Modal fixpoint logic:
some model theoretic questions

Gaëlle Fontaine

Modal fixpoint logic:
some model theoretic questions

ILLC Dissertation Series DS-2010-09

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 904
1098 XH Amsterdam
phone: +31-20-525 6051
fax: +31-20-525 5206
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

Modal fixpoint logic: some model theoretic questions

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof.dr. D.C. van den Boom
ten overstaan van een door het college voor
promoties ingestelde commissie, in het openbaar
te verdedigen in de Agnietenkapel
op donderdag 9 december 2010, te 10.00 uur

door

Gaëlle Marie Marguerite Fontaine

geboren te Messancy, België.

Promotiecommissie:

Promotores:
Prof.dr. J. F. A. K. van Benthem
Prof.dr. Y. Venema

Overige Leden:
Prof.dr. E. Grädel
Prof.dr. L. Santocanale
Prof.dr. J. Väänänen
Prof.dr. I. Walukiewicz
Dr. B. D. ten Cate
Dr. M. Marx
Dr. D. Niwiński

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The investigations were supported by VICI grant 639.073.501 of the Netherlands Organization for Scientific Research (NWO).

Printed and bound by Printondemand-worldwide.

ISBN: 978-90-5776-215-4

Contents

Acknowledgments

There is a tradition in Amsterdam of writing extensive acknowledgments to the people one feels indebted to. While I, too, am deeply grateful to my family, teachers, supervisors, and important others on my path, I have chosen another way to express how I feel. I simply did a collage, putting good intentions or love or whatever I felt appropriate in the frame of each photo. The collage can be found on my webpage or inside any physical copy of the thesis.

Chapter 1

Introduction

The μ-calculus is an extension of modal logic with least and greatest fixpoint operators. Modal logic was originally developed by philosophers in the beginning of the 20th century [BRV01]. It aimed at combining the concepts of possibility and necessity with propositional logic. In the 1950s, the possible world semantics was introduced (see for instance [BRV01, BS84]) and since then, modal logic has proved to be an appealing language to reason about transition systems. In addition to its philosophical motivations, modal logic appears to be of interest in many other areas: essentially in any area that uses relational models as representation means. Examples include artificial intelligence, economics, linguistics and computer science.

In computer science, labelled transition systems are used to represent processes or programs. The nodes of the transition systems model the possible states of the process, whereas the edges represent the possible transitions from one state to another. The label of a given node carries all the local information about the node. Within that perspective, logic seems to be a natural tool to describe the properties of programs. This approach turned out to be particularly useful for specification and verification purposes.

Verification is concerned with correctness of programs. More specifically, given a program represented by a labeled transition system and a formula (called the specification), representing the intended behavior of the program, we want to check whether the formula holds in the transition system. This is nothing but the model checking problem for the logic used as a specification language.

In order to reason about programs, especially non-terminating ones, standard modal logic lacks expressive power. Usual types of correctness properties that one would like to formulate are safety ("nothing bad ever happens") or fairness ("something good eventually happens"). Typically, such types of properties are expressed using a recursive definition (*nothing bad ever happens* if nothing bad happens now and it is the case that for the next states *nothing bad ever happens*). So it seems reasonable to enrich modal logic with operators capturing some form of recursive principle. At the end of the 1970s, Amir Pnueli [Pnu77] argued that linear temporal logic (LTL), which is obtained by restricting to models based on the natural numbers and by adding the "until" operator to modal logic, could be a useful formalism in that respect. Since then, other temporal logics have been introduced, the most famous ones being computation tree logic [CE81] (CTL) and CTL* [EL86], and are considered as appropriate specification languages.

Around the same time, Vaughan Pratt [Pra76] and Andrzej Salwicki [Sal70] independently introduced Dynamic Logic. The basic idea of Dynamic Logic is to associate a modality $[\theta]$ with each program θ; the intuitive meaning of a formula $[\theta]\varphi$ is that φ holds in all states reachable after an execution of θ. In 1977, a propositional version of Dynamic Logic (PDL) was introduced by Michael Fischer and Richard Ladner [FL79]. One disadvantage of Dynamic Logic is that unlike temporal logics, it is not adequate for modeling non-terminating programs. Extensions of PDL that can capture some specific infinite behaviors (see for instance [Har84]) have been studied by Robet S. Streett [Str81, Str82] (Delta-PDL), David Harel and Vaughan Pratt [HP78] (PDL with a loop construct).

Fixpoint logics are formalisms that can deal with both non-terminating behavior and recursion in its most general form. The basic idea of fixpoint logics is to explicitly add operators that allow us to consider solutions of an equation of the form $f(x) = x$. For example, safety is a solution of the equation "$x \leftrightarrow$ (*nothing bad happens now* $\wedge$ *for all successors,* x)".

The first logic that was extended by means of fixpoint operators was first-order logic [Mos74]. The initial purpose was to establish a generalized recursion theory. In the context of semantics of programming languages, the use of fixpoints to enrich first-order logic goes back to Dana Scott, Jaco de Bakker [SdB69, Bak80] and David Park [Par69]. However, this required the development of a complex mathematical theory. A few years later, arose the idea of considering fixpoint extensions of modal logic. The most successful logic that came out of this approach is the μ-calculus. Works of of E. Allen Emerson, Edmund Clarke [EC80], David Park [Par80] and Vaughan Pratt [Pra81] prefigured the actual definition of the μ-calculus which was given in 1983 by Dexter Kozen [Koz83].

The μ-calculus is obtained by adding the least fixpoint operator μx and its dual, the greatest fixpoint operator νx, to the standard syntax for modal logic. Intuitively, the formula $\mu x.\varphi(x)$ is the smallest solution of the equation $x \leftrightarrow \varphi(x)$. Similarly, $\nu x.\varphi(x)$ is the biggest solution of this equation.

Not surprisingly, adding fixpoint operators to modal logic results in a significant increase of the expressive power. Most temporal logics (including LTL, CTL and CTL*) can be defined in terms of the μ-calculus [Dam94, BC96]. In fact, these logics usually fall inside to a rather small syntactic fragment of the μ-calculus (the fragment of alternation depth at most 2).

Moreover, on binary trees, it follows from various results [Rab69, EJ91, Niw88, Niw97] that the μ-calculus is equivalent to monadic second-order logic (MSO).

MSO is an extension of first-order logic, which allows quantification over subsets of the domain. It is also one of the most expressive logics that is known to be decidable on trees, whether they are binary or unranked (that is, there is no restriction on the number a successors of a node). Hence, it is not surprising that most specification languages are fragments of MSO. This means that on binary trees, the μ-calculus subsumes most specification languages.

On arbitrary structures, it is easy to see that the μ-calculus is a proper fragment of MSO. A key result concerning the expressive power of the μ-calculus is the Janin-Walukiewicz theorem [JW96]: an MSO formula φ is equivalent to a μ-formula iff φ is invariant under bisimulation. Bisimulations are used to formalize the notion of behavioral equivalence. The idea is that when specifying behaviors, one is interested in the behavior of programs rather than the programs themselves. Hence, a specification language should not distinguish two programs displaying the same behavior. On a theoretical level, this boils down to the requirement that a formula used for specification is invariant under bisimulation. So the Janin-Walukiewicz theorem basically says that the μ-calculus is the "biggest" relevant specification language which a fragment of MSO.

It is also interesting to mention that the Janin-Walukiewicz theorem extends an important result of modal logic proved by Johan van Benthem [Ben76]: a first-order formula φ is equivalent to a modal formula iff φ is invariant under bisimulation. In the area of modal and temporal logics, the most common logics used as yardsticks (references against which the other logics are compared) are first-order logic and MSO. It follows from the Janin-Walukiewicz theorem and van Benthem characterization that the μ-calculus is the counterpart of MSO, in the same way that modal logic is the counterpart of first-order logic. From a theoretical point of view, this makes the μ-calculus an attractive extension of modal logic.

In order for a logic to be used as a specification language, it is important that there is a good balance between its expressive power and its complexity. By complexity, we usually refer to the complexities of the model checking problem and the satisfiability problem. The model checking problem was already mentioned before and consists in deciding whether a given formula holds on a given finite structure. The satisfiability problem consists in deciding whether for a given formula, there exists a structure in which the formula is true.

The model-checking problem for the μ-calculus is NP $\cap$ co-NP; the result can even be strengthened to[1] UP $\cap$ co-UP [Jur98]. To obtain this upper bound, the idea is to use the connection between parity games and the μ-calculus, which was observed by several authors, including E. Allen Emerson, Charanjit Jutla [EJ88]

[1]A non-deterministic Turing machine is unambiguous if for every input, there is at most one accepting computation. The complexity class UP (Unambiguous Non-deterministic Polynomial-time) is the class of languages problems solvable in polynomial time by an unambiguous non-deterministic Turing machine (for more details on this model of computation, see for instance [Pap94]).

and Colin Stirling [Sti95]. Parity games play a fundamental role in game theory. A parity game is a game of which the winning condition is specified by a map assigning a bounded priority to each position of the board game. The winner of an infinite match depends of the priorities encountered infinitely often during the match.

It can be shown that the model checking problem for the μ-calculus is equivalent to the problem of solving parity games, which consists in deciding which player has a winning strategy in a given parity game with an initial position. It was proved independently by Andrzej Mostowski [Mos91], E. Allen Ermerson and Charanjit Jutla [EJ91] that a winning strategy in a parity game may be assumed to be positional. That is, the move dictated by the strategy at a position of a match only depends on the actual position, and not on what has been played before reaching the position. This result implies that the complexity of solving a parity game is NP $\cap$ co-NP. Later Marcin Jurdziński gave a tighter complexity bound, which is UP $\cap$ co-UP [Jur98]. It is an important open problem what is the exact complexity of solving parity games and in particular, whether this complexity is polynomial.

The satisfiability problem for the μ-calculus is EXPTIME-complete [EJ88]. This was shown by E. Allen Ermerson and Charanjit Jutla, using automata theoretic methods. The basic idea of the automata theoretic approach is to associate with each formula an automaton that accepts exactly the structures in which the formula is true. It follows that solving the satisfiability problem for a formula is reduced to checking non-emptiness of an automaton. The non-emptiness problem for an automaton is to decide whether there exists a structure accepted by the automaton.

Similarly to the model checking problem, verifying whether there is a structure accepted by a given automaton is equivalent to checking whether a player has a winning strategy in an initialized infinite game associated with the automaton [NW96]. Furthermore, a winning strategy in the game would directly induce a structure accepted by the automaton.

The automata theoretic approach has also been useful for establishing other important results. For example, the Janin-Walukiewicz theorem mentioned earlier is proved using the correspondence between formulas and automata. In addition, David Janin and Igor Walukiewicz showed that there is a disjunctive normal form for the formulas of the μ-calculus [JW95a]; a crucial part of the proof is based on the fact that we can determinize automata operating on infinite words.

The goal of these last few paragraphs was not only to give some insight about the complexity of the μ-calculus, but also to illustrate how the theory of the μ-calculus benefits from the connections between different formalisms, such as game theory, automata theory and, obviously, logic. This feature is not specific to the μ-calculus: the same holds for all temporal logics and on a broader scale, this is a phenomenon that is characteristic of many areas of mathematics. Nevertheless, it is still a very enjoyable aspect of the μ-calculus.

Now, from what we have seen, the μ-calculus seems a well-suited specification language, as it combines a great expressive power and manageable decision procedures. But there is a drawback: the μ-calculus is probably not the most understandable way to specify behaviors. Most people would have a difficult time understanding the meaning of a formula of the μ-calculus with alternation depth greater than 2. In that respect, other temporal logics, such as LTL, CTL and CTL*, are more convenient.

That being said, it is still the case that: the μ-calculus provides a uniform framework containing all specification languages; it is characterized by a rich and interesting mathematical theory; despite its difficult interaction with human thinking, it has direct practical applications in the area of specification. For these reasons, the μ-calculus has become a significant formalism in the landscape of modal logic and specification.

In this thesis, we consider some important theoretical aspects of the μ-calculus, namely axiomatizability, expressivity, decidability and complexity. One running topic through the thesis is exploring the μ-calculus through its "fine-structure". Or to put it differently, we investigate the μ-calculus by focussing on restricted class of models, special fragments of the language, etc. This approach is motivated by the fact that the μ-calculus is a complex and powerful system.

In Chapters 3 and 4, we restrict our attention to special classes of models, namely trees. Trees are particularly relevant structures for any logic that is invariant under bisimulation. Such logics have the tree property; that is, a formula is satisfiable iff it is satisfiable in a tree. In Chapter 3, we consider the question of the axiomatization of the μ-calculus. This problem is notorious for its difficulty, but it turns out that when focussing on finite trees, the proof of the completeness of the axiomatization becomes much simpler. In Chapter 4, we deal with the question of the expressive power of the μ -calculus in the context of frames (which are transition systems without any labeling). Again we investigate this question in the restricted setting of trees.

In Chapter 5, instead of having restrictions on the structures, we consider some special fragments of the language. The main contribution of that chapter concerns a characterization of what we call the continuous fragment. As we will see, the continuous fragment is a good candidate for approximating the "computational part" of the μ-calculus. Chapter 6 is slightly different than the other chapters, as it concerns the formalism XPath [BK08]. The goal of that chapter is to show how results in the area of modal logic can help for the understanding of XPath. One of these results was shown in Chapter 5.

The last chapter is also concerned with special classes of models, but the perspective with respect to the "fine-structure" approach is in effect reversed. Instead of looking at specific classes of models, we consider more general structures, namely coalgebras. Coalgebras are an abstract version of evolving systems and generalize the notion of labelled transition systems or Kripke models. In

Chapter 7, we extend the automata theoretic approach for the μ-calculus to the setting of coalgebras.

We give now a more detailed overview of the content of each chapter.

Axiomatizability

Chapter 3 In the same paper where he introduced the μ-calculus [Koz83], Dexter Kozen also suggested an axiomatization. The completeness of that axiomatization remains an open problem for many years. Eventually Igor Walukiewicz [Wal95] provided a proof which is based on automata theory, game theory and classical logic tools such as tableaux. The proof is also well-known for its difficulty.

In Chapter 3, we propose an easier proof in the restricted setting of the μ-calculus on finite trees. On finite trees the expressive power of the μ-calculus is rather limited: any formula of the μ-calculus is equivalent to a formula of alternation depth 1. Nevertheless, the completeness proof we provide is not a simplification of the original proof given by Igor Walukiewicz. The technique we use consists in combining an Henkin-type semantics for the μ-calculus together with model theoretic methods (inspired by the work of Kees Doets [Doe89]).

We hope that this different approach towards completeness might contribute modestly to a better understanding of the problem. This method might also help to prove other completeness results and we give two examples in the chapter. The first one concerns a complete axiomatization of the graded μ-calculus on finite trees. The other example applies to extensions of the μ-calculus with shallow axioms [Cat05] on finite trees.

This chapter is based on the paper "An easy completeness proof for the μ-calculus on finite trees", co-authored by Balder ten Cate and published in the proceedings of FOSSACS 2010.

Expressive power

Chapter 4 The Janin-Walukiewicz theorem concerns the expressive power of the μ-calculus on the level of models, i.e. transition systems equipped with a valuation (stating which atomic propositions are true at each node). In Chapter 4, we shift to the context of frames, which are transition systems without any valuation. The truth of a formula in a frame involves a second-order quantification over all possible valuations.

As opposed to the case of modal logic, very little is known about the expressive power of the μ-calculus on frames. This chapter compares the expressive power of the μ-calculus and MSO on frames, in the particular case when the frames have a tree structure. More specifically, we provide a characterization of those MSO formulas that are equivalent on trees (seen as frames) to a formula of the

μ-calculus. This characterization is formulated in terms of natural structural criteria, namely closure under subtrees and p-morphic images. The result might be compared to the Janin-Walukiewicz theorem, the main differences being the context (frames vs. models), and our more restricted setting (trees vs. arbitrary models).

This chapter is based on the paper "Frame definability for classes of trees in the μ-calculus" co-authored by Thomas Place and published in the proceedings of MFCS 2010.

Chapter 5 We present syntactic characterizations of semantic properties of the μ-calculus, the two main ones being the continuous fragment and completely additive formulas. A formula φ of the μ-calculus is continuous in a proposition letter p iff the truth of φ at a given node only depends on the existence of finitely many points making p true. The name "continuity" originates from the direct connection between this fragment and the notion of Scott continuity, widely used in theoretical computer science. One of the most interesting features of a continuous formula is that its least fixpoint can be constructed in at most ω steps.

The completely additive fragment corresponds to distributivity over countable unions, which, in the case of the μ-calculus, was studied by Marco Hollenberg [Hol98b]. Using a characterization of this fragment, Marco Hollenberg obtained an extension of the Janin-Walukiewicz theorem for μ-programs [Hol98b] (which is what motivated the study of the completely additive fragment). Inspired by our results for the continuous fragment, we propose an alternative proof for the characterization of the completely additive fragment. Unlike the original argument, this proof provides a direct translation from the completely additive fragment to the adequate syntactic fragment.

This chapter is based on the paper "Continuous fragment of the μ-calculus" published in the proceedings of CSL 2008 and on a submitted paper "Syntactic characterizations of semantic fragments of the μ-calculus" co-authored by Yde Venema.

Chapter 6 This chapter is concerned with the expressive power of a fragment of CoreXPath. XPath is a navigation language for XML documents and CoreXPath has been introduced to capture the logical core of XPath [GKP05]. The basic idea of the chapter is to exploit the tight link between CoreXPath and modal logic. CoreXPath is essentially a modal logic evaluated on specific models (which are finite trees with two basic modalities). The main difference between modal logic and CoreXPath is that the syntax for XPath is two-sorted: it contains both formulas (which corresponds to subsets of the model) and programs (which corresponds to binary relations).

In this chapter, we combine well-known results of the μ-calculus in order to obtain results about the expressive power of CoreXPath. One of the results that we use is the adaptation of the Janin-Walukiewicz theorem for μ-programs. This result was (re-)proved in the previous chapter.

This part of the thesis is based on the paper "Modal aspects of XPath" co-authored by Balder ten Cate and Tadeusz Litak and which is an invited paper for M4M 2007.

Decidability and complexity

Chapter 7 In this chapter, we extend the notion of automaton to the setting of coalgebras. The aim of the theory of coalgebras is to provide a uniform framework to describe evolving systems, Kripke models being a key example. It is then not surprising that the definition of coalgebraic logic was inspired by modal logic. Roughly, there are two kinds of coalgebraic logic: one using nabla (∇) operators [Mos99], the other being based on the notion of predicate lifting [Pat03]. Similarly to what happens in modal logic, we can extend coalgebraic logic with fixpoint operators and obtain a coalgebraic μ-calculus.

As mentioned earlier, the automata theoretic approach has been very successful for the μ-calculus. Automata for the coalgebraic μ-calculus using nabla operators have been introduced by Yde Venema [Ven06b]. The goal of this chapter is to contribute to the development of the automata theoretic approach for coalgebraic μ-calculus based on predicate liftings. More specifically, we introduce the notion of an automaton associated with a set of predicate liftings. We use these automata to prove the decidability of the satisfiability problem and obtain a small model property. We also obtain a double exponential bound on the complexity of the satisfiability problem.

This chapter is based on the paper "Automata for coalgebras: an approach via predicate liftings" co-authored by Raul Leal and Yde Venema and published in the proceedings of ICALP 2010.

We mentioned earlier that there exist connections between the μ-calculus and other formalisms, the two major ones being game theory and automata theory. We can think of the exploitation of these connections as being methods for approaching the μ-calculus. From that point of view, the five chapters that we described can be seen, independently from the content, as a playground for these methods: how they interact and what they can be used for.

In the following chapters, we often make use of the links with automata theory and game theory. These two theories are themselves deeply connected to each other. One of the main reasons (in our context) is that the terminology of game theory is particularly adequate to describe the run of an automaton on branching structures. The fact that a tree is accepted by an automaton is usually reduced to the existence of a winning strategy for a player in a game associated with the automaton.

The most obvious place in the thesis where logic, automata and games are intertwined is Chapter 7. This chapter can be seen as an illustration of the efficiency of the automata theoretic approach. Automata are in effect an alternative way of thinking about formulas. One of their advantages is that they capture the algorithmic aspect of the μ-calculus, while not having the logical complexity resulting from an inductive definition (unlike formulas). Game theory also comes into play in this chapter: it offers a nice framework to interpret automata and formulate proofs.

Even though there is no explicit mention of automata in the Chapters 4 and 5, we could still think of these chapters as using the connections between automata, games and logic. A useful result in both chapters is the equivalence between the model checking problem for a formula and solving a certain parity game called the evaluation game. The evaluation game is the acceptance game of the alternating μ-automaton associated with the formula.

Now we do not only use game theory and automata theory as formalisms to represent formulas in a more intuitive or convenient way, but also as reservoirs of available results. For example, proofs in Chapter 7 rely on the fact that a strategy in a regular game may be assumed to be a finite memory strategy. Interestingly enough, a classical automata result (the determinization of automata on infinite words) plays an important role in the proof of this fact. Another (less direct) example is given in Chapters 4 and 5. The proofs in these chapters use in an essential way the existence of a disjunctive normal form for fixpoint logics. As mentioned earlier, a key ingredient for the proof of this last result is of automata theoretic nature (and is again the determinization of automata on infinite words).

Finally, to the existing arsenal, we add a new method in Chapter 3. As explained in the overview of this chapter, this method is inspired by model theory. The idea is to introduce a notion of rank which plays the same role in our proof as the notion of quantifier depth in model theory. Using this notion, we can transfer model theoretic arguments that work by induction on the alternation depth to the setting of the μ-calculus. In our case, the model theoretic argument originates from Kees Doets' work [Doe89].

Chapter 2

Preliminaries

We introduce the notation and results that we will need throughout this thesis. Not surprisingly, most of this chapter concerns the μ-calculus. For a detailed survey concerning the μ-calculus, we refer the reader to [BS07], [AN01], [GTW02] and [Ven08a].

2.1 Syntax of the μ-calculus

In this section, we introduce the syntax for the μ-calculus and some related terminology. We present the μ-calculus in two different syntactic formats. The first format that we denote by μML, consists in adding fixpoint operators to standard modal logic. The second format is obtained by replacing the modal operators $\Diamond$ and $\Box$ by the operator ∇ (nabla).

The set μML corresponds to the syntax for μ-calculus originally introduced by Dexter Kozen [Koz83]. The advantage is that the set μML corresponds to a natural way of enriching modal logic with fixpoints. On the other hand, the second format is closer to the automata theoretic approach (see Section 2.4).

μ-formulas Let *Prop* be a set of *proposition letters*, *Act* a set of *actions* and *Var* an infinite set of *variables*. We assume that $\mathit{Prop} \cap \mathit{Var} \neq \emptyset$. A *literal* is a formula of the form p or $\neg p$, where $p \in \mathit{Prop}$. The set μML of *μ-formulas* (over *Prop*, *Act* and *Var*) is inductively given as follows:

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

where p belongs to *Prop*, a belongs to *Act* and x belongs to *Var*.

The set $\mu\mathrm{ML}^\nabla$ of *μ-formulas in ∇-form* (over *Prop*, *Act* and *Var*), is inductively defined by:

$$\varphi ::= \top \mid x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \alpha \bullet \nabla_a \Phi \mid \mu x.\varphi \mid \nu x.\varphi,$$

where x belongs to *Var*, a belongs to *Act*, Φ is a finite subset of $\mu\mathrm{ML}^{\nabla}$ and α is a conjunction of literals and variables.

In general, by *μ-formula*, we mean either a formula in μML or in $\mu\mathrm{ML}^{\nabla}$. The notion of *subformula* is defined in the usual way and given a formula φ, we let $Sfor(\varphi)$ denote the collection of subformulas of φ. If ψ is a subformula of φ, we write $\psi \trianglelefteq \varphi$ and if in addition, $\varphi \neq \psi$, we write $\psi \triangleleft \varphi$.

A *modal formula* is a μ-formula φ such that $Sfor(\varphi)$ does not contain any formula of the form $\mu x.\psi$ or $\nu x.\psi$. A *propositional formula* is a μ-formula φ such that $Sfor(\varphi)$ does not contain any formula of the form $\mu x.\psi$, $\nu x.\psi$, $\Diamond_a \varphi$ or $\Box_a \varphi$.

An occurrence of a variable x in a formula φ is *bound* if x is in the scope of an operator μx or νx. Otherwise, the occurrence of x is *free*. A *μ-sentence* is a μ-formula that do not contain any free variable. *Substitutions* are defined as usual. If φ and ψ are μ-formulas and if v is either a proposition letter or a variable, we denote by $\varphi[v/\psi]$ the formula obtained by replacing in φ each free occurrence of v by ψ. Note that if all the occurrences of a variable x are bound in φ, we have $\varphi[x/\psi] = \varphi$.

A μ-formula φ is *well-named* if for every variable x the following holds

- every occurrence of x is free or
- every occurrence of x is bound and there is a unique subformula of the form $\eta_x x.\delta_x$. This unique subformula is called the *unfolding of x* and is denoted as $\eta_x^{\varphi} x.\delta_x^{\varphi}$. We call x a μ-variable if $\eta_x^{\varphi} = \mu$, and a ν-variable if $\eta_x^{\varphi} = \nu$. If φ is clear from the context, we simply write $\eta_x.\delta_x$.

Convention Throughout this thesis, unless specified otherwise, we fix a set *Prop* of proposition letters and an infinite set *Var* of variables. Moreover, in most of the chapters, we assume the set of actions to be a singleton and in this case, there is no confusion to write $\Diamond$ instead of $\Diamond_a$ and $\Box$ instead of $\Box_a$. The only reason for this restriction is to make the presentation smoother, but all the results can be extended to the case where we have more than one action.

The way we defined the set μML, only allows the application of the negation symbol to proposition letters. As we shall see in the next section, it is equivalent to allow the application of the negation to any formula but require that in formulas of the form $\mu x.\varphi$ and $\nu x.\varphi$, x is under the scope of an even number of negation symbols. The advantage of this last presentation is that it enables us to lower down the number of primitive symbols, as we may treat $\bot$, $\wedge$, $\Box_a$ and νx as definable symbols (instead of primitive symbols). However, the presentation we gave here, interacts better with the game semantics, which comes often into play in this thesis.

It will still be sometimes useful to be able to apply the negation symbol to a formula in μML. For that purpose, we introduce the abbreviation $\neg\varphi$ (where $\varphi \in \mu\mathrm{ML}$).

The symbol ¬ Given a formula φ in μML, we define $\neg\varphi$ by induction on φ as follows:

$$
\begin{array}{rcl|rcl}
\neg\top & = & \bot, & \neg\Box_a\varphi & = & \Diamond_a\neg\varphi, \\
\neg\bot & = & \top, & \neg\Diamond_a\varphi & = & \Box_a\neg\varphi, \\
\neg(\varphi\wedge\psi) & = & \neg\varphi\wedge\neg\psi, & \neg\mu x.\varphi & = & \nu x.\neg\varphi[\neg x/x], \\
\neg(\varphi\vee\psi) & = & \neg\varphi\wedge\neg\psi, & \neg\nu x.\varphi & = & \mu x.\neg\varphi[\neg x/x],
\end{array}
$$

where p belongs to *Prop*, x belongs to *Var* and $\varphi[\neg x/x]$ is the formula φ in which all occurrences of $\neg x$ are replaced with x.

If we consider a subformula ψ of a formula φ, it might be the case that some variables x bound in φ, become free in ψ. In a sense, these variables lose the role they played in φ. If we want to restore the role of these variables in ψ, we can simply replace each variable x by the formula $\eta_x.\delta_x$. Formally, we have the following definition.

Dependency order and expansion Given a well-named formula φ, we define the *dependency order* $<_\varphi$ on the bound variables of φ as the least strict partial order such that $x <_\varphi y$ if δ_x is a proper subformula of δ_y.

If $\{x_1, \dots, x_n\}$ is the set of variables occurring in φ_0, where we may assume that $i < j$ if $x_i <_\varphi x_j$, we define the *expansion* $e_\varphi(\psi)$ of a subformula ψ of φ as:

$$e(\psi) \;:=\; \psi[x_1/\eta_{x_1}.\delta_{x_1}]\dots[x_n/\eta_{x_n}.\delta_{x_n}].$$

That is, we substitute first x_1 by δ_{x_1} in ψ; in the obtained formula, we substitute x_2 by δ_{x_2}, etc. If no confusion is likely we write $e(\psi)$ instead of $e_\varphi(\psi)$.

