a \in A \cap \mathsf{ran} R_n. \tag{5.5}$$

Let $\mathsf{res}_A : B^\sharp \to A^\sharp$ denote the map sending R to $R \cap A^2$, and similarly for $\mathsf{res}_D : B^\sharp \to D^\sharp$.

By (5.5) and naturality of one-step formulas (Proposition 5.3.6) we get:

$$\mathsf{T}_{\mathsf{X}}\mathsf{res}_A(\gamma) \in \bigcap_{1 \le i \le k} [\![\Theta_A(a_i)]\!]_{a_i}$$

meaning that:

$$A^\sharp, \mathsf{T}_{\mathsf{X}}\mathsf{res}_A(\gamma), n^A_a \Vdash^1 \Theta_A(a), \ \text{for all } a \in A \cap \mathsf{ran} R_n \tag{5.6}$$

But then by Corollary 5.3.7 we have that

$$\mathrm{Base}(\Gamma), \mathsf{T}_{\mathsf{X}}\mathsf{res}_A(\gamma), n^A_a \Vdash^1 \Theta_A(a), \ \text{for all } a \in A \cap \mathsf{ran} R_n, \tag{5.7}$$

And hence $\mathsf{T}_{\mathsf{X}}\mathsf{res}_A(\gamma)$ is an admissible move for player I in the consequence game at position (S_n, S'_n). Thus by Player II's winning strategy σ we find a map $f : A^\sharp \to D^\sharp$ such that $\mathsf{T}_{\mathsf{X}}(f \circ \mathsf{res}_A)(\gamma) \in [\![\Theta_D(d)]\!]$. Denoting $\mathsf{T}_{\mathsf{X}}(f \circ \mathsf{res}_A)(\gamma)$ by $\gamma' = (\Upsilon', \Gamma')$ we get:

$$\mathrm{Base}(\Gamma'), \gamma', n^D_d \Vdash^1 \Theta_D(d), \tag{5.8}$$

We shall prove the following claim:

Claim. There is some $S \in \mathrm{Base}(\Gamma)$, and some $c \in D$ such that $(d, c) \in f(\mathsf{res}_A(S)) \cap \mathsf{res}_D(S)$.

Proof of Claim: Note that the map $f : A^\sharp \to D^\sharp$ is a one-step frame morphism. Hence, if we define a marking $m : \text{Base}(\Gamma) \to \mathcal{P}D$ by setting

$$m(S) := n_d^D(f(\mathsf{res}_A(S))),$$

then we may apply naturality (Proposition 5.3.6) to (5.8) and obtain

$$\text{Base}(\Gamma), \gamma, m \Vdash^1 \Theta_D(d). \tag{5.9}$$

But then by Proposition 5.6.8, it follows from (5.4) and (5.9) that there is some $c \in D$ and some $S \in \text{Base}(\Gamma)$ such that $c \in n_d^D(S) \cap m(S)$. Unravelling the definitions of n_d^D and m we find that, respectively, $(d, c) \in \mathsf{res}_D(S)$ and $(d, c) \in f(\mathsf{res}_A(S))$ as required. □

With this claim in place, we define the next move for ∀ prescribed by the strategy χ to be the relation $R_{n+1} := \mathsf{res}_A(S) \cup \{(d, c)\}$, where $S \in \text{Base}(\Gamma)$ and $c \in D$ are as described in the claim, so that $(d, c) \in f(\mathsf{res}_A(S)) \cap \mathsf{res}_D(S)$. Note that this R_{n+1} is a legitimate move in response to $\gamma = (\mathtt{Y}, \gamma)$ since $R_{n+1} \subseteq S \in \text{Base}(\Gamma)$. The shadow match is then extended by the pair $(S_{n+1}, S'_{n+1}) := (\mathsf{res}_A(S), f(\mathsf{res}_A(S))$ so that condition (iii)(c) of the induction hypothesis holds as an immediate consequence of the claim. For conditions (i) and (ii), it is obvious that $|\mathsf{ran}(R_{n+1}) \cap D| = 1$. By an argument similar to the one used in the proof of Proposition 5.6.4, and the assumption that $\mathbb{A}$ is semi-disjunctive, we can ensure that R_{n+1} is thin with respect to $\mathbb{A}$.

We now explain how to define R_1. The definition follows a similar line of argumentation, but we need to take care of the switch between initial states in $\mathbb{B}$, $\mathbb{A}$ and $\mathbb{D}$ such that we can relate traces in ρ with traces in the shadow match. First note that in position R_0, by the definition of $\mathbb{A} \wedge \neg\mathbb{D}$, an admissible ∃-move γ is such that:

$$B^\sharp, \gamma, n_{b_I}^B \Vdash^1 \Theta_A(a_I)$$

and

$$B^\sharp, \gamma, n_{b_I}^B \Vdash^1 \Theta_D(d_I)^\delta.$$

Now instead of res_A and res_D, we use the maps $\mathsf{res}_A^* : A^\sharp \to A^\sharp$ and $\mathsf{res}_D^* : B^\sharp \to D^\sharp$ defined by:

$$\mathsf{res}_A^*(S) = \{(a_I, a) \mid (b_I, a) \in S\},$$

and

$$\mathsf{res}_D^*(S) = \{(d_I, d) \mid (b_I, d) \in S\}.$$

Then by naturality (Proposition 5.3.6), we get

$$A^\sharp, (\mathsf{T}_{\mathtt{X}}\mathsf{res}_A^*)(\gamma), n_{a_I}^A \Vdash^1 \Theta_A(a_I)$$

and

$$D^\sharp, (\mathsf{T_X res}^*_D)(\gamma), n^D_{d_I} \Vdash^1 \Theta_D(d_I)^\delta.$$

So $(\mathsf{T_X res}^*_A)(\gamma)$ is a legal move for player I in $\mathcal{C}(\mathbb{A}, \mathbb{D})$. Let $f : A^\sharp \to D^\sharp$ be the σ-guided response by player II. Then we have

$$D^\sharp, \mathsf{T_X}(f \circ \mathsf{res}^*_A)(\gamma), n^D_{d_I} \Vdash^1 \Theta_D(d_I).$$

Now, as in the above Claim, define the marking $m := n^D_{d_I} \circ f \circ \mathsf{res}^*_A$. By naturality, we then get

$$\text{Base}(\gamma), \gamma, m \Vdash^1 \Theta_D(d_I).$$

Hence by Proposition 5.6.8, we obtain $S \in \text{Base}(\gamma)$ and $c \in D$ such that $c \in n^B_{b_I}(S) \cap m(S)$. Now define

$$R_1 := \{(b_I, a) \mid a \in A\} \cup \{(b_I, c)\}.$$

Note that R_1 is a legal $\forall$-move since $c \in n^B_{b_I}(S)$ and hence $(b_I, c) \in S$. We then extend the shadow match by setting $S_1 = \mathsf{res}^*_A(S)$ and $S'_1 = f(S_1)$. To see that condition (iii)(b) holds, note that:

$$\begin{array}{rcl}
\{(d_I, d) \in D^2 \mid (b_I, d) \in R_1\} & = & \{(d_I, c)\} \\
& \subseteq & \{(d_I, d) \in D^2 \mid d \in m(S)\} \\
& = & \{(d_I, d) \in D^2 \mid d \in (n^D_{d_I} \circ f \circ \mathsf{res}^*_A)(S)\} \\
& \subseteq & f(\mathsf{res}^*_A(S)) = S'_1.
\end{array}$$

To show that the thus defined strategy χ is winning for $\forall$, first observe that he never gets stuck, so that we may focus on infinite matches. It suffices to prove that every infinite χ-guided match contains a bad trace, so consider an arbitrary such match $\rho = (R_i)_{i \geq 0}$.

Clearly we may assume that all initial parts of ρ, corresponding to the partial matches $(R_i)_{0 \leq i \leq n}$, satisfy the conditions (i) to (iii). From this it follows that ρ itself has an infinite σ-guided shadow match $(S_i, S'_i)_{i \geq 0}$ satisfying the condition (iii)(a-c). In addition, it follows from (i) and (ii) that ρ will contain a *unique* trace in D, which by (iii) will also be a trace on the right side of the shadow match in the consequence game. That is, the match $R_0 R_1 R_n \dots$ contains a unique trace of the form $b_I d_1 d_2 d_3 \dots$ with each d_i in D, and by definition of S_1, this trace corresponds to a trace $d_I d_1 d_2 d_3 \dots$ through the stream $S'_0 S'_1 S'_2 \dots$. If this trace is bad, then we are done. If not, then given the priorities assigned to states in $\neg\mathbb{D}$ it must be bad as a trace in $\mathbb{D}$ since parities are swapped in $\neg\mathbb{D}$. Hence there must be a bad trace $a_I a_1 a_2 a_3 \dots$ on the left side $S_0 S_1 S_2 \dots$ of the shadow match in the consequence game, since this shadow match was guided by the winning strategy σ of Player II. But then this trace corresponds to a bad $b_I a_1 a_2 a_3 \dots$ a bad trace in the match $R_0 R_1 R_2 \dots$ of the satisfiability game. Summarizing, we see that either the unique trace through D in ρ is bad or there is some bad trace through A in ρ. In either case, ρ is a loss for $\exists$ as required. $\square$

5.7 A strong simulation theorem

The goal of this section is to prove a strengthened simulation theorem for coalgebra automata: we will present a construction $\mathsf{sim}(.)$ that turns an arbitrary modal automaton $\mathbb{A}$ into a disjunctive modal automaton $\mathsf{sim}(\mathbb{A})$ that is not only semantically equivalent to $\mathbb{A}$, but in fact game-equivalent to $\mathbb{A}$ in the strong sense as stated in Theorem 5.7.2 below. Roughly speaking, the idea is to define $\mathsf{sim}(\mathbb{A})$ via a variation of the power set construction such that a match of the acceptance game of $\mathsf{sim}(\mathbb{A})$ corresponds to $\exists$ simultaneously playing various matches of the acceptance game of $\mathbb{A}$.

5.7.1. Definition. Fix a modal $\mathtt{X}$-automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$. Given a set $\mathtt{Y} \subseteq \mathtt{X}$, let $\widehat{\mathtt{Y}}$ denote the following set of formulas:

$$\bigwedge\{p \mid p \in \mathtt{Y}\} \wedge \bigwedge\{\neg p \mid p \notin \mathtt{Y}\}.$$

The *pre-simulation* of $\mathbb{A}$ is defined to be the structure $\mathsf{pre}(\mathbb{A}) = (A^\sharp, \Theta', NBT_\Omega, a'_I)$ where $A^\sharp := \mathcal{P}(A \times A)$ as always, $a'_I := \{(a_I, a_I)\}$,

$$\Theta'(R) = \bigvee\{\widehat{\mathtt{Y}} \wedge \nabla\alpha \mid (\mathtt{Y}, \alpha) \in \bigcap_{b \in \mathsf{ran} R} [\![\Theta(b)]\!]_b\},$$

and NBT_Ω as usual is the set of streams over $A^\sharp$ that do not contain any bad traces.

Since the acceptance condition NBT_Ω is an ω-regular language with alphabet $A^\sharp$, we may pick some deterministic parity automaton $\mathbb{Z} = (Z, \delta, \Omega', z_I)$ on alphabet $A^\sharp$ that recognizes NBT_Ω. Finally, we define $\mathsf{sim}(\mathbb{A})$ to be the modal automaton $(D, \Theta'', \Omega'', d_I)$ where:

- $D = A^\sharp \times Z$,
- $d_I = (a'_I, z_I)$,
- $\Theta''(R, z) = \Theta'(R)[(Q, \delta(R, z))/Q \mid Q \in A^\sharp]$ and
- $\Omega''(R, z) = \Omega'(z)$.

We also define a "forgetful" map $G : D \to A^\sharp$ by sending a pair (R, z) in the product $A^\sharp \times Z$ to its left component R.

Note that the simulation construction is very tightly related to the satisfiability game; the states of the pre-simulation of $\mathbb{A}$ just are the basic positions of the satisfiability game for $\mathbb{A}$, and the acceptance condition for the pre-simulation of $\mathbb{A}$ is exactly the winning condition in $\mathcal{S}(\mathbb{A})$.

The main result of this section is the following result, which is one of the key lemmas in our completeness proof.

5.7.2. Theorem. *The map* $\mathsf{sim}(\cdot)$ *assigns to each modal automaton* $\mathbb{A}$ *a disjunctive automaton* $\mathsf{sim}(\mathbb{A})$ *such that:*

(1) $\mathbb{A} \models_{\mathcal{C}} \mathsf{sim}(\mathbb{A})$ *and* $\mathsf{sim}(\mathbb{A}) \models_{\mathcal{C}} \mathbb{A}$*,*

(2) $\mathbb{B}[\mathsf{sim}(\mathbb{A})/x] \models_{\mathcal{C}} \mathbb{B}[\mathbb{A}/x]$*, for any modal automaton* $\mathbb{B}$ *which is positive in* x*.*

Proof:
To prove that $\mathbb{A} \models_{\mathcal{C}} \mathsf{sim}(\mathbb{A})$, let $\mathbb{Z}$ be the fixed stream automaton that recognizes NBT_{Ω}. Every finite word $R_0 \ldots R_k$ over $A^\sharp$ determines an associated state of $\mathbb{Z}$ by simply running $\mathbb{Z}$ on the word $R_0 \ldots R_k$; so for ϵ (the empty word) the associated state is z_I, for R_0 the associated state is $\delta(R_0, z_I)$ etc. Since every partial match ρ of the consequence game $\mathcal{C}(\mathbb{A}, \mathsf{sim}(\mathbb{A}))$ determines a word $R_0 \ldots R_k$ over $A^\sharp$ in the obvious way, we can associate a state z_ρ of $\mathbb{Z}$ with each such partial match. If player I continues the match ρ consisting of basic positions $(R_0, R'_0) \ldots (R_k, R'_k)$ by choosing the move $(\mathtt{Y}, \alpha) \in \mathsf{T}_{\mathtt{X}} A^\sharp$, then we let player II respond with the map $f : \mathrm{Base}(\alpha) \to (A^\sharp \times Z)^\sharp$ (since only the action on $\mathrm{Base}(\alpha)$ will matter for legitimacy of the move) that is defined by mapping $R \in \mathrm{Base}(\alpha)$ to the singleton $\{((R_k, z_\rho), (R, \delta(R_k, z_\rho))\}$. It can be checked that this defines a functional winning strategy for Player II, and we leave the details to the reader.

The direction $\mathsf{sim}(\mathbb{A}) \models_{\mathcal{C}} \mathbb{A}$ of clause (1), which can be seen as a simple special case of clause (2), follows from Propositions 5.7.5 and 5.7.6, as will clause (2) itself. □

The difficult part of Theorem 5.7.2 is to prove clause (2), and this will be the focus of the rest of this section. It will be convenient to state more abstractly what the crucial properties are of the automaton $\mathsf{sim}(\mathbb{A})$ that we have associated with an arbitrary automaton $\mathbb{A}$. First we need an auxiliary definition, for which we recall the notion of a *true* state from Definition 5.3.22.

5.7.3. Definition. Given a disjunctive automaton $\mathbb{D} = (D, \Theta, \Omega, d_i)$, and a fixed true state $d_\top$ of $\mathbb{D}$, we let

$$\mathsf{Sing}_\top(d) := \begin{cases} \varnothing & \text{if } d = d_\top \\ \{d\} & \text{if } d \neq d_\top. \end{cases}$$

define the D-marking $\mathsf{Sing}_\top : D \to \mathcal{P}D$.

5.7.4. Definition. Let $\mathbb{A} = (A, \Theta_A, \Omega_A, a_I)$ and $\mathbb{D} = (D, \Theta_D, \Omega_D, d_I)$ be an arbitrary and a disjunctive modal automata respectively. We say $\mathbb{D}$ is a *disjunctive companion* of $\mathbb{A}$ if $\mathbb{D}$ has a true state $d_\top$ and there is a map $G : D \to A^\sharp$ satisfying the following conditions:

(DC1) $Gd_I = \{(a_I, a_I)\}$ and $G(d_\top) = \varnothing$.

(DC2) Let $\delta \in \mathsf{T_x}D$ and $D, \delta, \mathsf{Sing}_\top \vDash^1 \Theta_D(d)$. Then $\mathsf{T_x}G(\delta) \in [\![\Theta_A(a)]\!]_a$ for all $a \in \mathsf{ran}(Gd)$.

(DC3) If $(Gd_i)_{i\in\omega} \in (A^\sharp)^\omega$ contains a bad $\mathbb{A}$-trace, then $(d_i)_{i\in\omega}$ is itself a bad $\mathbb{D}$-trace.

The map G in this definition is intended as a witness of a tight structural relationship between the automaton $\mathbb{D}$ and the satisfiability game for $\mathbb{A}$. In particular the map G captures the intuition that every state of a disjunctive companion represents a macro-state of $\mathbb{A}$ (i.e. a position of the satisfiability game), plus possibly some extra information. In the concrete case of the automaton $\mathsf{sim}(\mathbb{A})$, this "extra information" is a state of the stream automaton that detects bad traces. Informally one can think of a state $d \in D$ as a conjunction of the states in $\mathsf{ran}(Gd)$, or differently put: for each for $a \in \mathsf{ran}(Gd)$, think of a as being "implied by" d.

Each of the clauses of this definition can thus be given an informal explanation that is consistent with this idea. The first clause (DC1) simply expresses that the start state of $\mathbb{D}$ is a representation of the start position of the satisfiability game $\mathcal{S}(\mathbb{A})$. The second clause (DC2) captures the idea that any state a $a \in \mathsf{ran}(Gd)$ is "entailed" by d in the following sense. Given an object $\delta = (\mathsf{Y}, \alpha) \in \mathsf{T_x}D$, we can see δ as a one-step model over D by taking $\mathsf{Sing}_\top : D \to \mathcal{P}D$ (restricted to the Base(α)) as a D-marking. Similarly, applying the map G to δ, we obtain $\mathsf{T_x}G(\delta) \in \mathsf{T_x}A^\sharp$. Recalling from Remark 5.3.14, $\mathsf{T_x}G(\delta)$ may be taken as a family $\{(A^\sharp, n^A_a, \mathsf{T_x}G(\delta)) \mid a \in A\}$ of one-step models. Now the condition (DC2) requires that if $D, \delta, \mathsf{Sing}_\top \vDash^1 \Theta_D(d)$, then $A^\sharp, n^A_a, \mathsf{T_x}G(\delta) \vDash^1 \Theta_A(a)$ for all $a \in \mathsf{ran}(Gd)$. Finally clause (DC3) makes sure that if $(d_i)_{i\in\omega}$ is a "good" $\mathbb{D}$-stream, in the sense that it satisfies the acceptance condition of $\mathbb{D}$, then the $A^\sharp$-stream $(Gd_i)_{i\in\omega}$ has no bad trace, and thus each of its traces is a win for $\exists$ in the satisfiability game.

5.7.5. Proposition. *The simulation map* $\mathsf{sim}(.)$ *assigns a disjunctive companion to any modal automaton.*

Proof:
It is fairly straightforward to check that the map $G : D \to A^\sharp$, specified in Definition 5.7.1, which simply forgets the states of the stream automata used in the product construction, has all the properties required to witness that $\mathsf{sim}(A)$ is a disjunctive companion of $\mathbb{A}$. □

5.7.6. Proposition. *Let* $\mathbb{A}$ *and* $\mathbb{B}$ *be arbitrary modal automata, let* $\mathbb{D}$ *be a disjunctive companion of* $\mathbb{A}$*, and assume that* $\mathbb{B}$ *is positive in* x*. Then:*

$$\mathbb{B}[\mathbb{D}/x] \vDash_{\mathcal{C}} \mathbb{B}[\mathbb{A}/x].$$

Before we set out to prove Proposition 5.7.6, we provide an auxiliary result from [ESV19], which helps us to make a simplifying assumption on the moves player I makes in the consequence game associated with $\mathbb{B}[\mathbb{D}/x]$ and $\mathbb{B}[\mathbb{A}/x]$. The proof is completely analogous to that of Proposition 8.7 in [ESV19], and so we omit the rather tedious details.

5.7.7. PROPOSITION. *Let Θ_{BD} denote the transition map of the automaton $\mathbb{B}[\mathbb{D}/x]$, where $\mathbb{B}$ is an arbitrary modal automaton (positive in x), and $\mathbb{D}$ is a disjunctive disjunctive automaton. Fix some $R \in (B \cup D)^\sharp$, $\mathcal{Q} \subseteq (B \cup D)^\sharp$, some $C \subseteq \mathsf{ran} R$ and $\Gamma \in \mathsf{T}_\mathtt{X}\mathcal{Q}$ such that*

$$\Gamma \in \bigcap_{a \in \mathsf{ran} R} [\![\Theta_{BD}(a)]\!]^1_a.$$

Then there are $\mathcal{Q}' \subseteq (B \cup D)^\sharp$ and $\Gamma' \in \mathsf{T}_\mathtt{X}\mathcal{Q}'$ such that $\Gamma' \in \bigcap_{a \in \mathsf{ran} R} [\![\Theta_{BD}(a)]\!]^1_a$, and for each $Q \in \mathcal{Q}'$ and $c \in C$, we have $|Q[c] \cap D| \leq 1$.

We are now ready for the proof of Proposition 5.7.6. We start by fixing a notation.

5.7.8. EXAMPLE. Let $\rho = R_0 R_1 \dots R_k$ be any stream over $A^\sharp$ such that all traces through ρ are functional. If $a \in \mathsf{dom}(R_n)$ for some $n < k$, then a has at most one R_{n+1}-successor. We denote this unique successor of a by a^+.

Proof of Proposition 5.7.6. Starting with notation, let $\mathbb{A} = (A, \Theta_A, \Omega_A, a_I)$, $\mathbb{B} = (B, \Theta_B, \Omega_B, b_I)$ and $\mathbb{D} = (D, \Theta_D, \Omega_D, d_I)$ and let $G : D \to A^\sharp$ be the map given by Definition 5.7.4 witnessing that $\mathbb{D}$ is a disjunctive companion of $\mathbb{A}$. We denote by Θ_{BD} and Θ_{BA} the transition map $\mathbb{B}[\mathbb{D}/x]$ and $\mathbb{B}[\mathbb{A}/x]$ respectively.

Our goal is to provide a winning strategy χ for player II in the consequence game $\mathcal{C}(\mathbb{B}[\mathbb{D}/x], \mathbb{B}[\mathbb{A}/x])$. Using Proposition 5.7.7, it will be convenient to make the following simplifying assumptions on player I's moves.

1. CLAIM. *Without loss of generality we may assume that in any partial match ρ ending with (R, R'), player I always pick an element $\alpha = (\Upsilon, \Gamma)$ such that:*

(1) $\mathsf{dom}(Q) \subseteq \mathsf{ran}(R)$ *for all* $Q \in \mathrm{Base}(\Gamma)$,

(2) $Q \cap (D \times B) = \emptyset$ *for all* $Q \in \mathrm{Base}(\Gamma)$,

(3) $|Q[d] \cap D| \leq 1$ *for all* $Q \in \mathrm{Base}(\Gamma)$ *and* $d \in B \cup D$.

For a given relation $R \in (B \cup D)^\sharp$, let $\mathsf{ran}_B R$ denote the set $\mathsf{ran}(R) \cap B$, and let $\mathsf{ran}_D R$ denote the set $\mathsf{ran}(R) \cap D$. Furthermore, we denote the sets $R \cap (B \times B)$ and $R \cap (D \times D)$ by $\mathsf{res}_B R$ and $\mathsf{res}_D R$ respectively. We use analogous notations for a relation $R' \in (B \cup A)^\sharp$.

To appreciate the above claim, consider an arbitrary partial match

$$\rho = (R_0, R'_0) \dots (R_k, R'_k),$$

in $\mathcal{C}(\mathbb{B}[\mathbb{D}/x], \mathbb{B}[\mathbb{A}/x])$ with $R_0 = R'_0 = \{(b_I, b_I)\}$. It follows by Claim 1 that we may assume each element $c \in \mathsf{ran} R_k$ to lie on some trace through $R_0 \dots R_k$, and that every trace through $R_0 \dots R_k$ is either a $\mathbb{B}$-trace, or else it consists of an initial, non-empty $\mathbb{B}$-trace, followed by a non-empty $\mathbb{D}$-trace. By the second and third assumption of the claim, traces are *D-functional*, in the sense that if $d \in \mathsf{ran}_D R_n$ for some $n < k$, then d has at most one R_{n+1}-successor. Following Convention 5.7.8 we will denote this successor as d^+, if it exists. As a consequence, every trace τ on $R_0 \dots R_n$ ending at d has at most one continuation through $R_{n+1} \dots R_k$.

A key role in our proof is played by a ρ-induced total order on $\mathsf{ran}_D R_k$ that we will introduce now. Intuitively, we say, for $d, d' \in \mathsf{ran}_D R_k$, that d is ρ-older than d' if d lies on a trace τ that entered D at an earlier stage than any trace arriving at d'.

For the formal definition of this ordering, we fix an arbitrary injective map $\mathsf{mnt} : D \to \omega$, which we will use to break "ties", i.e., situations where the longest D-traces leading to two different states in D are of the same length. The number $\mathsf{mnt}(d)$ is called the *birth minute* of d. We let $\mathsf{tb}_\rho(d)$ be the smallest pair of natural numbers (j, l) in the lexicographic order on $\omega \times \omega$ such that there is some $e \in \mathsf{ran}_D R_j$ with $\mathsf{mnt}(e) = l$, and such that the unique trace on $R_j \dots R_k$ beginning with e ends with d. Note that by Claim 1(1) and Claim 1(2) such an e is guaranteed to exist, and the corresponding trace is unique because of trace functionality in D. The pair $\mathsf{tb}_\rho(d) = (j, l)$ is called the *time of birth* of d relative to the match ρ; we simply write $\mathsf{tb}(d)$ if ρ is clear from context.

Note that tb_ρ is always an injective map. To see this, suppose that $\mathsf{tb}_\rho(d) = \mathsf{tb}_\rho(d') = (j, l)$. Then there are $e, e' \in \mathsf{ran}_D R_j$ such that the unique trace on $R_j \dots R_k$ beginning with e ends with d, and the unique trace beginning with e' ends with d', and such that $\mathsf{mnt}(e) = \mathsf{mnt}(e') = l$. By injectivity of mnt, we get $e = e'$, and so we get $d = d'$ by uniqueness of traces in the $\mathbb{D}$-part of $R_0 \dots R_k$.

Finally, we define a strict total ordering $\lhd_\rho$ on $\mathsf{ran}_D R_k$ relative to ρ by saying that d is *ρ-older* than d' (denoted by $d \lhd_\rho d'$) if $\mathsf{tb}_\rho(d)$ is smaller than $\mathsf{tb}_\rho(d')$ (in the lexicographic order). We may drop the reference to the match ρ when this is clear from context. We leave it for the reader to verify that, for $d \in \mathsf{ran} R_n$ with $n < k$, it holds that $\mathsf{tb}(d^+) \leq \mathsf{tb}(d)$.

We now turn to player II's winning strategy χ. By a simultaneous induction on the length of a partial χ-guided match

$$\rho = (R_0, R'_0) \dots (R_k, R'_k),$$

with $R_0 = R'_0 = \{(b_I, b_I)\}$, we will define maps

$$F_n : (B \cup D)^\sharp \to (B \cup A)^\sharp,$$

and

$$g_n : \mathsf{ran}_A R'_n \to \mathsf{ran}_D R_n.$$

We let the F-maps determine player II's strategy in the following sense. Suppose that in the mentioned partial play ρ, player I legitimately picks an element $\alpha = (\Upsilon, \Gamma) \in \mathsf{T}_{\mathtt{X}}(B \cup D)^\sharp$. Then player II's response will be the map $F_{n+1}\restriction_{\mathcal{R}}$, that is the map F_{n+1}, restricted to the set $\mathcal{R} := \text{Base}(\Gamma) \subseteq (B \cup D)^\sharp$, such that

$$\mathsf{T}_{\mathtt{X}}(F_{n+1}\restriction_{\mathcal{R}})(\alpha) \in \bigcap_{b \in \mathsf{ran}(R')} [\![\Theta_{BA}(b)]\!]^1_b.$$

Inductively we will ensure that the following conditions are maintained:

(∗) $F_n R_n = R'_n$, for all $n \geq 0$,

(†0) $R'_n = \mathsf{res}_B R'_n \cup (R'_n \cap (B \times A)) \cup \mathsf{res}_A R'_n$,

(†1) $\mathsf{res}_B R'_n \subseteq \mathsf{res}_B R_n$,

(†2) $R'_n \cap (B \times A) \subseteq \bigcup_{d \in D} \{(b, a) \mid (b, d) \in R_n \text{ and } (a_I, a) \in Gd\}$,

(†3) $\mathsf{res}_A R'_n \subseteq \bigcup \{Gd \mid d \in \mathsf{ran} R_n \cap D\}$,

(††) $a \in \mathsf{ran} G(g_n(a))$ for all $a \in \mathsf{ran} R'_n \cap A$.

For some explanation and motivation of these conditions, observe that (*) indicates that ρ itself is indeed χ-guided. For conditions (†0) to (†3) which we may refer to as condition (†), first observe that while by Claim 1, all $\mathbb{B}[\mathbb{D}/x]$-traces consist of an initial $\mathbb{B}$-part followed by an $\mathbb{D}$-tail, condition (†0) implies that similarly, all $\mathbb{B}[\mathbb{A}/x]$-traces consist of an initial $\mathbb{B}$-part followed by an $\mathbb{A}$-tail. Condition (†1) then implies that the $\mathbb{B}$-part on the left and right side of a $\mathcal{C}(\mathbb{B}[\mathbb{D}/x], \mathbb{B}[\mathbb{A}/x])$-match is the same, and condition (†3) implies that every pair $(a, b) \in \mathsf{res}_A \mathsf{ran} R'_n$ is 'covered' or 'implied' by some $d \in \mathsf{ran}_D R_n$. Finally, (††) states that, for every $a \in \mathsf{ran} R'_n$, the map g_n picks a specific element $d \in \mathsf{ran}_D R_n$ such that $a \in \mathsf{ran}(Gd)$. As we will see in Claim 4 below, it will be this condition, together with the condition on the reflection of traces in Definition 5.7.4 and the actual definition of the maps g_n, that is pivotal in proving that player II wins all infinite matches.

Setting up the induction, observe that $R_0 = R'_0 = \{(b_I, b_I)\}$. Defining F_0 as the map $R \mapsto \mathsf{res}_B R$ and g_0 as the empty map, we can easily check that (*), (†) and (††) hold.

In the inductive case we will define the maps F_{n+1} and g_{n+1} for a partial match $\rho = (R_0, R'_0) \ldots (R_n, R'_n)$ as above. For the definition of $F_{n+1} : (B \cup D)^\sharp \to (B \cup A)^\sharp$, first observe that that by (†0) we are only interested in relations $R \in (B \cup D)^\sharp$ that are of the form $R = \mathsf{res}_B R \cup (R \cap (B \times D)) \cup \mathsf{res}_D R$. We will define F_{n+1} by treating these three parts of R separately, using, respectively, the identity map on $B^\sharp$ and two auxiliary maps that we define now.

For the D-part of R, we define an auxiliary map $H_{n+1} : D \times D \to A^\sharp$:

$$H_{n+1} : (d, d') \mapsto \{(a, a') \in G(d') \mid d = g_n(a)\}.$$

For the $B \times D$-part of R, we need a second auxiliary map $L : B \times D \to \mathcal{P}(B \times A)$, given by

$$L(b, d) := \{(b, a) \in B \times A \mid (a_I, a) \in G(d)\}.$$

We now define the map $F_{n+1} : (B \cup D)^\sharp \to (B \cup A)^\sharp$ by

$$\begin{array}{rcl} F_{n+1}(R) & := & \mathsf{res}_B R \\ & \cup & \bigcup\{L(b, d) \mid (b, d) \in R \cap (B \times D)\} \\ & \cup & \bigcup\{H_{n+1}(d, d^+) \mid (d, d^+) \in \mathsf{res}_D R\}. \end{array}$$

That is, we define $F_{n+1}(R)$ as the union of three disjoint parts: a $B \times B$-part, a $B \times A$-part and an $A \times A$-part.

For the definition of g_{n+1}, consider an arbitrary position (R_{n+1}, R'_{n+1}) following the partial play ρ. Note that we may assume that R_{n+1} satisfies the assumptions formulated in Claim 1, and that we have $R'_{n+1} = F_{n+1}(R_{n+1})$ by the fact that player II's strategy is given by the map F_{n+1}. Given $a \in \mathsf{ran}_A R'_{n+1}$, distinguish cases:

Case 1 If a does have an R'_{n+1}-predecessor in A, that is, the set $\{b \in A \mid (b, a) \in R'_{n+1}\}$ is non-empty, we can define $g_{n+1}a$ to be the oldest element (with respect to the match $\rho \cdot (R_{n+1}, R'_{n+1}))$ of the set $\{(g_n b)^+ \mid (b, a) \in R'_{n+1}\} \subseteq D$. Note that this set is indeed non-empty, by definition of F_{n+1} and g_{n+1} is well-defined.

Case 2 If a has no R'_{n+1}-predecessor in A, then by definition of F_{n+1} and L, the set of states $d \in D$ for which there is a $b \in B$ with $(b, d) \in R_{n+1}$ and $(a_I, a) \in Gd$ is non-empty. We define $g_{n+1}a$ to be the *oldest* element of this set, that is, in this case, the element with the earliest birth minute.

To gain some intuitions concerning this definition, observe that in the second case, we cannot define $g_{n+1}a$ inductively on the basis of the map g_n applied to an R'_{n+1}-predecessor of a: we have to start from scratch. This case only applies, however, in a situation where a does have an R'_{n+1}-successor $b \in B$ such that in R_{n+1}, this same b has an R_{n+1}-successor $d \in D$ such that $(a_I, a) \in Gd$. In this case we simply define $g_{n+1}a := d$, and if there are more such pairs (b, d), then for $g_{n+1}a$ we may pick any of these d's, for instance the one with the earliest birth minute.

We now turn to the first clause of the definition of g_{n+1} — here lies, in fact, the heart of the proof of Proposition 5.7.6. Consider a situation where a_0 and a_1, both in A, are the two R_{n+1}-predecessors of $a \in A$. Both $g_n a_0$ and $g_n a_1$ are states in D, and therefore their R_{n+1}-successors in D, if existing, are unique, and will be denoted by $(g_n a_0)^+$ and $(g_n a_1)^+$, respectively. We want to define $g_{n+1}a$ as either $(g_n a_0)^+$ or $(g_n a_1)^+$, but then we are facing a *choice* between these two states of D in case they are *distinct*. It is here that our play-dependent ordering of states in D comes in: we will define $g_{n+1}a$ as the *oldest* element of the two, relative to the (extended) play $\rho \cdot (R_{n+1}, R'_{n+1})$. Suppose (without loss of generality) it holds that $(g_n a_0)^+$ is older than $(g_n a_1)^+$, so that we put $g_{n+1}a := (g_n a_0)^+$. In this case we say that the trace through $g_n a_0$ is *continued*, while there is also a *trace jump* witnessed by the fact that $(a_1, a) \in R'_{n+1}$ but $(g_n a_1, g_{n+1}a) \notin R_{n+1}$ (see Figure 5.2, where the dashed lines represent the g-maps, and the partial trace of white points on the right is not mapped to a partial trace on the left, due to a trace jump).