Let us mention that the order for performing the substitutions is crucial. We illustrate this by an example. Consider the μ-sentence $\varphi = \mu x.(\mu y.x \wedge y)$ and let ψ be the subformula $x \wedge y$. It is easy to see that $y <_\varphi x$. Now the formula $\psi_1 := \psi[\eta_y.\delta_y/y][\eta_x.\delta_x/x]$ is equal to $\mu y.(\varphi \wedge y) \wedge \varphi$, whereas the formula $\psi_2 := \psi[\eta_x.\delta_x/x][\eta_y.\delta_y/y]$ is equal to $\varphi \wedge \mu y.(x \wedge y)$. The variables play exactly the same role in φ and in ψ_1 (which is the expansion of ψ). This does not hold for the formula ψ_2: there is an occurrence of the variable x that is free in ψ_2. So we see that it is important to start the substitution with the variable the unfolding of which is the innermost subformula of φ.

Closure of a formula The *closure* $Cl(\varphi)$ of a formula φ is the smallest set of formulas such that

$\varphi \in Cl(\varphi)$,
if $\neg p \in Cl(\varphi)$, $p \in Cl(\varphi)$,
if $\psi \vee \chi$ or $\psi \wedge \chi$ belongs to $Cl(\varphi)$, then both $\psi, \chi \in Cl(\varphi)$,
if $\Diamond_a \psi$ or $\Box_a \varphi$ belongs to $Cl(\varphi)$, then $\psi \in Cl(\varphi)$,
if $\alpha \bullet \nabla_a \Phi$ belongs to $Cl(\varphi)$, then $\alpha \in Cl(\varphi)$ and $\Phi \subseteq Cl(\varphi)$,
if $\mu x.\psi \in Cl(\varphi)$, then $\psi[x/\mu x.\psi] \in Cl(\varphi)$,
if $\nu x.\psi \in Cl(\varphi)$, then $\psi[x/\nu x.\psi] \in Cl(\varphi)$.

So the closure of a formula can be seen as the analog of the Fischer-Ladner closure for PDL.

An immediate adaptation of a proof in [Koz95] shows that if φ is well-named, $Cl(\varphi)$ is equal to the set $\{e_\varphi(\psi) \mid \psi \in \mathit{Sfor}(\varphi)\}$.

Size of a formula Following [KVW00], we define the size of a μ-formula φ, notation: $\mathit{size}(\varphi)$, as the cardinality of the set $Cl(\varphi)$.

The size of a formula is related to the number of nodes in the DAG (directed acyclic graph) representation of the formula. There are two usual ways to represent a formula: as a tree or as a DAG. We give some intuition on how these two representations work.

Given a formula φ, we can define a tree the nodes of which are labeled with subformulas of φ. The root is labeled with the formula φ. Given a node labeled with a formula ψ, we create the children of the node according to the form of ψ. If ψ is a proposition letter or a variable, the node has no children. If ψ is a disjunction of the form $\psi_1 \vee \psi_2$, the node has two children, one labeled with ψ_1 while the other one is labeled with ψ_2. If ψ is of the form $\mu x.\chi$, the node has one child labeled with χ. It is easy to imagine how to proceed in the other cases.

Now we can also represent φ as DAG. Each node of the DAG corresponds to a subformula of φ. There is an edge from a node associated with a formula ψ to a node associated with a formula χ if χ is an immediate successor of ψ with respect to the order $\trianglelefteq$.

In the case we represent a formula as a tree, the number of nodes depends on the number of subformulas and how many times each subformula occurs in φ. In the case we represent a well-named formula φ as a DAG, the number of nodes is linear in the size of φ.

2.1.1. Remark. There are two main ways to define the size of a formula: either as the cardinality of the closure of the formula or as the number of symbols of the formula. As explained earlier, the first definition is related to the number of nodes in the DAG representation of the formula, while the latter definition corresponds to the number of nodes in the tree representation.

The fact that we opted for the first definition of the size is not crucial. In fact, most of the (few) complexity results presented here are also true with the other alternative definition of size. Our choice for the definition of the size is dictated by the fact that the proofs for the complexity results (of this thesis) are usually based on an automata theoretic approach. The number of states of an automaton associated to a formula is directed related to the cardinality of the closure of the formula (see proof of Proposition 2.4.2).

Finally, we introduce the notion of alternation depth. There are in fact several possibilities to define the alternation depth. Since this notion does not play a crucial role in this thesis, we use what is called the "simple-minded" definition in [BS07]. For other definitions of alternation depth, we also refer to [BS07].

Alternation depth Let φ be a μ-formula. An *alternating μ-chain* in φ of length k is a sequence

$$\varphi \trianglerighteq \mu x_1.\psi_1 \vartriangleright \nu x_2.\psi_2 \vartriangleright \cdots \vartriangleright \mu_k/\nu_k.\psi_k,$$

where for all $i \in \{1, \ldots, k\}$, x_i is not free in every ψ such that $\psi_i \geq \psi \geq \psi_{i+1}$. We let $max^{\mu}(\varphi)$ be the maximal length of an alternating μ-chain in φ. We define in a similar way $max^{\nu}(\varphi)$. The *alternation depth* of a μ-formula is the maximum of $max^{\mu}(\varphi)$ and $max^{\nu}(\varphi)$.

2.2 Semantics for the μ-calculus

The structures on which we interpret the μ-formulas are the usual structures for modal logic.

Kripke frames A *(Kripke) frame* is a pair $(W, (R_a)_{a\in Act})$, where W is a set and for all $a \in Act$, R_a a binary relation on W. W is the *domain* of the frame and R is the *accessibility relation* or *transition relation*. Elements of w are called *nodes*, *states* or *points*.

If $(W, (R_a)_{a\in Act})$ is a Kripke frame and (w, v) belongs to R_a, we say that w is an *a-predecessor* of v and v is an *a-successor* of w. Given a binary relation $R \subseteq W \times W$, we denote by $R[w]$ the set $\{v \in W \mid (w, v) \in R\}$. The transitive closure of R is denoted as R^+; elements of the set $R_a^+[w]$ are called (proper) *a-descendants* of w.

A *subframe* of a frame $(W, (R_a)_{a\in Act})$, is a frame of the form $(W', (R'_a)_{a\in Act})$, where $W' \subseteq W$ and for all $a \in Act$, $R'_a = R_a \cap (W' \times W')$. Given a point w in W, the *subframe generated* by w is the unique subframe, the domain of which is $\{w\} \cup \left(\bigcup_{a\in Act} R_a\right)^+[w]$.

Kripke models A *(Kripke) model* (over *Prop*) is a triple $(W, (R_a)_{a\in Act}, V)$ where $(W, (R_a)_{Act})$ is a Kripke frame and $V : Prop \longrightarrow \mathcal{P}(W)$ a *valuation*. A *pointed model* is a pair $(\mathcal{M}, w)$, where $\mathcal{M}$ is a Kripke model and w belongs to the domain of $\mathcal{M}$.

A *submodel* of a model $(W, (R_a)_{a\in Act}, V)$, is a model of the form $(W', (R'_a)_{a\in Act}, V')$, where $(W', (R'_a)_{a\in Act})$ is a subframe of $(W, (R_a)_{a\in Act})$ and for all $p \in Prop$, $V'(p) = V(p) \cap W'$. Given a point w in a model $\mathcal{M}$, the *submodel generated* by w is is the unique submodel of $\mathcal{M}$, the domain of which is $\{w\} \cup \left(\bigcup_{a\in Act} R_a\right)^+ [w]$.

We start by giving the semantics for the modal formulas. The boolean connectives are interpreted as usual. The operator $\Diamond_a$ and $\Box_a$ are interpreted as in the setting of modal logic. Next, a formula $\alpha \bullet \nabla_a\Phi$ is true at a point w in a model if α is true at w and

$$\begin{aligned} &\text{for all } v \in R_a[w], \text{ there is } \varphi \in \Phi \text{ such that } \varphi \text{ is true at } v, \\ &\text{for all } \varphi \in \Phi, \text{ there is } v \in R_a[w] \text{ such that } \varphi \text{ is true at } v. \end{aligned} \tag{2.1}$$

Alternatively, if we let $[\![\varphi]\!]$ be the set of points at which φ is true, (2.1) is equivalent to the fact that the set of $\bigcup\{[\![\varphi]\!] \mid \varphi \in \Phi\}$ contains the a-successors of w and that each set $[\![\varphi]\!]$ with $\varphi \in \Phi$, has a non-empty intersection with $R_a[w]$.

Given the semantics of $\alpha \bullet \nabla_a\Phi$, the notation $\alpha \wedge \nabla_a\Phi$ might seem more appropriate. The reason for writing $\alpha \bullet \nabla_a\Phi$ will become clear in Section 2.4.3. This notation allows us to formulate the notion of disjunctive formula in an easier way.

In the area of modal logic, the first explicit occurrences of the ∇ connective can be found in the work of Jon Barwise and Lawrence Moss [BM96] and in that of David Janin and Igor Walukiewicz [JW95b]. We also would like to mention that ∇ corresponds to the relation lifting of the satisfiability relation. Given a model with domain W, we can think of the satisfiability relation $\Vdash$ as a relation between W and the set $\mu\mathrm{ML}^\nabla$. A pair (w, φ) belongs to the satisfiability relation if φ is true at w. Using the notion of relation lifting, we can lift this relation into a relation $\overline{\mathcal{P}}(\Vdash)$ between $\mathcal{P}(W)$ and $\mathcal{P}(\mu\mathrm{ML})$ (for more details, see for instance [Ven06a]). It turns out that $\nabla_a\Phi$ is true at a point w iff $(R_a[w], \Phi)$ belongs to $\overline{\mathcal{P}}(\Vdash)$. This equivalence was the key ingredient for the definition of coalgebraic modal logic [Mos99].

Semantics for modal formulas Fix a Kripke model $\mathcal{M} = (W, (R_a)_{a\in Act}, V)$. Given a modal formula φ and an assignment $\tau : Var \longrightarrow \mathcal{P}(W)$, we will define the *meaning* of φ as a set $[\![\varphi]\!]_{\mathcal{M},\tau} \subseteq W$. In case a point w belongs to $[\![\varphi]\!]_{\mathcal{M},\tau}$, we say that φ is *true* at w. The definition of $[\![\varphi]\!]_{\mathcal{M},\tau}$ proceeds by induction on φ:

$$\begin{array}{rcl}
[\![\top]\!]_{\mathcal{M},\tau} & = & W, \\
[\![\bot]\!]_{\mathcal{M},\tau} & = & \emptyset, \\
[\![p]\!]_{\mathcal{M},\tau} & = & V(p), \\
[\![\neg p]\!]_{\mathcal{M},\tau} & = & W \setminus V(p), \\
[\![x]\!]_{\mathcal{M},\tau} & = & \tau(x), \\
[\![\varphi \vee \psi]\!]_{\mathcal{M},\tau} & = & [\![\varphi]\!]_{\mathcal{M},\tau} \cup [\![\psi]\!]_{\mathcal{M},\tau}, \\
[\![\varphi \wedge \psi]\!]_{\mathcal{M},\tau} & = & [\![\varphi]\!]_{\mathcal{M},\tau} \cap [\![\psi]\!]_{\mathcal{M},\tau}, \\
[\![\Diamond_a \varphi]\!]_{\mathcal{M},\tau} & = & \{w \in W \mid R_a[w] \cap [\![\varphi]\!]_{\mathcal{M},\tau} \neq \emptyset\}, \\
[\![\Box_a \varphi]\!]_{\mathcal{M},\tau} & = & \{w \in W \mid R_a[w] \subseteq [\![\varphi]\!]_{\mathcal{M},\tau}\}, \\
[\![\alpha \bullet \nabla\Phi]\!]_{\mathcal{M},\tau} & = & [\![\alpha]\!]_{\mathcal{M},\tau} \cap [\![\nabla\Phi]\!]_{\mathcal{M},\tau}, \\
[\![\nabla\Phi]\!]_{\mathcal{M},\tau} & = & \{w \in W \mid R[w] \subseteq \bigcup\{[\![\varphi]\!]_{\mathcal{M},\tau} \mid \varphi \in \Phi\} \\
 & & \text{and for all } \varphi \in \Phi, [\![\varphi]\!]_{\mathcal{M},\tau} \cap R[w] \neq \emptyset\}.
\end{array}$$

To define the semantics for the μ-formulas, it remains to interpret the fixpoint operators μ and ν. The idea is to view formulas as set-theoretic maps and to think of the connectives μx and νx as least and greatest fixpoints of these maps. We start by defining the map $\varphi_x^{\mathcal{M},\tau}$ associated with a given a formula φ and a variable x. Intuitively, this map captures how the meaning of φ depends on the meaning of x.

The map φ_x Formally, given a μ-formula φ, a model $\mathcal{M} = (W, (R_a)_{a \in Act})$ and an assignment $\tau : Var \to \mathcal{P}(W)$, we define the map $\varphi_x^{\mathcal{M},\tau}$ by:

$$\begin{array}{rl}
\varphi_x^{\mathcal{M},\tau} & : \mathcal{P}(W) \to \mathcal{P}(W) \\
 & \quad U \mapsto [\![\varphi]\!]_{\mathcal{M},\tau[x \mapsto U]},
\end{array}$$

where $\tau[x \mapsto U]$ is the assignment τ' such that $\tau'(x) = U$ and $\tau'(y) = \tau(y)$, for all variables $y \neq x$. If $\mathcal{M}$ and τ are clear from the context, we write φ_x instead of $\varphi_x^{\mathcal{M},\tau}$.

We would like to define the meaning of $\mu x.\varphi$ ($\nu x.\varphi$) as the least (greatest) fixpoint of the map φ_x. In general, not all maps admit a fixpoint and if they do, there is no guarantee that there is a least or a greatest one.

In order to show that there are always a least and greatest fixpoints for the map φ_x, we use the Knaster-Tarski theorem. We start by recalling the basic definitions needed to state the Knaster-Tarski theorem.

Fixpoints A *complete lattice* $\mathbb{P}$ is a partially ordered set $(P, \leq)$ in which each subset has a greatest lower bound (called the *meet*) and a least upper bound (called the *join*). If U is a subset of the lattice, we denote by $\bigwedge U$ the meet of U and by $\bigvee U$ the join of U. If $U = \{c, d\}$, we write $c \wedge d$ instead $\bigwedge U$. Similarly, we use the notation $c \vee d$ instead of $\bigvee U$.

Let $f : P \longrightarrow P$ be a map. A point $c \in P$ is a *fixpoint* of f if $f(c) = c$. Next, c is the *least fixpoint* of f if for all fixpoints d of f, we have $c \leq d$. The point c is the *greatest fixpoint* of f if for all fixpoints d of f, we have $d \leq c$. Finally, c is a *pre-fixpoint* of f if $f(c) \leq c$ and c is a *post-fixpoint* of f if $c \leq f(c)$.

2.2.1. Theorem ([Tar55]). *Let $(P, \leq)$ be a complete lattice and $f : P \longrightarrow P$ a monotone map (that is, for all $c, d \in P$ satisfying $c \leq d$, we have $f(c) \leq f(d)$). Then f admits a least fixpoint, which is given by*

$$\bigwedge\{c \in P \mid f(c) \leq c\}$$

and f admits a greatest fixpoint, which is given by

$$\bigvee\{c \in P \mid c \leq f(c)\}.$$

Now it is immediate that for all models $\mathcal{M}$ with domain W, the pair $(\mathcal{P}(W), \subseteq)$ is a complete lattice (with meet operator $\bigwedge$ given by $\bigcap$ and join operator $\bigvee$ given by $\bigcup$). Moreover, the syntax of the μ-formulas is defined such that no variable occurring in a μ-formula is in the scope of a negation symbol. We can easily derive that for all assignments $\tau : \mathit{Var} \longrightarrow \mathcal{P}(W)$, the map $\varphi_x^{\mathcal{M},\tau} : \mathcal{P}(W) \longrightarrow \mathcal{P}(W)$ is monotone. Combining this with the Knaster-Tarski theorem, we obtain that the map φ_x always admits a least and a greatest fixpoint. We are now ready to define the semantics for the μ-formulas.

Semantics for μ-formulas Let $\mathcal{M} = (W, (R_a)_{a \in \mathit{Act}}, V)$ be a Kripke model. Given an assignment $\tau : \mathit{Var} \longrightarrow \mathcal{P}(W)$ and a μ-formula φ, we define the *meaning* of φ, notation: $[\![\varphi]\!]_{\mathcal{M},\tau}$, by induction on the complexity of φ. The definition is as in the case of the semantics for modal formulas, with the extra clauses:

$$\begin{aligned}
[\![x]\!]_{\mathcal{M},\tau} &= \tau(x), \\
[\![\mu x.\varphi]\!]_{\mathcal{M},\tau} &= \bigcap\{U \subseteq W \mid [\![\varphi]\!]_{\mathcal{M},\tau[x \mapsto U]} \subseteq U\}, \\
[\![\nu x.\varphi]\!]_{\mathcal{M},\tau} &= \bigcup\{W \subseteq S \mid U \subseteq [\![\varphi]\!]_{\mathcal{M},\tau[x \mapsto U]}\},
\end{aligned}$$

where $\tau[x \mapsto U]$ is the assignment τ' such that $\tau'(x) = U$ and $\tau'(y) = \tau(y)$, for all variables $y \neq x$.

Note that the set $[\![\mu x.\varphi]\!]_{\mathcal{M},\tau}$ is defined as the intersection of all the pre-fixpoints of the map φ_x. Hence, by the Knaster-Tarski theorem, $[\![\mu x.\varphi]\!]_{\mathcal{M},\tau}$ is indeed the least fixpoint of the map φ_x. Similarly, $[\![\nu x.\varphi]\!]_{\mathcal{M},\tau}$ defined as the union of all the post-fixpoints of φ_x, is the greatest fixpoint of the map φ_x.

It is also interesting to observe that for all formulas $\varphi \in \mu$ML, for all models $\mathcal{M} = (W, (R_a)_{a\in Act}, V)$ and for all assignments $\tau : Var \longrightarrow \mathcal{P}(W)$, we have

$$[\![\neg\varphi]\!]_{\mathcal{M},\tau} = W \backslash [\![\varphi]\!]_{\mathcal{M},\tau}.$$

This can be shown by a standard induction on φ. We introduce now some general terminology related to the semantics.

Truth and validity If $w \in [\![\varphi]\!]_{\mathcal{M},\tau}$, we will write $\mathcal{M}, w \Vdash_\tau \varphi$ and say that φ is *true* at w under the assignment τ. If φ is a sentence, the meaning of φ does not depend on the assignment and so we write $\mathcal{M}, w \Vdash \varphi$, and say that φ is *true* at w. A μ-formula is *true* in a model $\mathcal{M}$ with domain W, if for all assignments $\tau : Var \longrightarrow \mathcal{P}(W)$ and for all $w \in W$, we have $\mathcal{M}, w \Vdash_\tau \varphi$. In this case, we write $\mathcal{M} \Vdash \varphi$. A μ-formula φ is *valid* at a point w of a frame $\mathbb{F} = (W, (R_a)_{a\in Act})$, notation: $\mathbb{F}, w \Vdash \varphi$, if for all valuations $V : Prop \longrightarrow \mathcal{P}(W)$ and for all assignments $\tau : Var \longrightarrow \mathcal{P}(W)$, we have $(W, (R_a)_{a\in Act}, V), w \Vdash_\tau \varphi$. If for all $w \in W$, we have $\mathbb{F}, w \Vdash \varphi$, then φ is *valid* in $\mathbb{F}$ and we write $\mathbb{F} \Vdash \varphi$.

Two formulas φ and ψ are *equivalent* on a class $\mathcal{C}$ of Kripke models, notation: $\varphi \equiv_{\mathcal{C}} \psi$, if for all models $\mathcal{M}$ in $\mathcal{C}$, and for all assignments τ, $[\![\varphi]\!]_{\mathcal{M},\tau} = [\![\psi]\!]_{\mathcal{M},\tau}$. If $\mathcal{C}$ is the class of all Kripke models, we simply write $\varphi \equiv \psi$. A μ-formula is *satisfiable in a model* $\mathcal{M}$ with domain W, if for some assignment $\tau : Var \longrightarrow \mathcal{P}(W)$ and some $w \in W$, φ is true at w under the assignment τ. A μ-formula φ is *satisfiable* if there is a model in which φ is satisfiable.

In the definition of the meaning of $\mu x.\varphi$ and $\nu x.\varphi$, we obtained the least and greatest fixpoints by taking intersections of pre-fixpoints and unions of post-fixpoints. Another way to obtain the least and greatest fixpoints is to approximate them.

Obtaining the fixpoints using approximations Fix a μ-formula φ, a model $\mathcal{M} = (W, (R_a)_{a\in Act}, V)$ and an assignment $\tau : Var \longrightarrow \mathcal{P}(W)$. For each ordinal β, we define the sets $(\varphi_x)^\beta_\mu(\emptyset)$ by induction on β in the following way:

$$\begin{aligned}
(\varphi_x)^0_\mu(\emptyset) &= \emptyset, \\
(\varphi_x)^{\beta+1}_\mu(\emptyset) &= \varphi_x((\varphi_x)^\beta_\mu(\emptyset)), \\
(\varphi_x)^\lambda_\mu(\emptyset) &= \bigcup\{(\varphi_x)^\beta_\mu(\emptyset) \mid \beta < \lambda\},
\end{aligned}$$

where λ is a limit ordinal. Dually, we define the sets $(\varphi_x)^\beta_\nu(W)$ in the following way:

$$\begin{array}{rcl} (\varphi_x)^0_\nu(W) & = & W, \\ (\varphi_x)^{\beta+1}_\nu(W) & = & \varphi_x((\varphi_x)^\beta_\nu(W)), \\ (\varphi_x)^\lambda_\nu(W) & = & \bigcap\{(\varphi_x)^\beta_\mu(W) \mid \beta < \lambda\}, \end{array}$$

where λ is a limit ordinal. Using the fact that φ_x is monotone, we can show that for some ordinal β_0, we have $(\varphi_x)^{\beta_0}_\mu(\emptyset) = (\varphi_x)^{\beta_0+1}_\mu(\emptyset)$. Moreover, the set $(\varphi_x)^{\beta_0}_\mu(\emptyset)$ is the least fixpoint of φ_x. The smallest ordinal β_0 satisfying $(\varphi_x)^{\beta_0}_\mu(\emptyset) = (\varphi_x)^{\beta_0+1}_\mu(\emptyset)$, is called the *closure ordinal* of φ.

Similarly, for some ordinal β_1, we have $(\varphi_x)^{\beta_1}_\nu(W) = (\varphi_x)^{\beta_1+1}_\nu(W)$ and $(\varphi_x)^{\beta_1}_\nu(W)$ is the greatest fixpoint of φ_x. For details about the proof, we refer the reader to [AN01].

Now that we defined the semantics, it is not hard to show that it is equivalent to use the standard modalities $\Box_a$ and $\Diamond_a$ in a fixpoint formula or to use the ∇ modality. Unravelling the semantics of ∇_a, we find that

$$\nabla_a\Phi \equiv \bigwedge \Diamond_a\Phi \wedge \Box_a \bigvee \Phi, \tag{2.2}$$

where $\Diamond_a\Phi = \{\Diamond_a\varphi \mid \varphi \in \Phi\}$. Conversely, it is easy to see that $\Diamond_a\varphi \equiv \nabla_a\{\varphi, \top\}$ and $\Box_a\varphi \equiv \nabla_a\emptyset \vee \nabla_a\{\varphi\}$. Based on this we can prove that the languages μML and μML^∇ are effectively equi-expressive.

2.2.2. FACT. For every formula in μML, we can compute an equivalent formula in μML^∇, and vice versa.

It also easily follows from the semantics that each μ-formula in μML (in μML^∇) is equivalent to a well-named μ-formula in μML (in μML^∇), which is simply obtained by renaming some variables in the original formula. We adopt the following convention.

Convention We always assume μ-formulas to be well-named.

Finally, there is another restriction one might want to make on the shape of the μ-formulas.

Guarded formula A μ-formula is *guarded* if each occurrence of a variable, that is in the scope of a fixpoint operator η (where η is either μ or ν), is also in in the scope of a modal operator (which is either $\Box_a$, $\Diamond_a$ or ∇_a), which is itself in the scope of η.

For example, $\mu x.(p \vee \Diamond x)$ is guarded but $\Diamond(\mu x.p \vee x)$ is not guarded. It is not hard to see that $\mu x.x \equiv \bot$ and $\nu x.x \equiv \top$. We can generalize this and show that if an occurrence of x in a μ-formula of the form $\eta x.\varphi$ (with $\eta \in \{\mu, \nu\}$), is not under the scope of a modal operator, we can basically get rid of this occurrence.

2.2.3. PROPOSITION ([Koz83]). *Each formula in* μML *(in* $\mu\mathrm{ML}^\nabla$*) can be transformed in linear time into an equivalent guarded formula in* μML *(in* $\mu\mathrm{ML}^\nabla$*).*

2.3 Game terminology and the evaluation game

An alternative way to define truth of μ-formulas, is to use games. The advantage of the game semantics is that it is more intuitive than the semantics introduced above. However, the latter semantics is usually more appropriate when proving results by induction on the complexity of the formulas. Another advantage of the game semantics is that it allows us to transfer results from game theory to μ-calculus. Let us also mention that game theory plays an important role when investigating the link between μ-calculus and automata (see the next section, but also Chapter 7).

We start by introducing some general terminology for graph games. These are board games that are played by two players (called $\exists$ and $\forall$).

General terminology Given a set G, we write G^* for the set of *finite* sequences of elements in G; we write G^ω for the set *infinite* sequences of elements in G.

Graph game A *graph game* $\mathbb{G}$ is a tuple $(G_\exists, G_\forall, E, \mathit{Win})$, where $G_\exists$ and $G_\forall$ are disjoint sets, E is a subset of $(G_\exists \cup G_\forall)^2$, and Win is a subset of $(G_\exists \cup G_\forall)^\omega$. An *initialized graph game* $\mathbb{G}_0$ is a tuple $(G_\exists, G_\forall, E, \mathit{Win}, z_I)$, where $(G_\exists, G_\forall, E, \mathit{Win})$ is a graph game and the *initial position* z_I belongs to $G_\exists \cup G_\forall$.

We write G for the set $G_\exists \cup G_\forall$ and call it *the board* of the game. An element z in G is a *position*. Moreover, if $z \in G_\exists$, z is a *position for* $\exists$, which means that $\exists$ is supposed to move at position z. Otherwise, z is a *position for* $\forall$. If no confusion is possible, we simply refer to graph games and initialized graph games as *games*.

Very often we present a graph game in a table. The first column contains the positions of the board. The second column specifies which player each position belongs to. In the third row, we can find the sets $E[z]$ of possible moves.

Match Let $\mathbb{G}$ be the graph game $(G_\exists, G_\forall, E, \mathit{Win})$ and let $\mathbb{G}_0$ be the initialized graph game $(G_\exists, G_\forall, E, \mathit{Win}, z_I)$. A $\mathbb{G}$-*match* is a sequence $(z_i)_{i<\kappa} \in G^* \cup G^\omega$ such that for all i with $i + 1 < \kappa$, $(z_i, z_{i+1}) \in E$. A $\mathbb{G}_0$-*match* is a sequence

$(z_i)_{i<\kappa} \in G^* \cup G^\omega$ that is a $\mathbb{G}$-match and such that $z_0 = z_I$. We call κ the *length* of the match.

A match $\pi = (z_i)_{i<\kappa}$ is *full* if either $\kappa = \omega$, or κ is finite and there is no $z \in G$ such that $(z_{\kappa-1}, z) \in E$. In the latter case, if z is a position that belongs to a player σ, we say that player σ *gets stuck*. A match that is not full is also called *partial*.

Every full match π has a *winner*; in case π is finite, the winner is the opponent of the player who got stuck. In case π is infinite, $\exists$ wins π if π belongs to *Win*. Otherwise, π is won by $\forall$.

The notion of strategy is central in game theory. Many important results concerns the existence of a strategy, or the existence of normal forms for strategies. A strategy for a player is a map that tells the player how to play. In case the strategy is winning, the player is ensured to win the match.

Strategy Let $\mathbb{G}$ be the graph game $(G_\exists, G_\forall, E, Win)$ and let $\mathbb{G}_0$ be the initialized graph game $(G_\exists, G_\forall, E, Win, z_I)$. If σ belongs to $\{\forall, \exists\}$, we denote by G^*G_σ the set of sequences $(z_i)_{i<\kappa}$ in G^*, with $1 < \kappa < \omega$ and $z_{\kappa-1} \in G_\sigma$.