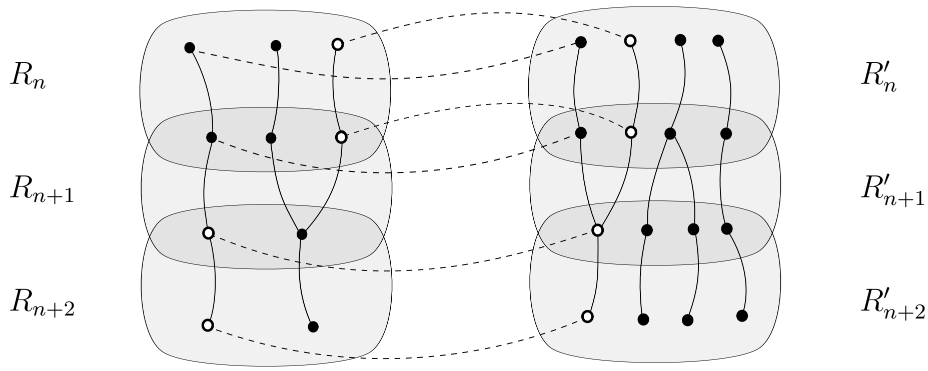


Figure 5.2: A trace merge results in a trace jump.

2. Claim. *By playing according to the strategy χ, player II indeed maintains the conditions* (∗), (†) *and* (††).

Proof of Claim 2:[1]
Let ρ be a partial χ-play satisfying the conditions (*), (†) and (††), and let $(R_{n+1}, R'_{n+1}) \in \mathrm{Gr}(F_{n+1})$ be any possible next position. It suffices to show that (R_{n+1}, R'_{n+1}) also satisfies (*), (†) and (††).

[1]The proof of this Claim is verbatim the same as that of Claim 2 in the proof of Proposition 8.6 in [ESV19].

The conditions (*), (†0), (†1) and (†2) are direct consequences of the definition of F_{n+1}, while (†3) is immediate by the fact that

$$(b, a) \in F_{n+1}R_{n+1} \Longleftrightarrow (b, a) \in G((g_n b)^+), \tag{5.10}$$

for all $b, a \in A$. To prove (5.10), consider the following chain of equivalences, which hold for all $b, a \in A$:

$$\begin{aligned} (b, a) \in F_{n+1}R_{n+1} &\overset{(\text{Def. } F_{n+1})}{\Longleftrightarrow} (b, a) \in H_{n+1}(d, d^+), \text{ some } (d, d^+) \in \mathsf{res}_D R_n \\ &\overset{(\text{Def. } H_{n+1})}{\Longleftrightarrow} (b, a) \in G(d^+), \text{ some } (d, d^+) \in \mathsf{res}_D R_n \text{ with } d = g_n b \\ &\overset{(\text{obvious})}{\Longrightarrow} (b, a) \in G((g_n b)^+). \end{aligned}$$

Finally, for condition (††), let $a \in \mathsf{ran}_A R'_{n+1}$ be arbitrary. If a has an R'_{n+1}-predecessor in A, then we are in case 1 of the definition of $g_{n+1}a$, where $g_{n+1}a$ is of the form $(g_n b)^+$ for some b with $(b, a) \in \mathsf{res}_A R'_{n+1}$. But then $(b, a) \in G((g_n b)^+)$ by (5.10), so that indeed we find $a \in \mathsf{ran} G(g_{n+1}a)$. If, on the other hand, a has no R'_{n+1}-predecessor in A, then we are in case 2 of the definition of $g_{n+1}a$. In this case, $g_{n+1}a$ is an element of a set, each of whose elements d satisfies $a \in \mathsf{ran} G(d)$; so we certainly have $a \in \mathsf{ran} G(g_{n+1}a)$.

3. Claim. *The moves for Player II prescribed by the strategy χ are legitimate.*

Proof of Claim 3:[2] Let Θ_{BD} and Θ_{BA} denote the transition maps of the automata $\mathbb{B}[\mathbb{D}/x]$ and $\mathbb{B}[\mathbb{A}/x]$, respectively. Consider a partial match ρ ending with the position (R_n, R'_n) and a subsequent move $\gamma = (\Upsilon, \Gamma) \in \mathsf{T}_{\mathtt{X}}(B \cup D)^\sharp$ by player I such that

$$(B \cup D)^\sharp, \gamma, n_e^{B \cup D} \Vdash^1 \Theta_{BD}(e), \tag{5.11}$$

for all $e \in \mathsf{ran} R_n$. By naturality of one-step formulas (Proposition 5.3.6), in order to prove the claim it suffices to show that, for an arbitrary element $c \in \mathsf{ran} R'_n = \mathsf{ran}(F_{n+1}R_n)$, we have

$$(B \cup A)^\sharp, \mathsf{T}_{\mathtt{X}} F_{n+1}(\gamma), n_c^{B \cup A} \Vdash^1 \Theta_{BA}(c). \tag{5.12}$$

But since $c \in B \cup A$ by definition of $\mathbb{B}[\mathbb{A}/p]$, one of the following two cases applies:

Case 1: $c \in A$. Then by (††) we find $c \in \mathsf{ran}(G(d))$, where $d := g_n(c)$ belongs to $\mathsf{ran}_D R_n$. As an immediate consequence of (5.11) and the fact that $\Theta_{BD}(d) = \Theta_D(d)$, we find

$$(B \cup D)^\sharp, \gamma, n_d^{B \cup D} \Vdash^1 \Theta_D(d), \tag{5.13}$$

[2] The proof of this Claim is verbatim the same as that of Claim 3 in the proof of Proposition 8.6 in [ESV19].

from which it follows by naturality that

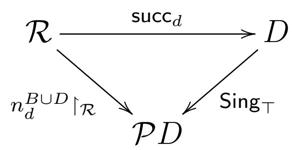
$$\mathcal{R}, \gamma, n_d^{B \cup D}\restriction_{\mathcal{R}} \Vdash^1 \Theta_D(d). \tag{5.14}$$

with $\mathcal{R} := \text{Base}(\Gamma)$.

Let the map $\mathsf{succ}_d : \mathcal{R} \to D$ be given by

$$\mathsf{succ}_d(Q) := \begin{cases} e & \text{if } Q[d] = \{e\}, \\ d_\top & \text{if } Q[d] = \emptyset. \end{cases}$$

Observe that this provides a well-defined (total) map by (Ass3) in Claim 1, and an easy calculation reveals that the diagram below commutes:



so that we may conclude that succ_d is a one-step model morphism:

$$\mathsf{succ}_d : (\mathcal{R}, \gamma, n_d^{B \cup D}\restriction_{\mathcal{R}}) \to (\mathsf{succ}_d[\mathcal{R}], (\mathsf{T_x}\mathsf{succ}_d)(\gamma), \mathsf{Sing}_\top).$$

From this, (5.14), the fact that $\Theta_D(d)$ is a one-step formula in D, and Corollary 5.3.7 we conclude that

$$D, (\mathsf{T_x}\mathsf{succ}_d)\gamma, \mathsf{Sing}_\top \Vdash^1 \Theta_D(d). \tag{5.15}$$

Now we may use the assumption that $\mathbb{D}$ is a disjunctive companion of $\mathbb{A}$, obtaining from clause (DC2) that

$$A^\sharp, (\mathsf{T_x}G)(\mathsf{T_x}\mathsf{succ}_d)(\gamma), n_c^A \Vdash^1 \Theta_A(c). \tag{5.16}$$

By functoriality of $\mathsf{T_x}$ and the fact that $\Theta_A(c) = \Theta_{BA}(c)$, this is equivalent to

$$A^\sharp, (\mathsf{T_x}(G \circ \mathsf{succ}_d))(\gamma), n_c^A \Vdash^1 \Theta_{BA}(c), \tag{5.17}$$

and so by Corollary 5.3.7 and Proposition 5.3.8 we obtain

$$(B \cup A)^\sharp, (\mathsf{T_x}(G \circ \mathsf{succ}_d))(\gamma), n_c^{B \cup A} \Vdash^1 \Theta_{BA}(c). \tag{5.18}$$

From here on for conciseness we will write n_c for $n_c^{B \cup A}$. We now claim that, comparing the two A-markings $n_c \circ (G \circ \mathsf{succ}_d)$ and $n_c \circ F_{n+1}$, we have

$$\big(n_c \circ (G \circ \mathsf{succ}_d)\big)(Q) \subseteq \big(n_c \circ F_{n+1}\big)(Q) \tag{5.19}$$

for all $Q \in \mathcal{R}$. To see this, recall that $d = g_n(c)$. Now assume that $a \in \big(n_c \circ (G \circ \mathsf{succ}_d)\big)(Q)$, that is, $(c, a) \in G(\mathsf{succ}_d(Q))$. Observe that since $G(d_\top) = \varnothing$ by (DC1), by definition of the map succ_d it must be the case that $\mathsf{succ}_d(Q) = e$

for some unique $e = d_Q^+ \in D$ such that $Q[d] = \{d_Q^+\}$. Then (c, a) belongs to $H_{n+1}(d, d_Q^+)$ by definition of H_{n+1}, and to $F_{n+1}Q$ by definition of F_{n+1}. But from $(c, a) \in F_{n+1}(Q)$ we immediately obtain $a \in \big(n_c \circ F_{n+1}\big)(Q)$. This proves (5.19).

We use this observation in the following line of reasoning, where the key observation is that in fact both maps $G \circ \mathsf{succ}_d$ and F_{n+1} are one-step model morphisms.

$$
\begin{array}{cl}
(B \cup A)^\sharp, (\mathsf{T}_{\mathtt{X}}(G \circ \mathsf{succ}_d))\gamma, n_c \Vdash^1 \Theta_{BA}(c) & \\
\Updownarrow & \text{(Prop. 5.3.6)} \\
(B \cup A)^\sharp, \gamma, n_c \circ (G \circ \mathsf{succ}_d) \Vdash^1 \Theta_{BA}(c) & \\
\Updownarrow & \text{(Corollary 5.3.7)} \\
\mathcal{R}, \gamma, \big(n_c \circ (G \circ \mathsf{succ}_d)\big)\restriction_{\mathcal{R}} \Vdash^1 \Theta_{BA}(c) & \\
\Downarrow & \text{((5.19), Prop. 5.3.10)} \\
\mathcal{R}, \gamma, \big(n_c \circ F_{n+1}\big)\restriction_{\mathcal{R}} \Vdash^1 \Theta_{BA}(c) & \\
\Updownarrow & \text{(Corollary 5.3.7)} \\
(B \cup A)^\sharp, \gamma, n_c \circ F_{n+1} \Vdash^1 \Theta_{BA}(c) & \\
\Updownarrow & \text{(Prop. 5.3.6)} \\
(B \cup A)^\sharp, (\mathsf{T}_{\mathtt{X}} F_{n+1})(\gamma), n_c \Vdash^1 \Theta_{BA}(c). &
\end{array}
$$

This proves (5.12), as required.

Case 2 $c \in B$. Note that in this case we have $\Theta_{BA}(c) = \Theta_B(c)[\Theta_A(a_I)/x]$ and $\Theta_{BD}(c) = \Theta_B(c)[\Theta_D(d_I)/x]$. Thus by assumption we know that $(B \cup D)^\sharp, \gamma, n_c^{B \cup D} \Vdash^1 \Theta_B(c)[\Theta_D(d_I)/x]$, while we need to establish that

$$(B \cup A)^\sharp, \mathsf{T}_{\mathtt{X}} F_{n+1}(\gamma), n_c^{B \cup A} \Vdash^1 \Theta_B(c)[\Theta_A(a_I)/x].$$

To achieve this it clearly suffices to show that

$$(B \cup D)^\sharp, \gamma, n_c^{B \cup D} \Vdash^1 \alpha[\Theta_D(d_I)/x] \text{ implies } (B \cup A)^\sharp, \mathsf{T}_{\mathtt{X}} F_{n+1}(\gamma), n_c^{B \cup A} \Vdash^1 \alpha[\Theta_A(a_I)/x] \tag{5.20}$$

for all $\alpha \in \mathtt{1ML}_\nabla^+(\mathtt{X}, B)$. We will prove (5.20) by induction on the one-step formula α, taken as a lattice term over the set $\{x\} \cup \mathtt{1ML}_\nabla^+(\mathtt{X} \setminus \{x\}, B)$. This perspective allows us to distinguish the following two cases in the induction base.

Base Case a: $\alpha = x$. Here we find $\alpha[\Theta_D(d_I)/x] = \Theta_D(d_I)$ and $\alpha[\Theta_A(a_I)/x] = \Theta_A(a_I)$. In other words, in order to prove (5.20) we assume that

$$(B \cup D)^\sharp, \gamma, n_c^{B \cup D} \Vdash^1 \Theta_D(d_I), \tag{5.21}$$

and we need to show that

$$(B \cup A)^\sharp, \mathsf{T}_{\mathtt{X}} F_{n+1}(\gamma), n_c^{B \cup A} \Vdash^1 \Theta_A(a_I). \tag{5.22}$$

Our line of reasoning here will be close to that in Case 1, and for this reason we are a bit more sketchy. By (Ass3) we may define a map $\mathsf{succ}_c : \mathcal{R} \to D$ by setting $\mathsf{succ}_c(Q)$ to be the unique Q-successor of c if $Q[c]$ is nonempty, and the true state $d_\top$ otherwise. As in Case 1 this map is a one-step morphism of models:

$$\mathsf{succ}_c : (\mathcal{R}, \gamma, n_c^{B\cup D}{\restriction_{\mathcal{R}}}) \to (D, \mathsf{T}_\mathtt{X}\mathsf{succ}_c(\gamma), \mathsf{Sing}_\top). \tag{5.23}$$

We also claim that our definition of the map F_{n+1} has been tailored towards the following inclusion:

$$\big(n_{a_I}^{B\cup A} \circ (G \circ \mathsf{succ}_c)\big)(Q) \subseteq \big(n_c^{B\cup A} \circ F_{n+1}\big)(Q) \tag{5.24}$$

for all $Q \in \mathcal{R}$. For a proof of (5.24), assume that $a \in \big(n_{a_I}^{B\cup A} \circ (G \circ \mathsf{succ}_c)\big)(Q)$ for some $Q \in \text{Base}(\Gamma)$. In other words, we have $(a_I, a) \in G(\mathsf{succ}_c(Q))$, and so by definition of succ_c there is a unique $d \neq d_\top \in D$ such that $(c, d) \in Q$. But then we obtain $(a_I, a) \in L(b, d)$ by definition of the map L, and since $(c, d) \in Q \cap (B \times D)$ this gives $(c, a) \in F_{n+1}Q$ by definition of F_{n+1}. But from $(c, a) \in F_{n+1}Q$ we directly see that $a \in n_c^{B\cup A}(F_{n+1}Q)$, as required. This proves (5.24).

We can now show how to prove (5.22) from (5.21):

$$
\begin{array}{cl}
(B \cup D)^\sharp, \gamma, n_c^{B\cup D} \Vdash^1 \Theta_D(d_I) & \\
\Updownarrow & \text{(Corollary 5.3.7)} \\
\mathcal{R}, \gamma, n_c^{B\cup D}{\restriction_{\mathcal{R}}} \Vdash^1 \Theta_D(d_I) & \\
\Updownarrow & \text{(Prop. 5.3.6, (5.23))} \\
D, (\mathsf{T}_\mathtt{X}\mathsf{succ}_c)(\gamma), \mathsf{Sing}_\top \Vdash^1 \Theta_D(d_I) & \\
\Downarrow & \text{(DC1,DC2)} \\
A^\sharp, (\mathsf{T}_\mathtt{X}G)((\mathsf{T}_\mathtt{X}\mathsf{succ}_c(\gamma)), n_{a_I}^A \Vdash^1 \Theta_A(a_I) & \\
\Updownarrow & \text{(functoriality)} \\
A^\sharp, (\mathsf{T}_\mathtt{X}(G \circ \mathsf{succ}_c))(\gamma), n_{a_I}^A \Vdash^1 \Theta_A(a_I) & \\
\Updownarrow & \text{(as in Case 1)} \\
(B \cup A)^\sharp, (\mathsf{T}_\mathtt{X}(G \circ \mathsf{succ}_c))(\gamma), n_{a_I}^{B\cup A} \Vdash^1 \Theta_A(a_I) & \\
\Downarrow & \text{(as in Case 1, by (5.24))} \\
(B \cup A)^\sharp, (\mathsf{T}_\mathtt{X}F_{n+1})(\gamma), n_c^{B\cup A} \Vdash^1 \Theta_A(a_I). &
\end{array}
$$

Base Case b: $\alpha \in \mathtt{1ML}^{+}_{\nabla}(\mathtt{X}\backslash\{x\}, B)$, that is, α is a x-free one-step formula over B. In this case the proof of (5.20) is straightforward: clearly the substitutions in (5.20) have no effect, so what we have to prove is that

$$(B \cup D)^\sharp, \gamma, n_c^{B\cup D} \Vdash^1 \alpha \text{ implies } (B \cup A)^\sharp, (\mathsf{T}_\mathtt{X}F_{n+1})(\gamma), n_c^{B\cup A} \Vdash^1 \alpha. \tag{5.25}$$

But intuitively this is clear, since α only uses variables from B, and 'when restricted to B', the two models in (5.25) are the same.

Formally, our proof of (5.25) proceeds as follows:

$$
\begin{array}{cl}
(B \cup D)^\sharp, \gamma, n_c^{B \cup D} \Vdash^1 \alpha & \\
\Updownarrow & \text{(Proposition 5.3.8)} \\
(B \cup D)^\sharp, \gamma, n_c^B \circ \mathsf{res}_B \Vdash^1 \alpha & \\
\Updownarrow & \text{(Prop. 5.3.6)} \\
B^\sharp, (\mathsf{T_x res}_B)(\gamma), n_c^B \Vdash^1 \alpha & \\
\Updownarrow & (\dagger 1) \\
B^\sharp, (\mathsf{T_x}(\mathsf{res}_B \circ F_{n+1}))(\gamma), n_c^B \Vdash^1 \alpha & \\
\Updownarrow & \text{(functoriality)} \\
B^\sharp, (\mathsf{T_x res}_B)((\mathsf{T_x} F_{n+1})(\gamma)), n_c^B \Vdash^1 \alpha & \\
\Updownarrow & \text{(Prop. 5.3.6)} \\
(B \cup A)^\sharp, (\mathsf{T_x} F_{n+1})(\gamma), n_c^B \circ \mathsf{res}_B \Vdash^1 \alpha & \\
\Updownarrow & \text{(Prop. 5.3.8)} \\
(B \cup A)^\sharp, (\mathsf{T_x} F_{n+1})(\gamma), n_c^{B \cup A} \Vdash^1 \alpha &
\end{array}
$$

Inductive case: The inductive cases in the proof of (5.20), where α is of the form $\alpha_0 \vee \alpha_1$ or $\alpha_0 \wedge \alpha_1$, are trivial.

To finish the proof of Proposition 5.7.6 we need to show that χ is a winning strategy. Since it is clear that player II never gets stuck playing the strategy χ, it is enough to show that all infinite χ-guided matches are won by player II.

4. Claim. *Suppose ρ is an infinite χ-guided match with basic positions*

$$(R_0, R_0')(R_1, R_1')(R_2, R_2')...$$

If there is a bad trace on $R_0' R_1' R_2'...$, there is also a bad trace on $R_0 R_1 R_2....$

Proof of Claim 4:
For the proof of this claim, fix a χ-guided match ρ as above. There are two possibilities for a trace τ on $R_0' R_1' R_2'...$: either τ stays entirely in B, or from some finite stage onwards τ stays entirely in A. Since the first case is easy we focus on the second: so suppose τ is an infinite trace of the form

$$b_0 b_1 ... b_n a_{n+1} a_{n+2} a_{n+3} ...,$$

where the b_j are all in B, and the a_i are all in A. It now suffices to prove the following:

($\star$) There exists an index k such that for all $j \geq k$, we have
$$g_{j+1}(a_{j+1}) = (g_j(a_j))^+.$$

Before we prove ($\star$), let us see how it entails Claim 4. Suppose there exists such an index k, and consider $g_k(a_k) \in \mathsf{ran}_D R_k$. Pick an arbitrary initial trace $b_0...b_n d_{n+1}...d_k$ of $R_0...R_k$ such that $g_k(a_k) = d_k$ (as mentioned already after Claim 1, the existence of such a trace follows from our assumptions on player I's strategy). Then the stream

$$b_0...d_{k-1}g_k(a_k)g_{k+1}(a_{k+1})g_{k+2}(a_{k+2})...,$$

is a trace of $R_0R_1R_2...$ by the property of the index k described in ($\star$). Furthermore, it follows that $a_k a_{k+1} a_{k+2}....$ is a trace of the stream

$$G(g_k(a_k))G(g_{k+1}(a_{k+1}))G(g_{k+2}(a_{k+2}))...$$

To see why, consider the pair (a_j, a_{j+1}) where $j \geq k$. Then $(a_j, a_{j+1}) \in R'_{j+1} = F_k(R_{j+1})$, so there is some $(d, d') \in R_{j+1}$ with $(a_j, a_{j+1}) \in H_{j+1}(d, d')$. Hence $d = g_j(a_j)$ and $(a_j, a_{j+1}) \in G(d')$.

But $d' = d^+$ by functionality of traces on D (which follows from the third assumption in Claim 1), and so we find $d' = d^+ = (g_j a_j)^+ = g_{j+1}a_{j+1}$. From this we get $(a_j, a_{j+1}) \in G(g_{j+1}a_{j+1})$ as required. Note too that $a_k a_{k+1} a_{k+2} \ldots$ has the same tail as τ, and hence it is a *bad* trace too. It now follows from the trace reflection clause (DC3) of Definition 5.7.4 that $g_k a_k, g_{k+1}a_{k+1}, g_{k+2}a_{k+2}, \ldots$ is itself a bad trace, and so we have found a bad trace on $R_0R_1R_2\ldots$ as required.

We now prove ($\star$). Say that a *trace jump* occurs at the index $j \geq k$ if we have $g_{j+1}(a_{j+1}) \neq g_j(a_j)^+$. We want to show that there can only be finitely many j at which a trace jump occurs. Since, clearly, if no trace jump occurs at j, then

$$\mathsf{tb}(g_j(a_j)) \geq \mathsf{tb}(g_j(a_j)^+) = \mathsf{tb}(g_{j+1}(a_{j+1})),$$

it suffices to prove that if a trace jump occurs at j then $\mathsf{tb}(g_{j+1}(a_{j+1}))$ is strictly smaller than $\mathsf{tb}(g_j(a_j))$ in the lexicographic order. It follows that the stream

$$\mathsf{tb}(g_k(a_k))\mathsf{tb}(g_{k+1}(a_{k+1}))\mathsf{tb}(g_{k+2}(a_{k+2}))...,$$

is a stream of pairs of natural numbers that never increases, and strictly decreases at each j at which a trace jump occurs. By well-foundedness of the lexicographic order on $\omega \times \omega$ this can therefore only happen finitely many times, as required.

So, we are left with the task of proving that tb is strictly decreasing at each index j for which a trace jump occurs. To see this is indeed so, suppose that $g_{j+1}(a_{j+1}) \neq g_j(a_j)^+$. Recall that by Case 1 of definition of g_{n+1}, we define $g_{j+1}(a_{j+1})$ to be the oldest element of the following set

$$\{(g_j(c))^+ \mid (c, a_{j+1}) \in R'_{j+1})\}.$$

But since $(a_j, a_{j+1}) \in R'_{j+1}$, it follows that $g_{j+1}(a_{j+1})$ must be older than $(g_j(a_j))^+$, with respect to the order induced by the match $(R_0, R'_0), \ldots, (R_{j+1}, R'_{j+1})$, and so $\mathsf{tb}(g_{j+1}(a_{j+1}))$ must be strictly smaller than $\mathsf{tb}((g_j(a_j))^+) \leq \mathsf{tb}(g_j(a_j))$, as required. This completes the proof of $(\star)$. □

Finally, the proof of of Proposition 5.7.6 is immediate by the last two claims: it follows from Claim 3 that player II never gets stuck, so that we need not to worry about finite plays. But Claim 4 states that II wins all infinite plays of $\mathbb{B}[\mathbb{D}/x] \vDash_{\mathcal{C}} \mathbb{B}[\mathbb{A}/x]$ as well. □

5.8 From automata to formulas

In section 5.3.4, we defined a translation from formulas to modal automata by induction on the complexity of formulas, and based on operations on automata corresponding to the Boolean connectives, modalities and fixpoint operators. In this section[3], we provide a translation tr in the opposite direction, that is, from automata to formulas, and we establish some properties of this translation. Our definition of the translation map is based on a more or less standard induction on the complexity of the automaton [GTW02].

The main purpose of this translation is to prove the following proposition:

5.8.1. Proposition. *For every formula φ, we have $\varphi \equiv_{\mathbf{K}} \mathsf{tr}(\mathbb{A}_\varphi)$.*

The proof of this proposition will proceed by induction on the complexity of formulas. As a central auxiliary result (Lemma 5.8.16 below) we will show that the translation commutes with the logical operations on automata and formulas, and with the operation of substitution.

The key point about the translation tr and Proposition 5.8.1 is that, allowing us to apply proof-theoretic notions such as derivability or consistency to automata, they open the door to proof theory for automata.

5.8.2. Definition. A modal automaton $\mathbb{A}$ will be called *consistent* if the formula $\mathsf{tr}(\mathbb{A})$ is consistent. Given two modal automata $\mathbb{A}$ and $\mathbb{B}$, we say that $\mathbb{A}$ provably implies $\mathbb{B}$, notation: $\mathbb{A} \leq_{\mathbf{K}} \mathbb{B}$, if $\mathsf{tr}(\mathbb{A}) \leq_{\mathbf{K}} \mathsf{tr}(\mathbb{B})$, and that $\mathbb{A}$ and $\mathbb{B}$ are provably equivalent if $\mathsf{tr}(\mathbb{A}) \equiv_{\mathbf{K}} \mathsf{tr}(\mathbb{B})$. We will use similar notation and terminology relating formulas and automata, for instance we will say that φ *provably implies* $\mathbb{A}$ and write $\varphi \leq_{\mathbf{K}} \mathbb{A}$ if $\varphi \leq_{\mathbf{K}} \mathsf{tr}(\mathbb{A})$, etc.

In order to provide the translation $\mathsf{tr}(\mathbb{A})$ of an automaton $\mathbb{A}$, we first define a map $\mathtt{tr}_{\mathbb{A}}$ assigning a formula to *each* state of $\mathbb{A}$. The formula $\mathsf{tr}(\mathbb{A})$ is then obtained by applying the map $\mathtt{tr}_{\mathbb{A}}$ to the initial state of $\mathbb{A}$. For a proper inductive

[3]This section is verbatim the same as that of Chapter 8 in [ESV18], except for minor modifications concerning the ∇ modality in a more general coalgebraic setting.

formulation of this definition it is convenient to extend the class of automata, allowing states of the automaton to appear in the scope of a modality in a one-step formula.

5.8.3. Definition. A *generalized modal automaton* is a structure $\mathbb{A} = (A, \Theta, \Omega, a_I)$ where A, Ω and a_I are as in the definition of standard modal automata, and the transition map Θ is of type $\Theta : A \to \mathtt{1ML}^{+}_{\nabla}(\mathtt{X}, A \cup \mathtt{X})$.

The notion of acceptance for generalized automata is a straightforward generalization of the one for standard modal automata. For completeness we provide a definition here — one that stays close to our approach in terms of one-step models is the following.

5.8.4. Definition. A *generalized one-step* $\mathsf{T}_{\mathtt{X}}$*-model* is a structure $(S, \mathtt{Y}, \alpha, m)$ such that S is some set, $\mathtt{Y} : S \uplus \{\star\} \to \mathcal{P}\mathtt{X}$ is an $\mathtt{X}$-marking on the set $S \uplus \{\star\}$, $\alpha \in \mathsf{T}S$, and m is an A-marking on the set S. The one-step satisfaction relation $\Vdash^1$ for generalized one-step formulas in $\mathtt{1ML}^{+}_{\nabla}(\mathtt{X}, A \cup \mathtt{X})$ is defined in the most obvious way: we treat a generalized one-step model $(S, \mathtt{Y}, \alpha, m)$ as if it were the standard one-step $\mathsf{T}_{\mathtt{X}}$-model $(S, (\mathtt{Y}(\star), \alpha), (\mathtt{Y}{\restriction_S} \cup m))$ over $\mathtt{X} \cup A$, with $(\mathtt{Y}(\star), \alpha) \in \mathsf{T}_{\mathtt{X}}S$, and $\mathtt{Y}{\restriction_S} \cup m$ being the $(\mathtt{X} \cup A)$-marking defined by $(\mathtt{Y}{\restriction_S} \cup m)(s) := \mathtt{Y}{\restriction_S}(s) \cup m(s)$.

It will make sense to define the mentioned translation map $\mathtt{tr}_{\mathbb{A}}$ for 'uninitialized' automata, i.e., structures (A, Θ, Ω) that could be called (generalised) automata if they did not lack an initial state.

5.8.5. Definition. An *automaton structure* is a triple $\mathbb{A} = (A, \Theta, \Omega)$ such that A is a finite, non-empty set endowed with a transition map $\Theta : A \to \mathtt{ML}^{1}_{\mathsf{T}}(\mathtt{X}, A \cup \mathtt{X})$ and a priority function $\Omega : A \to \omega$.

The *underlying automaton structure* of a (generalized) modal automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$ is given as the triple $\underline{\mathbb{A}} := (A, \Theta, \Omega)$. Conversely, given an automaton structure $\mathbb{A} = (A, \Theta, \Omega)$ and a state a in A, we let $\mathbb{A}\langle a \rangle$ denote the initialized automaton (A, Θ, Ω, a).

Many concepts that we defined for automata in fact apply to automaton structures in the most obvious way, and we will use this observation without further notice.

Finally, the restriction that we announced is that for our definition of the translation map $\mathtt{tr}_{\mathbb{A}}$ we will first confine our attention to so-called *linear* automaton structures.

5.8.6. Definition. An automaton structure $\mathbb{A} = (A, \Theta, \Omega)$ will be called *linear* if the relation $\sqsubset_{\mathbb{A}}$ is a strict linear order satisfying $(\lhd_{\mathbb{A}} \setminus \rhd_{\mathbb{A}}) \subseteq \sqsubset_{\mathbb{A}}$.

Given two automaton structures $\mathbb{A} = (A, \Theta, \Omega)$ and $\mathbb{A}' = (A, \Theta, \Omega')$, we say that $\mathbb{A}'$ is a *refinement* of $\mathbb{A}$ if

(1) the partial order $\sqsubseteq_{\mathbb{A}}$ is clusterwise contained in $\sqsubseteq_{\mathbb{A}'}$, i.e., $a \bowtie b$ and $a \sqsubseteq_{\mathbb{A}} b$ imply $a \sqsubseteq_{\mathbb{A}'} b$; and

(2) $\Omega'(a')$ has the same parity as $\Omega(a)$, for all $a \in A$.

A linear refinement is called a *linearization*.

In words: linear automata structures have an injective priority map Ω, and satisfy the condition that if one state a is active in another state b, but not vice versa, then $a \sqsubset b$. In other words, the priority of states goes down if a match of the acceptance game passes from one cluster to the next. Our focus on linear automaton structures is justified by the observation that all linearizations of an automaton $\mathbb{A}$ are equivalent to $\mathbb{A}$ (and hence, to one another). In the sequel we shall need a formulation of this equivalence in terms of the consequence game.

5.8.7. PROPOSITION. *Every automaton structure $\mathbb{A}$ has a linearization $\mathbb{A}^l$ such that, for all $a \in A$*

(1) $\mathbb{A}\langle a\rangle \models_{\mathcal{C}} \mathbb{A}^l\langle a\rangle$ *and* $\mathbb{A}^l\langle a\rangle \models_{\mathcal{C}} \mathbb{A}\langle a\rangle$*;*

(2) each player $\Pi \in \{\exists, \forall\}$ *has a winning strategy in* $\mathcal{S}(\mathbb{A}\langle a\rangle)$ *(resp.* $\mathcal{S}_{thin}(\mathbb{A}\langle a\rangle)$*) iff she/he has a winning strategy in* $\mathcal{S}(\mathbb{A}^l\langle a\rangle)$ *(resp.* $\mathcal{S}_{thin}(\mathbb{A}^l\langle a\rangle)$*).*

Proof:
One may easily obtain a linearization $\mathbb{A}^l$ of $\mathbb{A}$, so it suffices to prove that the statements in (1) and (2) hold for an arbitrary refinement $\mathbb{A}^l$ of $\mathbb{A}$ and an arbitrary state a in $\mathbb{A}$. To prove (1), it is straightforward to verify that the identity map on $A^\sharp$ provides a winning strategy for player II in both $\mathcal{C}(\mathbb{A}\langle a\rangle, \mathbb{A}^l\langle a\rangle)$ and $\mathcal{C}(\mathbb{A}^l\langle a\rangle, \mathbb{A}\langle a\rangle)$. And to prove (2), it is equally straightforward to verify that a winning strategy for $\exists$ in the (thin) satisfiability game for $\mathbb{A}\langle a\rangle$ is also a winning strategy for her in the (thin) satisfiability game for $\mathbb{A}^l\langle a\rangle$, and vice versa. Part (2) then easily follows by the determinacy of the (thin) satisfiability game. □

The advantage of working with linear automaton structures is that we may define the translation map by a simple induction on the *size* of the structure. For its definition, we recall that our notation for formula substitution has been given in Definition 5.2.2.

5.8.8. DEFINITION. By induction on the size of a linear modal $\mathtt{X}$-automaton structure $\mathbb{A}$ we define a map $\mathtt{tr}_{\mathbb{A}} : A \to \mu\mathtt{ML}^{\mathtt{I}}_{\nabla}(\mathtt{X})$.

In the base case of the induction we are dealing with an automaton structure $\mathbb{A}$ based on a single state a. Then we define

$$\mathtt{tr}_{\mathbb{A}}(a) := \eta_a a.\Theta(a),$$

where $\eta_a \in \{\mu, \nu\}$.