A *strategy* for a player σ in $\mathbb{G}$ (resp. in $\mathbb{G}_0$) is a partial map $f : G^*G_\sigma \longrightarrow G$ such that for all $\pi = (z_i)_{i<\kappa}$ in the domain of f, $(z_{\kappa-1}, f(\pi))$ belongs to E. We denote by $Dom(f)$ the domain of f. A $\mathbb{G}$-match (resp. $\mathbb{G}_0$-match) $\pi = (z_i)_{i<\kappa}$ is *f-conform* if for all $i + 1 < \kappa$ such that $z_i \in G_\sigma$, we have $z_{i+1} = f(z_0 \dots z_i)$.

A position $z \in G$ is *winning with respect to f* in $\mathbb{G}$ if the two following conditions hold. For all f-conform partial $\mathbb{G}$-matches $(z_i)_{i<\kappa}$ with $z_0 = z$ and $z_{\kappa-1} \in G_\sigma$, we have $(z_i)_{i<\kappa} \in Dom(f)$. Moreover, for all full f-conform $\mathbb{G}$-matches $\pi = (z_i)_{i<\kappa}$ with $z_0 = z$, π is won by σ.

We say that a strategy f for σ in $\mathbb{G}_0$ is *a winning strategy* if z_I is a winning position with respect to f in $\mathbb{G}$. Next, a position $z \in G$ is *winning for a player σ* in $\mathbb{G}$ if there is a strategy f for σ in $\mathbb{G}$ such that z is winning with respect to f in $\mathbb{G}$. We denote by $Win_\sigma(\mathbb{G})$ the set of all positions $z \in G$ that are winning for σ.

Finally, a strategy f for player σ is a *maximal winning strategy* in $\mathbb{G}$ if all winning positions z of σ are winning with respect to f in $\mathbb{G}$.

All the games that we consider are either parity games or can be linked to parity games. The notion of a parity game is fundamental in game theory. It captures a large class of graph games and still enjoys a powerful property: positional determinacy. This means that a given position is either winning for $\exists$ or $\forall$ (the game is determined) and that each strategy may be assumed to be positional; that is, the decision dictated by the strategy at a position of a match does not depend on what has been played before reaching the position.

Parity games and positional strategy A *parity game* is a tuple $(G_\exists, G_\forall, E, \Omega)$, where $G_\exists$ and $G_\forall$ are disjoint sets, E is a subset of $(G_\exists \cup G_\forall)^2$, and Ω is a map from $G_\exists \cup G_\forall$ to $\mathbb{N}$. We write G for the set $G_\exists \cup G_\forall$ and call it *the board* of the game. If $z \in G$, we say that $\Omega(z)$ is the priority of z.

With each parity game we can associate a graph game $\mathbb{G} = (G_\exists, G_\forall, E, \mathit{Win})$ defining Win such that for all sequences π in G^ω,

$$\pi \in \mathit{Win} \quad \text{iff} \quad \mathit{max}\{\Omega(z) \mid z \in \mathit{Inf}(\pi)\} \text{ is even},$$

where $\mathit{Inf}(\pi)$ is the set of elements in G that appear infinitely often in π. Hence, we can easily adapt all the definitions introduced earlier, to the case of parity games.

A strategy f for a player σ is *positional* if there is a partial map $f_p : G_\sigma \longrightarrow G$ such that for all $z_0 \dots z_n \in \mathit{Dom}(f)$, $f(z_0 \dots z_n) = f_p(z_n)$. We usually identify f and f_p.

We give now the result stating that parity games enjoy positional determinacy.

2.3.1. Theorem ([EJ91],[Mos91]). *Let $\mathbb{G}$ be a parity game. There exist positional strategies $f_\exists$ and $f_\forall$ for $\exists$ and $\forall$ respectively such that for all positions z on the board of $\mathbb{G}$, z is winning either with respect to $f_\exists$ or with respect to $f_\forall$.*

Moreover, given a position z, there is an effective procedure to determine whether z is winning for $\exists$ or $\forall$.

2.3.2. Theorem. *[Jur00] Let $\mathbb{G} = (G_\exists, G_\forall, E, \Omega)$ be a parity game and let n, m and d be the size of G, E and the range of Ω, respectively. Then for each player σ, the problem, whether a given position $z \in G$ is winning for σ, is decidable in time $\mathcal{O}\left(d \cdot m \cdot \left(\frac{n}{\lfloor d/2 \rfloor}\right)^{\lfloor d/2 \rfloor}\right)$.*

Next we introduce a parity game (called the evaluation game) which provides an alternative semantics for the μ-calculus. The connection between μ-calculus and games was first observed by E. Allen Emerson and Charanjit Jutla [EJ88] and Colin Stirling [Sti95].

A position in the evaluation game is a pair (w, φ), where w is a point in a model and φ a formula. The goal of $\exists$ is to show that φ is true at w, whereas $\forall$ wants to prove the opposite.

Evaluation game Let $\mathcal{M} = (W, R, V)$ be a Kripke model and let φ be a sentence in μML or in μML^∇. We also fix a map $pr : \mathit{Var} \longrightarrow \mathbb{N}$, which assigns a priority to each variable such the two following conditions hold. If a variable x in φ is a μ-variable, then $pr(x)$ is even and if x is a ν-variable, then $pr(x)$ is odd. Moreover, if for some variables x and y, we have $x <_\varphi y$, then $pr(x) < pr(y)$. It is easy to see that there is always such a map.

Next, given a set of formulas Φ and a point $w \in W$, we say that a map $m : \Phi \longrightarrow \mathcal{P}(R[w])$ is a ∇_a-*marking* if

- for all $\varphi \in \Phi$, there exists $v \in R_a[w]$ such that $v \in m(\varphi)$,
- for all $v \in R_a[w]$, there exists $\varphi \in \Phi$ such that $v \in m(\varphi)$.

We define the *evaluation game* $\mathcal{E}(\mathcal{M}, \varphi)$ as a parity game. The board of the game consists of the pairs (w, ψ), where $w \in W$ and ψ is a subformula of φ.

The game is given in Table 2.1, where $w \in W$, $p \in Prop$, $x \in Var$, $\eta \in \{\mu, \nu\}$, $\psi, \psi_1, \psi_2 \in Sfor(\varphi)$, Φ is a subset of $Sfor(\varphi)$ and α is a conjunction of literals.

Position z	Player	Possible moves $E[z]$	$\Omega(z)$
$(w, \top)$	$\forall$	$\emptyset$	0
$(w, \bot)$	$\exists$	$\emptyset$	0
(w, x)	-	$\{(w, \delta_x)\}$	$pr(x)$
(w, p) and $w \in V(p)$	$\forall$	$\emptyset$	0
(w, p) and $w \notin V(p)$	$\exists$	$\emptyset$	0
$(w, \neg p)$ and $w \notin V(p)$	$\exists$	$\emptyset$	0
$(w, \neg p)$ and $w \in V(p)$	$\forall$	$\emptyset$	0
$(w, \psi_1 \wedge \psi_2)$	$\forall$	$\{(w, \psi_1), (w, \psi_2)\}$	0
$(w, \psi_1 \vee \psi_2)$	$\exists$	$\{(w, \psi_1), (w, \psi_2)\}$	0
$(w, \eta x.\psi)$	-	$\{(w, \psi)\}$	0
$(w, \Diamond_a \psi)$	$\exists$	$\{(v, \psi) \mid v \in R[w]\}$	0
$(w, \Box_a \psi)$	$\forall$	$\{(v, \psi) \mid v \in R[w]\}$	0
$(w, \alpha \bullet \nabla_a \Phi)$	$\forall$	$\{(w, \alpha), (w, \nabla_a \Phi)\}$	0
$(w, \nabla_a \Phi)$	$\exists$	$\{m : \Phi \longrightarrow \mathcal{P}(R[w]) \mid$ m is a ∇_a-marking $\}$	0
$m : \Phi \longrightarrow \mathcal{P}(R[w])$	$\forall$	$\{(u, \psi) \mid u \in m(\psi)\}$	0

Figure 2.1: The evaluation game

Given the fact that at position (w, ψ), $\exists$'s goal to prove that ψ is true at w, the definitions of $G_\exists$, $G_\forall$ and E given above, seem fairly natural. The winning condition might seem a bit more intricate. The intuition is that a μ-variable can be unfolded only finitely many times, whereas a ν-variable corresponds to possibly infinite unfolding (what we call "unfolding" of a variable x in the evaluation game, is the move from a position (w, x) to position (w, δ_x)).

Obviously it could be the case that there is more than one variable that is unfolded infinitely many times during a match. The point is that among all the variables unfolded infinitely many times, there is a unique one which is highest in the dependency order. $\exists$ wins iff this variable is a ν-variable. This is exactly what is expressed by the parity condition.

2.3.3. Theorem. *Let φ be a sentence and let $(\mathcal{M}, w)$ be some pointed Kripke model. For the game $\mathcal{E}(\mathcal{M}, \varphi)$, there are positional strategies $f_\exists$ and $f_\forall$ such that*

$f_\exists$ is winning for $\exists$, $f_\forall$ is winning for $\forall$ and for every position z, z is a winning either with respect to $f_\exists$ or with respect to $f_\forall$.

In addition, for all $w \in \mathcal{M}$,

$$\mathcal{M}, w \Vdash \varphi \quad \textit{iff} \quad (w, \varphi) \in \mathit{Win}_\exists(\mathcal{E}(\mathcal{M}, \varphi)).$$

More generally, for all subformulas ψ of φ and for all $w \in \mathcal{M}$, we have

$$\mathcal{M}, w \Vdash e_\varphi(\psi) \quad \textit{iff} \quad (w, \psi) \in \mathit{Win}_\exists(\mathcal{E}(\mathcal{M}, \varphi)).$$

It follows from this proposition that a procedure for deciding which player has a winning strategy in an initialized parity game, would give us a procedure for the model checking problem for the μ-calculus. The model checking problem consists in deciding whether a given μ-sentence is true at a given point in a finite model. Using the result proved by Marcin Jurdziński in [Jur98] about the complexity of parity games, we obtain the following upper bound for the complexity of the model checking problem.

2.3.4. Theorem ([Jur98]). *The model checking problem for the μ-calculus is* UP ∩ co-UP.

A non-deterministic Turing machine is unambiguous if for every input, there is at most one accepting computation. The complexity class UP (Unambiguous Non-deterministic Polynomial-time) is the class of languages problems solvable in polynomial time by an unambiguous non-deterministic Turing machine (for more details on this model of computation, see for instance [Pap94]).

The result above only gives us an upper bound for the model checking. It is an important open problem to obtain the exact complexity (and in particular, whether model checking can be done in polynomial time).

In fact, the problem of finding the exact complexity of the model checking problem for the μ-calculus is equivalent to the problem of finding the exact complexity for solving parity games. The problem of solving a parity game consists in deciding which player has a winning strategy from a given position in a parity game. The fact that there is a reduction from the model checking problem to the problem of solving parity games immediately follows from Theorem 2.3.3. The converse reduction follows from the fact that given a parity game and a player, there is a μ-formula describing the set of winning positions for the player. This result was proved by E. Allen Emerson and Charanjit Jutla [EJ91]; another proof can be found in [Wal02].

2.4 μ-Automata

Another way to approach μ-formulas, is to use μ-automata. Automata theory is a vast area, but we only present the material that we need in this thesis. For more details about automata, see for instance [Tho97].

2.4.1 ω-Automata

We start by recalling the notion of ω-automaton, which will be mostly used to specify winning conditions for graph games.

ω-Automaton Fix a finite alphabet Σ. The elements of Σ^ω are the infinite *words* over Σ. A *non-deterministic parity ω-automaton* over a finite alphabet Σ is a tuple (Q, q_I, δ, Ω), where Q is a finite set of states, $q_I \in Q$ is the *initial state*, δ is a function $\delta : Q \times \Sigma \longrightarrow \mathcal{P}(Q)$ called the *transition map*, and $\Omega : Q \longrightarrow \mathbb{N}$ is the *parity map*.

The automaton $\mathbb{A}$ is a *deterministic parity ω-automaton* if for all $(q, c) \in Q \times \Sigma$, the set $\delta(q, c)$ is a singleton. In this case, we can think of δ as a map from $Q \times \Sigma$ to Q.

We define the *size* of the automaton as the cardinality of the set Q and the *index* of $\mathbb{A}$ is the size of the range of Ω.

ω-Automata operate on infinite words. We recall the notions of run and accepting run.

Run and acceptance Let $c_0c_1\dots$ be a word over Σ. A *run* of the automaton (Q, q_I, δ, Ω) on $c_0c_1\dots$ is a sequence $q_0q_1\dots$ in Q^ω such that $q_0 = q_I$ and for all $i \in \mathbb{N}$, $q_{i+1} \in \delta(q_i, c_i)$.

A word $c_0c_1\dots$ is *accepted* by the automaton $\mathbb{A}$ if there is a run $q_0q_1\dots$ of the automaton on $c_0c_1\dots$ such that the maximum of the set $\{\Omega(q) \mid q \in \mathit{Inf}(q_0q_1\dots)\}$ is even. Recall that $\mathit{Inf}(q_0q_1\dots)$ is the set of elements in Q, that occur infinitely often in the sequence $q_0q_1\dots$.

A subset L of Σ^ω is an *ω-regular language* over Σ if there is a non-deterministic parity automaton $\mathbb{A}$ such that L is exactly the set of words accepted by $\mathbb{A}$.

Assuming ω-automata to be non-deterministic turns to be convenient for proving certain results. Examples of such results include the fact that there is a disjunctive normal form for the μ-calculus (see the end of Section 2.4) and results from Chapter 7. This assumption can be made without loss of generality, as shown by the following result.

2.4.1. THEOREM ([McN66, Saf92, Pit06]). *Given a non-deterministic parity ω-automaton $\mathbb{A}$ with size n and index k, we can construct in time exponential in the size of $\mathbb{A}$, a deterministic parity ω-automaton $\mathbb{A}'$ such that $\mathbb{A}$ and $\mathbb{A}'$ recognize the same language and the index of $\mathbb{A}'$ is linear in n and k.*

2.4.2 μ-automata

We present the notion of μ-automaton, which is basically an alternative way to think of a μ-formula. Michael Rabin introduced non-deterministic automata operating on infinite binary trees [Rab69] in order to show that the monadic second-order theory of infinite binary trees is decidable (see Section 2.6). Later David Muller and Paul Schupp considered alternating automata on infinite binary trees [MS87]. In [SE89], Robert Streett and E. Allen Emerson gave a transformation from μ-formulas to Rabin automata, while a converse translation was established by Damian Niwiński in [Niw88]. These notions of automata have been extended to the setting of Kripke models (instead of binary trees) by David Janin and Igor Walukiewicz in [Jan97].

Given the nature of fixpoint logics, a linear representation, as offered by μ-formulas, is not always well-suited. Automata provide a graph theoretical representation which has been useful for establishing fundamental results concerning the μ-calculus. Let us for example mention the satisfiability problem (see Chapter 7 where the satisfiability problem is presented in the framework of coalgebras). A proof of the μ-calculus hierarchy theorem (see [Arn99]) is also based on the connection between μ-automata and μ-formulas. Another example, the fact that there is a disjunctive normal form, can be found in the next subsection.

μ-automata Given a finite set Q, we define the set $TC(Q)$ of *transition conditions* as the set of formulas φ given by:

$$\varphi \ ::= \ \top \mid \bot \mid p \mid \neg p \mid \mid \Diamond_a q \mid \Box_a q \mid q \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

where $a \in Act$, $p \in Prop$ and $q \in Q$. The set $TC^n(Q)$ is the set of formulas of the form $\alpha \wedge \varphi$, where α is a conjunction of literals and φ belongs to the set of formulas given by:

$$\begin{aligned} \psi \ &::= \ \top \mid \bot \mid \mid \Diamond_a q \mid \Box_a q \mid \psi \wedge \psi, \\ \varphi \ &::= \ \psi \mid \varphi \vee \varphi, \end{aligned}$$

where $a \in Act$ and $q \in Q$. Basically a formula in $TC^n(Q)$ is a formula of the form $\alpha \wedge \varphi$, where α is a conjunction of literals and φ is a disjunction of conjunctions of formulas of the form $\Diamond_a q$, $\Box_a q$, $\top$ or $\bot$. It is easy to see that $TC^n(Q)$ is a subset of $TC(Q)$.

Next, we define the set $TC^d(Q)$ of *disjunctive transition conditions* as the set of formulas that are disjunctions of formulas of the form $\alpha \bullet \nabla Q'_a$, where $a \in Act$, $Q' \subseteq Q$ and α is a conjunction of literals over *Prop*.

An *alternating μ-automaton* is a tuple $\mathbb{A} = (Q, q_I, \delta, \Omega)$, where Q is a finite set of states, $q_I \in Q$ is the *initial state*, $\delta : Q \to TC(Q)$ is the *transition map* and $\Omega : Q \to \mathbb{N}$ is the *parity map*. In case δ is a map from Q to $TC^n(Q)$, $\mathbb{A}$ is a *normalized alternating μ-automaton.* Finally, if δ is a map from Q to $TC^d(Q)$, $\mathbb{A}$ is a *non-deterministic μ-automaton.* The *size* of $\mathbb{A}$ is the size of A and the *index* of $\mathbb{A}$ is the size of the range of Ω.

The input for a μ-automaton is a pointed model. Whether an automaton accepts a pointed model or not, depends on the existence of a winning strategy for player $\exists$ in a parity game, that we call the acceptance game.

Acceptance game Let $\mathcal{M} = (W, R, V)$ be a model and let $\mathbb{A} = (Q, q_I, \delta, \Omega)$ be a normalized alternating or non-deterministic μ-automaton. The associated *acceptance game* $\mathcal{A}(\mathcal{M}, \mathbb{A})$ is the parity game given by the table below.

Position	Player	Admissible moves	Priority
$(w, q) \in W \times Q$	$\exists$	$\{m : Q \to \mathcal{P}(W) \mid (W, R, m), w \Vdash \delta(q)\}$	$\Omega(q)$
$m : Q \to \mathcal{P}(W)$	$\forall$	$\{(v, q') \mid v \in m(q')\}$	0

A pointed model $(\mathcal{M}, w_0)$ is *accepted* by the automaton $\mathbb{A}$ if the pair (w_0, q_I) is a winning position for player $\exists$ in $\mathcal{A}(\mathcal{M}, \mathbb{A})$.

The acceptance game of μ-automata proceeds in rounds, moving from one basic position in $W \times Q$ to another. Each round consists of two moves. At position (w, q), $\exists$ has to come up with a marking m that assigns states of the automaton to each point in W. The marking should be such that the formula $\delta(q)$ is true at w. Given the shape of the formulas in $TC(Q)$, we may assume that if $\mathbb{A}$ is alternating, then for all $v \notin R[w] \cup \{w\}$, $m(v) = \emptyset$. Similarly, if $\mathbb{A}$ is either normalized or non-deterministic, then we may suppose that for all $v \notin R[w]$, $m(v) = \emptyset$.

Intuitively, after the marking is chosen, the automaton is split in several copies, each of them corresponding to a pair (v, q') with $v \in m(q')$. Now $\forall$ can pick one of these copies and we move to a new basic position.

Since $TC^n(Q)$ is a subset of $TC(Q)$, a normalized alternating automaton is an alternating automaton, but the converse is not true. The main difference between these two notions of μ-automata is that alternating automata allow "empty" transitions in the model. This means that during a round, while the automaton is moving from one state to another, it can happen that there is no move in the model from the current point to one of its successors.

Equivalence between μ-automata Two μ-automata $\mathbb{A}$ and $\mathbb{A}'$ are *equivalent*if for all pointed models $(\mathcal{M}, w_0)$, $\mathbb{A}$ accepts $(\mathcal{M}, w_0)$ iff $\mathbb{A}'$ accepts $(\mathcal{M}, w_0)$.

For example, it is easy to see that each non-deterministic μ-automaton is equivalent to a normalized μ-automaton. This follows from the facts that ∇_a can be expressed using $\Box_a$ and $\Diamond_a$ (see Fact (2.2)) and that every positive formula of propositional logic is equivalent to a disjunction of conjunctions of proposition letters.

Now that we defined the way automata operate on pointed models, we can show that there are effective truth-preserving transformations from μ-formulas to μ-automata and vice-versa.

Equivalence between μ-automata and μ-formula A μ-sentence φ *is equivalent to a μ-automaton* $\mathbb{A}$if for all pointed Kripke model $(\mathcal{M}, w)$, we have

$$\mathcal{M}, w \Vdash \varphi \quad \text{iff} \quad \mathbb{A} \text{ accepts } (\mathcal{M}, w).$$

When this happens, we write $\varphi \equiv \mathbb{A}$.

Using Theorem 2.3.3, it is easy to transform a sentence into an alternating automaton. We do give the construction (which is short in any case), as it illustrates how close the notions of sentence and alternating automaton are.

2.4.2. Proposition. *] There is a procedure transforming a μ-sentence φ into an equivalent alternating μ-automaton $\mathbb{A}_\varphi$ of size n and index d, where n is the number of subformulas of φ and d is the alternation depth of φ.*

Proof Let φ be a sentence in μML. We define an alternating automaton $\mathbb{A} = (Q, q_I, \delta, \Omega)$ in the following way. We define Q as $\{\widehat{\psi} \mid \psi \text{ subformulas of } \varphi\}$. The initial state q_I is the state $\hat{\varphi}$. For all subformulas ψ of φ, the formula $\delta(\widehat{\psi})$ is defined by induction on ψ in the following way:

$$\begin{array}{ll|ll}
\delta(\widehat{x}) & = \widehat{\delta_x}, & \delta(\widehat{\psi_1 \wedge \psi_2}) & = \widehat{\psi_1} \wedge \widehat{\psi_2}, \\
\delta(\widehat{\top}) & = \top, & \delta(\widehat{\Box_a \psi}) & = \Box_a \widehat{\psi} \\
\delta(\widehat{\bot}) & = \bot, & \delta(\widehat{\Diamond_a \psi}) & = \Diamond_a \widehat{\psi} \\
\delta(\widehat{\psi_1 \vee \psi_2}) & = \widehat{\psi_1} \vee \widehat{\psi_2}, & \delta(\widehat{\eta x.\psi}) & = \widehat{\psi}, \\
\delta(\widehat{p}) & = p & \delta(\widehat{\neg p}) & = \neg p
\end{array}$$

where $p \in \mathit{Prop}$, $a \in \mathit{Act}$ and η belongs to $\{\mu, \nu\}$. We also fix a map Ω which assigns a priority to each state $\widehat{\psi}$ such the three following conditions hold. If a variable x in φ is a μ-variable, then $\Omega(\widehat{x})$ is even and if x is a ν-variable, then $\Omega(\widehat{x})$ is odd. Moreover, if for some variables x and y, we have $x <_\varphi y$, then $\Omega(\widehat{x}) < \Omega(\widehat{y})$. Finally, if ψ is not a variable, $\Omega(\widehat{\psi}) = 0$.

With such a definition of $\mathbb{A}$, the acceptance game associated with $\mathbb{A}$ is extremely similar to the evaluation game associated with φ. Using Theorem 2.3.3, we can show that φ and $\mathbb{A}$ are equivalent.

The alternating automaton constructed in the proof above is almost the DAG of the formula, except that we add back edges from the variables to their unfoldings. The point is that the alternating automata format does not really differ from the linear representation of the μ-calculus. Normalized alternating automata make a better use of the possibilities that a graph theoretical representation of a formula has to offer. It follows from the next result that each μ-formula is equivalent to a normalized alternating μ-automaton.

2.4.3. Proposition ([EJ91, VW07]). *There is a procedure transforming an alternating μ-automaton $\mathbb{A}$ into an equivalent normalized alternating μ-automaton of size dn and index d, where n is the size and $\mathbb{A}$ is the alternation depth of $\mathbb{A}$.*

In some occasions (as we shall see in Chapter 5), it is handy to associate with a formula an even stronger notion of automata, namely non-deterministic automata. The difference between these automata and the normalized alternating automata that in the transition map of a non-deterministic automaton, the use of the operator $\wedge$ is rather restricted. Recall that in the evaluation game, the positions of the form $(\psi_1 \wedge \psi_2, w)$ belong to $\forall$. Hence, if we want to show that a formula is true at point (that is, provide $\exists$ with a winning strategy), these are "bad" positions as we cannot control them with the strategy. This suggests that showing that a pointed model is accepted by an automaton is easier in case the automaton is non-deterministic.

The notions of normalized alternating automata and non-deterministic automata are equivalent, as witnessed by the following result. A key ingredient for proving the result is the fact that we can determinize ω-automata (Theorem 2.4.1).

2.4.4. PROPOSITION ([EJ91, VW07]). *A normalized alternating μ-automaton $\mathbb{A}$ of size n and index d can be transformed into a non-deterministic μ-automaton $\mathbb{A}'$, of size exponential in n and index polynomial in d.*

We observed earlier that normalized alternating automata are alternating automata and that each non-deterministic automaton is equivalent to a normalized alternating automaton. Hence, it follows from Propositions 2.4.3 and 2.4.4 that the three notions of automata introduced are equivalent.

To finish the loop, it remains to transform an automaton into an equivalent formula.

2.4.5. PROPOSITION ([NIW88]). *There is a procedure transforming a normalized alternating μ-automaton $\mathbb{A}$ into an equivalent μ-sentence.*

As we briefly mentioned earlier, some results concerning the μ-calculus were obtained by using the tight connection between the μ-calculus and automata. For example, the automata theoretic approach was used to give an exponential decision procedure for the satisfiability problem (recall that the satisfiability problem consists in determining whether for a given formula φ, there exists a point in a model at which φ is true).

The satisfiability problem can be reduced to the non emptiness problem for μ-automata (that is, given a μ-automaton, decide wether there exists a pointed model accepted by the automaton). We will use the same approach in the setting of coalgebras in Chapter 7.

2.4.6. THEOREM ([EJ88]). *The satisfiability problem for the μ-calculus is* EXPTIME-*complete.*

A related property of the μ-calculus is the small model property. This was established in an earlier paper by Robert Streett and E. Allen Emerson [SE89], also using methods based on automata theory.

2.4.7. THEOREM ([SE89]). *If a μ-sentence φ is satisfiable, then φ is satisfiable in a model, the size of which is at most exponential in the size of φ.*

2.4.3 Disjunctive formulas

We give an example of the use of the automata theoretic approach for the μ-calculus, by showing that each μ-formula is equivalent to a formula in disjunctive normal form (defined below). This result was proved by David Janin and Igor Walukiewicz in [JW95b].

Similarly to non-deterministic μ-automata, the use of the connector $\wedge$ is restricted in disjunctive formulas. Hence, it is not surprising that when defining a winning strategy for $\exists$ in the evaluation game, it might be easier to assume formulas to be disjunctive. Disjunctive formulas enjoy some other nice properties. For example, the satisfiability problem for disjunctive formulas is linear time [Jan97], whereas it is exponential time in the general case.

Disjunctive formula The set of *disjunctive μ-formulas* is given by:

$$\varphi \ ::= \ \top \ \mid \ x \ \mid \ \varphi \vee \varphi \ \mid \ \alpha \bullet \bigwedge\{\nabla_{a_i}\Phi_i \mid i \in I\} \ \mid \ \mu x.\varphi \ \mid \ \nu x.\varphi,$$

where x belongs to *Var*, α is a conjunction of literals, I is a finite set, for all $i \in I$, Φ_i is a finite subset of $\mu\mathrm{ML}^{\nabla}$, a_i belongs to *Act* and for all $i \neq j$, $a_i \neq a_j$.

2.4.8. THEOREM ([JW95A]). *A μ-sentence can be effectively transformed into an equivalent disjunctive sentence.*

In [JW95a], the proof that each μ-formula φ can be transformed into an equivalent disjunctive formula, uses tableaux and ω-automata. We can also obtain the result by first transforming φ into an equivalent normalized alternating automaton (Propositions 2.4.2 and 2.4.3), second transform this automaton into a non-deterministic automaton (Proposition 2.4.4) and finally show that if we use the procedure described in the proof of Proposition 2.4.5, a non-deterministic automaton is transformed into a disjunctive formula. This proof is based on the idea that the transformation of a formula into a disjunctive formula corresponds, at the level of automata, to the transformation of a normalized alternating automaton into a non-deterministic automaton.