In the inductive case, where $|\mathbb{A}| > 1$, by injectivity of Ω there is a unique state $n \in A$ that reaches the maximal priority, that is, with $\Omega(n) = \max \Omega[A]$. Let $\mathbb{A}^-$ to be the $\mathtt{X} \cup \{n\}$-automaton structure $(A^-, \Theta^-, \Omega^-)$ with

$$A^- := A \setminus \{n\}$$
$$\Theta^- := \Theta\restriction_{A^-}$$
$$\Omega^- := \Omega\restriction_{A^-}.$$

Clearly we have $|\mathbb{A}^-| < |\mathbb{A}|$, so that inductively we may assume a map $\mathtt{tr}_{\mathbb{A}^-} : A^- \to \mu\mathtt{ML}^{\mathtt{T}}_{\nabla}(\mathtt{X} \cup \{n\})$.[4]

The map $\mathtt{tr}_{\mathbb{A}}$ is now defined in two steps. First we define $\mathtt{tr}_{\mathbb{A}}(n)$ as follows:

$$\mathtt{tr}_{\mathbb{A}}(n) := \eta n.\Theta(n)[\mathtt{tr}_{\mathbb{A}^-}(a)/a \mid a \in A^-].$$

Second, by putting

$$\mathtt{tr}_{\mathbb{A}}(a) := \mathtt{tr}_{\mathbb{A}^-}(a)[\mathtt{tr}_{\mathbb{A}}(n)/n]$$

we define $\mathtt{tr}_{\mathbb{A}}(a)$ for each $a \neq n$.

We now turn to the translation map for arbitrary automaton structures. By standard order theory every automaton structure has at least one linearization. Furthermore, by the following result the translation maps of different linearizations of the same structure are provably equivalent.

5.8.9. PROPOSITION. *Let $\mathbb{A}' = (A, \Theta, \Omega')$ and $\mathbb{A}'' = (A, \Theta, \Omega'')$ be two linearizations of the automaton structure $\mathbb{A} = (A, \Theta, \Omega)$. Then*

$$\mathtt{tr}_{\mathbb{A}'}(a) \equiv_{\mathbf{K}} \mathtt{tr}_{\mathbb{A}''}(a)$$

for all $a \in A$.

Proof:
The proof of this proposition is conceptually straightforward, boiling down to the observation that, where M is the set $\{m_1, ..., m_n\}$ of maximal states of an automaton $\mathbb{A}$, it does not matter in which way we order the states in M to obtain a linearization of $\mathbb{A}$, in the sense that all choices provide provably equivalent translations. To prove this, it suffices to show that the Bekič principle holds in the class of modal μ-algebras, which can be done by verifying that the proofs in [AN01, section 1.4] in fact do not rely on completeness of the underlying lattices, but only on the existence of all least- and greatest fixpoints. □

[4]Our motivation for introducing generalized modal automata stems from the observation that $\Theta^-(a)$ generally will have guarded occurrences of m, which in $\mathbb{A}^-$ is no longer a state of the automaton but a proposition letter.

Proposition 5.8.9 ensures that modulo provable equivalence the following definition of $\mathsf{tr}(\mathbb{A})$ for an arbitrary automaton $\mathbb{A}$ does not depend on the particular choice of a linearization for the underlying automaton structure of $\mathbb{A}$.

5.8.10. DEFINITION. With each automaton structure $\mathbb{A} = (A, \Theta, \Omega)$ we associate an arbitrary but fixed linearization $\mathbb{A}^l$ of $\mathbb{A}$ (with the understanding that $\mathbb{A}^l = \mathbb{A}$ in case $\mathbb{A}$ itself is linear). We then define $\mathtt{tr}_{\mathbb{A}} := \mathtt{tr}_{\mathbb{A}^l}$.

Finally, given an arbitrary modal automaton $\mathbb{A} = (A, \Theta, \Omega, a_I)$, we let

$$\mathsf{tr}(\mathbb{A}) := \mathsf{tr}_{\mathbb{A}^l}(a_I)$$

define the translation of the automaton $\mathbb{A}$ itself.

5.8.11. REMARK. The translation given in Definition 5.8.10 is reasonably standard. A particular feature of the formula $\mathsf{tr}(\mathbb{A})$ is that it will always be *strongly guarded* in the sense that there is a modality between any two occurrences of a fixpoint operator. Its alternation depth will not exceed the maximal size of a cluster in the automaton $\mathbb{A}$.

5.8.12. REMARK. An alternative approach would be to define the translation by induction on the *index* of an automaton, i.e., the size of the range of the priority map. In this approach, one would not have a unique maximal state, but a *set* of maximal states $\{m_1, ..., m_n\}$, and the automaton structure $\mathbb{A}^-$ would remove all the maximal states. We would then get a set of "equations" $m_i = \Theta(m_i)[\mathtt{tr}_{\mathbb{A}^-}(b) \mid b \sqsubset_{\mathbb{A}} m_i]$, which is solved by a formula of the *vectorial μ-calculus* [AN01], and this formula can then be translated into the one-dimensional μ-calculus using the Bekič principle for simultaneous fixpoints.

The following lemma gives two useful representations of the translation map $\mathtt{tr}_{\mathbb{A}}$ associated with an automaton structure $\mathbb{A}$. The point of the second result is that it displays each formula $\mathtt{tr}_{\mathbb{A}}(a)$ as a fixpoint formula; this characterization will be of crucial importance in the next section. For its formulation we need to consider *restrictions* of linear automaton structures, and it is for this definition that we needed to introduce the notion of an automaton structure: initialized automata will not necessarily be closed under this operation, but automata structures are.

5.8.13. DEFINITION. Let $\mathbb{A} = (A, \Theta, \Omega)$ be a linear automaton structure, and let $a \in \mathbb{A}$. The *a-restriction* of $\mathbb{A}$ is the automaton structure $\mathbb{A}{\downarrow}a := (B, \Theta{\restriction}_B, \Omega{\restriction}_B)$ of which the carrier is given as $B := \{b \in A \mid b \sqsubseteq a\}$.

5.8.14. LEMMA. *Let $\mathbb{A}$ be any automaton structure and let $a \in A$. Then:*

$$\mathtt{tr}_{\mathbb{A}}(a) \equiv_{\mathbf{K}} \Theta(a)[\mathtt{tr}_{\mathbb{A}}(b)/b \mid b \in A]. \tag{5.26}$$

If $\mathbb{A}$ is linear, we have in addition

$$\mathtt{tr}_{\mathbb{A}}(a) \equiv_{\mathbf{K}} \eta_a a.\Theta(a)[\mathtt{tr}_{(\mathbb{A}\downarrow a)^-}(b)/b \mid b \sqsubset a][\mathtt{tr}_{\mathbb{A}}(b)/b \mid a \sqsubset b] \tag{5.27}$$

Before moving on to prove this Lemma, we quickly note that for a linear automaton structure $\mathbb{A}$, a is the maximal priority state of $\mathbb{A}\downarrow a$, so that we find

$$\mathtt{tr}_{\mathbb{A}\downarrow a}(a) = \eta_a a.\Theta(a)[\mathtt{tr}_{(\mathbb{A}\downarrow a)^-}(b)/b \mid b \sqsubset a]$$

by definition of $\mathtt{tr}_{(\mathbb{A}\downarrow a)^-}$. Hence, we may read (5.27) as stating that

$$\mathtt{tr}_{\mathbb{A}}(a) \equiv_{\mathbf{K}} \mathtt{tr}_{\mathbb{A}\downarrow a}(a)[\mathtt{tr}_{\mathbb{A}}(b)/b \mid a \sqsubset b],$$

which may be of help to understand this characterization.

Proof:
For the first part of the lemma, we reason by induction on the size of $\mathbb{A}$. By Proposition 5.8.9 we may without loss of generality assume that $\mathbb{A}$ is linear. The case for automaton structures of size 1 is simple, so we focus on the case of a structure $\mathbb{A}$ with $|\mathbb{A}| > 1$. Let n be the (by linearity unique) state that reaches the maximal priority of $\mathbb{A}$, that is, $\Omega(n) = \max \Omega[A]$. For this state n we obtain:

$$\begin{aligned}
\mathtt{tr}_{\mathbb{A}}(n) &= \eta_n n.\Theta(n)[\mathtt{tr}_{\mathbb{A}^-}(b)/b \mid b \sqsubset n] && \text{(Definition } \mathtt{tr}_{\mathbb{A}}\text{)}\\
&\equiv_{\mathbf{K}} \Theta(n)[\mathtt{tr}_{\mathbb{A}^-}(b)/b \mid b \sqsubset n][\mathtt{tr}_{\mathbb{A}}(n)/n] && \text{(fixpoint logic)}\\
&= \Theta(n)[\mathtt{tr}_{\mathbb{A}^-}(b)[\mathtt{tr}_{\mathbb{A}}(n)/n]/b \mid b \sqsubset n, \mathtt{tr}_{\mathbb{A}}(n)/n] && \text{(Fact 5.2.3)}\\
&= \Theta(n)[\mathtt{tr}_{\mathbb{A}}(b)/b \mid b \sqsubset n, \mathtt{tr}_{\mathbb{A}}(n)/n] && \text{(Definition } \mathtt{tr}_{\mathbb{A}}\text{)}\\
&= \Theta(n)[\mathtt{tr}_{\mathbb{A}}(a)/a \mid a \in A]
\end{aligned}$$

For $a \neq n$, we have:

$$\begin{aligned}
\mathtt{tr}_{\mathbb{A}}(a) &= \mathtt{tr}_{\mathbb{A}^-}(a)[\mathtt{tr}_{\mathbb{A}}(n)/n] && \text{(Definition } \mathtt{tr}_{\mathbb{A}}\text{)}\\
&\equiv_{\mathbf{K}} \Theta(a)[\mathtt{tr}_{\mathbb{A}^-}(b)/b \mid b \sqsubset n][\mathtt{tr}_{\mathbb{A}}(n)/n] && \text{(inductive hypothesis)}\\
&= \Theta(a)[\mathtt{tr}_{\mathbb{A}^-}(b)[\mathtt{tr}_{\mathbb{A}}(n)/n]/b \mid b \sqsubset n, \mathtt{tr}_{\mathbb{A}}(n)/n] && \text{(Fact 5.2.3)}\\
&= \Theta(a)[\mathtt{tr}_{\mathbb{A}}(b)/b \mid b \sqsubset n, \mathtt{tr}_{\mathbb{A}}(n)/n] && \text{(Definition } \mathtt{tr}_{\mathbb{A}}\text{)}\\
&= \Theta(a)[\mathtt{tr}_{\mathbb{A}}(b)/b \mid b \in A]
\end{aligned}$$

The second part of the lemma is also proved by induction on the size of the automaton structure, and again we only consider the inductive case of the argument. Supposing that the result holds for automaton structures smaller than $\mathbb{A}$, we prove the result for $\mathbb{A}$.

For the unique state n of maximal priority, the result is immediate from the definition since in this case $\mathbb{A}\downarrow n = \mathbb{A}$.

For a non-maximal state a, assuming that the induction hypothesis holds for states b with $b \sqsubset a$, we get:

$$
\begin{aligned}
&\mathtt{tr}_{\mathbb{A}}(a) \\
&\equiv_{\mathbf{K}} \mathtt{tr}_{\mathbb{A}^-}(a)[\mathtt{tr}_{\mathbb{A}}(n)/n] && \text{(Definition } \mathtt{tr}_{\mathbb{A}}) \\
&= \eta_a a.\Theta^-(a)[\mathtt{tr}_{(\mathbb{A}^-\downarrow a}(b)/b \mid b \sqsubset a][\mathtt{tr}_{\mathbb{A}^-}(b)/b \mid a \sqsubset b \sqsubset n][\mathtt{tr}_{\mathbb{A}}(n)/n] && \text{(inductive hyp.)} \\
&= \eta_a a.\Theta^-(a)[\mathtt{tr}_{(\mathbb{A}\downarrow a)^-}(b)/b \mid b \sqsubset a][\mathtt{tr}_{\mathbb{A}^-}(b)/b \mid a \sqsubset b \sqsubset n][\mathtt{tr}_{\mathbb{A}}(n)/n] && ((\mathbb{A}\downarrow a)^- = \mathbb{A}^-\downarrow a) \\
&= \eta_a a.\Theta(a)[\mathtt{tr}_{(\mathbb{A}\downarrow a)^-}(b)/b \mid b \sqsubset a][\mathtt{tr}_{\mathbb{A}^-}(b)/b \mid a \sqsubset b \sqsubset n][\mathtt{tr}_{\mathbb{A}}(n)/n] && (\Theta(a) = \Theta^-(a)) \\
&= \eta_a a.\Theta(a)[\mathtt{tr}_{(\mathbb{A}\downarrow a)^-}(b)/b \mid b \sqsubset a][\mathtt{tr}_{\mathbb{A}}(b)/b \mid a \sqsubset b] && \text{(Fact 5.2.3, Def. } \mathtt{tr}_{\mathbb{A}})
\end{aligned}
$$

as required. □

The translation map interacts well with the operation on automata that we defined in section 5.3.3. As an auxiliary result we need the following observation, the proof of which can be found in the appendix of the report version [ESV16a] of [ESV18].

5.8.15. PROPOSITION. *Let $\mathbb{A}$ be a modal automaton with x free and positive. Then we have:*

$$\mathsf{tr}(\mathbb{A}) \equiv_{\mathbf{K}} (x \wedge \mathtt{tr}_{\mathbb{A}^x}((a_I)_0) \vee \mathtt{tr}_{\mathbb{A}^x}((a_I)_1) \tag{5.28}$$
$$\mathsf{tr}(\mu x.\mathbb{A}) \equiv_{\mathbf{K}} \mu x.\mathtt{tr}_{\mathbb{A}^x}((a_I)_1) \tag{5.29}$$
$$\mathsf{tr}(\nu x.\mathbb{A}) \equiv_{\mathbf{K}} \nu x.(\mathtt{tr}_{\mathbb{A}^x}((a_I)_0) \vee \mathtt{tr}_{\mathbb{A}^x}((a_I)_1)). \tag{5.30}$$

Note that we can alternatively write (5.29) as:

$$\mathsf{tr}(\mu x.\mathbb{A}) \equiv_{\mathbf{K}} \mu x.\mathsf{tr}(\mathbb{A}^x),$$

since we chose $(a_I)_1$ as the start state of $\mathbb{A}^x$. (The corresponding equation for $\nu x.\mathbb{A}$ does however *not* hold. To see this, take the $\{x, p, q\}$-automaton $\mathbb{A}$ to have just a single state mapped to the formula $(x \wedge p) \vee q$. We have $\mathsf{tr}(\nu x.\mathbb{A}) \equiv_{\mathbf{K}} p \vee q$, but $\nu x.\mathsf{tr}(\mathbb{A}^x) \equiv_{\mathbf{K}} q$.)

As mentioned, the central result of this section is the following.

5.8.16. PROPOSITION. *The following claims hold, for all modal automata $\mathbb{A}, \mathbb{B}$:*

(1) $\mathsf{tr}(\mathbb{A} \wedge \mathbb{B}) \equiv_{\mathbf{K}} \mathsf{tr}(\mathbb{A}) \wedge \mathsf{tr}(\mathbb{B})$ *and* $\mathsf{tr}(\mathbb{A} \vee \mathbb{B}) \equiv_{\mathbf{K}} \mathsf{tr}(\mathbb{A}) \vee \mathsf{tr}(\mathbb{B})$*;*

(2) $\mathsf{tr}(\neg\mathbb{A}) \equiv_{\mathbf{K}} \neg\mathsf{tr}(\mathbb{A})$*;*

(3) for automata $\mathbb{A}_1, \mathbb{A}_2, ..., \mathbb{A}_n$ and $\alpha \in \mathsf{T}\{\mathbb{A}_1, \mathbb{A}_2, ..., \mathbb{A}_n\}$ we get that $\mathsf{tr}(\nabla\alpha) \equiv_{\mathbf{K}} \nabla(\mathsf{T}\mathsf{tr})(\alpha)$*;*

(4) if $\mathbb{A}$ is positive in p then $\mathsf{tr}(\eta p.\mathbb{A}) \equiv_{\mathbf{K}} \eta p.\mathsf{tr}(\mathbb{A})$ *for* $\eta \in \{\mu, \nu\}$*;*

(5) if $\mathbb{A}$ is positive in p then $\mathtt{tr}(\mathbb{A}[\mathbb{B}/p]) \equiv_{\mathbf{K}} \mathtt{tr}(\mathbb{A})[\mathtt{tr}(\mathbb{B})/p]$.

Proof:

A full proof can be found in the appendix of [ESV16a]. Here we include only the proof for Clause (4).

Clause (4) For this clause we will use Proposition 5.8.15. We first consider the case where $\eta = \mu$. We have:

$$\begin{aligned}
\mathtt{tr}(\mu x.\mathbb{A}) &\equiv_{\mathbf{K}} \mu x.\mathtt{tr}_{\mathbb{A}^x}((a_I)_1) && (5.29)\\
&= \mu x.(x \wedge \mathtt{tr}_{\mathbb{A}^x}((a_I)_0)) \vee \mathtt{tr}_{\mathbb{A}^x}((a_I)_1) && \text{Proposition 5.3.27}\\
&\equiv_{\mathbf{K}} \mu x.\mathtt{tr}(\mathbb{A}) && (5.28)
\end{aligned}$$

Next, for the case of $\eta = \nu$, we have:

$$\begin{aligned}
\mathtt{tr}(\nu x.\mathbb{A}) &\equiv_{\mathbf{K}} \nu x.(\mathtt{tr}_{\mathbb{A}^x}((a_I)_0) \vee \mathtt{tr}_{\mathbb{A}^x}((a_I)_1) && (5.30)\\
&= \nu x.((x \wedge \mathtt{tr}_{\mathbb{A}^x}((a_I)_0)) \vee \mathtt{tr}_{\mathbb{A}^x}((a_I)_1) && \text{Proposition 5.3.27}\\
&\equiv_{\mathbf{K}} \nu x.\mathtt{tr}(\mathbb{A}) && (5.28)
\end{aligned}$$

and the proof is done. □

From this result, Proposition 5.8.1 follows easily.

Proof of Proposition 5.8.1: By induction on the complexity of a formula. For atomic formulas the result is easily checked, and for the inductive clauses we use the properties established in Proposition 5.8.16. For example, for a fixpoint formula $\mu x.\varphi(x)$, we have $\mathbb{A}_{\mu x.\varphi(x)} = \mu x.\mathbb{A}_{\varphi(x)}$ by definition, and we get

$$\mathtt{tr}(\mathbb{A}_{\mu x.\varphi(x)}) = \mathtt{tr}(\mu x.\mathbb{A}_{\varphi(x)}) \equiv_K \mu x.\mathtt{tr}(\mathbb{A}_{\varphi(x)}) \equiv_K \mu x.\varphi(x).$$

The other cases are similar. □

5.9 Completeness

In this section, we give an overview of the completeness proof for $\mu\mathtt{ML}_{\nabla}^{\overline{\top}}$ with respect to the derivation system **K**. In [Koz83] Kozen proved the completeness of his proof system for a fragment of the modal μ-calculus: he showed that for *aconjunctive* formulas consistency implies satisfiability. The following lemma, which is the one result missing to prove our main result, can be seen as an automata-theoretic version of Walukiewicz' rendering of Kozen's result.

5.9.1. LEMMA. *[Kozen's Lemma] Given an automaton $\mathbb{A}$, if* $\mathtt{tr}(\mathbb{A})$ *is consistent, then $\exists$ has a winning strategy in the thin satisfiability game for $\mathbb{A}$.*

Proof:
The proof of this Lemma is almost verbatim a copy of the proof of the analogous result, viz., Theorem 5, in [ESV18]: the only difference is that here we need the one-step completeness result, Proposition 5.3.15. □

We are now ready to proof our main technical result. Observe that here, and in the sequel, we will use the notation of Definition 5.8.2.

5.9.2. THEOREM. *For every formula $\varphi \in \mu\mathtt{ML}_{\nabla}^{\top}$, there is a semantically equivalent disjunctive automaton $\mathbb{D}$ such that $\varphi \leq_{\mathbf{K}} \mathbb{D}$.*

To prove this theorem, recall from Proposition 5.8.1 that every formula $\varphi \in \mu\mathtt{ML}_{\nabla}^{\top}$ is provably equivalent to the translation of the modal automaton $\mathbb{A}_{\varphi}$. Thus in particular we have $\varphi \leq_{\mathbf{K}} \mathbb{A}_{\varphi}$. We now want to apply the automata-theoretic machinery that we developed in previous sections, to strengthen this result, showing that for any formula φ there is a *disjunctive* automaton $\mathbb{D}_{\varphi}$ such that $\varphi \leq_{\mathbf{K}} \mathbb{D}_{\varphi}$. The following lemma shows that whenever φ is the translation of a semi-disjunctive automaton this result can be proved.

5.9.3. LEMMA. *Let $\mathbb{A}$ be any semi-disjunctive automaton. Then $\mathbb{A} \leq_{\mathbf{K}} \mathsf{sim}(\mathbb{A})$.*

Proof:
By Proposition 5.7.2 there is a winning strategy for player II in the consequence game $\mathcal{C}(\mathbb{A}, \mathsf{sim}(\mathbb{A}))$. Since $\mathbb{A}$ is semi-disjunctive, it follows from Theorem 5.6.7 that there is a winning strategy for $\forall$ in the thin satisfiability game for $\mathbb{A} \wedge \neg\mathsf{sim}(\mathbb{A})$. By Kozen's Lemma (Lemma 5.9.1) it follows that the automaton $\mathbb{A} \wedge \neg\mathsf{sim}(\mathbb{A})$ is inconsistent. Then from this and clause (2) of Lemma 5.8.16 it is immediate that $\mathbb{A} \leq_{\mathbf{K}} \mathsf{sim}(\mathbb{A})$ as required. □

Proof of Theorem 5.9.2:
Since any fixpoint formula is provably equivalent to a formula in negation normal form, without loss of generality we may prove the theorem for formulas in this shape, and proceed by an induction on the complexity of such formulas. That is, the base cases of the induction are the literals, and we need to consider induction steps for conjunctions, disjunctions, the modal operator and both fixpoint operators. The base case for literals follows immediately since it is easy to see that the modal automaton $\mathbb{A}_{\varphi}$ corresponding to a literal φ is already disjunctive.

Disjunction and modality are easy since the operations $\vee$ and ∇ on automata preserve the property of being disjunctive. For conjunctions, given formulas α, α' we have disjunctive automata $\mathbb{D} \equiv \alpha$ and $\mathbb{D}' \equiv \alpha'$ such that $\alpha \leq_{\mathbf{K}} \mathbb{D}$ and $\alpha' \leq_{\mathbf{K}} \mathbb{D}'$. By the first clause of Proposition 5.8.1 we get $\alpha \wedge \alpha' \leq_{\mathbf{K}} \mathbb{D} \wedge \mathbb{D}'$. But $\mathbb{D} \wedge \mathbb{D}'$ is semi-disjunctive by the fourth clause of Lemma 5.6.6, and we can apply Lemma 5.9.3 to obtain the desired conclusion.

Finally we turn to the fixpoint operators. For the greatest fixpoint operator, consider the formula $\varphi = \nu x.\alpha(x)$, and assume inductively that there is a disjunctive automaton $\mathbb{A}$ for α such that $\alpha \equiv \mathbb{A}$ and $\alpha \leq_{\mathbf{K}} \mathbb{A}$. It follows by Proposition 5.8.16(4) that $\varphi = \nu x.\alpha \leq_{\mathbf{K}} \nu x.\mathbb{A}$, and since $\nu x.\mathbb{A}$ is semi-disjunctive modulo provable equivalence by Proposition 5.6.6(6), by Lemma 5.9.3 we are done.

Now we consider the crucial case where $\varphi = \mu x.\alpha(x)$. By the induction hypothesis there is a semantically equivalent disjunctive automaton $\mathbb{A}$ for α such that $\alpha \leq_{\mathbf{K}} \mathbb{A}$. Let $\mathbb{D} := \mathsf{sim}(\mu x.\mathbb{A})$. This automaton is clearly semantically equivalent to φ. We want to show that:

$$\mu x.\mathbb{A} \leq_{\mathbf{K}} \mathbb{D},$$

from which the result follows since $\varphi \leq_{\mathbf{K}} \mu x.\mathbb{A}$. By Proposition 5.8.1 and (5.29) we obtain $\mu x.\mathbb{A} \equiv_{\mathbf{K}} \mathsf{tr}(\mu x.\mathbb{A}) \equiv_{\mathbf{K}} \mu x.\mathsf{tr}(\mathbb{A}^x)$, and by Proposition 5.8.1 we also have $\mathbb{D} \equiv_{\mathbf{K}} \mathsf{tr}(\mathbb{D})$, so in fact it suffices to show that:

$$\mu x.\mathsf{tr}(\mathbb{A}^x) \leq_{\mathbf{K}} \mathsf{tr}(\mathbb{D}).$$

Hence by the fixpoint rule it suffices to prove that:

$$\mathsf{tr}(\mathbb{A}^x)[\mathsf{tr}(\mathbb{D})/x] \leq_{\mathbf{K}} \mathsf{tr}(\mathbb{D}).$$

But using clause (5) of Proposition 5.8.16 we get

$$\mathsf{tr}(\mathbb{A}^x)[\mathsf{tr}(\mathbb{D})/x] \equiv_{\mathbf{K}} \mathsf{tr}(\mathbb{A}^x[\mathbb{D}/x]),$$

so it suffices to prove $\mathbb{A}^x[\mathbb{D}/x] \leq_{\mathbf{K}} \mathbb{D}$, or equivalently:

$$\vdash_{\mathbf{K}} \neg(\mathsf{tr}(\mathbb{A}^x[\mathbb{D}/x]) \wedge \neg\mathsf{tr}(\mathbb{D})).$$

We can now apply the clauses (1) and (2) of Proposition 5.8.16 to see that this is equivalent to $\vdash_{\mathbf{K}} \neg\mathsf{tr}(\mathbb{A}^x[\mathbb{D}/x] \wedge \neg\mathbb{D})$, and by Lemma 5.9.1 it therefore suffices to prove that $\forall$ has a winning strategy in the thin satisfiability game for the automaton $\mathbb{A}^x[\mathbb{D}/x] \wedge \neg\mathbb{D}$. Note that by clause (6) of Proposition 5.6.6 $\mathbb{A}^x$ is semi-disjunctive since $\mathbb{A}$ is disjunctive. Now since $\mathbb{D}$ is disjunctive and $\mathbb{A}^x$ is semi-disjunctive, from Proposition 5.6.6 clause (5) it follows that $\mathbb{A}^x[\mathbb{D}/x]$ is semi-disjunctive too. Hence, by Theorem 5.6.7 the required conclusion follows if we can show that $\mathbb{A}^x[\mathbb{D}/x] \vDash_{\mathcal{C}} \mathbb{D}$. But from Proposition 5.4.13 and Theorem 5.7.2 we get by transitivity of game consequence:

$$\mathbb{A}^x[\mu x.\mathbb{A}/x] \vDash_{\mathcal{C}} \mu x.\mathbb{A} \vDash_{\mathcal{C}} \mathsf{sim}(\mu x.\mathbb{A}) = \mathbb{D},$$

so it suffices to show that

$$\mathbb{A}^x[\mathbb{D}/x] \vDash_{\mathcal{C}} \mathbb{A}^x[\mu x.\mathbb{A}/x].$$

But this is an instance of Proposition 5.7.6, and so we are done. □

Finally we have all the pieces in place to prove completeness.

5.9.4. THEOREM. *[Completeness] Every consistent formula $\varphi \in \mu\mathtt{ML}_{\nabla}^{\mathsf{T}}$ is satisfiable.*

Proof:
Given a consistent formula φ, by Theorem 5.9.2 there exists a semantically equivalent disjunctive automaton $\mathbb{D}$ such that $\mathtt{tr}(\mathbb{D})$ is consistent too. Now by Lemma 5.9.1 $\exists$ has a winning strategy in $\mathcal{S}_{thin}(\mathbb{D})$. But $\mathbb{D}$ is disjunctive and hence semi-disjunctive, and so by Proposition 5.6.4 $\exists$ also has a winning strategy in $\mathcal{S}(\mathbb{D})$. □

5.10 Conclusion

In this chapter, we brought ideas from automata theory and the theory of coalgebras together and proved a completeness result for the nabla-based coalgebraic fixpoint logic $\mu\mathtt{ML}_{\mathsf{T}}^{\mathsf{T}}$, where T weakly preserves pullbacks. Following up on the results of this chapter, which was based on [ESV18] and [ESV16b], we set up a more general framework for proving completeness results for variants of the modal μ-calculus. In [ESV19], taking the predicate lifting approach towards coalgebraic modal logic, we illustrate the method by proving two new completeness results: for the graded μ-calculus (which is equivalent to monadic second-order logic on the class of unranked tree models), and for the monotone modal mu-calculus. Besides these main applications, our result in [ESV19] covers also the Kozen-Walukiewicz completeness theorem for the standard modal μ-calculus, as well as the linear-time μ-calculus and modal fixpoint logics on ranked trees. Completeness of the linear-time μ-calculus is known, but the proof we obtain here is different, and places the result under a common roof with Walukiewicz' result.

Bibliography

[Acz88] Peter Aczel, *Non-well-founded sets*, CSLI Lecture Notes, vol. 14, 1988.

[AGT09] J. Adámek, H. P. Gumm, and V. Trnková, *Presentation of set functors: A coalgebraic perspective*, Journal of Logic and Computation **20** (2009), no. 5, 991–1015.

[AM89] Peter Aczel and Nax Paul Mendler, *A final coalgebra theorem*, Category Theory and Computer Science, 1989, pp. 357–365.

[AN01] A. Arnold and D. Niwiński, *Rudiments of μ-calculus*, Studies in Logic and the Foundations of Mathematics, vol. 146, North-Holland Publishing Co., Amsterdam, 2001.

[AT90] Jiří Adámek and Věra Trnková, *Automata and algebras in categories*, Mathematics and its Applications (East European Series), vol. 37, Kluwer Academic Publishers Group, Dordrecht, 1990.

[Bal00] Alexandru Baltag, *A logic for coalgebraic simulation*, Electronic Notes in Theoretical Computer Science **33** (2000), 42–60.

[Bar93] Michael Barr, *Terminal coalgebras in well-founded set theory*, Theoretical Computer Science **114** (1993), no. 2, 299 – 315.

[BdRV02] P. Blackburn, M. de Rijke, and Y. Venema, *Modal logic*, Cambridge Tracts in Theoretical Computer Science, vol. 53, 2002.

[Ben76] J. van Benthem, *Modal correspondence theory*, Ph.D. thesis, Mathematisch Instituut & Instituut voor Grondslagenonderzoek, University of Amsterdam, 1976.

[Bet56] E. W. Beth, *On padoa's method in the theory of definition*, Journal of Symbolic Logic **21** (1956), no. 2, 194–195.

[Bíl07] Marta Bílková, *Uniform interpolation and propositional quantifiers in modal logics*, Studia Logica **85** (2007), 1–31.

[BK08] Christel Baier and Joost-Pieter Katoen, *Principles of model checking (representation and mind series)*, The MIT Press, 2008.

[BL69] J.R. Büchi and L.H. Landweber, *Solving sequential conditions by finite state strategies*, Transactions of the American Mathematical Society **138** (1969), 295–311.

[Bla92] A. Blass, *A game semantics for linear logic*, Annals of Pure and Applied Logic **56** (1992), no. 1-3, 183–220.

[BM96] Jon Barwise and Lawrence Moss, *Vicious circles*, CSLI Lecture Notes, vol. 60, 1996.

[BS01] Julian Bradfield and Colin Stirling, *Modal logics and mu-calculi: An introduction*, 2001.

[Büc60] J. R. Büchi, *Weak second-order arithmetic and finite automata*, Mathematical Logic Quarterly **6** (1960), no. 1-6, 66–92.

[BW95] Michael Barr and Charles Wells (eds.), *Category theory for computing science, 2nd ed.*, Prentice Hall International (UK) Ltd., Hertfordshire, UK, UK, 1995.

[CE80] Edmund M. Clarke and E. Allen Emerson, *Characterizing correctness properties of parallel programs using fixpoints*, Proceedings of the 7th Colloquium on Automata, Languages and Programming, Springer-Verlag, 1980, p. 169?181.

[CE81] ———, *Design and synthesis of synchronization skeletons using branching-time temporal logic*, Logic of Programs, Workshop, Springer-Verlag, 1981, p. 52?71.

[Che80] Brian F. Chellas, *Modal logic: An introduction*, Cambridge University Press, 1980.

[CKP⁺08] Corina Cirstea, Alexander Kurz, Dirk Pattinson, Lutz Schröder, and Yde Venema, *Modal logics are coalgebraic*, Visions of Computer Science, BCS International Academic Research Conference (BCS 2008) (Samson Abramsky, Erol Gelenbe, and Vladimiro Sassone, eds.), British Computer Society, 2008, pp. 129–140.

[CKP09] Corina Cîrstea, Clemens Kupke, and Dirk Pattinson, *EXPTIME tableaux for the coalgebraic μ-calculus*, Computer Science Logic, 23rd international Workshop, CSL 2009, 18th Annual Conference of the

EACSL, Coimbra, Portugal, September 7-11, 2009. Proceedings, 2009, pp. 179 – 193.

[CKW91] A. Carboni, G. M. Kelly, and R. J. Wood, *A 2-categorical approach to change of base and geometric morphisms i*, Cahiers de Topologie et Géométrie Différentielle Catégoriques **32** (1991), no. 1, 47–95.

[CP04] Corina Cîrstea and Dirk Pattinson, *Modular construction of modal logics*, CONCUR 2004 - Concurrency Theory, Springer Berlin Heidelberg, 2004, pp. 258–275.

[Cra57] William Craig, *Three uses of the Herbrand-Gentzen theorem in relating model theory and proof theory*, Journal of Symbolic Logic **22** (1957), no. 3, 269–285.

[D'A08] Giovanna D'Agostino, *Interpolation in non-classical logics*, Synthese **164** (2008), no. 3, 421–435.

[DH00] Giovanna D'Agostino and Marco Hollenberg, *Logical questions concerning the μ-calculus: Interpolation, lyndon and łoś-tarski*, Journal of Symbolic Logic **65** (2000), no. 1, 310–332.

[DV02] Giovanna D'Agostino and Albert Visser, *Finality regained: A coalgebraic study of scott-sets and multisets*, Archive for Mathematical Logic **41** (2002), no. 3, 267–298.

[EJ91a] E. A. Emerson and C. S. Jutla, *Tree automata, mu-calculus and determinacy*, Proceedings of the 32nd Annual Symposium on Foundations of Computer Science, 1991, p. 368?377.

[EJ91b] E. Allen Emerson and Charanjit S. Jutla, *Tree automata, mu-calculus and determinacy*, [1991] Proceedings 32nd Annual Symposium of Foundations of Computer Science (1991), 368–377.

[EL86] E. Allen Emerson and Chin-Laung Lei, *Efficient model checking in fragments of the propositional mu-calculus (extended abstract)*, Proceedings of the First Annual IEEE Symposium on Logic in Computer Science (LICS 1986), IEEE Computer Society Press, June 1986, pp. 267–278.