2.5 Axiomatization of the μ-calculus

In the same paper where he introduced the μ-calculus, Dexter Kozen proposed an axiomatization for the formalism [Koz83]. This axiomatization is very natural and consists in enriching the standard axiomatization for modal logic, with a fixpoint axiom and a fixpoint rule. Dexter Kozen also proved completeness of that axiomatization with respect to a fragment of the μ-calculus. The problem of showing completeness of the axiomatization with respect to the full μ-calculus turned out to be very difficult, but eventually it was solved by Igor Walukiewicz [Wal95].

Kozen's axiomatization The axiomatization of the Kozen system K^μ consists of the following axioms and rules

propositional tautologies,	
$\neg\Box_a\neg p \leftrightarrow \Diamond_a p$,	(Dual-mod),
if $\vdash \varphi \to \psi$ and $\vdash \varphi$, then $\vdash \psi$	(Modus ponens),
if $\vdash \varphi$, then $\vdash \varphi[p/\psi]$	(Substitution),
$\vdash \Box_a(p \to p') \to (\Box_a p \to \Box_a p')$	(K-axiom),
if $\vdash \varphi$, then $\vdash \Box_a\varphi$	(Necessitation),
$\nu x.\varphi \leftrightarrow \neg\mu x.\neg\varphi[x/\neg x]$	(Dual-fix),
$\vdash \varphi[x/\mu x.\varphi] \to \mu x.\varphi$	(Fixpoint axiom),
if $\vdash \varphi[x/\psi] \to \psi$, then $\vdash \mu x.\varphi \to \psi$	(Fixpoint rule),

where p, p' belong to *Prop*, a belongs to *Act*, φ, ψ belong to μML, x is not a bound variable of φ and no free variable of ψ is bound in φ.

A μ-formula is *provable* in K^μ if φ belongs to the smallest set of formulas, which contains the propositional tautologies, the K-axiom, the Fixpoint axiom, the definitions of $\Diamond$ and ν and is closed under the Modus Ponens, the Substitution, the Necessitation and the Fixpoint rules.

2.5.1. Remark. The presence of the axioms (Dual-mod) and (Dual-fix) is due to the fact the connectives $\Diamond$ and $\Box$ and the operators μx and νx are primitive symbols of our language. For instance, if $\Diamond$ and $\neg$ would be primitive symbols but not $\Box$, then the symbol $\Box$ would be introduced as an abbreviation for $\neg\Diamond\neg$. As a consequence, we would not need the axiom (Dual-mod).

2.5.2. Theorem ([Wal95]). *Kozen's axiomatization is complete with respect to the μ-calculus on frames. That is, for all μ-formulas $\varphi \in \mu$ML, φ is provable in K^μ iff φ is valid on all frames.*

2.6 Expressivity of the μ-calculus

2.6.1 Bisimulation

A fundamental notion in the semantics of modal logic is the notion of bisimulation, which was defined by Johan van Benthem [Ben84]. It was also introduced independently by David Park [Par81] in the area of process algebra.

One of the motivations for modal logic is that Kripke models can be used to represent processes. From that perspective, modal logic and its extensions are languages for describing certain conditions, also called specifications, met by a process. A natural requirement would be that a formula should not be able to distinguish points that display the same behavior. The notion of "having the same behavior" is formalized by the notion of bisimulation.

Bisimulation Let $\mathcal{M} = (W, (R_a)_{a\in Act}, V)$ and $\mathcal{M}' = (S', (R'_a)_{a\in Act}, V')$ be two models. A relation $B \subseteq W \times W'$ is a *bisimulation* if for all $(w, w') \in B$, for all $p \in Prop$ and for all $a \in Act$, we have

- $w \in V(p)$ iff $w' \in V'(p)$,
- for all $v \in W$ such that wR_av, there exists $v' \in W'$ such that $(v, v') \in B$ and $w'R'v'$,
- for all $v' \in W'$ such that $w'R'v'$, there exists $v \in W$ such that $(v, v') \in B$ and wRv.

A relation B is a *bisimulation* between two pointed models $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$ if B is a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$ such that $(w, w') \in B$. When this happens, we say that $(\mathcal{M}, w)$ and $(\mathcal{M}', w')$ are *bisimilar* and we write $\mathcal{M}, w \leftrightarroweq \mathcal{M}', w'$.

The following result shows that indeed μ-calculus formulas cannot distinguish two points with the same behavior.

2.6.1. Proposition. *If $\mathcal{M}, w \leftrightarroweq \mathcal{M}', w'$, then for all μ-formulas φ, $\mathcal{M}, w \Vdash \varphi$ iff $\mathcal{M}', w' \Vdash \varphi$.*

Using the property above, we can show that the μ-calculus has the tree model property. That is, every formula that is satisfiable, is satisfiable in a tree.

The proof relies on Proposition 2.6.1 together with the fact that given a pointed model $(\mathcal{M}, w)$, it unravels into a tree, the root of which is bisimilar to w. In fact, we can even generalize this unravelling construction and prove something stronger: for all cardinals κ and all pointed models $(\mathcal{M}, w)$, we can construct a tree, the root of which is bisimilar to w and such that each node of the tree (except the root) has at least κ "copies". This tree is called the κ-expansion of $(\mathcal{M}, w)$.

We recall the definition of a tree and the related terminology. We also define the notion of κ-expansion and unravelling.

Tree A pair (T, R), where T is a set and $R \subseteq T \times T$, is a *tree* if for some point $r \in T$, $T = \{r\} \cup R^+[r]$, r does not have a predecessor and every state $t \neq r$ has a unique predecessor. A pair (T', R'), where T' is a set and $R' \subseteq T' \times T'$ is a *transitive tree* if for some tree (T, R), we have $(T', R') = (T, R^+)$.

If $(T, (R_a)_{a \in Act})$ is a Kripke frame, we denote by R the relation $\bigcup\{R_a \mid a \in Act\}$. A Kripke frame $(T, (R_a)_{a \in Act})$ is a *(transitive) tree* if the pair (T, R) is a (transitive) tree. A Kripke frame is a *finite (transitive) tree* if it is a (transitive) tree and its domain is finite.

A node u is a *child* of a node t in a tree if $(t, u) \in R$. A *sibling* of a node u in a tree is a node $u' \neq u$ such that for some node t, u and u' are children of t.

A model $(T, (R_a)_{a \in Act}, V)$ is a *tree model* if $(T, (R_a)_{a \in Act})$ is a tree. Similarly, we can define the notions of *finite tree model*, *transitive tree model* and *finite transitive tree model*.

κ-Expansion Let κ be an cardinal. A tree model is *κ-expanded* if every node (apart from the root) has at least κ many bisimilar siblings. Given a model $\mathcal{M} = (W, (R_a)_{a \in Act}, V)$ and a point $w \in W$, the *κ-expansion* of $(\mathcal{M}, w)$ is the structure $\mathcal{M}^\kappa_w := (W', (R'_a)_{a \in Act}, V')$, defined as follows. The set W' is the set of all finite sequences $w_0 a_1 k_1 w_1 \dots a_n k_n w_n$ $(n \geq 0)$ such that $w_0 = w$ and $k_i < \kappa$, $w_i \in W$, $a_i \in Act$ and $w_{i-1} R_{a_i} w_i$ for all $i > 0$. For all $a \in Act$, the relation R'_a is given by

$$\{(w_0 a_1 k_1 s_1 \dots a_n k_n w_n, w_0 a_1 k_1 w_1 \dots a_n k_n w_n a k v) \mid k < \kappa \text{ and } w_n R_a v\}.$$

Finally, for all $p \in Prop$, $V'(p)$ is the set $\{w_0 a_1 k_1 w_1 \dots a_n k_n w_n \mid w_n \in V(p)\}$.

The *canonical bisimulation* between $\mathcal{M}^\omega_w$ and $\mathcal{M}$ is the relation linking any point $w_0 a_1 k_1 w_1 \dots a_n k_n w_n$ to w_n. The *unravelling* of a pointed model $(\mathcal{M}, w)$ is the 1-expansion of $(\mathcal{M}, w)$.

In case Act is a singleton, we can simply "forget" about the a_is in the definitions of κ-expansion, canonical bisimulation and unravelling. For example, the domain of the κ-extension becomes the set of all finite sequences $w_0 k_1 w_1 \dots k_n w_n$ $(n \geq 0)$ such that $w_0 = w$ and $k_i < \kappa$, $w_i \in W$ and $w_{i-1} R w_i$ for all $i > 0$.

2.6.2. Fact. Given a pointed model $(\mathcal{M}, w)$ and a cardinal κ, the structure $(\mathcal{M}^\kappa_w, w)$ is an κ-expanded tree which is bisimilar to $(\mathcal{M}, w)$ via the canonical bisimulation.

Putting this result together with Proposition 2.6.1, we obtain the following result.

2.6.3. PROPOSITION. *Let φ be a μ-formula and κ be a cardinal. Then φ is true in all models iff φ is true in all κ-expanded tree models.*

In particular, this shows the tree model property for the μ-calculus. Often we will also make use of Proposition 2.6.3 in Chapter 5, in the case where $\kappa = \omega$.

In the same chapter, it will also be very handy to assume the formulas to be disjunctive. The reason is that when evaluating a disjunctive formula on an ω-expanded tree, we may assume that a strategy for $\exists$ satisfies some nice property.

Scattered strategy for the evaluation game Let $\mathcal{M}$ be a model and φ a μ-sentence. Given a state $w \in \mathcal{M}$, a strategy f for a player σ in the game $\mathcal{E}(\mathcal{M}, \varphi)$ with initial position (w, φ), is *scattered* [KV05] if for all states v in $\mathcal{M}$, for all f conform matches $\pi = (z_i)_{i<\kappa}$ and $\pi' = (z'_i)_{i<\kappa'}$ and for all μ-formulas φ and φ',

$$z_{\kappa-1} = (v, \psi) \text{ and } z_{\kappa'-1} = (v, \psi') \quad \text{implies} \quad \pi \sqsubseteq \pi' \text{ or } \pi' \sqsubseteq \pi,$$

where $\sqsubseteq$ is the prefix (initial segment) relation.

2.6.4. PROPOSITION. *If $\varphi \in \mu\mathrm{ML}^{\nabla}$ is disjunctive and $\mathcal{T}$ is an ω-expanded tree model with root r, then $\mathcal{T}, r \Vdash \varphi$ iff there is a winning strategy f for $\exists$ in the game $\mathcal{E}(\mathcal{M}, \varphi)$ with initial position (r, φ), which is scattered.*

2.6.2 Expressivity results

Most expressivity results consist in identifying a given logic X as a fragment of a well-known logic, playing the role of a yardstick. In case of modal logic and its extensions, first-order logic (FO) and monadic second-order logic (MSO) are the usual yardsticks.

Monadic second-order logic extends first-order logic by allowing quantification over subsets of the domain of the model. The introduction of MSO is related to the first decidability and undecidability results for logics of arithmetics. In the beginning of the 20th century, several decidability results were established, a famous and still useful example being the decidability of Presburger arithmetic [Rab77]. Presburger arithmetic is the set of valid first-order formulas in the structure $(\omega, +)$. Surprisingly, it turned out that that the first-order theory of $(\omega, +, \cdot)$ is undecidable [Göd31]. In the sixties, using automata theoretic methods, J. Richard Büchi [Büc60] and Calvin C. Elgot [Elg61] showed independently that the monadic second-order theory of $(\omega, <)$ is decidable. The result was extended to infinite binary trees (with a signature consisting of a “left child” relation and “right child” relation) by Michael Rabin [Rab69]. Both results gave a particularly nice status to MSO, as full second-order logic is usually undecidable, even over very simple classes of models.

Monadic second-order logic Let *Sig* be a set of predicates, each of them with a given arity. Fix also an infinite set Var_1 of first order variables and an infinite set Var_2 of second order variables. The set of *first-order formulas* (FO) over *Sig* is given by:

$$\varphi \ ::= \ (x = y) \ | \ P(x_1, \dots, x_n) \ | \ \varphi \vee \varphi \ | \ \neg\varphi \ | \ \exists x \varphi,$$

where $x, y, x_1, \dots, x_n$ belongs to Var_1 and P is a n-ary predicate in *Sig*.

The set of *monadic second-order* (MSO) formulas over the signature *Sig*, is defined inductively as follows:

$$\varphi \ ::= \ (x = y) \ | \ P(x_1, \dots, x_n) \ | \ x \in X \ | \ \varphi \vee \varphi \ | \ \neg\varphi \ | \ \exists x \varphi \ | \ \exists X \varphi,$$

where $x, y, x_1, \dots, x_n$ belongs to Var_1, X belongs to Var_2 and P is a n-ary predicate in *Sig*. As usual, we let $\forall x \varphi$ be an abbreviation for $\neg\exists x \neg\varphi$ and $\varphi \rightarrow \psi$ be an abbreviation for $\psi \vee \neg\varphi$.

Sometimes instead of φ, we may write $\varphi(x_1, \dots, x_n)$; this means that the free variables in φ are among $x_1, \dots, x_n$ in Var_1.

Semantics for MSO Let $\mathcal{M} = (W, P^{\mathcal{M}})$ be a structure for *Sig*. That is, W is a set and for each predicate P in *Sig*, $P^{\mathcal{M}}$ is a subset of W^n, where n is the arity of P.

Also let φ be an MSO formula over *Sig* and let $\tau : Var_1 \cup Var_2 \rightarrow W \cup \mathcal{P}(W)$ be an assignment such that for all $x \in Var_1$, $\tau(x) \in W$ and for all $X \in Var_2$, $\tau(X) \in \mathcal{P}(W)$. We define the relation $\mathcal{M} \vDash_\tau \varphi$ by induction on φ, in the following way:

$$\begin{array}{ll}
\mathcal{M} \vDash_\tau (x = y) & \text{if } \tau(x) = \tau(y), \\
\mathcal{M} \vDash P(x_1, \dots, x_n) & \text{if } (\tau(x_1), \dots, \tau(x_n)) \in P^{\mathcal{M}}, \\
\mathcal{M}, \vDash x \in X & \text{if } \tau(x) \in \tau(X), \\
\mathcal{M} \vDash_\tau \varphi \vee \psi & \text{if } \mathcal{M} \vDash_\tau \varphi \text{ or } \mathcal{M} \vDash_\tau \psi, \\
\mathcal{M} \vDash_\tau \neg\varphi & \text{if } \mathcal{M} \nvDash_\tau \varphi, \\
\mathcal{M} \vDash_\tau \exists x \varphi & \text{if there is } w \in W \text{ such that } \mathcal{M} \vDash_{\tau[x \mapsto w]} \varphi, \\
\mathcal{M} \vDash_\tau \exists X \varphi & \text{if there is } U \subseteq W \text{ such that } \mathcal{M} \vDash_{\tau[x \mapsto U]} \varphi,
\end{array}$$

where $x, y, x_1, \dots, x_n$ belong to Var_1, X belongs to Var_2 and P is a n-ary predicate in *Sig*. The map $\tau[x \mapsto w]$ is the assignment τ' such that $\tau'(x) = w$, for all $y \in Var_1 \backslash \{x\}$, $\tau'(y) = \tau(y)$ and for all $X \in Var_2$, $\tau'(X) = \tau(X)$. The map $\tau[X \mapsto U]$ is the assignment τ' such that $\tau'(X) = U$, $\tau(x) = \tau(x)$ for all $x \in Var_1$ and $\tau'(Y) = \tau(Y)$, for all $Y \in Var_2 \backslash \{X\}$.

If $\mathcal{M} \vDash_\tau \varphi$, we say that φ *holds* in $\mathcal{M}$ under the assignment τ. In case $\varphi = \varphi(x_1, \dots, x_n)$ (with $x_1, \dots, x_n \in Var_1$) and $\tau(x_i) = u_i$ for all $1 \leq i \leq n$, we might write $\mathcal{M}, (u_1, \dots, u_n) \vDash \varphi$ or $\mathcal{M} \vDash \varphi(u_1, \dots, u_n)$ instead of $\mathcal{M} \vDash_\tau \varphi$.

Kripke frames and Kripke models were introduced for interpreting formulas of modal logic. But they can also be seen as structures in the usual model theoretic sense. We fix here the signatures associated with these structures.

Monadic second-order logic over frames and models A Kripke frame $\mathbb{F} = (W, (R_a)_{a \in Act})$ can be seen as a structure over the signature $Sig := \{R_a \mid a \in Act\}$, where for all $a \in Act$, R_a is a binary predicate. For all $a \in Act$, we can define $R_a^{\mathbb{F}}$ as the relation $R_a \subseteq W^2$.

Similarly, a Kripke model $\mathcal{M} = (W, (R_a)_{a \in Act}, V)$ can be seen as structure over the signature $Sig := \{R_a \mid a \in Act\} \cup \{P \mid p \in Prop\}$, where for all $p \in Prop$, P is a unary predicate. We define $R_a^{\mathcal{M}}$ as $R_a \subseteq W^2$ (for all $a \in Act$) and $P^{\mathcal{M}}$ as the set $V(p)$ (for all $p \in Prop$).

As mentioned earlier, one of the nicest features of MSO is that it is decidable on some widely used classes of models. We state here the result in the case of tree models.

As mentioned, the original result was proved by Michael Rabin for binary tree models (instead of arbitrary tree models). However, using a standard construction which allows us to encode tree models into binary tree models, we can easily derive from Rabin's theorem the following result.

2.6.5. Theorem ([Rab69]). MSO *is decidable on tree models. That is, there is an algorithm that decides whether for a given* MSO $\varphi(x)$ *formula (over the appropriate signature for Kripke models), there exists a tree model* $\mathcal{T}$ *with root* r *such that* $\mathcal{T}, r \vDash \varphi(x)$.

The result remains true when we restrict our attention to finite tree models. That is, MSO is decidable on finite tree models (see [TW68] and [Don70]).

It is easy to see that on Kripke models (and on Kripke frames), MSO subsumes the μ-calculus. We start by introducing the notion of equivalence between a μ-sentence and an MSO formula. Next we define the standard translation, which allows us to embed the μ-calculus into MSO. Let us also remark that the expressive power of μ-calculus goes beyond first order logic; a simple μ-sentence like $\mu x.\Diamond x \vee p$ (there exists a path on which p is eventually true) cannot be expressed in first-order logic.

Equivalence between MSO and μ-calculus An MSO formula $\varphi(x)$ is *equivalent* on a class $\mathcal{C}$ of Kripke models to a μ-sentence ψ if for all models $\mathcal{M}$ in $\mathcal{C}$ and for all nodes $w \in \mathcal{M}$,

$$\mathcal{M}, w \vDash \varphi(x) \quad \text{iff} \quad \mathcal{M}, w \Vdash \psi.$$

If $\mathcal{C}$ is the class of all models, we simply say that $\varphi(x)$ and ψ are equivalent.

Standard translation Given a μ-formula φ and a variable x in Var_1, we define the MSO formula $ST_{x,\rho}(\varphi)$ (with one free variable x) by induction on the complexity of φ:

$$
\begin{array}{lcl}
ST_x(\top) & = & (x = x),\\
ST_x(\bot) & = & \neg(x = x),\\
ST_x(p) & = & P(x),\\
ST_x(\neg p) & = & \neg P(x),\\
ST_x(y) & = & x \in Y,\\
ST_x(\varphi \vee \psi) & = & ST_x(\varphi) \vee ST_x(\psi),\\
ST_x(\varphi \wedge \psi) & = & ST_x(\varphi) \wedge ST_x(\psi),\\
ST_x(\Diamond_a \varphi) & = & \exists z(xR_a z \wedge (ST_z(\varphi))),\\
ST_x(\Box_a \varphi) & = & \forall z(xR_a z \rightarrow ST_z(\varphi)),\\
ST_x(\alpha \bullet \nabla_a \Phi) & = & ST_x(\alpha) \wedge ST_x(\nabla_a \Phi),\\
ST_x(\nabla_a\{\varphi_1, \dots, \varphi_n\}) & = & \forall z(xR_a z \rightarrow (ST_z(\varphi_1) \vee \cdots \vee ST_z(\varphi_n)))\wedge\\
 & & \bigwedge\{\exists z(xR_a z \wedge ST_z(\varphi_i)) \mid 1 \leq i \leq n\},\\
ST_x(\mu y.\varphi) & = & \forall Y((\forall z(ST_z(\varphi) \rightarrow z \in Y)) \rightarrow x \in Y),\\
ST_x(\nu y.\varphi) & = & \exists Y((\forall z(z \in Y \rightarrow ST_z(\varphi)) \wedge x \in Y),
\end{array}
$$

where y belongs to Var, x, z belong to Var_1, Y belongs to Var_2 and a belongs to Act. $ST_x(\varphi)$ is called the *standard translation* of φ.

It is crucial that in the fifth equality and in last two equalities, we use specifically the second-order variable Y. The reason is that a bound variable y in a μ-formula φ corresponds to a second-order variable. Hence, in order to translate φ into a second-order formula, we have to fix an injective map that sends a variable in Var to a second-order variable in Var_2. We implicitly fix such a map by sending a variable y to Y, a variable y' to Y', etc.

2.6.6. Proposition. *For all μ-sentences φ, $ST_x(\varphi)$ and φ are equivalent.*

Note that in case φ is a modal formula, $ST_x(\varphi)$ is a first-order formula. In other words, the standard translation also embeds modal logic into first-order logic.

As we have seen, MSO is a good benchmark for the expressive power of μ-calculus. Obviously, not all MSO formulas are expressible in the μ-calculus. Examples are counting of successors ("a point has at least 2 a-transitions") or cyclicity ("a point has a sequence of transition that is eventually a cycle"). Previously we mentioned that an important feature of the μ-calculus is that a μ-formula cannot distinguish two bisimilar points. David Janin and Igor Walukiewicz showed that this property characterizes completely the μ-calculus as a fragment of MSO [JW96].

This characterization extends a result proved by Johan van Benthem: modal logic corresponds to the bisimulation invariant fragment of FO [Ben76].

Bisimulation invariance An MSO formula $\varphi(x)$ is *invariant* under bisimulation on a class $\mathcal{C}$ of models if for all bisimulations B between two models $\mathcal{M}$ and $\mathcal{M}'$ in $\mathcal{C}$ and for $(w, w') \in B$, we have

$$\mathcal{M}, w \vDash \varphi(x) \quad \text{iff} \quad \mathcal{M}', w' \vDash \varphi(x).$$

If $\mathcal{C}$ is the class of all models, we simply say that $\varphi(x)$ is invariant under bisimulation.

2.6.7. Theorem ([JW96]). *An* MSO *formula $\varphi(x)$ is equivalent to a μ-sentence iff $\varphi(x)$ is invariant under bisimulation.*

Moreover, given an MSO *formula $\varphi(x)$, we can compute a μ-sentence ψ such that $\varphi(x)$ and ψ are equivalent iff $\varphi(x)$ is equivalent to a μ-sentence.*

We sometimes refer to this theorem as the Janin-Walukiewicz theorem. The result remains true if we only consider tree models, or finitely branching models (instead of arbitrary Kripke models). However, it is unknown whether the μ-calculus is still the bisimulation invariant fragment of MSO on finite models.

2.6.3 Expressivity results for μ-programs

Sometimes it will be convenient (specially in Chapter 6) to talk about μ-programs [Hol98b], which are a natural generalization of PDL programs.

PDL was first defined by Michael Fischer and Robert Ladner [FL79] and was designed to reason about programs. The basic idea is to associate a modality with each program. The programs are obtained by combining atomic programs. The atomic programs are interpreted by the relations R_as, while the "combination operations" usually include composition of programs, union of programs, test on formulas and iteration of programs.

PDL The *PDL formulas and PDL programs* are defined by simultaneous induction as follows:

$$\begin{array}{lcl} \varphi & ::= & \top \mid \bot \mid p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle\theta\rangle\varphi \mid [\theta]\varphi, \\ \theta & ::= & R_a \mid \varphi? \mid \theta;\theta \mid \theta \cup \theta \mid \theta^*, \end{array}$$

where p belongs to *Prop* and a belongs to *Act*.

Semantics for PDL Fix a Kripke model $\mathcal{M} = (W, (R_a)_{a\in Act}, V)$. Given a PDL formula φ, we define the *meaning* of φ as a set $[\![\varphi]\!]_{\mathcal{M}} \subseteq W$. The meaning $[\![\theta]\!]_{\mathcal{M}}$ of a PDL program θ is a binary relation over W.

The definitions of $[\![\varphi]\!]_{\mathcal{M}}$ and $[\![\theta]\!]_{\mathcal{M}}$ proceed by simultaneous induction on the complexity of φ and θ. We add the following clauses to the clauses for the definition of the semantics of modal logic:

$$
\begin{aligned}
[\![\langle\theta\rangle\varphi]\!]_{\mathcal{M}} &= \{w \in W \mid \text{ there is } v \in [\![\varphi]\!]_{\mathcal{M}} \text{ such that } (w,v) \in [\![\theta]\!]_{\mathcal{M}} \},\\
[\![[\theta]\varphi]\!]_{\mathcal{M}} &= \{w \in W \mid \text{ for all } v \text{ such that } (w,v) \in [\![\theta]\!]_{\mathcal{M}}, \text{ we have } v \in [\![\varphi]\!]_{\mathcal{M}}\},\\
[\![R_a]\!]_{\mathcal{M}} &= R_a,\\
[\![\varphi?]\!]_{\mathcal{M}} &= \{(w,w) \mid w \in [\![\varphi]\!]_{\mathcal{M}}\},\\
[\![\theta;\gamma]\!]_{\mathcal{M}} &= \{(w,u) \in W^2 \mid \text{ for some } v \in W,\ (w,v) \in [\![\theta]\!]_{\mathcal{M}} \text{ and } (v,u) \in [\![\gamma]\!]_{\mathcal{M}}\},\\
[\![\theta \cup \gamma]\!]_{\mathcal{M}} &= [\![\theta]\!]_{\mathcal{M}} \cup [\![\gamma]\!]_{\mathcal{M}},\\
[\![\theta^*]\!]_{\mathcal{M}} &= \{(w,w) \mid w \in W\} \cup \{(w,v) \in W^2 \mid \text{ for some } n > 0 \text{ and } v_0, \dots, v_n,\\
&\ v_0 = w, v_n = v \text{ and for all } 0 \leq i \leq n-1,\ (v_i, v_{i+1}) \in [\![\theta]\!]_{\mathcal{M}}\},
\end{aligned}
$$

where a belongs to Act.

We can now define the μ-programs. They are defined in the same way as the PDL programs, except that we are able to test over μ-sentences.

μ-Program The *μ-programs* are given by

$$\theta \quad ::= \quad R \mid \varphi? \mid \theta;\theta \mid \theta \cup \theta \mid \theta^*,$$

where φ is a μ-sentence.

Semantics for μ-programs Given a model $\mathcal{M}$ with domain W, the *meaning* $[\![\theta]\!]_{\mathcal{M}}$ of a μ-program θ is a subset of W^2, which is defined by induction on θ. All the induction steps are the same as the ones for the definition of the semantics of PDL.

As mentioned earlier, the μ-programs do not increase the expressive power of the μ-calculus. With a μ-program θ, we can associate an universal modality $[\theta]$ and an existential modality $\langle\theta\rangle$. Given a μ-formula φ, the meanings of $[\theta]\varphi$ and $\langle\theta\rangle\varphi$ are defined similarly to the case of PDL. It is not hard to prove by induction on θ that $[\theta]\varphi$ and $\langle\theta\rangle\varphi$ are equivalent to μ-formulas (for more details, see [Hol98b]).

This immediately implies that each PDL formula is equivalent to a μ-formula. However, not all μ-formulas are equivalent to a PDL formula. It is shown in [Koz83] that the μ-formula $\mu x.\Box x$ (there is no infinite path starting from the point) is not expressible in PDL.

In fact, not only PDL can be seen a fragment of the μ-calculus, but most temporal logics. It is easy to translate each formula of CTL [CE81] into an equivalent μ-formula. A much harder problem is to find a truth preserving translation for the state formulas of CTL* [EL86] into the μ-calculus. The most optimal translation (which is exponential in the size of the original formula) was given by Girish Bhat and Rance Cleaveland [BC96].

Next we state a result that is an extension of the Janin-Walukiewicz theorem to the setting of μ-programs. That is, we characterize the μ-programs as a fragment of MSO.

We start by giving a formal definition of equivalence between MSO formulas and μ-programs. Next we define the notion of safety for bisimulations, which was introduced by Johan van Benthem [Ben98]. Safety for bisimulation is basically a generalization of the two last clauses in the definition of bisimulation. Finally we state the expressiveness result for the μ-programs, which was proved by Marco Hollenberg [Hol98b]. We will use this result in Chapter 6.

Equivalence between MSO and the μ-programs An MSO formula $\varphi(x, y)$ is *equivalent* to a μ-program θ on a class $\mathcal{C}$ of models, if for all models $\mathcal{M} \in \mathcal{C}$ and for all w, v in the domain of $\mathcal{M}$

$$\mathcal{M}, (w, v) \models \varphi(x, y) \quad \text{iff} \quad (w, v) \in [\![\theta]\!]_{\mathcal{M}}.$$

Safety for bisimulations An MSO formula $\varphi(x, y)$ is *safe for bisimulations* on a class $\mathcal{C}$ of models if for all bisimulations B between two models $\mathcal{M}$ and $\mathcal{M}'$ in $\mathcal{C}$ and for all $(w, w') \in B$,

- if there is $v \in W$ such that $(w, v) \in [\![\theta]\!]_{\mathcal{M}}$, then there is $v' \in W'$ such that $(w', v') \in [\![\theta]\!]_{\mathcal{M}'}$,
- if there is $v' \in W'$ such that $(w', v') \in [\![\theta]\!]_{\mathcal{M}'}$, then there is $v \in W$ such that $(w, v) \in [\![\theta]\!]_{\mathcal{M}}$,

where W is the domain of $\mathcal{M}$ and W' is the domain of $\mathcal{M}'$.

2.6.8. THEOREM ([HOL98B]). *An MSO formula $\varphi(x, y)$ is equivalent to a μ-program iff $\varphi(x, y)$ is safe for bisimulations.*

Moreover, given an MSO formula $\varphi(x, y)$, we can compute a μ-program θ such that $\varphi(x, y)$ and θ are equivalent iff $\varphi(x, y)$ is equivalent to a μ-program.

2.7 Graded μ-calculus

The graded μ-calculus is obtained by adding graded modalities to the μ-calculus (or fixpoint operators to graded modal logic). The idea of adding graded modalities to modal logic (and its extensions) originates in the 1970s [Fin72, Gob70]. Graded modalities generalize the usual universal and existential modalities in that they can express that a point has at most or at least k successors making true a certain formula.

Syntax The set μGL of *graded μ-formulas* is defined by induction as follows:

$$\varphi ::= \top \mid \bot \mid p \mid \neg p \mid x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond_a^k \varphi \mid \Box_a^k \varphi \mid \mu x.\varphi \mid \nu x.\varphi,$$

where p belongs to *Prop*, a belongs to *Act*, k is a natural number and x belongs to *Var*.

Semantics for μGL Fix a Kripke model $\mathcal{M} = (W, (R_a)_{a \in Act}, V)$ and an assignment $\tau : Var \to \mathcal{P}(W)$. Given a graded μ-formula φ, we define by induction on φ the *meaning* of φ as a set $[\![\varphi]\!]_{\mathcal{M}} \subseteq W$. The definition of the meaning is obtained by adding the following clauses to the definition of the semantics for μML:

$$\begin{array}{rcl} [\![\Diamond_a^k \varphi]\!]_{\mathcal{M},\tau}\varphi & = & \{w \in W \mid |R_a[w] \cap [\![\varphi]\!]_{\mathcal{M},\tau}| > k\}, \\ [\![\Box_a^k \varphi]\!]_{\mathcal{M},\tau}\varphi & = & \{w \in W \mid |R_a[w] \cap (W \backslash [\![\varphi]\!]_{\mathcal{M},\tau})| \leq k\}, \end{array}$$

where a belongs to *Act* and k is a natural number.

Concerning the expressive power of the graded μ-calculus, there exists a result similar to the Janin-Walukiewicz theorem. It should be clear from their definitions, that the graded μ-formulas are not bisimulation invariant. A relevant notion of bisimulation can be obtained by strengthening the notion of bisimulation with some condition related to counting. Counting bisimulations were introduced by David Janin and Giacomo Lenzi [JL03].

Let us also mention that a notion of g-bisimulation for graded modal logic has been introduced by Maarten de Rijke [Rij00]. Two points linked by a counting bisimulation are g-bisimilar. However, the converse is not true. For our purpose, it is more convenient to use the notion of counting bisimulation than the notion of g-bisimulation.

Counting bisimulation Let $\mathcal{M} = (W, (R_a)_{a \in Act}, V)$ and $\mathcal{M}' = (W', (R'_a)_{a \in Act}, V')$ be two Kripke models. A relation $B \subseteq W \times W'$ is a *counting bisimulation* between $\mathcal{M}$ and $\mathcal{M}'$ if

- B is a bisimulation between $\mathcal{M}$ and $\mathcal{M}'$,
- for all $a \in Act$ and all $(w, w') \in B$, there exists a bijection $f : R_a[w] \to R'_a[w']$ such that for all $v \in R_a[w]$, we have $(v, f(v)) \in B$.

We define the equivalence between an MSO formula $\varphi(x)$ and a graded μ-calculus φ, in the same way we defined the equivalence between an MSO formula $\varphi(x)$ and a μ-formula.

2.7.1. Theorem (from [Wal02],[Jan06]). *An* MSO *formula $\varphi(x)$ is equivalent to a graded μ-formula iff $\varphi(x)$ is invariant under counting bisimulation.*

Moreover, given an MSO *formula $\varphi(x)$, we can compute a graded μ-formula ψ such that $\varphi(x)$ and ψ are equivalent iff $\varphi(x)$ is equivalent to a graded μ-formula.*

It is worth noting that invariance under counting bisimulation is equivalent to invariance under unravelling, since two pointed models are counting bisimilar iff their unravelling are isomorphic.

Chapter 3

An easy completeness proof for the μ-calculus on finite trees

When investigating a logic, a natural question to ask is whether there exists a nice axiomatization for this logic. The proof might provide some insight about the logic. Moreover, a nice axiomatization would already be a good indication that the logic is well-behaved. For example, the existence of a finite axiomatization implies that the satisfiability problem for the logic is semi-decidable: there is an algorithm that, given a formula, always terminates if the formula is not satisfiable and otherwise, keeps on running (simply consider the algorithm that examines all the possible formal proofs with respect to the axiomatization and check whether the conclusion is the input formula).

For the μ-calculus, a natural axiomatization (that we denote by K^{μ}) was proposed by Dexter Kozen in 1983 [Koz95]. This axiomatization consists in enriching the standard axiomatization for basic modal logic with one axiom (with the intended meaning that $\mu x.\varphi$ is a pre-fixpoint of the map induced by φ) and a rule (ensuring that all the pre-fixpoints of the map induced by φ, are implied by $\mu x.\varphi$). In the same paper, Dexter Kozen showed completeness for a fragment of the μ-calculus, called the aconjunctive fragment. The proof was essentially using tableau constructions.

Showing completeness of the μ-calculus(with respect to Kozen's axiomatization) appeared to be a hard problem, which was solved after ten years by Igor Walukiewicz [Wal95]. The proof is involved and uses the disjunctive normal form for μ-formulas and the weak aconjunctive fragment (which is an extension of the aconjunctive fragment). The disjunctive formulas behave nicely with respect to provability and one can show that if a disjunctive formula is not provable, then the negation of the formula is satisfiable. Hence, in order to obtain completeness for the full μ-calculus, it is sufficient to prove that each μ-sentence is provably equivalent to a disjunctive formula (in fact, even something weaker than that is sufficient). This is the hard part of the proof and it uses the notion of weak aconjunctivity.

In this chapter, we prove completeness of the Kozen's axiomatization together with the axiom $\mu x.\Box x$ with respect to the class of finite trees. It is easy to see that $\mu x.\Box x$ is valid on a frame iff the frame does not contain any infinite path. Adding this formula to the axiomatization helps us to obtain an easier proof for completeness. It should be mentioned that our result can be derived from the completeness result proved by Igor Walukiewicz (we give more details at the end of Section 3.4). It is also fair to say that on finite trees, the expressive power of the μ-calculus is rather limited. This is illustrated by the fact that for all graded μ-formulas φ, $\mu x.\varphi$ and $\nu x.\varphi$ are equivalent on finite trees.

However, we believe that the alternative proof proposed here is of interest for the following reasons. As mentioned already, it is a much simpler proof (but let us emphasize again that the setting is restricted in comparison with the original proof). Moreover, our proof can be extended to other settings: We obtain a similar result for the graded μ-calculus and we also show that when we add finitely many shallow axioms (as defined in [Cat05]) to the logic $\mathsf{K}^\mu + \mu x.\Box x$, we get a complete axiomatization for the corresponding class of finite trees. The probably most relevant feature of our proof are the tools that we use. These tools are completely different than the ones of the original proof, which uses tableau and automata (as the disjunctive normal form is deeply linked to the automata perspective on μ-calculus).

Our approach is inspired by model theoretic methods. In order to illustrate this, let us give a short summary of the structure of our proof. The proof is in three steps. The first step consists of defining a notion of rank for μ-formulas, which plays the same role as the modal depth for modal formulas and the quantifier depth for first order formulas. The second step is to prove completeness of the μ-calculus with respect to generalized models, which are Kripke models augmented with a set of admissible subsets, in the style of Henkin semantics for second order logic.

The last step is the one which has most of the connections with model theory. In fact, it is inspired by the work of Kees Doets [Doe89]. More specifically, by the method that he uses to provide complete axiomatizations of the monadic Π^1_1-theories of well-founded linear orders, the reals and certain classes of trees. An essential notion for this last step is the notion of n-goodness. As we will see from its definition, n-goodness shares some similarities with the model theoretic notion of n-elementary equivalence. Moreover, a crucial part of this last step consists in "gluing" together a collection of finite trees and during this process, we also want the truth of formulas of rank n to be preserved. This is a typical model theoretic argument: constructing new structures out of existing ones and preserving elementary equivalence (see for example the Feferman-Vaught theorem in [FV59]).

The chapter is organized as follows. In Section 3.1, we recall Kozen's axiomatization and fix some notation. In Section 3.2, we define the notion of rank for a formula. In Section 3.3, we introduce the notion of generalized model and we show completeness of K^μ with respect to the class of generalized models. In Section 3.4, we use Kees Doets' argument to obtain completeness of $\mathsf{K}^\mu + \mu x.\Box x$ with respect to the class of finite trees. In the next two sections, we give some examples of extensions of $\mathsf{K}^\mu + \mu x.\Box x$ to which we can apply our method in order to prove completeness.

3.1 Preliminaries

We start by recalling Kozen's axiomatization (already mentioned in Chapter 2) and fixing some notation relative to axiomatic systems.

Convention In this chapter, we do not use the ∇ operator. So whenever we write "μ-formula", we refer to a formula in μML (and not in μML^∇).

Kozen's axiomatization The axiomatization of the Kozen system K^μ consists of the following axioms and rules

$$\begin{array}{lr}
\text{propositional tautologies,} & \\
\text{if } \vdash \varphi \to \psi \text{ and } \vdash \varphi, \text{ then } \vdash \psi & \text{(Modus Ponens),}\\
\text{if } \vdash \varphi, \text{ then } \vdash \varphi[p/\psi] & \text{(Substitution),}\\
\vdash \Box(p \to q) \to (\Box p \to \Box q) & \text{(}\mathsf{K}\text{-axiom),}\\
\text{if } \vdash \varphi, \text{ then } \vdash \Box\varphi & \text{(Necessitation),}\\
\vdash \varphi[x/\mu x.\varphi] \to \mu x.\varphi & \text{(Fixpoint axiom),}\\
\text{if } \vdash \varphi[x/\psi] \to \psi, \text{ then } \vdash \mu x.\varphi \to \psi & \text{(Fixpoint rule),}
\end{array}$$

where p and q are proposition letters, φ and ψ are μ-formulas, x is not a bound variable of φ and no free variable of ψ is bound in φ.

If Φ is a set of modal formulas, we write $\mathsf{K} + \Phi$ for the smallest set of modal formulas that contains the propositional tautologies, the K-axiom and is closed under the Modus Ponens, Substitution and Necessitation rules. We say that $\mathsf{K}+\Phi$ is the *extension* of K by Φ. If Φ is empty, we simply write K.

Next, if Φ is a set of modal formulas, we denote by $\mathsf{K} +_r \Phi$ the smallest set of formulas that contains both K and Φ and is closed under the Modus Ponens and Necessitation rules. We call $\mathsf{K} +_r \Phi$ the *restricted extension* of K by Φ.

Finally, if Φ is a set of μ-sentences, we write $\mathsf{K}^\mu + \Phi$ for the smallest set of formulas that contains both K^μ and Φ and is closed under the Modus Ponens, Substitution, Necessitation and Fixpoint rules. We say that $\mathsf{K}^\mu + \Phi$ is the *extension* of K^μ by Φ.

As mentioned in the introduction, a key formula in this chapter is the formula $\mu x.\Box x$.

The formula $\mu x.\Box x$ A *path* in a model $\mathcal{M} = (W, R, V)$ is a sequence $(w_i)_{i<\kappa}$ such that $\kappa \in \mathbb{N} \cup \{\omega\}$ and for all $i + 1 < \kappa$, $(w_i, w_{i+1}) \in R$. It is easy to see that the formula $\mu x.\Box x$ is true at a point w in a model $\mathcal{M}$ iff there is no infinite path $(w_i)_{i<\omega}$ such that $w_0 = w$. In particular, $\mu x.\Box x$ is valid in a frame iff the frame does not contain any infinite path.

3.2 Rank of a formula

The goal of this section is to introduce a definition of rank that would be the analogue of the depth of a modal formula. For modal logic, the truth of a formula φ of modal depth n at a point w is determined by the proposition letters that are true at w and by the truth of the formulas of depth $\leq n$ at the successors of w. In our proof, we will need something similar for the μ-calculus.

The most natural idea would be to look at the nesting depth of modal and fixpoint operators (which is defined below, but intuitively, this is the most straightforward generalization of the notion of modal depth). However, this definition does not have the required properties[1].

The notion of rank that we develop in this section is in fact related to the closure of a formula, which has been introduced by Michael Fischer and Robert Ladner [FL79] and already mentioned in Chapter 2.

Closure, depth and rank of a formula We recall that the *closure* $Cl(\varphi)$ of a formula $\varphi \in \mu$ML is the smallest set of formulas such that

$\varphi \in Cl(\varphi)$,
if $\neg p \in Cl(\varphi)$, $p \in Cl(\varphi)$,
if $\psi \vee \chi$ or $\psi \wedge \chi$ belongs to $Cl(\varphi)$, then both $\psi, \chi \in Cl(\varphi)$,
if $\Diamond\psi$ or $\Box\varphi$ belongs to $Cl(\varphi)$, then $\psi \in Cl(\varphi)$,
if $\mu x.\psi \in Cl(\varphi)$, then $\psi[x/\mu x.\psi] \in Cl(\varphi)$,
if $\nu x.\psi \in Cl(\varphi)$, then $\psi[x/\nu x.\psi] \in Cl(\varphi)$.

It is also proved in [Koz95] that the closure $Cl(\varphi)$ of a formula φ is finite. In order to define the rank, we also need to introduce the notion of (nesting) depth of a formula.

[1]Look for example at the formula $\varphi = \mu x.\Diamond\Diamond(x \vee p)$ (φ is true at a point w in a model (W, R, V) iff there exists a natural number i and a path $(w_j)_{j<2i+1}$ such that $w_0 = w$ and p is true at w_{2i}). Obviously, φ is equivalent to $\Diamond\Diamond(\varphi \vee p)$. Thus, the truth of φ at a point w only depends on the truth of $\psi = \Diamond(\varphi \vee p)$ at the successors of w. However, the nesting depth of ψ is greater than the nesting depth of φ.

The *depth* $d(\varphi)$ of a formula φ is defined by induction as follows:

$$\begin{aligned} d(p) = d(\neg p) = d(x) &= 0, \\ d(\varphi \vee \psi) = d(\varphi \wedge \psi) &= max\{d(\varphi), d(\psi)\}, \\ d(\Diamond\varphi) = d(\Box\varphi) &= d(\varphi) + 1, \\ d(\mu x.\varphi) = d(\nu x.\varphi) &= d(\varphi) + 1. \end{aligned}$$

Finally, the *rank* of a formula φ is defined by:

$$rank(\varphi) = max\{d(\psi) \mid \psi \in Cl(\varphi)\}.$$

Remark that since $Cl(\varphi)$ is finite, $rank(\varphi)$ is well-defined (that is, the max is actually reached). All we will use later are the following properties of the rank.

3.2.1. Proposition. *If the set Prop of proposition letters is finite, then for all natural numbers n, there are only finitely many sentences of rank n (up to logical equivalence).*

Proof Fix a natural number n. If $rank(\varphi) = n$, then in particular, $d(\varphi) \leq n$. Hence, it is sufficient to show that there only finitely many sentences of depth less or equal to n (up to logical equivalence). There is a finite set of variables $\{x_1, \ldots, x_k\}$ such that the variables occurring in φ are among $x_1, \ldots, x_k$. So we can restrict ourselves to show that there are finitely many formulas (up to logical equivalence) of depth n over the set *Prop* and the set of variables $\{x_1, \ldots, x_k\}$. The proof is by induction on k. We do not give details, as it is similar to the proof that there only finitely many modal formulas (up to logical equivalence) of modal depth n over a finite set of proposition letters. □

3.2.2. Proposition. *The rank is closed under boolean combination. That is, for any n, a boolean combination of formulas of rank at most n is a formula of rank at most n.*

3.2.3. Proposition. *Every formula φ is provably equivalent to a boolean combination of proposition letters and formulas of the form $\Diamond\psi$ or $\Box\psi$, with $rank(\psi) \leq rank(\varphi)$.*

Proof Let φ be a μ-formula. As mentioned in the preliminaries, every μ-formula is equivalent to a guarded formula. In fact, the two formulas are even provably equivalent. A close inspection of the proof in [Koz83] shows that when transforming a formula into a provably equivalent guarded formula, the rank remains unchanged. Hence, we may assume that φ is guarded. We define a map G by induction as follows:

$$\begin{aligned} G(p) &= p, \\ G(\neg p) &= \neg p, \\ G(x) &= x \\ G(\psi \wedge \psi') &= G(\psi) \wedge G(\psi') \\ G(\psi \vee \psi') &= G(\psi) \vee G(\psi'), \\ G(\Diamond \psi) &= \Diamond \psi, \\ G(\Box \psi) &= \Box \psi, \\ G(\mu x.\psi) &= G(\psi[x/\mu x.\psi]), \\ G(\nu x.\psi) &= G(\psi[x/\nu x.\psi]), \end{aligned}$$

where p is a proposition letter and x is a variable. Using the fact that φ is guarded, one can show that the computation of $G(\varphi)$ does terminate. It is not hard to see that $G(\varphi)$ is equivalent to φ. So it remains to show that $G(\varphi)$ is a boolean combination of proposition letters and formulas of the form $\Diamond\psi$ and $\Box\psi$, with $rank(\psi) \leq rank(\varphi)$. It follows from the definition of G and the fact that φ is guarded that $G(\varphi)$ is a boolean combination of proposition letters and formulas of the form $\Diamond\psi$ and $\Box\psi$, where ψ belongs to the closure of φ. Now if a formula ψ belongs to $Cl(\varphi)$, then $Cl(\psi)$ is a subset of $Cl(\varphi)$. As a corollary, the rank of ψ is less or equal to the rank of φ, which concludes the proof. □

3.3 Completeness for generalized models

We introduce a semantics for the μ-calculus that is based on structures that we call generalized models. The goal of this section is to prove completeness of K^μ with respect to the class of generalized models.

The semantics presented here is the analogue of the Henkin semantics for second order logic (see [Hen50]). The "Henkin trick" consists in restricting the range of the predicates in the formulas. In general, such a restriction brings the complexity of the logic down, as illustrated by the case of second order logic. Another example can be found in [Ben05], where Johan van Benthem shows that an adaptation of the Henkin semantics makes the fixpoint extension of first order logic decidable.

However, in the case of modal logic and μ-calculus, the situation is different. It follows from the completeness theorem of this section (Corollary 3.3.3) and the completeness of μ-calculus (proved in [Wal95]) that a formula is true in all Kripke models iff it is true in all generalized models. In particular, the logic corresponding to the standard semantics and the logic corresponding to the "Henkin semantics" coincide. Finally let us emphasize that all the material presented in this section is not specific to the setting of finite trees.

Generalized models Consider a quadruple $\mathcal{M} = (W, R, V, \mathbb{A})$, where (W, R) is a Kripke frame, $\mathbb{A}$ is a subset of $\mathcal{P}(W)$ and $V : \mathit{Prop} \longrightarrow \mathbb{A}$ is a valuation. A set which belongs to $\mathbb{A}$ is called *admissible*.

We define the truth of a formula φ under an assignment $\tau : \mathit{Var} \longrightarrow \mathbb{A}$ by induction. All the clauses are the same as usual, except the one defining the truth of $\mu x.\varphi$. Normally, we define the set $[\![\mu x.\varphi]\!]_{\mathcal{M},\tau}$ as the least pre-fixpoint of the map φ_x. But here, we define it as the intersection of all the admissible pre-fixpoints of φ_x. The set $[\![\varphi]\!]_{\mathcal{M},\tau}$ is defined by induction on φ as follows:

$$
\begin{aligned}
[\![p]\!]_{\mathcal{M},\tau} &= V(p),\\
[\![\neg p]\!]_{\mathcal{M},\tau} &= S\backslash V(p),\\
[\![x]\!]_{\mathcal{M},\tau} &= \tau(x),\\
[\![\varphi \vee \psi]\!]_{\mathcal{M},\tau} &= [\![\varphi]\!]_{\mathcal{M},\tau} \cup [\![\psi]\!]_{\mathcal{M},\tau},\\
[\![\varphi \wedge \psi]\!]_{\mathcal{M},\tau} &= [\![\varphi]\!]_{\mathcal{M},\tau} \cap [\![\psi]\!]_{\mathcal{M},\tau},\\
[\![\Diamond\varphi]\!]_{\mathcal{M},\tau} &= \{w \in W \mid \text{ there exists a successor of } w \text{ in } [\![\varphi]\!]_{\mathcal{M},\tau}\},\\
[\![\Box\varphi]\!]_{\mathcal{M},\tau} &= \{w \in W \mid \text{ all the successors of } w \text{ belongs to } [\![\varphi]\!]_{\mathcal{M},\tau}\},\\
[\![\mu x.\varphi]\!]_{\mathcal{M},\tau} &= \bigcap\{U \in \mathbb{A} \mid [\![\varphi]\!]_{\mathcal{M},\tau[x\mapsto U]} \subseteq U\},\\
[\![\nu x.\varphi]\!]_{\mathcal{M},\tau} &= \bigcup\{U \in \mathbb{A} \mid U \subseteq [\![\varphi]\!]_{\mathcal{M},\tau[x\mapsto U]}\}.
\end{aligned}
$$

If $w \in [\![\varphi]\!]_{\mathcal{M},\tau}$, we write $\mathcal{M}, w \Vdash_\tau \varphi$ and we say that φ is *true* at w under the assignment τ. If φ is a sentence, the truth of φ does not depend on the assignment τ and we simply write $\mathcal{M}, w \Vdash \varphi$. A formula φ is *true* in $\mathcal{M}$ under an assignment τ if for all $w \in W$, we have $\mathcal{M}, w \Vdash_\tau \varphi$. In this case, we write $\mathcal{M} \Vdash_\tau \varphi$.

Note that if the set $\mathbb{A}$ of admissible sets is arbitrary, it is possible that for some formulas φ, $[\![\varphi]\!]_{\mathcal{M},\tau}$ does not belong to $\mathbb{A}$. We say that a quadruple $\mathcal{M} = (W, R, V, \mathbb{A})$ is a *generalized model* if for all formulas φ and all assignments $\tau : \mathit{Var} \longrightarrow \mathbb{A}$, the set $[\![\varphi]\!]_{\mathcal{M},\tau}$ belongs to $\mathbb{A}$. A triple $\mathbb{F} = (W, R, \mathbb{A})$ is a *generalized frame* if for every valuation $V : \mathit{Prop} \longrightarrow \mathbb{A}$, the quadruple $(W, R, V, \mathbb{A})$ is a generalized model. For people familiar with modal algebras, we observe that in particular, $\mathbb{A}$ is a modal algebra, which is not necessarily complete, but for which some special infinite meets and joins exist (namely the ones corresponding to the construction of the least and greatest fixpoints, as described above).

If $\mathbb{F} = (W, R, \mathbb{A})$ is a generalized frame, we call (W, R) the *underlying Kripke frame* of $\mathbb{F}$. A formula φ is *valid* in a generalized frame $\mathbb{F} = (W, R, \mathbb{A})$, notation: $\mathbb{F} \Vdash \varphi$, if for all valuations $V : \mathit{Prop} \longrightarrow \mathbb{A}$ and all assignments $\tau : \mathit{Var} \longrightarrow \mathbb{A}$, the formula φ is true in $(W, R, V, \mathbb{A})$ under the assignment τ.

Any Kripke model $\mathcal{M} = (W, R, V)$ can be seen as the generalized model $\mathcal{M}' = (W, R, V, \mathcal{P}(W))$. It follows easily from our definition that for all formulas φ and all points $w \in W$,

$$\mathcal{M}, w \Vdash \varphi \quad \text{iff} \quad \mathcal{M}', w \Vdash \varphi.$$

Now we show that for all sets Φ of formulas, the logic $\mathsf{K}^\mu + \Phi$ is complete with respect to a particular generalized model. This is actually more general than what we need in order to prove the completeness of $\mathsf{K}^\mu = \mu x.\Box x$ (it would be sufficient to know that K^μ is complete with respect to a particular generalized model). But showing this more general result will help us to extend the completeness of K^μ to other settings.

First we introduce some definitions and recall some results of modal logic.

3.3.1. Theorem (from [BRV01]). *Let Φ be a set of modal formulas over a set Prop of proposition letters. There exists a model $\mathcal{M}$ over Prop such that for all modal formulas φ over Prop, φ is provable in $\mathsf{K} +_r \Phi$ iff $\mathcal{M} \Vdash \varphi$.*

Proof Fix a set Φ of modal formulas. We define the *canonical model* $\mathcal{M} = (W, R, V)$ for Φ as follows. A set Γ of modal formulas over *Prop* is Φ-consistent if $\bot$ does not belong to $\mathsf{K} +_r \Phi$ and there is no finite subset $\{\gamma_1, \dots, \gamma_n\}$ of Γ such that $\gamma_1 \wedge \dots \gamma_n \rightarrow \bot$ belongs to $\mathsf{K} +_r \Phi$. Moreover, Γ is *maximal Φ-consistent* if for all Φ-consistent sets Γ' such that $\Gamma \subseteq \Gamma'$, we have $\Gamma = \Gamma'$.

The set W is the set of maximal Φ-consistent sets of modal formulas over *Prop*. The relation R is defined such that for $\Gamma, \Gamma' \in W$, $\Gamma R \Gamma'$ iff for all $\varphi \in \Gamma'$, $\Diamond\varphi$ belongs to Γ. Finally, we define V such that for all proposition letters p, $V(p) = \{\Gamma \in W \mid p \in \Gamma\}$.

It follows from the proof of in [BRV01] that for all formulas φ, $[\![\varphi]\!]_{\mathcal{M}} = \{\Gamma \in W \mid \varphi \in \Gamma\}$. As a corollary, φ is provable in $\mathsf{K} +_r \Phi$ iff $\mathcal{M} \Vdash \varphi$. $\Box$

Replacement of a formula Let *Prop* be a set of proposition letters and *Var* a set of variables. We let μFL be the set of sentences of the form $\mu x.\varphi$ or $\nu x.\varphi$, for some μ-formula φ over *Prop*. We denote by $Prop^+$ the set $Prop \cup \{p_\varphi \mid \varphi \in \mu\mathrm{FL}\}$. Given a μ-formula φ over $Prop^+$, we say that a subformula ψ of φ is a *maximal subformula* of φ in μFL if ψ belongs to μFL and there is no subformula ξ of φ such that ξ belongs to μFL and ψ is a subformula of ξ.

If φ is a μ-formula over $Prop^+$, we define $s(\varphi)$ as the formula obtained by replacing each proposition letter of the form p_ψ ($\psi \in \mu$FL), by the formula ψ. We call $s(\varphi)$ the *source* of φ.

Next, if φ is a μ-formula over $Prop^+$, we say that a modal formula ψ over $Prop^+$ is the *replacement* of φ if ψ is obtained by replacing, in the formula $s(\varphi)$, all maximal subformulas χ in μFL by the proposition letter p_χ. In this case, we use the notation $repl(\varphi)$. Finally, if Φ is a set of μ-formulas over *Prop*, we let $repl(\Phi)$ be the set $\{repl(\varphi) \mid \varphi \in \Phi\}$.