[ESV15] Sebastian Enqvist, Fatemeh Seifan, and Yde Venema, *Monadic second-order logic and bisimulation invariance for coalgebras*, Proceedings of the 30th Annual ACM/IEEE Symposium on Logic in Computer Science (LICS), LICS '15, IEEE Computer Society, 2015, pp. 353 – 365.

[ESV16a] S. Enqvist, F. Seifan, and Y. Venema, *Completeness for the modal μ-calculus: separating the combinatorics from the dynamics*, Tech. Report PP-2016-33, Institute for Logic, Language and Computation, Universiteit van Amsterdam, 2016, (Submitted).

[ESV16b] Sebastian Enqvist, Fatemeh Seifan, and Yde Venema, *Completeness for coalgebraic fixpoint logic*, 25th EACSL Annual Conference on Computer Science Logic (CSL 2016) (Dagstuhl, Germany), Leibniz International Proceedings in Informatics (LIPIcs), vol. 62, Schloss Dagstuhl–Leibniz-Zentrum fuer Informatik, 2016, pp. 7:1– 7:19.

[ESV17] ———, *An expressive completeness theorem for coalgebraic modal mu-calculi*, Logical Methods in Computer Science **13** (2017), no. 2.

[ESV18] ———, *Completeness for the modal mu-calculus: Separating the combinatorics from the dynamics*, Theor. Comput. Sci. **727** (2018), 37–100.

[ESV19] Sebastian Enqvist, Fatemeh Seifan, and Yde Venema, *Completeness for μ-calculi : A coalgebraic approach*, Annals of Pure and Applied Logic **170** (2019), no. 5, 578–641.

[EV17] Sebastian Enqvist and Yde Venema, *Disjunctive bases: Normal forms for modal logics*, 7th Conference on Algebra and Coalgebra in Computer Science, (CALCO 2017), vol. 72, 2017, pp. 11:1–11:16.

[Fin72] Kit Fine, *In so many possible worlds.*, Notre Dame J. Formal Logic **13** (1972), no. 4, 516–520.

[FL79] Michael J. Fischer and Richard E. Ladner, *Propositional dynamic logic of regular programs*, J. Comput. Syst. Sci. **18** (1979), 194–211.

[FLV10] Gäelle Fontaine, Raul Leal, and Yde Venema, *Automata for coalgebras: An approach using predicate liftings*, ICALP'10: Automata, Languages and Programming: Bordeaux, France, July, 2010. Proceedings, Part II, 2010, pp. 381–392.

[Fon10] Gabriel A Fontaine, *Modal fixpoint logic: some model theoretic questions*, 2010.

[Fre06] Tim French, *Bisimulation quantifiers for modal logic*, Ph.D. thesis, School of Computer Science and Software Engineering, University of Western Australia, 2006.

[FV18] Gaëlle Fontaine and Yde Venema, *Some model theory for the modal μ-calculus: syntactic characterisations of semantic properties*, Logical Methods in Computer Science **14** (2018), no. 1.

[Ghi95] Silvio Ghilardi, *An algebraic theory of normal forms*, Annals of Pure and Applied Logic **71** (1995), no. 3, 189–245.

[GS05] H. P. Gumm and T. Schröder, *Types and coalgebraic structure*, Journal of Algebra Universalis **53** (2005), no. 2, 229–252.

[GTW02] E. Grädel, W. Thomas, and T. Wilke (eds.), *Automata, logic, and infinite games*, LNCS, vol. 2500, Springer, 2002.

[Gum01] Peter H. Gumm, *Functors for coalgebras*, Algebra Universalis **45** (2001), no. 2, 135–147.

[Gum05] H. Peter Gumm, *From t-coalgebras to filter structures and transition systems*, Algebra and Coalgebra in Computer Science, Springer, 2005, pp. 194–212.

[GZ95] Silvio Ghilardi and Marek Zawadowski, *Undefinability of propositional quantifiers in the modal system **S4***, Journal of Studia Logica **55** (1995), no. 2, 259–271.

[Han03] Helle Hvid Hansen, *Monotonic modal logics*, Master's thesis, ILLC, Universiteit van Amsterdam, the Netherlands, 10 2003.

[Hen63] Leon Henkin, *An extension of the Craig-Lyndon interpolation theorem*, Journal of Symbolic Logic **28** (1963), no. 3, 201–216.

[HK04] Helle Hvid Hansen and Clemens Kupke, *A coalgebraic perspective on monotone modal logic*, Electronic Notes in Theoretical Computer Science **106** (2004), 121–143.

[HKP09] Helle Hvid Hansen, Clemens Kupke, and Eric Pacuit, *Neighbourhood structures: Bisimilarity and basic model theory*, Logical Methods in Computer Science **5** (2009), no. 2.

[HM80] Matthew Hennessy and Robin Milner, *On observing nondeterminism and concurrency*, Automata, Languages and Programming, Springer Berlin Heidelberg, 1980, pp. 299–309.

[HM98] Aviad Heifetz and Philippe Mongin, *The modal logic of probability*, Proceedings of the 7th Conference on Theoretical Aspects of Rationality and Knowledge, TARK '98, Morgan Kaufmann Publishers Inc., 1998, pp. 175–185.

[HTK00] David Harel, Jerzy Tiuryn, and Dexter Kozen, *Dynamic logic*, MIT Press, 2000.

[Jac16] Bart Jacobs, *Introduction to coalgebra: Towards mathematics of states and observation*, Cambridge Tracts in Theoretical Computer Science, Cambridge University Press, 2016.

[Joy95] A. Joyal, *Free lattices, communication and money games*, Proceedings of the 10th International Congress of Logic, Methodology, and Philosophy of Science, 1995.

[JR97] Bart Jacobs and Jan Rutten, *A tutorial on (co)algebras and (co)induction*, EATCS Bulletin **62** (1997), 62–222.

[JW95] David Janin and Igor Walukiewicz, *Automata for the modal μ-calculus and related results*, MFCS'95 Symposium on Mathematical Foundations of Computer Science, Prague, Czech Republic, August-September, 1995. Proceedings, vol. 969, 1995, pp. 552–562.

[JW96] ______, *On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic*, CONCUR '96: Concurrency Theory: Pisa, Italy, August, 1996. Proceedings, 1996, pp. 263–277.

[KKV03] Clemens Kupke, Alexander Kurz, and Yde Venema, *Stone coalgebras*, Electronic Notes in Theoretical Computer Science **82** (2003), no. 1, 170–190, CMCS'03, Coalgebraic Methods in Computer Science (Satellite Event for ETAPS 2003).

[KKV08] ______, *Completeness of the finitary Moss logic*, AiML'08: Advances in Modal Logic, Nancy, France, September, 2008. Proceedings, 2008, pp. 193–217.

[KKV12] ______, *Completeness for the coalgebraic cover modality*, Logical Methods in Computer Science **8** (2012), no. 3, 1–76.

[Koz83] Dexter Kozen, *Results on the propositional μ-calculus*, Theoretical Computer Science **27** (1983), no. 3, 333–354.

[KP11] C. Kupke and D. Pattinson, *Coalgebraic semantics of modal logics: An overview*, Theoretical Computer Science **412** (2011), no. 38, 5070 – 5094, CMCS Tenth Anniversary Meeting.

[Kri59] Saul A. Kripke, *A completeness theorem in modal logic*, The Journal of Symbolic Logic **24** (1959), no. 1, 1–14.

[Kup06] Clemens Kupke, *Finitary coalgebraic logics*, Ph.D. thesis, ILLC, Universiteit van Amsterdam, 3 2006.

[KV08] Clemens Kupke and Yde Venema, *Coalgebraic automata theory: Basic results*, Logical Methods in Computer Science **4** (2008), no. 4.

[KV09] Christian Kissig and Yde Venema, *Complementation of coalgebra automata*, CALCO'09: Algebra and Coalgebra in Computer Science, Udine, Italy, September, 2009. Proceedings, 2009, pp. 81–96.

[LPSS12] Tadeusz Litak, Dirk Pattinson, Lutz Schröder, and Katsuhiko Sano, *Coalgebraic predicate logic*, 39th International Colloquium on Automata, Languages and Programming (ICALP 2012) (Kurt Mehlhorn, Andrew Pitts, and Roger Wattenhofer, eds.), vol. 7392, 2012, pp. 299–311.

[Lyn59] Roger C. Lyndon, *An interpolation theorem in the predicate calculus*, Pacific Journal of Mathematics **9** (1959), no. 1, 129–142.

[Mar11] Johannes Marti, *Relation liftings in coalgebraic modal logic*, Master's thesis, ILLC, Universiteit van Amsterdam, the Netherlands, 9 2011.

[McM18] Kenneth L. McMillan, *Interpolation and model checking*, pp. 421–446, Springer International Publishing, 2018.

[Mil81] R. Milner, *A modal characterisation of observable machine-behaviour*, CAAP '81, Springer Berlin Heidelberg, 1981, pp. 25–34.

[ML71] Saunders Mac Lane, *Categories for the working mathematician*, Graduate Texts in Mathematics, no. 5, Springer-Verlag, 1971.

[Mon70] R. Montague, *Universal grammar*, Theoria **36** (1970), no. 3, 373–398.

[Mon74] Richard Montague, *Universal grammar*, Theoria **36** (1974), 373 – 398.

[Mos84] Andrzej Wlodzimierz Mostowski, *Regular expressions for infinite trees and a standard form of automata*, Symposium on Computation Theory, 1984.

[Mos91] ———, *Games with forbidden positions*, Technical Report 78, University of Gdań sk, 1991.

[Mos99] Larry Moss, *Coalgebraic logic*, Annals of Pure and Applied Logic **96** (1999), 277–317, (Erratum published *Ann.P.Appl.Log.* 99:241–259, 1999).

[MS87] David E. Muller and Paul E. Schupp, *Alternating automata on infinite trees*, Theoretical Computer Science **54** (1987), no. 2, 267 – 276.

[MS95] ———, *Simulating alternating tree automata by nondeterministic automata: New results and new proofs of the theorems of rabin, mcnaughton and safra*, Theoretical Computer Science **141** (1995), no. 1-2, 69–107.

[MSV15] Johannes Marti, Fatemeh Seifan, and Yde Venema, *Uniform interpolation for coalgebraic fixpoint logic*, CALCO'15: Algebra and Coalgebra in Computer Science, Nijmegen, Netherlands, June, 2015. Proceedings, 2015, pp. 238–252.

[MV12] Johannes Marti and Yde Venema, *Lax extensions of coalgebra functors*, Proceedings of the International Workshop on Coalgebraic Methods in Computer Science (CMCS) (2012), 150–160.

[Niw88] Damian Niwinski, *Fixed points vs. infinite generation*, Proceedings of the Eleventh Annual IEEE Symposium on Logic in Computer Science (LICS 1988), IEEE Computer Society Press, 1988, pp. 402 – 409.

[Niw97] ———, *Fixed point characterization of infinite behaviour of finite-state systems*, Theoretical Computer Science **189** (1997), no. 1, 1 – 69.

[NW96] D. Niwínski and I. Walukiewicz, *Games for the μ-calculus*, Theoretical Computer Science **163** (1996), 99–116.

[Par80] David Park, *On the semantics of fair parallelism*, Abstract Software Specifications, Springer Berlin Heidelberg, 1980, pp. 504–526.

[Par85] Rohit Parikh, *The logic of games and its applications*, Annals of Discrete Mathematics, Elsevier, 1985, pp. 111 – 140.

[Pat03a] Dirk Pattincon, *An introduction to the theory of coalgebras*, Course notes for NASSLLI, 2003.

[Pat03b] Dirk Pattinson, *Coalgebraic modal logic: soundness, completeness and decidability of local consequence*, Theor. Comput. Sci. **309** (2003), no. 1-3, 177–193.

[Pau99] M. Pauly, *Bisimulation for general non-normal modal logic*, Manuscript (unpublished) (1999).

[Pau02] ———, *A modal logic for coalitional power in games*, J. Logic Comput. **12** (2002), no. 1, 149–166.

[Pit92] Andrew M. Pitts, *On an interpretation of second order quantification in first order intuitionistic propositional logic*, Journal of Symbolic Logic **57** (1992), no. 1, 33–52.

[Pnu77] Amir Pnueli, *The temporal logic of programs*, Proceedings of the 18th Annual Symposium on Foundations of Computer Science, SFCS 77, IEEE Computer Society, 1977, pp. 46–57.

[Pra81] Vaughan R. Pratt, *A decidable mu-calculus: Preliminary report*, 22nd Annual Symposium on Foundations of Computer Science (sfcs 1981) (1981), 421–427.

[Rab69] Michael O. Rabin, *Decidability of second-order theories and automata on infinite trees*, Transactions of the American Mathematical Society **141** (1969), 1–35.

[Rut98] J.J.M.M. Rutten, *Relators and metric bisimulations*, Electronic Notes in Theoretical Computer Science **11** (1998), 252 –258.

[Rut00] ———, *Universal coalgebra: a theory of systems*, Theoretical Computer Science **249** (2000), no. 1, 3–80.

[Rut19] ———, *The method of coalgebra: exercises in coinduction*, CWI, February 2019.

[San02] Luigi Santocanale, *Free μ-lattices*, Journal of Pure and Applied Algebra **168** (2002), no. 2-3, 227–264.

[SB69] D. Scott and J. W. De Bakker, *A theory of programs*, Unpublished manuscript. IBM (1969).

[Sch08] Lutz Schröder, *Expressivity of coalgebraic modal logic: The limits and beyond*, Theoretical Computer Science **390** (2008), no. 2, 230 – 247.

[Sco70] D. Scott, *Advice on modal logic*, Philosophical Problems in Logic: Some Recent Developments (K. Lambert, ed.), 1970, pp. 143–173.

[SE89] Robert S. Streett and E. Allen Emerson, *An automata theoretic decision procedure for the propositional mu-calculus*, Information and Computation **81** (1989), no. 3, 249 – 264.

[Sha94] Vladimir Yurievich Shavrukov, *Adventures in diagonalizable algebras*, Ph.D. thesis, ILLC, Universiteit van Amsterdam, 6 1994.

[SP08] L. Schröder and D. Pattinson, *Shallow models for non-iterative modal logics*, KI 2008: Advances in Artificial Intelligence (Berlin, Heidelberg), Springer Berlin Heidelberg, 2008, pp. 324–331.

[SP09a] Lutz Schröder and Dirk Pattinson, *PSPACE bounds for rank-1 modal logics*, ACM Trans. Comput. Log. **10** (2009), no. 2, 13:1–13:33.

[SP09b] Lutz Schröder and Dirk Pattinson, *Strong completeness of coalgebraic modal logics*, 26th International Symposium on Theoretical Aspects of Computer Science (STACS 2009) (Susanne Albers and Jean-Yves Marion, eds.), 2009, pp. 673–684.

[SP10] Lutz Schröder and Dirk Pattinson, *Rank-1 modal logics are coalgebraic*, J. Log. Comput. **20** (2010), no. 5, 1113–1147.

[SPL15] Lutz Schröder, Dirk Pattinson, and Tadeusz Litak, *A van benthem/rosen theorem for coalgebraic predicate logic*, J. Log. Comput. **27** (2015), 749–773.

[SSP17] L. Schröder, F. Seifan, and D. Pattinson, *Uniform interpolation in coalgebraic modal logic*, 7th Conference on Algebra and Coalgebra in Computer Science (CALCO 2017), Leibniz International Proceedings in Informatics (LIPIcs), vol. 72, 2017, pp. 21:1–21:16.

[SV10] Luigi Santocanale and Yde Venema, *Uniform interpolation for monotone modal logic*, AiML'10: Advances in Modal Logic, Moscow, Russia, August, 2010. Proceedings, 2010, pp. 350–370.

[Tar57] Alfred Tarski, *A lattice-theoretical fixpoint theorem and its applications*, Pacific journal of mathematics **22** (1957).

[Trn69] V. Trnková, *Some properties of set functors*, Commentationes Mathematicae Universitatis Carolinae **10** (1969), no. 2, 323–352.

[Var96] Moshe Vardi, *Why is modal logic so robustly decidable?*, Descriptive Complex Finite Models **31** (1996).

[vB76] Johan van Benthem, *Modal correspondence theory*, Ph.D. thesis, ILLC, Universiteit van Amsterdam, 1976.

[vB10] J. van Benthem, *Modal logic for open minds*, Bibliovault OAI Repository, the University of Chicago Press, 2010.

[Ven06] Yde Venema, *Automata and fixed point logic: a coalgebraic perspective*, Information and Computation **204** (2006), 637–678.

[Ven14] Yde Venema, *Expressiveness modulo bisimilarity: A coalgebraic perspective.*, Johan van Benthem on Logic and Information Dynamics, Springer, 2014, pp. 33–65.

[Ven19] Yde Venema, *Coalgebra and modal logic: an introduction*, Lecture Notes, ILLC, University of Amsterdam, 2019.

[Vis96a] Albert Visser, *Bisimulations, model descriptions and propositional quantifiers*, vol. 161, Logic Group Preprint Series, 1996.

[Vis96b] ______, *Uniform interpolation and layered bisimulation*, Gödel'96: Logical foundations of mathematics, computer science and physics—Kurt Gödel's legacy, Brno, Czech Republic, August, 1996. Proceedings, Lecture Notes in Logic, vol. 6, 1996, pp. 139–164.

[Wal96] Igor Walukiewicz, *Monadic second order logic on tree-like structures*, 1996.

[Wal00] ______, *Completeness of Kozen's axiomatisation of the propositional μ-calculus*, Information and Computation **157** (2000), 142–182.