For example, let φ be the formula $\Diamond(\mu x.p_{\nu y.x \wedge y})$. Then $s(\varphi)$ is the formula $\Diamond(\mu x.\nu y.(x \wedge y))$ and $repl(\varphi)$ is the formula $\Diamond p_{\mu x.\nu y.(x \wedge y)}$.

We observe that $repl(\varphi)$ is a modal formula over $Prop^+$. Moreover, if φ is a modal formula, then $repl(\varphi) = \varphi$. Let us also mention that $s(repl(\varphi)) = s(\varphi)$ and $repl(s(\varphi)) = repl(\varphi)$.

Now we prove that for all sets of formulas Φ, the logic $\mathsf{K}^\mu + \Phi$ is complete with respect to the class of generalized models in which Φ is true. An easy way to show this would be to do a standard canonical model construction (inspired by the one used for the completeness of the modal logic K).

We give here another proof. The idea is to use the replacement map introduced previously in order to translate the completeness result for modal logic into a completeness result for generalized Kripke models. This proof might seem a bit more tedious but it will help us to extend our result to other settings (like graded μ-calculus).

3.3.2. Theorem. *Let Φ be a set of μ-formulas over a set Prop. There is a generalized model $\mathcal{M} = (W, R, V, \mathbb{A})$ such that for all sentences φ, φ is provable in $\mathsf{K}^\mu + \Phi$ iff $\mathcal{M} \Vdash \varphi$.*

Proof By Theorem 3.3.1, there is a Kripke model $\mathcal{N} = (W, R, V^+)$ (over $Prop^+$) such that for all modal formulas θ over $Prop^+$, θ is provable in $\mathsf{K} +_r repl(\mathsf{K}^\mu + \Phi)$ iff $\mathcal{N} \Vdash \theta$. Now let $\mathbb{A}$ be the set $\{[\![\delta]\!]_{\mathcal{N}} \mid \delta$ modal formula over $Prop^+\}$. We define $\mathcal{M}$ as the quadruple $(W, R, V^+, \mathbb{A})$.

First, we show that for all μ-formulas φ over $Prop^+$, all w in W and all assignments $\tau : Var \to \mathbb{A}$, we have

$$\mathcal{N}, w \Vdash_\tau repl(\varphi) \quad \text{iff} \quad \mathcal{M}, w \Vdash_\tau \varphi. \tag{3.1}$$

The proof is by induction on the alternation depth of φ. The basic case where φ is a modal formula over $Prop^+$ is immediate. For the induction step, fix a formula φ of alternation depth $n + 1$. To show that equivalence (3.1) holds for φ, we start a second induction on the complexity of φ. We only give details for the most difficult case. That is, φ is a formula of the form $\mu x.\chi$, where χ is a μ-formula over $Prop^+$ the alternation depth of which is less or equal to n.

For the direction from left to right of equivalence (3.1), suppose that $\mathcal{N}, w \Vdash_\tau repl(\mu x.\chi)$. In order to show that $\mathcal{M}, w \Vdash \mu x.\chi$, we have to prove that for all U in $\mathbb{A}$ such that $[\![\chi]\!]_{\mathcal{M},\tau[x \mapsto U]} \subseteq U$, w belongs to U.

Fix an admissible set U in $\mathbb{A}$ such that $[\![\chi]\!]_{\mathcal{M},\tau[x \mapsto U]} \subseteq U$. By definition of $\mathbb{A}$, there is a modal formula δ over $Prop^+$ such that $U = [\![\delta]\!]$. Now, since $[\![\chi]\!]_{\mathcal{M},\tau[x \mapsto [\![\delta]\!]]} \subseteq [\![\delta]\!]$, we have that for all $v \in W$, if $\mathcal{M}, v \Vdash_{\tau[x \mapsto [\![\delta]\!]]} \chi$, then $\mathcal{M}, v \Vdash_\tau \delta$. By a standard induction, we can show that for all $v \in W$, $\mathcal{M}, v \Vdash_{\tau[x \mapsto [\![\delta]\!]]} \chi$ iff $\mathcal{M}, v \Vdash_\tau \chi[\delta/x]$. Therefore, for all $v \in W$, if $\mathcal{M}, v \Vdash_\tau \chi[\delta/x]$, then $\mathcal{M}, v \Vdash_\tau \delta$. That is, for all $v \in W$,

$$\mathcal{M}, v \Vdash_\tau \chi[\delta/x] \to \delta.$$

Using the (first) induction hypothesis, we get that for all $v \in W$, $\mathcal{M}, v \Vdash_\tau \chi[\delta/x]$ iff $\mathcal{N}, v \Vdash_\tau repl(\chi[\delta/x])$. It also follows from the induction hypothesis that for all $v \in W$, $\mathcal{M}, v \Vdash_\tau \delta$ iff $\mathcal{N}, v \Vdash_\tau repl(\delta)$. Moreover, since δ is a modal formula, we have $repl(\delta) = \delta$. Putting everything together, we obtain that for all $v \in W$,

$$\mathcal{N}, v \Vdash_\tau repl(\chi[\delta/x]) \to \delta.$$

By definition of $\mathcal{N}$, this means that $\mathit{repl}(\chi[\delta/x]) \longrightarrow \delta$ is provable in $\mathsf{K} +_r \mathit{repl}(\mathsf{K}^\mu + \Phi)$.

We show that it implies that

$$s(\mathit{repl}(\chi[\delta/x]) \longrightarrow \delta) \text{ is provable in } \mathsf{K}^\mu + \Phi. \tag{3.2}$$

Since $\mathit{repl}(\chi[\delta/x]) \longrightarrow \delta$ is provable in $\mathsf{K} +_r \mathit{repl}(\mathsf{K}^\mu + \Phi)$, there is a formal proof for $\mathit{repl}(\chi[\delta/x]) \longrightarrow \delta$ in the system $\mathsf{K} +_r \mathit{repl}(\mathsf{K}^\mu + \Phi)$. By replacing each formula ψ in each line of this formal proof by its source $s(\psi)$, we obtain a formal proof for $s(\mathit{repl}(\chi[\delta/x]) \longrightarrow \delta)$ in $\mathsf{K}^\mu + \Phi$. This finishes the proof of (3.2).

Now we also have that

$$\begin{aligned} s(\mathit{repl}(\chi[\delta/x]) \longrightarrow \delta) &= s(\mathit{repl}(\chi[\delta/x])) \longrightarrow s(\delta), \\ &= s(\chi[\delta/x]) \longrightarrow s(\delta), \qquad \text{(for all } \theta,\ s(\mathit{repl}(\theta)) = s(\theta)) \\ &= s(\chi)[s(\delta)/x] \longrightarrow s(\delta). \end{aligned}$$

Putting this together with (3.2), we obtain that $s(\chi)[s(\delta)/x] \longrightarrow s(\delta)$ is provable in $\mathsf{K}^\mu + \Phi$.

It follows from the Fixpoint rule that the formula $\mu x.s(\chi) \longrightarrow s(\delta)$ is provable in $\mathsf{K}^\mu + \Phi$. That is, $\mathit{repl}(\mu x.s(\chi) \longrightarrow s(\delta))$ belongs to $\mathit{repl}(\mathsf{K}^\mu + \Phi)$. Moreover,

$$\begin{aligned} \mathit{repl}(\mu x.s(\chi) \longrightarrow s(\delta)) &= \mathit{repl}(\mu x.s(\chi)) \longrightarrow \mathit{repl}(s(\delta)), \\ &= \mathit{repl}(\mu x.\chi) \longrightarrow \mathit{repl}(\delta), \\ &= \mathit{repl}(\mu x.\chi) \longrightarrow \delta. \qquad (\delta \text{ is a modal formula}) \end{aligned}$$

Putting everything together, we obtain that $\mathit{repl}(\mu x.\chi) \longrightarrow \delta$ is provable in $\mathsf{K}^\mu + \Phi$.

Since $\mathcal{N} \Vdash \mathit{repl}(\mathsf{K}^\mu + \Phi)$ and $\mathit{repl}(\mu x.\chi) \longrightarrow \delta$ belongs to $\mathit{repl}(\mathsf{K}^\mu + \Phi)$, we have that $\mathcal{N} \Vdash_\tau \mathit{repl}(\mu x.\chi) \longrightarrow \delta$. Using the fact that $\mathcal{N}, w \Vdash_\tau \mathit{repl}(\mu x.\chi)$, we obtain that $\mathcal{N}, w \Vdash_\tau \delta$. That is, w belongs to $[\![\delta]\!]$ and this finishes the proof of the implication from left to right of equivalence (3.1).

For the direction from right to left of equivalence (3.1), suppose that $\mathcal{M}, w \Vdash_\tau \mu x.\chi$. We have to show that $\mathcal{N}, w \Vdash_\tau \mathit{repl}(\mu x.\chi)$. Since $\mathcal{M}, w \Vdash_\tau \mu x.\chi$, we have that for all admissible sets U in $\mathbb{A}$ such that $[\![\chi]\!]_{\mathcal{M},\tau[x \mapsto U]} \subseteq U$, w belongs to U. Therefore, if we let U_0 be the admissible set $[\![\mathit{repl}(\mu x.\chi)]\!]$ and show that $[\![\chi]\!]_{\mathcal{M},\tau[x \mapsto U_0]} \subseteq U_0$, we will obtain that w belongs to U_0. That is, $\mathcal{N}, w \Vdash_\tau \mathit{repl}(\mu x.\chi)$.

So in order to prove the direction from right to left of equivalence (3.1), it is sufficient to show that $[\![\chi]\!]_{\mathcal{M},\tau[x \mapsto U_0]} \subseteq U_0$. Suppose that v belongs to $[\![\chi]\!]_{\mathcal{M},\tau[x \mapsto U_0]}$. This means that $\mathcal{M}, v \Vdash_{\tau[x \mapsto U_0]} \chi$. That is,

$$\mathcal{M}, v \Vdash_\tau \chi[\mathit{repl}(\mu x.\chi)/x].$$

By (the first) induction hypothesis, this happens iff $\mathcal{N}, v \Vdash_\tau \mathit{repl}(\chi[\mathit{repl}(\mu x.\chi)/x])$. As the sources of the formulas $\chi[\mathit{repl}(\mu x.\chi)/x]$ and $\chi[\mu x.\chi/x]$ are the same, we have

$$\mathit{repl}(\chi[\mathit{repl}(\mu x.\chi)/x]) = \mathit{repl}(\chi[\mu x.\chi/x]).$$

Since $\chi[\mu x.\chi/x] \to \mu x.\chi$ is provable in K^μ, $repl(\chi[\mu x.\chi/x]) \to repl(\mu x.\chi)$ belongs to $repl(\mathsf{K}^\mu + \Phi)$. Recall also that $\mathcal{N} \Vdash repl(\mathsf{K}^\mu + \Phi)$. Therefore,

$$\mathcal{N}, v \Vdash repl(\chi[\mu x.\chi/x]) \to repl(\mu x.\chi).$$

Now since $\mathcal{N}, v \Vdash repl(\chi[\mu x.\chi/x])$, we also have that $\mathcal{N}, v \Vdash repl(\mu x.\chi)$. In other words, v belongs to U_0. This finishes the proof of equivalence (3.1).

Next, we prove that for all μ-sentences φ (over *Prop*), we have

$$\mathcal{M} \Vdash \varphi \quad \text{iff} \quad \varphi \text{ is provable in } \mathsf{K}^\mu + \Phi.$$

For the direction from left to right, suppose that φ is not provable in $\mathsf{K}^\mu + \Phi$. Using a proof similar to the one for (3.2), we can show that this implies that $repl(\varphi)$ is not provable in $\mathsf{K} +_r repl(\mathsf{K}^\mu + \Phi)$. Therefore, the formula $repl(\varphi)$ is not true in $\mathcal{N}$. By equivalence (3.1), this means that φ is not true in $\mathcal{M}$.

For the direction from right to left, assume that φ is provable in $\mathsf{K}^\mu + \Phi$. It is routine to show that for all generalized models $\mathcal{M}'$ such that $\mathcal{M}' \Vdash \Phi$, we have that $\mathcal{M}' \Vdash \varphi$. Moreover, using equivalence (3.1) together with the fact that $repl(\Phi)$ is true in $\mathcal{N}$, we obtain that Φ is true in $\mathcal{M}$. Putting everything together, we get that φ is true in $\mathcal{M}$.

To finish the proof, it remains to show that $\mathcal{M}$ is a generalized model. That is, for all μ-formulas φ over *Prop*, the set $[\![\varphi]\!]_\mathcal{M}$ belongs to $\mathbb{A}$. Fix a μ-formula φ over *Prop*. By equivalence (3.1), the set $[\![\varphi]\!]_\mathcal{M}$ is equal to $[\![repl(\varphi)]\!]_\mathcal{N}$. By definition of $\mathbb{A}$, this set belongs to $\mathbb{A}$. □

3.3.3. Corollary. *The logic $\mathsf{K}^\mu + \Phi$ is complete with respect to the class of generalized models in which Φ is true. That is, for all sentences φ, φ is provable in $\mathsf{K}^\mu + \Phi$ iff for all generalized models $\mathcal{M}$ such that $\mathcal{M} \Vdash \Phi$, we have $\mathcal{M} \Vdash \varphi$.*

3.4 Completeness for finite trees

In the style of Kees Doets [Doe89], we prove completeness of $\mathsf{K}^\mu + \mu x.\Box x$ with respect to the class of finite trees. The argument is as follows. First, we say that a point w in a generalized model is n-good if there is a point t in a finite tree model $\mathcal{T}$ such that no formula of rank at most n can distinguish w from t. Let us emphasize that $\mathcal{T}$ is a Kripke model, not a generalized model.

Next, we show that "being n-good" is a property that can be expressed by a formula γ_n of rank at most n. Afterward, we prove that each point in a generalized model satisfying $\mu x.\Box x$, is n-good. Finally, using completeness for generalized models, we obtain completeness of $\mathsf{K}^\mu + \mu x.\Box x$ with respect to the class of finite trees.

n-goodness Fix a natural number n. Let $\mathcal{M}$ and $\mathcal{M}'$ be two generalized models. A world $w \in \mathcal{M}$ is *rank n-indistinguishable* from a world $w' \in \mathcal{M}'$ if for all formulas φ of rank at most n, we have

$$\mathcal{M}, w \Vdash \varphi \quad \text{iff} \quad \mathcal{M}', w' \Vdash \varphi.$$

In case this happens, we write $(\mathcal{M}, w) \sim_n (\mathcal{M}', w')$. Finally, we say that $w \in \mathcal{M}$ is *n-good* if there exists a finite tree model $\mathcal{T}$ and some $t \in \mathcal{T}$ such that $(\mathcal{M}, w) \sim_n (\mathcal{T}, t)$.

Let n be a natural number and let Φ_n be the set of formulas of rank at most n. For any generalized model $\mathcal{M}$ and any $w \in \mathcal{M}$, we define the *n-type* $\theta_n(w)$ as the set of formulas in Φ_n that are true at w.

By Proposition 3.2.1, Φ_n is finite (up to logical equivalence) and in particular, there are only finitely many distinct n-types.

3.4.1. Lemma. *Let n be a natural number. There exists a formula γ_n of rank n such that for any generalized model $\mathcal{M}$ and any $w \in \mathcal{M}$, we have*

$$\mathcal{M}, w \Vdash \gamma_n \quad \textit{iff} \quad (\mathcal{M}, w) \textit{ is } n\textit{-good.}$$

Proof Let n be a natural number and let γ_n be the formula defined by

$$\gamma_n = \bigvee\{\bigwedge \theta_n(w) \mid w \text{ is } n\text{-good}\}.$$

Since there are only finitely many distinct n-types, the formula γ_n is well-defined. Moreover, from Proposition 3.2.2, it follows that the rank of γ_n is n.

It remains to check that γ_n has the required properties. It is immediate to see that if a point w in a generalized model is n-good, then γ_n is true at w. For the other direction, assume that γ_n is true at a point w in a generalized model $\mathcal{M}$. Therefore, there is a point w' in a generalized model $\mathcal{M}'$ such that w' is n-good and $\theta_n(w')$ is true at w. Since w' is n-good, there is a point t in a finite tree $\mathcal{T}$ such that w' and t are rank n-indistinguishable. Using the fact that w and w' have the same n-type, we obtain that w and t are also rank n-indistinguishable. That is, w is n-good. □

3.4.2. Lemma. *For all natural numbers n, $\vdash_{\mathsf{K}^\mu} \Box\gamma_n \rightarrow \gamma_n$.*

Proof Let n be a natural number. By Corollary 3.3.3, it is sufficient to show that the formula $\Box\gamma_n \rightarrow \gamma_n$ is valid in all generalized models. Let $\mathcal{M}$ be a generalized model and let w be a point in $\mathcal{M}$. We have to show $\mathcal{M}, w \Vdash \Box\gamma_n \rightarrow \gamma_n$. So suppose $\mathcal{M}, w \Vdash \Box\gamma_n$. If w is a reflexive point, we immediately obtain $\mathcal{M}, w \Vdash \gamma_n$ and this finishes the proof. Assume now that w is irreflexive. We have to prove that $(\mathcal{M}, w)$ is n-good. That is, we have to find a finite tree model $\mathcal{T}$ and some $t \in \mathcal{T}$ such that $(\mathcal{M}, w) \sim_n (\mathcal{T}, t)$.

Now for any successor v of w, we have $\mathcal{M}, v \Vdash \gamma_n$. Therefore, $(\mathcal{M}, v)$ is n-good and there exists a finite tree model $\mathcal{T}_v = (T_v, R_v, V_v)$ and some $t_v \in T_v$ such that $(\mathcal{M}, v) \sim_n (\mathcal{T}_v, t_v)$. Without loss of generality, we may assume that t_v is the root of $\mathcal{T}_v$.

The idea is now to look at the disjoint union of these models and to add a root t (that would be rank n-indistinguishable from w). However, this new model might not be a finite tree model (t might have infinitely many successors). The solution is to restrict ourselves to finitely many successors of w. More precisely, for each n-type θ, we pick at most one successor of w the n-type of which is θ.

So let W_0 be a set of successors of w such that for all successors v of w, there is exactly one point w_0 of W_0 satisfying $\theta_n(v) = \theta_n(w_0)$. Remark that since there are only finitely many distinct n-types, W_0 is finite. Let $\mathcal{T} = (T, S, U)$ be the model defined by

$$\begin{aligned} T &= \{t\} \cup \biguplus \{T_{w_0} \mid w_0 \in W_0\}, \\ S &= \{(t, t_{w_0}) \mid w_0 \in W_0\} \cup \bigcup \{R_{w_0} \mid w_0 \in W_0\}, \\ U(p) &= \begin{cases} \{t\} \cup \bigcup \{V_{w_0}(p) \mid w_0 \in W_0\} & \text{if } \mathcal{M}, w \Vdash p, \\ \bigcup \{V_{w_0}(p) \mid w_0 \in W_0\} & \text{otherwise,} \end{cases} \end{aligned}$$

for all proposition letters p. Since W_0 is finite, $\mathcal{T}$ is a finite tree model. Thus, it is sufficient to check that for all formulas φ of rank at most n, we have

$$\mathcal{M}, w \Vdash \varphi \quad \text{iff} \quad \mathcal{T}, t \Vdash \varphi.$$

By Proposition 3.2.3, φ is provably equivalent to a boolean combination of proposition letters and formulas of the form $\Diamond\psi$ and $\Box\psi$, where $rank(\psi)$ is at most n. Thus, it is sufficient to show that w and t satisfy exactly the same proposition letters and the same formulas $\Diamond\psi$ and $\Box\psi$, with $rank(\psi) \leq n$.

By definition of U, it is immediate that w and t satisfy the same proposition letters. Now let ψ be a formula of rank at most n. We show that

$$\mathcal{M}, w \Vdash \Diamond\psi \quad \text{iff} \quad \mathcal{T}, t \Vdash \Diamond\psi.$$

The proof is similar for the formula $\Box\psi$. For the direction from left to right, suppose that $\mathcal{M}, w \Vdash \Diamond\psi$. Thus, there exists a successor v of w such that $\mathcal{M}, v \Vdash \psi$. By definition of W_0, there is $w_0 \in W_0$ such that $(\mathcal{M}, v) \sim_n (\mathcal{M}, w_0)$. Thus, $(\mathcal{M}, v) \sim_n (\mathcal{T}_{w_0}, t_{w_0})$ and in particular, $\mathcal{T}_{w_0}, t_{w_0} \Vdash \psi$. By definition of S, it follows that $\mathcal{T}, t \Vdash \Diamond\psi$. The direction from right to left is similar. □

3.4.3. Proposition. *For all natural numbers n, $\vdash_{\mathsf{K}^\mu} \mu x.\Box x \longrightarrow \gamma_n$.*

Proof By Lemma 3.4.2, we know that $\Box\gamma_n \longrightarrow \gamma_n$ is provable in K^μ. By the Fixpoint rule, we obtain that $\mu x.\Box x \longrightarrow \gamma_n$ is provable in K^μ. □

3.4.4. THEOREM. $\mathsf{K}^\mu + \mu x.\Box x$ *is complete with respect to the class of finite trees.*

Proof For all finite tree models $\mathcal{T}$, we have $\mathcal{T} \Vdash \mathsf{K}^\mu$ and $\mathcal{T} \Vdash \mu x.\Box x$. Thus, it is sufficient to show that if φ is not provable in $\mathsf{K}^\mu + \mu x.\Box x$, there exists a finite tree model $\mathcal{T}$ such that $\mathcal{T} \nVdash \varphi$. Let φ be such a formula. In particular, $\nvdash_{\mathsf{K}^\mu} \mu x.\Box x \to \varphi$. By Corollary 3.3.3, we have $\mathcal{M}, w \nVdash \mu x.\Box x \to \varphi$, for some generalized model $\mathcal{M}$ and some $w \in \mathcal{M}$.

Let n be the rank of φ. By Corollary 3.3.3 and Proposition 3.4.3, we get that $\mathcal{M}, w \Vdash \mu x.\Box x \to \gamma_n$. Since $\mathcal{M}, w \Vdash \mu x.\Box x$, it follows that $\mathcal{M}, w \Vdash \gamma_n$. Therefore, there exists a finite tree model $\mathcal{T}$ and some $t \in \mathcal{T}$ such that $(\mathcal{M}, w) \sim_n (\mathcal{T}, t)$. Since $\mathcal{M}, w \nVdash \varphi$, we have $\mathcal{T}, t \nVdash \varphi$. $\Box$

3.4.5. REMARK. As mentioned before, this result also follows from the completeness of K^μ showed by Igor Walukiewicz in [Wal95]. We briefly explain how to derive Theorem 3.4.4 from the completeness of K^μ. Recall that in [Wal95], Igor Walukiewicz showed that a sentence φ is provable in K^μ iff it is true in all tree models.

Suppose that a sentence φ is not provable in $\mathsf{K}^\mu + \mu x.\Box x$. In particular, the formula $\mu x.\Box x \to \varphi$ is not provable in K^μ. It follows from the completeness of K^μ that there is a tree model $\mathcal{T} = (T, R, V)$ and a point t in T such that $\mu x.\Box x \to \varphi$ is not true at t. We may assume that t is the root of $\mathcal{T}$.

Since $\mu x.\Box x$ is true at t and since t is the root, the model $\mathcal{T}$ does not contain any infinite path. Let n be the rank of φ. Now, if a point u in T has more than one successor of a given n-type θ, we can pick one successor of n-type θ and delete all the other successors of n-type θ. This would not modify the fact that φ is not true at t. By repeating this operation in an appropriate way, we can prove that the tree $\mathcal{T}$ may be assumed to be finite. Therefore, there is a finite tree $\mathcal{T}$ in which φ is not true.

3.5 Adding shallow axioms to $\mathsf{K}^\mu + \mu x.\Box x$

By slightly modifying our method, it is also possible to prove that when we extend the logic $\mathsf{K}^\mu + \mu x.\Box x$ with axioms that are shallow, as defined in [Cat05], we obtain a complete axiomatization for the corresponding class of finite trees.

Shallow formulas A formula is *Prop-free* if it is a sentence that does not contain any proposition letter. A formula is *propositional* if it is a sentence of the μ-calculus that contains neither $\Diamond$ nor μ.

A formula is *shallow* if no occurrence of a proposition letter is in the scope of a fixpoint operator and each occurrence of a proposition letter is in the scope of at most one modality. In other words, the shallow formulas is the language defined by

$$\varphi \ ::= \ \psi \mid \Diamond\psi \mid \Box\psi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi,$$

where ψ is either a *Prop*-free formula or a propositional formula.

For example, $\Diamond p \rightarrow \Box p$ is a shallow formula. Other examples are formulas expressing that each point has at most two successors ($\Diamond p \wedge \Diamond q \rightarrow \Box(p \vee q)$), or that each point has at most one blind successor ($\Diamond(p \wedge \Box\bot) \wedge \Box(\Box\bot \rightarrow p)$).

The remaining of the section is devoted to the proof of the following completeness result.

3.5.1. THEOREM. *Let* φ *be a shallow formula. Then the logic* $\mathsf{K}^\mu + \mu x.\Box x + \varphi$ *is complete with respect to the class of finite trees in which* φ *is valid.*

In order to prove this result, as for the logic $\mathsf{K}^\mu + \mu x.\Box x$, we first show that the logic is complete with respect to a class of generalized frames. An important tool for this proof is the fact that the shallow formulas are persistent with respect to refined frames, which was proved by Balder ten Cate in [Cat05]. We start by recalling the definitions of persistency and refinedness.

Refined frames and persistent formulas A generalized frame $\mathbb{F} = (W, R, \mathbb{A})$ is *differentiated* if for all $w, v \in W$ with $w \neq v$, there exists $A \in \mathbb{A}$ such that $w \in A$ and $v \notin A$. A generalized model $\mathbb{F} = (W, R, \mathbb{A})$ is *tight* if for all $w, v \in W$ such that $(w, v) \notin R$, there exists $A \in \mathbb{A}$ such that $v \in A$ and for all $u \in A$, $(w, u) \notin R$. A generalized frame is *refined* if it is differentiated and tight.

A formula φ is *persistent* with respect to refined frames if for all refined frame $\mathbb{F}$ such that $\mathbb{F} \Vdash \varphi$, the formula φ is valid on the underlying Kripke frame of $\mathbb{F}$.

3.5.2. THEOREM ([CAT05]). *Every shallow formula is persistent with respect to refined frames.*

3.5.3. THEOREM. *Let* φ *be a shallow formula. The logic* $\mathsf{K}^\mu + \varphi$ *is complete with respect to the class of generalized frames* $\mathbb{F}$ *such that* φ *is valid in the underlying Kripke frame of* $\mathbb{F}$.

Proof By Theorem 3.3.2, we know that the logic $\mathsf{K}^\mu + \varphi$ is complete with respect to a generalized model $\mathcal{M} = (W, R, V^+, \mathbb{A})$. It follows from the proofs of this theorem and Theorem 3.3.1 that we may assume that W is the set of Φ-consistent maximal sets of modal formulas over $Prop^+$, where $\Phi = \{repl(\varphi)\}$. Moreover, R is such that for all $w, w' \in W$, wRw' iff for all $\psi \in w'$, $\Diamond\psi$ belongs to w. The valuation $V : Prop^+ \rightarrow \mathcal{P}(W)$ is such that for all $p \in Prop^+$ and all $w \in W$, $w \in V(p)$ iff $p \in w$. Finally, $\mathbb{A}$ is the set $\{[\![\psi]\!]_{\mathcal{N}} \mid \psi$ is a modal formula over $Prop^+\}$, where $\mathcal{N} = (W, R, V^+)$.

Since $\mathsf{K}^\mu + \varphi$ is complete with respect to $\mathcal{M}$, it is sufficient to show that φ is valid in the underlying Kripke frame (W, R). It is easy to check that $(W, R, \mathbb{A})$ is refined. Hence, by Theorem 3.5.2, we can restrict ourselves to show that φ is valid in the generalized Kripke frame $(W, R, \mathbb{A})$.

Let $V' : \mathit{Prop} \longrightarrow \mathbb{A}$ be a valuation and let $\mathcal{M}'$ be the generalized model $(W, R, V', \mathbb{A})$. We have to show that $\mathcal{M}' \Vdash \varphi$. It follows from the definition of $\mathbb{A}$ that for all proposition letters p, there is a modal formula φ_p over Prop^+ such that $V'(p) = [\![\varphi_p]\!]_{\mathcal{N}}$. Recall that in the proof of Theorem 3.3.2, we showed that for all μ-formulas ψ over Prop^+ and for all $w \in W$,

$$\mathcal{N}, w \Vdash_\tau \mathit{repl}(\psi) \quad \text{iff} \quad \mathcal{M}, w \Vdash_\tau \psi.$$

In particular, for all $w \in W$, $\mathcal{N}, w \Vdash_\tau \mathit{repl}(s(\varphi_p))$ iff $\mathcal{M}, w \Vdash_\tau s(\varphi_p)$. That is $[\![\mathit{repl}(s(\varphi_p))]\!]_{\mathcal{N}} = [\![s(\varphi_p)]\!]_{\mathcal{M}}$.

Recall also that for all μ-formulas ψ over Prop^+, we have $\mathit{repl}(s(\psi)) = \mathit{repl}(\psi)$ and in case ψ is a modal formula, $\mathit{repl}(\psi) = \psi$. It follows that $\mathit{repl}(s(\varphi_p))$ is equal to φ_p. Putting this together with the fact that $[\![\mathit{repl}(s(\varphi_p))]\!]_{\mathcal{N}} = [\![s(\varphi_p)]\!]_{\mathcal{M}}$, we obtain that $[\![\varphi_p]\!]_{\mathcal{N}} = [\![s(\varphi_p)]\!]_{\mathcal{M}}$.

Now we show that for all μ-formulas ψ over Prop, for all assignments $\tau : \mathit{Var} \longrightarrow \mathcal{P}(W)$ and for all $w \in W$,

$$\mathcal{M}', w \Vdash_\tau \psi \quad \text{iff} \quad \mathcal{M}, w \Vdash_\tau \psi[p/s(\varphi_p)], \tag{3.3}$$

where $\psi[p/s(\varphi_p)]$ is a formula obtained by simultaneously replacing each proposition letter p in ψ by the formula $s(\varphi_p)$. The basic step follows from the facts that $[\![\varphi_p]\!]_{\mathcal{N}} = [\![s(\varphi_p)]\!]_{\mathcal{M}}$ and $V'(p) = [\![\varphi_p]\!]_{\mathcal{N}}$. The induction steps are straightforward.

It follows from (3.3) that $\mathcal{M}' \Vdash \varphi$ iff $\mathcal{M} \Vdash \varphi[p/s(\varphi_p)]$. Since the logic $\mathsf{K}^\mu + \varphi$ is closed under substitution, $\varphi[p/s(\varphi_p)]$ belongs to $\mathsf{K}^\mu + \varphi$. Putting that together with the fact that $\mathsf{K}^\mu + \varphi$ is complete with respect to $\mathcal{M}$, we obtain that $\mathcal{M} \Vdash \varphi[p/s(\varphi_p)]$. Hence, $\mathcal{M}' \Vdash \varphi$. This finishes the proof that φ is valid in the generalized Kripke frame $(W, R, \mathbb{A})$. □

n-goodness Let φ be a formula and let $\mathcal{M}$ be a generalized model. A point $w \in \mathcal{M}$ is *n-good for φ* if there exist a finite tree model $\mathcal{T}$ such that $\mathcal{T} \Vdash \varphi$ and $(\mathcal{M}, w) \sim_n (\mathcal{T}, t)$, for some $t \in \mathcal{T}$.

The formula δ_n For all $n \in \mathbb{N}$, we define the formula δ_n as the formula $\gamma_n \wedge \mu x.(\varphi \wedge \Box x)$, where γ_n is the formula given by Lemma 3.5.4.

With such a definition of δ_n, the next lemma is immediate.

3.5.4. LEMMA. *Let φ be a formula and let n be a natural number strictly greater than the rank of φ. For all generalized models $\mathcal{M}$ and all $w \in \mathcal{M}$, we have*

$$\mathcal{M}, w \Vdash \gamma_n \quad \textit{iff} \quad (\mathcal{M}, w) \textit{ is } n\textit{-good for } \varphi.$$

The proof of the next lemma is an adaptation of the proof of Lemma 3.4.2.

3.5.5. LEMMA. *Let φ be a shallow formula and let n be a natural number strictly greater than the rank of φ. We have $\vdash_{\mathsf{K}^\mu + \varphi} \Box\delta_n \longrightarrow \delta_n$.*

Proof Let φ be a shallow formula and let n be a natural number strictly greater than the rank of φ. By Theorem 3.5.3, it is enough to show that for all generalized frames $\mathbb{F} = (W, R, \mathbb{A})$ such that $(W, R) \Vdash \varphi$, we have $\mathbb{F} \Vdash \Box\delta_n \to \delta_n$. Let $\mathbb{F} = (W, R, \mathbb{A})$ be a generalized frame such that $(W, R) \Vdash \varphi$. We also fix a valuation $V : \mathit{Prop} \to \mathbb{A}$ and a point $w \in W$. We prove that $\mathcal{M}, w \Vdash \Box\delta_n \to \delta_n$, where $\mathcal{M} = (W, R, V, \mathbb{A})$. So suppose that $\mathcal{M}, w \Vdash \Box\delta_n$.

If $(w, w) \in R$, it is immediate that $\mathcal{M}, w \Vdash \delta_n$. Next assume that $(w, w) \notin R$. We have to show that $\mathcal{M}, w \Vdash \delta_n$. That is, w is n-good for φ.

For all successors v of w, we have $\mathcal{M}, v \Vdash \delta_n$. Therefore, $(\mathcal{M}, v)$ is n-good for φ and there exists a finite tree model $\mathcal{T}_v = (T_v, R_v, V_v)$ and some $t_v \in T_v$ such that $(T_v, R_v) \Vdash \varphi$ and $(\mathcal{M}, v) \sim_n (\mathcal{T}_v, t_v)$. Without loss of generality, we may assume that t_v is the root of $\mathcal{T}_v$.

Now we define the set W_0 and the finite tree model $\mathcal{T} = (T, S, U)$ exactly as in the proof of Lemma 3.4.2. Recall that W_0 is a set of successors of w such that for all successors v of w, there is exactly one point w_0 of W_0 satisfying $\theta_n(v) = \theta_n(w_0)$. The model $\mathcal{T} = (T, S, U)$ is defined by

$$\begin{aligned} T &= \{t\} \cup \biguplus \{T_{w_0} \mid w_0 \in W_0\}, \\ S &= \{(t, t_{w_0}) \mid w_0 \in W_0\} \cup \bigcup \{R_{w_0} \mid w_0 \in W_0\}, \\ U(p) &= \begin{cases} \{t\} \cup \bigcup \{V_{w_0}(p) \mid w_0 \in W_0\} & \text{if } \mathcal{M}, w \Vdash p, \\ \bigcup \{V_{w_0}(p) \mid w_0 \in W_0\} & \text{otherwise,} \end{cases} \end{aligned}$$

for all proposition letters p. Since W_0 is finite, $\mathbb{T} = (T, S)$ is a finite tree. Now it remains to check that $\mathbb{T} \Vdash \varphi$ and $(\mathcal{M}, w) \sim_n (\mathcal{T}, t)$. The proof that $(\mathcal{M}, w) \sim_n (\mathcal{T}, t)$ is exactly the same as in the proof of Lemma 3.4.2.

In order to show that $\mathbb{T} \Vdash \varphi$, fix a point $u \in T$. We prove that $\mathbb{T}, u \Vdash \varphi$. First, if $u \neq t$, then there exists a successor w_0 of w such that $w_0 \in W_0$ and $u \in T_{w_0}$. It follows from the construction of $\mathbb{T}$ that for all formulas ψ, $\mathbb{T}, u \Vdash \psi$ iff $(T_{w_0}, R_{w_0}), u \Vdash \psi$. Putting this together with the fact that $(T_{w_0}, R_{w_0}) \Vdash \varphi$, we obtain that $\mathbb{T}, u \Vdash \varphi$.

Next suppose that $u = t$ and let $\mathcal{T}' = (T, S, U')$ be a model based on $\mathbb{T}$. We show that $\mathcal{T}', t \Vdash \varphi$. We let $\mathcal{M}'$ be the model (W, R, V'), where $V' : \mathit{Prop} \to \mathcal{P}(W)$ is a valuation such that for all proposition letters p, the two following equivalences hold. The point w belongs to $V'(p)$ iff t belongs to $U'(p)$. For all successors v of w, there exists $w_0 \in W_0$ such that $\theta_n(v) = \theta_n(w_0)$ and v belongs to $V'(p)$ iff t_{w_0} belongs to $U'(p)$. We prove the following claim.

1. CLAIM. $\mathcal{M}', w \Vdash \varphi$ iff $\mathcal{T}', t \Vdash \varphi$.

It will immediately follow from this claim that $\mathcal{T}', t \Vdash \varphi$. Indeed, since $(W, R) \Vdash \varphi$, we have $\mathcal{M}', w \Vdash \varphi$. So we can conclude using the claim.

Proof of Claim The proof is by induction on the complexity of φ. If φ is a propositional formula, the claim follows immediately from the definition of V'. The induction cases where φ is a disjunction or a conjunction are straightforward.

Next assume that φ is a *Prop*-free formula. Since φ is *Prop*-free, $\mathcal{M}', w \Vdash \varphi$ iff $\mathcal{M}, w \Vdash \varphi$. Moreover, as $(\mathcal{M}, w) \sim_n (\mathcal{T}, t)$, we also have $\mathcal{M}, w \Vdash \varphi$ iff $\mathcal{T}, t \Vdash \varphi$. Using again the fact that φ is *Prop*-free, we get $\mathcal{T}, t \Vdash \varphi$ iff $\mathcal{T}', t \Vdash \varphi$. Putting everything together, we obtain $\mathcal{M}', w \Vdash \varphi$ iff $\mathcal{T}', t \Vdash \varphi$.

Now suppose that φ is a formula of the form $\Diamond\psi$, where ψ is either a propositional formula or a *Prop*-free formula. We have to show $\mathcal{M}', w \Vdash \Diamond\psi$ iff $\mathcal{T}', t \Vdash \Diamond\psi$. We only show the implication from left to right, as the proof for the other implication is similar. So suppose that $\mathcal{M}', w \Vdash \Diamond\psi$. Thus, there exists a successor w_0 of w such that $\mathcal{M}', w_0 \Vdash \psi$. Without loss of generality, we may assume that w_0 belongs to W_0. If ψ is a propositional formula, it follows from the definition of V' that $\mathcal{T}', t_{w_0} \Vdash \psi$ and therefore, $\mathcal{T}', t \Vdash \Diamond\psi$.

Finally, assume that ψ is a *Prop*-free formula. Since ψ does not contain any proposition letter, $\mathcal{M}', w_0 \Vdash \psi$ iff $\mathcal{M}, w_0 \Vdash \psi$. Hence, $\mathcal{M}, w_0 \Vdash \psi$. Putting this with the fact that $(\mathcal{M}, w_0) \sim_n (\mathcal{T}, t_{w_0})$, we have $\mathcal{T}, t_{w_0} \Vdash \psi$. Using again the fact that ψ is *Prop*-free, it follows that $\mathcal{T}, t_{w_0} \Vdash \psi$ iff $\mathcal{T}', t_{w_0} \Vdash \psi$. We can conclude that $\mathcal{T}', t_{w_0} \Vdash \psi$.

The case where φ of the form $\Diamond\psi$, where ψ is a *Prop*-free formula, is similar to the previous case. ◀

□

3.5.6. Proposition. *Let φ be a shallow formula and let n be a natural number strictly greater than the rank of φ. Then $\vdash_{\mathsf{K}^\mu} \mu x.\Box x \longrightarrow \gamma_n$.*

Proof By Lemma 3.5.5, we know that $\Box\delta_n \longrightarrow \delta_n$ is provable in $\mathsf{K}^\mu + \varphi$. By the Fixpoint rule, we obtain that $\mu x.\Box x \longrightarrow \delta_n$ is provable in $\mathsf{K}^\mu + \varphi$. □

3.5.7. Theorem. *Let φ be a shallow formula. The logic $\mathsf{K}^\mu + \mu x.\Box x + \varphi$ is complete with respect to the class of finite trees in which φ is valid.*

Proof It is easy to see that every formula of the logic $\mathsf{K}^\mu + \mu x.\Box x + \varphi$ is valid in a finite tree $\mathbb{T}$ satisfying $\mathbb{T} \Vdash \varphi$. Thus, it is sufficient to show that if ψ is not provable in $\mathsf{K}^\mu + \mu x.\Box x + \varphi$, there exists a finite tree $\mathbb{T}$ such that $\mathbb{T} \Vdash \varphi$ and $\mathbb{T} \nVdash \psi$. Let ψ be such a formula. In particular, $\nvdash_{\mathsf{K}^\mu + \varphi} \mu x.\Box x \longrightarrow \psi$. By Theorem 3.5.3, there exist a generalized frame $\mathbb{F}$ and a generalized model $\mathcal{M}$ based on $\mathbb{F}$ such that $\mathbb{F} \Vdash \varphi$ and $\mathcal{M}, w \nVdash \mu x.\Box x \longrightarrow \varphi$, for some $w \in \mathbb{F}$.

Let n be a natural number strictly greater than the rank of φ and greater or equal to the rank of ψ. By Theorem 3.5.3 and Proposition 3.5.6, we get that $\mathcal{M}, w \Vdash \mu x.\Box x \longrightarrow \delta_n$. Since $\mathcal{M}, w \Vdash \mu x.\Box x$, it follows that $\mathcal{M}, w \Vdash \delta_n$. Therefore, there exists a finite tree $\mathbb{T}$, a Kripke model $\mathcal{T}$ based on $\mathbb{T}$ and $t \in \mathbb{T}$ such that $\mathbb{T} \Vdash \varphi$ and $(\mathcal{M}, w) \sim_n (\mathcal{M}, t)$. Since $\mathcal{M}, w \nVdash \psi$, we have $\mathcal{T}, t \nVdash \psi$. Hence, $\mathbb{T}$ is a finite tree such that $\mathbb{T} \Vdash \varphi$ and $\mathbb{T} \nVdash \psi$. □

3.6 Graded μ-calculus

By adapting the definition of rank to the setting of the graded μ-formulas, we can use the same proof to show that the graded μ-calculus together with the axiom $\mu x.\Box x$ is complete with respect to the class of finite trees.

Axiomatization for the graded μ-calculus The axiomatization of the system GK^μ consists of the following axioms and rules:

propositional tautologies,	
If $\vdash \varphi \to \psi$ and $\vdash \varphi$, then $\vdash \psi$	(Modus ponens),
If $\vdash \varphi$, then $\vdash \varphi[p/\psi]$	(Substitution),
If $\vdash \varphi$, then $\vdash \Box^0\varphi$	(Necessitation),
$\vdash \varphi[x/\mu x.\varphi] \to \mu x.\varphi$	(Fixpoint rule),
$\Diamond^{k+1}p \to \Diamond^k p$	(axiom $G1$),
$\Box^0(p \to q) \to (\Diamond^n p \to \Diamond^n q)$	(axiom $G2$),
$\Diamond^{!0}(p \wedge q) \to ((\Diamond^{!k}p \wedge \Diamond^{!l}q) \to \Diamond^{!k+l}(p \vee q))$	(axiom $G3$),
$\vdash \varphi[x/\mu x.\varphi] \to \mu x.\varphi$	(Fixpoint axiom),
If $\vdash \varphi[x/\psi] \to \psi$, then $\vdash \mu x.\varphi \to \psi$	(Fixpoint rule),

where p and q are proposition letters, φ and ψ are μ-formulas, x is not a bound variable of φ and no free variable of ψ is bound in φ. For all $k > 0$ and for all graded μ-formula, $\Diamond^{!k}$ is an abbreviation for $\Diamond^k\varphi \wedge \neg\Diamond^{k-1}\varphi$.

The logic GK is the smallest set of formulas which contains the propositional tautologies, the axioms $G1$, $G2$ and $G3$ and is closed under the Substitution, the Modus ponens and the Necessitation rules.

3.6.1. Theorem ([FBC85]). *The logic* GK *is complete with respect to a single model. That is, there is a Kripke model* $\mathcal{M}$ *such that a graded modal formula is provable in* GK *iff it is true in* $\mathcal{M}$.

3.6.2. Theorem. *The logic* $\mathsf{GK}^\mu + \mu x.\Box^0 x$ *is complete with respect to the class of finite trees. That is, a graded μ-formula is provable in* $\mathsf{GK}^\mu + \mu x.\Box^0 x$ *iff it is valid in all finite tree frames.*

Proof The structure of the proof is the same as the one for the proof of Theorem 3.4.4. So first, we need to define a notion of rank for graded μ-formulas. As before, we start by defining the closure and the depth of a formula. The closure of graded formula is defined as in the case of μ-calculus, except that we replace $\Diamond$ by $\Diamond^k$. The depth of a graded μ-formula is defined by induction as follows:

$$\begin{aligned} d(p) = d(\neg p) = d(x) &= 0 \\ d(\varphi \vee \psi) = d(\varphi \wedge \psi) &= max\{d(\varphi), d(\psi)\}, \\ d(\Diamond^k\varphi) = d(\Box^k\varphi) &= d(\varphi) + k + 1, \\ d(\mu x.\varphi) = d(\nu x.\varphi) &= d(\varphi) + 1. \end{aligned}$$

Finally, we can define the rank of a graded μ-formula as in the case of μ-calculus.

The second step is to prove completeness of GK^μ with respect to the class of generalized frames. Using Theorem 3.6.1, it is possible to show this by using a proof that is identical to the proof of Theorem 3.3.2 (except that $\Phi = \emptyset$). We do not give details.

As mentioned earlier, the proof that we gave of Theorem 3.3.2 is a bit tedious, and not the most direct proof. The most direct proof would involve an adaptation of the canonical model construction. Here we see the advantage of using our more difficult proof for Theorem 3.3.2: it can be immediately adapted to the setting of graded μ-calculus. A proof using an adaptation of the canonical model construction would have been hard to extend, as the canonical model construction for graded modal logic is already very involved.

The last step is to show the completeness of $\mathsf{GK}^\mu + \mu x.\Box^0 x$ with respect to the class of finite tree frames. This is done by extending all the notions and results of Section 3.4 to the setting of the graded μ-calculus. It is immediate how to proceed. $\Box$

3.7 Conclusions

We showed that Kozen's axiomatization together with the axiom $\mu x.\Box x$ is complete with respect to the class of finite trees. We also gave two examples of settings to which our proof can be adapted. If we add finitely many shallow axioms Φ to the system $\mathsf{K}^\mu + \mu x.\Box x$, we obtain an axiomatization that is complete with respect to the class of trees in which Φ is valid. Finally, we proved that if we add $\mu x.\Box x$, the fixpoint axiom and the fixpoint rule to the standard axiomatization for graded modal logic, we obtain a complete axiomatization for graded μ-calculus with respect to the class of finite trees.

We believe that this method could be adapted to other cases. In fact, Balder ten Cate and Amélie Gheerbrant used the same method to show that MSO, $\mathsf{FO}(\mathsf{TC}^1)$ (the extension of FO under the reflexive transitive closure of binary definable relations) and $\mathsf{FO}(\mathsf{LFP}^1)$ (the extension of FO with a unary least fixpoint operator) admit complete axiomatizations on finite trees. So this technique might be useful for proving completeness of logics (in the model theory and modal logic areas) with respect to structures which do not contain infinite paths.

It would also be nice if the method could be extended for proving completeness of Kozen's axiomatization with respect to arbitrary structures. A natural first step would be to show completeness of $\mathsf{K}^\mu + \Diamond p \to \Box p$ with respect to linear structures. This has been already established by Rope Kaivola [Kai97], using a proof inspired by the one proposed by Igor Walukiewicz [Wal95] in the general case. The proof of Rope Kaivola is considerably simpler than the proof in the general case.

However, adapting the method that we presented in this chapter to the case of linear orders is already complicated. Remember that the proofs of the main results of this chapter are in three steps. The first step is the definition of rank, which do not need to be adapted. The second step is proving completeness with respect to generalized models and the last step is to go from generalized models to usual structures. Again, we can find inspiration for adapting the last step to the case of linear orders in Kees Doets' paper [Doe89]. However, in order to be able to apply the method presented in that paper, we need the generalized models to satisfy some conditions (which correspond to the fact that "approximately", they are linear orders). We could construct a canonical generalized model for $\mathsf{K}^\mu + \Diamond p \rightarrow \Box p$ and try to massage this model until it satisfies the required conditions. These generalized models are so complex that it is very hard to modify them, while keeping the truth of a fixed formula at a given point. It might be possible but the proof would be technical.

Finally, we would like to mention that Yde Venema and Stéphane Demri raised the question to which class of coalgebras this proof could be adapted. It does not seem immediate how to do this. First, one should find the adequate equivalent of the notion of "tree" in the coalgebraic setting. Moreover, the last step of the proof involves some manipulation of the structures (taking a disjoint union of trees and adding a root to this disjoint union). It is not really clear how one could understand this from a coalgebraic perspective.

Chapter 4

The μ-calculus and frame definability on trees

As we saw in the preliminaries, modal logic and μ-calculus offer different levels of semantics: there are various kind of structures on which formulas can be interpreted. There are basically two orthogonal choices to be made: (i) whether the structures are models or frames and (ii) whether the perspective is local or global. By local vs global perspective, we mean whether or not there is a point that is distinguished in the structure. Recall that a model or a frame with a distinguished point is called pointed.

On the level of models, modal logic is essentially a fragment of first-order logic, as the operators $\Box$ and $\Diamond$ correspond to quantification over the points in the model. Concerning the μ-calculus, the fixpoint operators capture a second-order quantification over the subsets that are fixpoints of a certain map. More formally, given a μ-sentence φ, we can define the standard translation $ST_x(\varphi)$ (see Section 2.6) such that $ST_x(\varphi)$ is a monadic second-order formula with one free variable x and for all pointed models $(\mathcal{M}, w)$, $\mathcal{M}, w \Vdash \varphi$ iff $\mathcal{M} \vDash ST_x(\varphi)(w)$. In case φ is a modal formula, the standard translation is a first-order formula.

If we shift from the context of pointed models to the context of pointed frames, this results in a second-order quantification over all possible valuations. That is, a μ-sentence φ with propositions letter $p_1, \dots, p_n$ corresponds to the second-order formula $\forall P_1 \dots \forall P_n ST_x(\varphi)$. In that case, modal logic is not longer a fragment of first-order logic, but a fragment of MSO, as the μ-calculus. Finally, if we consider global truth, we add a first-order quantification over all points of the model. A μ-sentence φ with propositions letter $p_1, \dots, p_n$ corresponds on (non-pointed) frames to the formula $\forall x \forall P_1 \dots \forall P_n ST_x(\varphi)$.

The correspondences we mentioned provide various links between modal logics and classical first- and second-order logic. A more in-depth study of these links produced some of the most beautiful results in the areas of modal logic and μ-calculus. In the context of pointed models, the situation is now well understood. Johan van Benthem proved that modal logic is the bisimulation invariant fragment of first-order logic [Ben76], while David Janin and Igor Walukiewicz extended this result to the setting of the μ-calculus [JW96] (that is to say, the μ-calculus is the bisimulation invariant fragment of MSO).

Since the introduction of the relational semantics for modal logic, the expressive power of modal logic in the context of frames has been studied as well. One of the motivations is that modal logic, when evaluated on frames, can express natural graph-theoretic properties, such as reflexivity, transitivity, etc. Even though modal logic is a fragment of second-order logic, the focus has always been more on the link with first-order logic. The most important works in that respect consist of Sahlqvist theory and semantic characterizations of modally definable classes of frames. The Sahlqvist fragment [Sah75] is a rather large syntactic fragment of modal logic that contain formulas corresponding, ar the level of pointed frames, to first-order formulas. It turned out that the formulas in this fragment also enjoy another desirable property: canonicity. The most famous example of a semantic characterization of modally definable classes of frames is the Goldblatt-Thomasom theorem [GT75]. It provides necessary and sufficient conditions for an elementary class of frames (that is, definable by a set of first-order formulas) to be definable by a set of modal formulas.

Even though there is a long tradition of studying modal logic in the context of frames, nothing is known for the μ-calculus. In this chapter, we contribute to the understanding of the expressive power of the μ-calculus at the level of frames. Since the question is relatively complex, we concentrate on trees, which are fundamental structure in many areas of logic. More precisely, we investigate under which conditions an MSO formula is *frame definable* in the particular case where the frames are trees. An MSO formula φ is said to be (globally) frame definable if there exists a μ-sentence ψ with proposition letters $p_1, \dots, p_n$ such that φ is equivalent to $\forall x \forall P_1, \dots, \forall P_n ST_x(\psi)$. Local frame definability is defined in a similar fashion, except that we replace the last formula by $\forall P_1, \dots, \forall P_n ST_x(\psi)$.

The results we provide can be expressed in a simple way: an MSO formula is frame definable on trees iff it is preserved under p-morphic images on trees and under taking subtrees. Moreover, an MSO formula is locally frame definable on trees iff it is preserved under p-morphic images on trees. Basically p-morphisms are functional bisimulations. Using these results, we show that it is decidable whether a given MSO formula is (locally) frame definable on trees.

The proof of the results proceeds in three steps. The first step is the most straightforward: we show that a characterization of local frame definability on trees induces a characterization of frame definability on trees. Hence, we can restrict ourselves to the study of local frame definability on trees.

For the second step, we use the connection between MSO and the graded μ-calculus proved by Igor Walukiewicz [Wal02]. We establish a correspondence between the MSO formulas that are preserved under p-morphic images on trees and a fragment lying between the μ-calculus and the graded μ-calculus. We denote this fragment by $\mu\mathrm{ML}^{\nabla'}$.

The second step consists in showing that each formula φ in $\mu\mathrm{ML}^{\nabla'}$ the propositions letters of which belongs to a set *Prop*, can be translated into a μ-sentence ψ, which may contain fresh proposition letters (that is, proposition letters which do not appear in *Prop*), and such that the following correspondence holds. At a given point u in a tree model over *Prop*, the *truth* of φ at u corresponds to the *validity* of ψ at u (that is, we quantify over all possible valuations for the fresh proposition letters). In other words, this step is a shift from the model perspective to the frame perspective.

We also present two variations of our main result. We ask exactly the same question, but with a different definition of frame definability. The standard notion of frame definability involves a universal second order quantification over all proposition letters, or to put it in a different way, over all possible valuations. Here, we replace this universal quantification by an existential quantification. As we will see later in this chapter, there are two ways to implement this existential quantification: the first one corresponds to negative definability (introduced by Yde Venema in [Ven93]) and the second one corresponds to projective definability.

The chapter is organized as follows. In the first section, we introduce notation and state the main results. We also show that a characterization of local frame definability on trees easily implies a characterization of global frame definability on trees. In the second section, we recall the link between MSO and the graded μ-calculus. We also show that there exists a disjunctive normal form for the graded μ-calculus. In the next section, we establish a correspondence, on the level of tree models, between the fragment of MSO which is closed under p-morphic images and $\mu\mathrm{ML}^{\nabla'}$. In Section 4, we show how to derive our main result from this correspondence. In the last section, we give two variations of our main results, which concern negative definability and projective definability.

We would like to thank Balder ten Cate for his precious help in correcting a mistake in one of the proofs of this chapter.

4.1 μMLF-definability on trees

Our goal is to characterize the MSO formulas that are equivalent in the context of frames to μ-sentences. The characterization we propose is very natural and only involves two well-known operations of modal logic: taking subtrees and p-morphic images. We recall these notions together with some basic terminology and state our main result.

Signature for MSO In the preliminaries, we mentioned that when interpreting MSO on Kripke models, the signature consists of a binary relation and a unary predicate for each proposition letter. In this chapter, we will mostly deal with MSO on tree models. We modify the signature by adding a unary predicate for the root. Given a tree model, the interpretation of that unary predicate is the singleton consisting of the root of the tree model.

Moreover, in this chapter, we will deal with two sets of proposition letters, *Prop* and *Prop'*. The models we consider are usually models over *Prop* or over $Prop \cup Prop'$. Sometimes it might confusing which signature we have in mind. However, the models on which we evaluate MSO formulas are *always* models over *Prop*. Or to put it in other words, the signature for MSO in this chapter is as follows: it consists of a binary relation, a unary predicate for each proposition letter in *Prop* and a unary predicate for the root.

Trees A Kripke frame (T, R) is a *tree* if for some point $r \in T$, $T = \{r\} \cup R^+[r]$, r does not have a predecessor and every state $t \neq r$ has a unique predecessor. If (T, R) is a tree, then v is a *child* of u if $(u, v) \in R$ and v is a *descendant* of u if $(u, v) \in R^+$.