[Wal02] I. Walukiewicz, *Monadic second-order logic on tree-like structures*, Theoretical Computer Science **275** (2002), 311–346.

[Wil01] Thomas Wilke, *Alternating tree automata, parity games, and modal μ-calculus*, Bulletin of the Belgian Mathematical Society - Simon Stevin **8** (2001), no. 2, 359–391.

[Zie98] Wieslaw Zielonka, *Infinite games on finitely coloured graphs with applications to automata on infinite trees*, Theoretical Computer Science **200** (1998), no. 1-2, 135–183.

Samenvatting

Dit proefschrift bestudeert de expressiviteit en volledigheid van de coalgebraïsche μ-calculus. Met deze logica, een coalgebraïsche generalisatie van de standaard μ-calculus, creëren we een uniform raamwerk voor verschillende modale dekpuntlogica's. Ons belangrijkste doel is om te laten zien dat verscheidene belangrijke resultaten, zoals uniforme interpolatie, expressieve volledigheid en axiomatische volledigheid van de standaard μ-calculus kunnen worden gegeneraliseerd naar het niveau van coalgebra's. Om dit doel te bereiken ontwikkelen we automaten- en speltheoretische methodes om eigenschappen van de coalgebraïsche μ-calculus te bestuderen.

In Hoofdstuk 3 bewijzen we een uniforme-interpolatiestelling voor de coalgebraïsche μ-calculus. Deze stelling generaliseert een resultaat van D'Agostino en Hollenberg [DH00] naar een bredere klasse van dekpuntlogica's, waaronder de monotone μ-calculus: de uitbreiding van de monotone modale logica met dekpuntoperatoren. Om dit doel te bereiken beschouwen we eerst een belangrijke eigenschap van automaten, namelijk afsluiting onder projectie. We bewijzen dat deze eigenschap, waarvan bekend is dat deze geldt voor functoren die zwakke pullbacks behouden, ook geldt voor een bredere klasse van functoren, te weten de functoren met een zogeheten quasi-functionele lakse relatielifting. Vervolgens laten we zien dat afsluiting onder projectie impliceert dat de bisimulatiekwantor definieerbaar is in de taal van de coalgebraïsche μ-calculus. Ten slotte gebruiken we dit resultaat om een uniforme-interpolatiestelling te bewijzen voor de coalgebraïsche μ-calculus.

In hoofdstuk 4 generaliseren we de stelling van Janin-Walukiewicz [JW96] die stelt dat de modale μ-calculus precies het bisimulatie-invariante fragment van de monadische tweede-orde logica vangt, naar het niveau van coalgebra's. Daarvoor introduceren we eerst een notie van coalgebraïsche monadische tweede-orde logica $\mathtt{MSO}_{\mathsf{T}}$ voor coalgebra's van type T. In navolging van automaten-theoretische benaderingen van de gewone monadische tweede-orde logica definiëren we een klasse van pariteitsautomaten die correspondeert met $\mathtt{MSO}_{\mathsf{T}}$. Vergelijkbaar met bekende

resultaten voor de monadische tweede-orde-logica over bomen, geven we een vertaling van $\mathtt{MSO}_\mathsf{T}$ naar deze automaten die waarheidbehoudend is over de klasse van boomachtige T-coalgebra's. Vervolgens identificeren we functoren T waarvoor de coalgebraïsche μ-calculus $\mu\mathtt{ML}_\mathsf{T}$ overeenkomt met het fragment van $\mathtt{MSO}_\mathsf{T}$ dat invariant is onder gedragsequivalentie. We benaderen dit op het niveau van éénstapstalen en laten zien dat het, om een coalgebraïsche karakteriseringsstelling te bewijzen, voldoende is om een adequate uniforme constructie voor de functor T te vinden. Als toepassing van dit resultaat verkrijgen we een gedeeltelijk nieuw bewijs voor de stelling van Janin-Walukiewicz, en bisimulatie-invariantieresultaten voor de multiset-functor (gegradeerde modale logica), en alle exponentiële polynomiale functoren. In het laatste deel van dit hoofdstuk gaan we dieper in op de monotone omgevingsfunctor $\mathcal{M}$, die een coalgebraïsche semantiek verschaft voor de monotone modale logica. Het blijkt dat er geen adequate uniforme constructie is voor $\mathcal{M}$. We lossen dit probleem op door een aangepaste versie van onze algemene karakteriseringsstelling te bewijzen.

In Hoofdstuk 5 bewijzen we een axiomatisch volledigheidsresultaat voor de coalgebraïsche μ-calculus. Hier volgen we dezelfde lijn als in hoofdstuk 4: een cruciale rol in onze bewijzen is weggelegd voor automaten en de notie van éénstapslogica. Door ideeën uit de automatentheorie en de coalgebra toe te passen, is het ons doel om Walukiewicz' bewijs van volledigheid voor de modale μ-calculus [Wal00] te generaliseren naar het niveau van coalgebra's. Onze belangrijkste bijdrage is dat we automaten expliciet in de bewijstheorie onderbrengen. Binnen deze automaten-theoretische benadering kunnen we twee belangrijke aspecten van de coalgebraïsche μ-calculus (en de standaard μ-calculus) onderscheiden:
de éénstapsdynamiek die gecodeerd ligt in de semantiek van de modale operatoren, en de combinatoriek die een rol speelt bij het omgaan met geneste dekpuntoperatoren. Dit onderscheid stelt ons in staat om grotendeels onafhankelijk met deze twee aspecten te werken. Meer in detail zijn de belangrijkste instrumenten die we gebruiken in onze automatentheoretische benadering twee soorten spellen voor modale automaten: het vervulbaarheidsspel en het consequentiespel, en twee speciale soorten modale automaten: disjunctieve en semi-disjunctieve automaten. Het consequentiespel tussen twee automaten is een originele bijdrage van onze aanpak. Het is een oneindig spel voor twee spelers dat gericht is op het tot stand brengen van structurele verbindingen tussen de automaten. Naast de disjunctieve automaten die bekend zijn uit het werk van Janin en Walukiewicz [JW95], definiëren we de klasse van semi-disjunctieve automaten en laten we zien dat deze semi-disjunctieve automaten, net als de disjunctieve, een relatief eenvoudige combinatorische sporentheorie hebben met betrekking tot de vervulbaarheids- en consequentiespellen. Als onze belangrijkste bijdrage generaliseren we het belangrijkste technische resultaat van Walukiewicz, namelijk, dat elke formule van de modale μ-calculus bewijsbaar de vertaling impliceert van een disjunctieve automaat, naar het niveau van coalgebras. Hieruit volgt de volledigheidsstelling vrijwel onmiddellijk.

Abstract

This dissertation studies the expressivity and completeness of the coalgebraic μ-calculus. This logic is a coalgebraic generalization of the standard μ-calculus, which creates a uniform framework to study different modal fixpoint logics. Our main objective is to show that several important results, such as uniform interpolation, expressive completeness and axiomatic completeness of the standard μ-calculus can be generalized to the level of coalgebras. To achieve this goal we develop automata and game-theoretic tools to study properties of coalgebraic μ-calculus.

In Chapter 3 we prove a uniform interpolation theorem for the coalgebraic μ-calculus. This theorem generalizes a result by D'Agostino and Hollenberg [DH00] to a wider class of fixpoint logics, including the monotone μ-calculus, which is the extension of monotone modal logic with fixpoint operators. To this aim, first we consider a key property of automata, namely closure under projection. We prove this property, which is known to hold for weak pullback-preserving functors, for a wider class of functors, i.e., functors with a certain type of relation lifting called a *quasi-functorial lax extension*. Then we show that closure under projection implies definability of the bisimulation quantifier in the language of the coalgebraic μ-calculus. Finally, we use this result to prove a uniform interpolation theorem for the coalgebraic μ-calculus.

In Chapter 4 we generalize the Janin-Walukiewicz theorem [JW96], which states that the modal μ-calculus captures exactly the bisimulation invariant fragment of monadic second-order logic, to the level of coalgebras. For that, we first introduce a notion of coalgebraic monadic second-order logic $\mathtt{MSO}_\mathsf{T}$ for coalgebras of type T. Following the tradition of automata-theoretic approaches toward monadic second-order logic we define a class of parity automata that corresponds to $\mathtt{MSO}_\mathsf{T}$. Similar to well-known results for monadic second-order logic over trees, we provide a translation from $\mathtt{MSO}_\mathsf{T}$ to these automata, which is truth-preserving on the class of *tree-like* T-coalgebras. We then proceed by identifying the class of functors T for which the coalgebraic μ-calculus $\mu\mathtt{ML}_\mathsf{T}$ corresponds to the fragment

of $\mathtt{MSO}_{\mathsf{T}}$ that is invariant under behavioural equivalence. We approach this at the level of one-step languages and show that to prove a coalgebraic characterization theorem, it suffices to find what we call an *adequate uniform construction* for the functor T. As applications of this result we obtain a partly new proof of the Janin-Walukiewicz theorem, and bisimulation invariance results for the bag functor (graded modal logic), and all exponential polynomial functors. In the last part of this chapter we consider in some detail the monotone neighborhood functor $\mathcal{M}$, which provides coalgebraic semantics for monotone modal logic. It turns out that there is no adequate uniform construction for $\mathcal{M}$. We resolve this problem by proving a second version of our general characterization theorem.

In Chapter 5 we prove an axiomatic completeness result for the coalgebraic μ-calculus. Here we follow the same track as in Chapter 4: a crucial role in our proofs is played by the notions of one-step logic and automata. Applying ideas from automata theory and coalgebra, our aim is to generalise Walukiewicz' proof of completeness for the modal μ-calculus [Wal00] to the level of coalgebras. Our main contribution is to bring automata explicitly into the proof theory. This automata-theoretic approach lets us distinguish two key aspects of the coalgebraic μ-calculus (and the standard μ-calculus): the *one-step dynamic* encoded in the semantics of the modal operators, and the *combinatorics* involved in dealing with nested fixpoints. This distinction allows us to work with these two features in a largely independent manner. More in detail, the main tools that we employ in our automata-theoretic approach are two kinds of games for modal automata: the *satisfiability game* and the *consequence game*, and two special kinds of modal automata: *disjunctive* and *semi-disjunctive automata*. The consequence game between two automata is an original contribution of our approach. It is an infinite two-player game concentrating on establishing structural connections between the automata. In addition to the disjunctive automata that are known from the work of Janin and Walukiewicz [JW95], we define the class of semi-disjunctive automata and show that similar to the disjunctive ones, semi-disjunctive automata also have a fairly simple *trace theory* (combinatorics) with regards to the satisfiability and consequence games. As our main result we then provide a generalization of Walukiewicz' main technical result, which states that every formula of the modal μ-calculus *provably* implies the translation of a disjunctive automaton, to the level of coalgebras. From this the completeness theorem is almost immediate.

Titles in the ILLC Dissertation Series:

ILLC DS-2018-13: **Jeroen Zuiddam**
Algebraic complexity, asymptotic spectra and entanglement polytopes

ILLC DS-2019-01: **Carlos Vaquero**
What Makes A Performer Unique? Idiosyncrasies and commonalities in expressive music performance

ILLC DS-2019-02: **Jort Bergfeld**
Quantum logics for expressing and proving the correctness of quantum programs

ILLC DS-2019-03: **András Gilyén**
Quantum Singular Value Transformation & Its Algorithmic Applications

ILLC DS-2019-04: **Lorenzo Galeotti**
The theory of the generalised real numbers and other topics in logic

ILLC DS-2019-05: **Nadine Theiler**
Taking a unified perspective: Resolutions and highlighting in the semantics of attitudes and particles

ILLC DS-2019-06: **Peter T.S. van der Gulik**
Considerations in Evolutionary Biochemistry

ILLC DS-2019-07: **Frederik Möllerström Lauridsen**
Cuts and Completions: Algebraic aspects of structural proof theory

ILLC DS-2020-01: **Mostafa Dehghani**
Learning with Imperfect Supervision for Language Understanding

ILLC DS-2020-02: **Koen Groenland**
Quantum protocols for few-qubit devices

ILLC DS-2020-03: **Jouke Witteveen**
Parameterized Analysis of Complexity

ILLC DS-2020-04: **Joran van Apeldoorn**
A Quantum View on Convex Optimization

ILLC DS-2020-05: **Tom Bannink**
Quantum and stochastic processes

ILLC DS-2020-06: **Dieuwke Hupkes**
Hierarchy and interpretability in neural models of language processing

ILLC DS-2020-07: **Ana Lucia Vargas Sandoval**
On the Path to the Truth: Logical & Computational Aspects of Learning

ILLC DS-2020-08: **Philip Schulz**
Latent Variable Models for Machine Translation and How to Learn Them

ILLC DS-2020-09: **Jasmijn Bastings**
A Tale of Two Sequences: Interpretable and Linguistically-Informed Deep Learning for Natural Language Processing

ILLC DS-2020-10: **Arnold Kochari**
Perceiving and communicating magnitudes: Behavioral and electrophysiological studies

ILLC DS-2020-11: **Marco Del Tredici**
Linguistic Variation in Online Communities: A Computational Perspective

ILLC DS-2020-12: **Bastiaan van der Weij**
Experienced listeners: Modeling the influence of long-term musical exposure on rhythm perception

ILLC DS-2020-13: **Thom van Gessel**
Questions in Context

ILLC DS-2020-14: **Gianluca Grilletti**
Questions & Quantification: A study of first order inquisitive logic

ILLC DS-2020-15: **Tom Schoonen**
Tales of Similarity and Imagination. A modest epistemology of possibility

ILLC DS-2020-16: **Ilaria Canavotto**
Where Responsibility Takes You: Logics of Agency, Counterfactuals and Norms

ILLC DS-2020-17: **Francesca Zaffora Blando**
Patterns and Probabilities: A Study in Algorithmic Randomness and Computable Learning

ILLC DS-2021-01: **Yfke Dulek**
Delegated and Distributed Quantum Computation

ILLC DS-2021-02: **Elbert J. Booij**
The Things Before Us: On What it Is to Be an Object

ILLC DS-2021-03: **Seyyed Hadi Hashemi**
Modeling Users Interacting with Smart Devices

ILLC DS-2021-04: **Sophie Arnoult**
Adjunction in Hierarchical Phrase-Based Translation

ILLC DS-2021-05: **Cian Guilfoyle Chartier**
A Pragmatic Defense of Logical Pluralism

ILLC DS-2021-06: **Zoi Terzopoulou**
Collective Decisions with Incomplete Individual Opinions

ILLC DS-2021-07: **Anthia Solaki**
Logical Models for Bounded Reasoners

ILLC DS-2021-08: **Michael Sejr Schlichtkrull**
Incorporating Structure into Neural Models for Language Processing

ILLC DS-2021-09: **Taichi Uemura**
Abstract and Concrete Type Theories

ILLC DS-2021-10: **Levin Hornischer**
Dynamical Systems via Domains: Toward a Unified Foundation of Symbolic and Non-symbolic Computation

ILLC DS-2021-11: **Sirin Botan**
Strategyproof Social Choice for Restricted Domains

ILLC DS-2021-12: **Michael Cohen**
Dynamic Introspection

ILLC DS-2021-13: **Dazhu Li**
Formal Threads in the Social Fabric: Studies in the Logical Dynamics of Multi-Agent Interaction

ILLC DS-2022-01: **Anna Bellomo**
Sums, Numbers and Infinity: Collections in Bolzano's Mathematics and Philosophy

ILLC DS-2022-02: **Jan Czajkowski**
Post-Quantum Security of Hash Functions

ILLC DS-2022-03: **Sonia Ramotowska**
Quantifying quantifier representations: Experimental studies, computational modeling, and individual differences

ILLC DS-2022-04: **Ruben Brokkelkamp**
How Close Does It Get?: From Near-Optimal Network Algorithms to Suboptimal Equilibrium Outcomes

ILLC DS-2022-05: **Lwenn Bussière-Carae**
No means No! Speech Acts in Conflict

ILLC DS-2023-01: **Subhasree Patro**
Quantum Fine-Grained Complexity

ILLC DS-2023-02: **Arjan Cornelissen**
Quantum multivariate estimation and span program algorithms

ILLC DS-2023-03: **Robert Paßmann**
Logical Structure of Constructive Set Theories

ILLC DS-2023-04: **Samira Abnar**
Inductive Biases for Learning Natural Language

ILLC DS-2023-05: **Dean McHugh**
Causation and Modality: Models and Meanings

ILLC DS-2023-06: **Jialiang Yan**
Monotonicity in Intensional Contexts: Weakening and: Pragmatic Effects under Modals and Attitudes

ILLC DS-2023-07: **Yiyan Wang**
Collective Agency: From Philosophical and Logical Perspectives

ILLC DS-2023-08: **Lei Li**
Games, Boards and Play: A Logical Perspective

ILLC DS-2023-09: **Simon Rey**
Variations on Participatory Budgeting

ILLC DS-2023-10: **Mario Giulianelli**
Neural Models of Language Use: Studies of Language Comprehension and Production in Context

ILLC DS-2023-11: **Guillermo Menéndez Turata**
Cyclic Proof Systems for Modal Fixpoint Logics

ILLC DS-2023-12: **Ned J.H. Wontner**
Views From a Peak: Generalisations and Descriptive Set Theory

ILLC DS-2024-01: **Jan Rooduijn**
Fragments and Frame Classes: Towards a Uniform Proof Theory for Modal Fixed Point Logics

ILLC DS-2024-02: **Bas Cornelissen**
Measuring musics: Notes on modes, motifs, and melodies

ILLC DS-2024-03: **Nicola De Cao**
Entity Centric Neural Models for Natural Language Processing