If (T, R, V) is a Kripke model over a set *Prop* and (T, R) is a tree, we say that (T, R, V) is a *tree model* over *Prop*. If a tree model $\mathcal{T}$ is a Kripke model over a set $Prop \cup Prop'$ of propositions, we may represent $\mathcal{T}$ as a tuple (T, R, V, V') where (T, R) is a tree, V is a map from *Prop* to $\mathcal{P}(T)$ and V' is a map from *Prop'* to $\mathcal{P}(T)$. In this chapter, a tree will always be a tree model. Since the set of proposition letters is not fixed, we always try to specify whether a tree is a tree over *Prop* or over $Prop \cup Prop'$.

μMLF**-definability** An MSO formula φ is μMLF-*definable on trees* if there are a set *Prop'* and a μ-sentence ψ over $Prop \cup Prop'$ such that for all trees $\mathcal{T} = (T, R, V)$ over *Prop*,

$$\mathcal{T} \vDash \varphi \quad \text{iff} \quad \text{for all } u \in T, \text{ for all } V' : Prop' \to \mathcal{P}(T),\ (T, R, V, V'), u \Vdash \psi. \tag{4.1}$$

In case $Prop = \emptyset$, this means that for all trees (T, R), φ is satisfied in (T, R) iff ψ is valid in (T, R). The letter F in the abbreviation μMLF stands for "frames" (and μML stands for modal fixpoint logic, as usual).

An MSO formula φ is *locally* μMLF-*definable on trees* if there are a set *Prop'* and a μ-sentence ψ over $Prop \cup Prop'$ such that for all trees $\mathcal{T} = (T, R, V)$ over *Prop*,

$$\mathcal{T} \vDash \varphi \quad \text{iff} \quad \text{for all } V' : Prop' \to \mathcal{P}(T),\ (T, R, V, V'), r \Vdash \psi, \tag{4.2}$$

where r is the root of $\mathcal{T}$. When this happens, we say that φ is *locally* μMLF-*definable on trees by* ψ.

p**-morphisms** Let $\mathcal{M} = (W, R, V)$ and $\mathcal{M}' = (W', R', V')$ be two models over *Prop*. A map $f : W \to W'$ is a *p-morphism* between $\mathcal{M}$ and $\mathcal{M}'$ if the two following conditions hold. For all $w, v \in W$ such that wRv, we have $f(w)R'f(v)$.

For all $w \in W$ and $v' \in W'$ such that $f(w)R'v'$, there exists $v \in W$ such that $f(v) = v'$ and wRv.

Essentially p-morphisms are functional bisimulations. An MSO formula φ is *preserved under p-morphic images on trees* if for all surjective p-morphisms f between two trees $\mathcal{T}$ and $\mathcal{T}'$ over *Prop*,

$$\mathcal{T} \vDash \varphi \quad \text{implies} \quad \mathcal{T}' \vDash \varphi.$$

Preservation under taking subtrees A *subtree* of a tree $\mathcal{T} = (T, R, V)$ over *Prop* is a submodel of $\mathcal{T}$, the domain of which consists of a node u of $\mathcal{T}$ and all the descendants of u. If $\mathcal{T}$ is a tree over *Prop* and u is a node of $\mathcal{T}$, we let $\mathcal{T}_u$ denote the subtree of $\mathcal{T}$ with root u.

An MSO formula φ is *preserved under taking subtrees* if for all trees $\mathcal{T}$ over *Prop* and for all nodes u of $\mathcal{T}$,

$$\mathcal{T} \vDash \varphi \quad \text{implies} \quad \mathcal{T}_u \vDash \varphi.$$

We can now state the main result of this chapter.

4.1.1. THEOREM. *An* MSO *formula φ is μMLF-definable on trees iff φ is preserved under p-morphic images on trees and under taking subtrees. An* MSO *formula φ is locally μMLF-definable on trees iff φ is preserved under p-morphic images on trees.*

Most of the remaining part of this chapter is devoted to the proof of this result. We concentrate on the direction from right to left, since the other direction is easy. Our first observation is that it is sufficient to obtain a characterization of the formulas locally μMLF-definable on trees.

4.1.2. LEMMA. *If an* MSO *formula φ is locally μMLF-definable on trees and is preserved under taking subtrees, then φ is μMLF-definable on trees.*

Proof Let φ be an MSO formula that is locally μMLF-definable on trees and preserved under taking subtrees. Since φ is locally μMLF-definable on trees, there are a set $Prop'$ and a μ-sentence ψ over $Prop \cup Prop'$ such that for all trees $\mathcal{T} = (T, R, V)$ over *Prop*, (4.2) holds. It is sufficient to show that equivalence (4.1) holds. The only direction that is not immediate is from right to left. Suppose that $\mathcal{T} \vDash \varphi$. Let u be a node of $\mathcal{T}$. Since $\mathcal{T} \vDash \varphi$ and φ is preserved under taking subtrees, we also have $\mathcal{T}_u \vDash \varphi$. Putting this together with equivalence (4.2), we obtain that for for all $V' : Prop' \to \mathcal{P}(T)$, $(T, R, V, V'), u \Vdash \psi$.

It follows from the lemma that in order to obtain Theorem 4.1.1, we can restrict ourselves to show that if an MSO formula φ is preserved under p-morphic

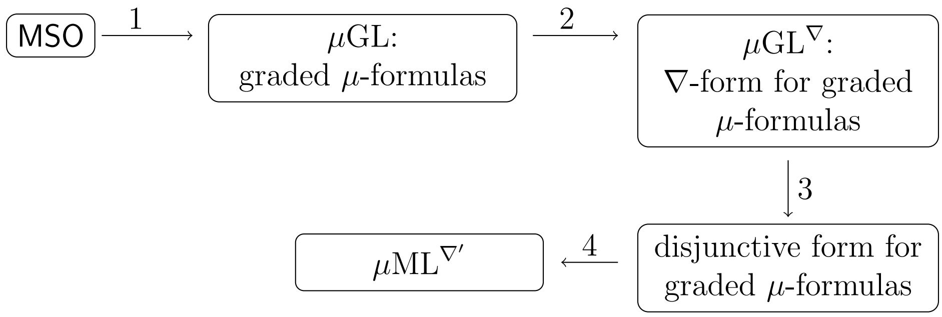


Figure 4.1: General scheme of the first part of the construction of ψ.

images on trees, then φ is locally μMLF-definable on trees. So suppose that φ is an MSO formula that is preserved under p-morphic images on trees. We have to find a set *Prop*′ and a μ-sentence ψ such that (4.2) holds. The route we take to go from φ to ψ involves many intermediate logics between MSO and the μ-calculus. So it might help for the understanding of the proof to give some indication about the intuition and the main steps for the construction of ψ.

The basic idea is that the quantification over all valuations in (4.2) allows some form of counting. This is the reason why the graded μ-calculus plays such an important role in this chapter. However, the counting provided by μMLF-definability is weaker than the usual counting of the graded μ-calculus. The counting that we can express with μMLF-definability is captured by an operator that we denote by ∇'. This operator is a stronger version of the usual ∇ operator. We define $\mu\mathrm{ML}^{\nabla'}$ as the logic obtained by replacing ∇ by ∇' in the formulas of $\mu\mathrm{ML}^{\nabla}$.

The construction of ψ is in two parts. The first part consists in applying to φ several successive transformations. With each transformation, we associate a language, as depicted in Figure 4.1. For example, the logic associated with the first transformation is the graded μ-calculus. After each transformation, we should obtain a formula that belongs to the logic corresponding to the transformation. Moreover, under the conditions that φ is preserved under p-morphic images and that we restrict to trees, this formula is equivalent to φ. In fact, for some of the transformations, this equivalence remains true under some milder conditions. Later in the chapter, we will give a more detailed table stating the exact links between the logics of Figure 4.1.

As shown in Figure 4.1, the formula resulting after all the transformations is a formula in $\mu\mathrm{ML}^{\nabla'}$. That is, if we replace the usual ∇ operator by ∇', we obtain exactly the MSO formulas that are μMLF-definable on trees. The second part of the construction of ψ (presented in Section 4.4) consists in showing that for all sentences χ in $\mu\mathrm{ML}^{\nabla'}$, we can find a set *Prop*′ and a μ-sentence ψ such that for all trees $\mathcal{T} = (T, R, V)$ over *Prop*,

$$\mathcal{T}, r \Vdash \chi \quad \text{iff} \quad \text{for all } V' : \mathit{Prop}' \to \mathcal{P}(T),\ (T, R, V, V'), r \Vdash \psi.$$

where r is the root of $\mathcal{T}$.

4.2 Graded μ-calculus: connection with MSO and disjunctive normal form

In this section, we recall the connection between MSO and the graded μ-calculus, established in [Wal02] and [Jan06]. We introduce a ∇-like operator for the graded μ-calculus, inspired by [Wal02]. Using results from [AN01], we also show that there is a disjunctive normal for the graded μ-calculus. This section basically corresponds to the arrows (1), (2) and (3) in Figure 4.1.

For the first transformation of Figure 4.1, we can simply rely on existing results. Igor Walukiewicz established that on trees, MSO has the same expressive power as a special sort of automaton, called later non-deterministic counting automaton by David Janin [Jan06]. David Janin [Jan06] also showed that for all counting automata, we can compute an equivalent graded μ-sentence. Hence, MSO and the graded μ-calculus are equi-expressive on trees. As observed in [Jan06] and [JL03], we can derive from this result that on arbitrary models, the graded μ-calculus is the fragment of MSO that is preserved under counting bisimulation (see Section 2.7).

Equivalence between MSO and the graded μ-calculus An MSO sentence φ is *equivalent on trees* to a sentence ψ in μGL if for all trees $\mathcal{T}$ with root r,

$$\mathcal{T} \vDash \varphi \quad \text{iff} \quad \mathcal{T}, r \Vdash \psi.$$

4.2.1. Theorem (from [Wal02], [Jan06]). *MSO and the graded μ-calculus are effectively equi-expressive on trees. That is, for all MSO formulas φ, we can compute a graded μ-sentence ψ over Prop such that φ and ψ are equivalent on trees over Prop, and vice-versa.*

For the second transformation of Figure 4.1, we should introduce a ∇-like operator for the μ-calculus. The definition is inspired by the automata presented by Igor Walukiewicz in [Wal02].

∇ operator for the graded μ-calculus Given a set A, a *finite tuple* $\vec{a}$ over A is a tuple of the form $(a_1, \ldots, a_k)$, where $k \in \mathbb{N}$ and for all $0 \leq i \leq k$, a_i belongs to A. In case $k = 0$, then $\vec{a} = \emptyset$.

The set $\mu\mathrm{GL}^{\nabla}$ of ∇-formulas of the graded μ-calculus (over a set *Prop* of proposition letters and a set *Var* of variables) is given by:

$$\varphi \ ::= \ x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \alpha \bullet \nabla^g(\vec{\varphi};\Psi) \mid \mu x.\varphi \mid \nu x.\varphi,$$

where $x \in \mathit{Var}$, α is a conjunction literals over *Prop*, Ψ is a finite set of formulas and $\vec{\varphi}$ is a finite tuple of formulas.

Given a formula φ, a model $\mathcal{M} = (W, R, V)$, an assignment $\tau : \mathit{Var} \to \mathcal{P}(W)$ and a point $w \in W$, the relation $\mathcal{M}, w \Vdash_\tau \varphi$ is defined by induction as in the case of the μ-calculus with the extra condition:

$\mathcal{M}, w \Vdash_\tau \alpha \bullet \nabla^g(\vec{\varphi}, \Psi)$ iff $\mathcal{M}, w \Vdash_\tau \alpha$ and $\mathcal{M}, w \Vdash_\tau \nabla^g(\vec{\varphi}, \Psi)$,
$\mathcal{M}, w \Vdash_\tau \nabla^g(\vec{\varphi}, \Psi)$ iff there exists a tuple $(w_1, \dots w_k)$ over $R[w]$ such that,

1. for all $1 \leq i < j \leq k$, $w_i \neq w_j$,
2. for all $1 \leq i \leq k$, $\mathcal{M}, w_i \Vdash_\tau \varphi_i$,
3. for all u in $R[w] \backslash \{w_i \mid 1 \leq i \leq k\}$,
$\mathcal{M}, u \Vdash_\tau \bigvee \Psi$.

where $\vec{\varphi} = (\varphi_1, \dots, \varphi_k)$. We say that $(w_1, \dots, w_k)$ is a tuple of ∇^g-*witnesses* for the pair $(\vec{\varphi}, \Psi)$ and the point w. If w is clear from the context, we simply say that $(w_1, \dots, w_k)$ is a tuple of ∇^g-*witnesses* for the pair $(\vec{\varphi}, \Psi)$. Moreover, for all $1 \leq i \leq n$, w_i is a ∇^g-*witness associated with* φ_i.

A map $m : \mu\mathrm{GL}^{\nabla} \to \mathcal{P}(R[w])$ is a ∇^g-*marking* for $((\varphi_1, \dots, \varphi_k), \Psi)$ if there exists a tuple $(w_1, \dots, w_k)$ such that for all $1 \leq i < j \leq k$, $w_i \neq w_j$, $w_i \in m(\varphi_i)$ and for all successors u of w such that $u \notin \{w_i \mid 1 \leq i \leq k\}$, there is $\psi \in \Psi$ such that $u \in m(\psi)$.

A sentence φ in $\mu\mathrm{GL}^{\nabla}$ is *equivalent on trees* to an MSO formula ψ if for all trees $\mathcal{T}$ over *Prop*, ψ is valid on $\mathcal{T}$ iff φ is true at the root of $\mathcal{T}$. A formula φ in $\mu\mathrm{GL}^{\nabla}$ is *equivalent* to a formula ψ in $\mu\mathrm{GL}$ if for all Kripke models $\mathcal{M}$, for all assignments $\tau : \mathit{Var} \to \mathcal{P}(\mathcal{M})$ and for all $w \in \mathcal{M}$, we have $\mathcal{M}, w \Vdash_\tau \varphi$ iff $\mathcal{M}, w \Vdash_\tau \psi$.

The set of formulas in $\mu\mathrm{GL}^{\nabla}$ in *disjunctive normal form* is defined by induction in the following way:

$$\varphi \ ::= \ x \mid \varphi \vee \varphi \mid \alpha \bullet \nabla^g(\vec{\varphi};\Psi) \mid \mu x.\varphi \mid \nu x.\varphi,$$

where $x \in \mathit{Var}$, α is a conjunction of literals over *Prop*, Ψ is a finite set of formulas and $\vec{\varphi}$ is a finite tuple of formulas. This essentially means that the only conjunctions that are allowed are conjunctions of literals.

4.2.2. Proposition. *$\mu\mathrm{GL}$ and $\mu\mathrm{GL}^{\nabla}$ are effectively equi-expressive. That is, for all formulas in $\mu\mathrm{GL}$, we can compute an equivalent formula in $\mu\mathrm{GL}^{\nabla}$, and vice-versa.*

Proof First, we show that each formula φ of the graded μ-calculus is equivalent to a ∇-formula of the graded μ-calculus. The proof is by induction on the complexity of φ. We only treat the cases where φ is of the form $\Diamond^k\psi$ or $\Box^k\psi$ (which are the only cases that are not straightforward). One can check that $\Diamond^k\psi$ is equivalent to $\top \bullet \nabla^g((\varphi, \dots, \varphi), \{\top\})$, where $(\varphi, \dots, \varphi)$ is a tuple of length $k+1$. The formula $\Box^k\psi$ is equivalent to $\top \bullet \nabla^g((\top, \dots, \top), \{\psi\})$, where $(\top, \dots, \top)$ is a tuple of length k.

Second, we have to verify that each ∇-formula φ of the graded μ-calculus is equivalent to a graded μ-calculus formula. for all $v \in R[w]$, we have $(v, f(v)) \in B$. We have to show that for all $(w, w') \in B$, we have

The proof is also by induction on the complexity of the formulas. We only treat the most difficult case, where $\varphi = \alpha \bullet \nabla^g(\vec{\varphi}, \Psi)$. Suppose that $\vec{\varphi}$ is the tuple $(\varphi_1, \dots, \varphi_k)$. We also define φ_{k+1} as the formula $\bigvee \Psi$. If $i > 0$, we abbreviate the set $\{1, \dots i\}$ by $[i]$.

We start by giving some intuition. Assume that the formula $\nabla^g(\vec{\varphi}, \psi)$ is true at a point w in a model $\mathcal{M} = (W, R, V)$. This means that there exist pairwise distinct successors $w_1, \dots, w_k$ of w such for all $1 \leq i \leq k$, $\mathcal{M}, w_i \Vdash \varphi_i$ and for all $u \in R[w] \backslash \{w_1, \dots, w_n\}$, we have $\mathcal{M}, u \Vdash \varphi_{k+1}$. Now we have to use the operators $\Box^n$ and $\Diamond^n$ to describe the situation. The way we use these operators depends very much on which formulas among $\varphi_1, \dots, \varphi_{k+1}$ are true at the points $w_1, \dots, w_k$. In order to encode this information, we introduce a map $f : [k] \longrightarrow \mathcal{P}[k+1]$ with the intended meaning that φ_j is true at w_i iff $j \in f(i)$. So in particular, for all $i \in [k]$, i belongs to $f(i)$. Given a subset N of $[k+1]$, we define the formula $\psi'(N)$ by:

$$\psi'(N) = \left(\bigwedge \{\varphi_j \mid j \in N\} \wedge \bigwedge \{\neg\varphi_j \mid j \notin N\} \right).$$

Given a point v in a model, there is a unique $N \subseteq [k+1]$ such $\psi'(N)$ is true at v; $\psi'(N)$ is essentially the type of v with respect to the set of formulas $\{\varphi_1, \dots, \varphi_{k+1}\}$. So if the map f is defined as mentioned before, the type of the point w_i is the formula $\psi'(f(i))$.

Now we want to use the operator $\Diamond^n$ in order to express that we have enough witnesses making $\varphi_1, \dots, \varphi_k$ true. If $i \in [k]$, how many successors of w make $\psi'(f(i))$ true (and in particular φ_i)? By definition of f, the number of such successors is at least equal to $n(i, f)$ given by

$$n(i, f) := |\{j \in [k] \mid f(i) = f(j)\}|.$$

Hence, we should require that $\Diamond^{n(i,f)}\psi'(f(i))$ holds at w. We also want to use the operator $\Box^n$ to express that the successors of w that do not belong to $\{w_1, \dots, w_n\}$ make φ_{k+1} true. Which successors of w do not make φ_{k+1} true? Only the w_is such that $k+1 \notin f(i)$. Therefore, if we define $m(f)$ by:

$$m(f) = |\{i \in [k] \mid k+1 \notin f(i)\}|,$$

the formula $\Box^{m(f)}\varphi_{k+1}$ is true at w. Putting everything together, we have that the formula $\psi(f)$ given by:

$$\psi(f) = \bigwedge\{\Diamond^{n(i,f)}\psi'(f(i)) \mid i \in [k]\} \wedge \Box^{m(f)}\varphi_{k+1}$$

is true at w.

We are now ready to define ψ. We let ψ be the formula given by:

$$\psi = \bigvee\{\psi(f) \mid f : [k] \longrightarrow \mathcal{P}[k+1] \text{ and for all } i \in [k],\, i \in f(i)\}.$$

We prove that the formula $\nabla^g(\vec{\varphi}, \Psi)$ is equivalent to the formula ψ. We do not give details for the fact that $\nabla^g(\vec{\varphi}, \Psi)$ implies ψ, as the proof is similar to the intuition that we gave above.

Now we prove that ψ implies $\nabla^g(\vec{\varphi}, \Psi)$. Let w be a point in a model $\mathcal{M} = (W, R, V)$ and let $\tau : \mathit{Var} \longrightarrow \mathcal{P}(W)$ be a valuation such that $\mathcal{M}, w \Vdash_\tau \psi$. Thus, there is a map $f : [k] \longrightarrow \mathcal{P}[k+1]$ such that for all $i \in [k]$, i belongs to $f(i)$ and the formula $\psi(f)$ is true at w.

Now take a set $N \subseteq [k+1]$ in the range of f. Hence, there is $i \in [k]$ such that $N = f(i)$. Since $\psi(f)$ is true at w, the formula $\Diamond^{n(i,f)}\psi'(f(i))$ is true at w. That is, there are at least $n(i, f)$ successors of w at which $\psi'(N)$ is true. Recall that $n(i, f)$ is the size of the set $\{j \in [k] \mid f(i) = f(j)\}$. That is, $n(i, f)$ the size of the set $\{j \in [k] \mid f(j) = N\}$. Hence, we can fix an injective map $g_N : \{j \in [k] \mid f(j) = N\} \longrightarrow W$ such that for all j in the domain of g_N,

$$\mathcal{M}, g_N(j) \Vdash_\tau \psi'(N). \tag{4.3}$$

For each $j \in [k]$, we define w_j as the point $g_{f(j)}(j)$. It follows from (4.3) that $\psi'(f(j))$ is true at w_j. In order to show that $\nabla^g(\vec{\varphi}, \Psi)$ is true at w, it is sufficient to prove that

(i) for all $i, j \in [k]$ such that $i \neq j$, we have $w_i \neq w_j$,

(ii) for all $i \in [k]$, $\mathcal{M}, w_i \Vdash_\tau \varphi_i$,

(iii) for all successors u of w such that $u \notin \{w_i \mid i \in [k]\}$, $\mathcal{M}, u \Vdash_\tau \bigvee \Psi$.

We start by showing that for all $i, j \in [k]$ such that $i \neq j$, we have $w_i \neq w_j$. Take $i, j \in [k]$ such that $i \neq j$. First assume that $f(i) \neq f(j)$ and suppose for contradiction that $w_i = w_j$. Recall that given a point u in $\mathcal{M}$, there is a unique subset N of $[k+1]$ such that $\psi'(N)$ is true at u. We observed earlier that $\psi'(f(i))$ is true at w_i and $\psi'(f(j))$ is true at w_j. Since $w_i = w_j$, $\psi'(f(i))$ and $\psi'(f(j))$ are true at w_i. This is contradiction as $f(i) \neq f(j)$. Next suppose that $f(i) = f(j)$. By definition, w_i is equal to $g_{f(i)}(i)$ and w_j is equal to $g_{f(j)}(j)$. Since $f(i) = f(j)$, w_j is equal to $g_{f(i)}(j)$. Since $g_{f(i)}$ is an injective map and $i \neq j$, we have $g_{f(i)}(i) \neq g_{f(i)}(j)$. That is, $w_i \neq w_j$.

For (ii), take $i \in \{1, \dots, n\}$. We observed earlier that $\psi'(i, f)$ is true at w_i. Since i belongs to $f(i)$, it is immediate from the definition of $\psi'(i, f)$ that if $\psi'(i, f)$ is true at a point, then φ_i is also true at that point. In particular, φ_i is true at w_i.

So it remains to prove (iii). That is, for all successors u of w such that $u \notin \{w_i \mid i \in [k]\}$, $\mathcal{M}, u \Vdash_\tau \bigvee \Psi$. Since $\Box^{m(f)} \bigvee \Psi$ is true at w, there are most $m(f)$ successors of w at which $\bigvee \Psi$ is not true. Recall that for all $i \in [k]$, $\psi'(f(i))$ is true at w_i. In particular, if $k+1$ does not belong to $f(i)$, $\bigvee \Psi$ is not true at w_i. Or in other words, $\bigvee \Psi$ is not true at any point of the set $\{w_i \mid i \in [k], k+1 \notin f(i)\}$. By definition, the size of this set is $m(f)$. Putting this together with the fact that there are most $m(f)$ successors of w at which $\bigvee \Psi$ is not true, we obtain that $\bigvee \Psi$ is true at all successors of w which do not belong to $\{w_i \mid i \in [k], k+1 \notin f(i)\}$. In particular, for all successors u of w such that $u \notin \{w_i \mid i \in [k]\}$, $\mathcal{M}, u \Vdash_\tau \bigvee \Psi$. This finishes the proof that ψ implies $\nabla^g(\vec{\varphi}, \Psi)$. □

4.2.3. Remark. Another (shorter) proof for Proposition 4.2.2 is to use the fact that the graded μ-calculus is the fragment of MSO invariant under counting bisimulation and to show that the formulas in $\mu\mathrm{GL}^\nabla$ are equivalent to MSO formulas invariant under counting bisimulation. However, in the proof of the fact that the graded μ-calculus is the fragment of MSO invariant under counting bisimulation, David Janin [Jan06] skips the proof that a non-deterministic counting automaton is equivalent to a counting automaton. The latter proof corresponds exactly to the proof that each formula in $\mu\mathrm{GL}^\nabla$ is equivalent to a formula in $\mu\mathrm{GL}$. This motivated our decision to give details for the proof of Proposition 4.2.2, instead of using the shortcut provided by the the fact that the graded μ-calculus is the fragment of MSO invariant under counting bisimulation.

Now we move on to the third arrow of Figure 4.1. That is, we show that there is a normal form for the graded μ-calculus.

4.2.4. Theorem. *For each formula of the graded μ-calculus, we can compute an equivalent formula of the graded μ-calculus in disjunctive normal form.*

This theorem follows from an application of a result from [AN01]. We start by recalling the definitions required to state the result from this book.

Fixpoint algebras A *signature* is a set *Sig* of function symbols equipped with an arity function $\rho : Sig \longrightarrow \mathbb{N}$. Let $\mathbb{P} = (P, \leq_\mathbb{P})$ be a complete lattice (see Section 2.2). We say that $\mathbb{P} = (P, \leq_\mathbb{P})$ is *distributive* if for all $c, d, e \in P$, we have

$$c \wedge (d \vee e) = (c \wedge d) \vee (c \wedge e).$$

Over a signature *Sig*, a *fixpoint algebra* $\mathbb{P}$ is a complete lattice $(P, \leq_\mathbb{P})$ together with, for each symbol $f \in Sig$, a monotone function $f_\mathbb{P} : P^{\rho(f)} \longrightarrow P$. We always assume that the binary operations $\wedge$ and $\vee$ and the 0-ary operations $\bot$ and $\top$ belong to *Sig*. We let $\mathcal{F}_{Sig}$ denote the set of *terms* that contains the identity operation (which is a unary operation and maps an element $d \in P$ to d), all the operations in *Sig* and that is closed under the composition operation.

Over a signature *Sig* and a set *Var* of variables, the *fixpoint formulas* are defined by

$$\varphi ::= x \mid f(\varphi, \ldots, \varphi) \mid \mu x.\varphi \mid \nu x.\varphi,$$

where $x \in Var$ and $f \in Sig$. Given a fixpoint algebra $\mathbb{P}$ and an assignment $\tau : Var \longrightarrow \mathbb{P}$, we define the set $[\![\varphi]\!]_{\mathbb{P},\tau}$ similarly to the case of modal fixpoint logic. More specifically, we define $[\![\varphi]\!]_{\mathbb{P},\tau}$ by induction on φ in the following way:

$$\begin{aligned} [\![x]\!]_{\mathbb{P},\tau} &= \tau(x), \\ [\![f(\varphi_1, \ldots, \varphi_n)]\!]_{\mathbb{P},\tau} &= f_{\mathbb{P}}([\![\varphi_1]\!]_{\mathbb{P},\tau}, \ldots, [\![\varphi_n]\!]_{\mathbb{P},\tau}), \\ [\![\mu x.\varphi]\!]_{\mathbb{P},\tau} &= \bigwedge\{c \in \mathbb{P} \mid c \leq_{\mathbb{P}} [\![\varphi]\!]_{\mathbb{P},\tau[x\mapsto c]}\}\ , \\ [\![\nu x.\psi]\!]_{\mathbb{P},\tau} &= \bigvee\{c \in \mathbb{P} \mid c \geq_{\mathbb{P}} [\![\varphi]\!]_{\mathbb{P},\tau[x\mapsto c]}\}, \end{aligned}$$

where $x \in Var$, f is a symbol in $Sig \cup \{\wedge, \vee, \bot, \top\}$ of arity n and $\tau[x \mapsto c]$ is the assignment $\tau' : Var \longrightarrow \mathbb{A}$ such that $\tau'(x) = c$ and for all variables $y \neq x$, $\tau'(y) = \tau(y)$. We say that $[\![\varphi]\!]_{\mathbb{P},\tau}$ is the *interpretation* of φ in $\mathbb{P}$ under the assignment τ. As usual, a fixpoint formula is a *fixpoint sentence* if each variable occurring in the formula is bound.

If φ and ψ are fixpoint formulas, φ and ψ are *equivalent* over a class $\mathcal{L}$ of fixpoint algebras if for all fixpoint algebras $\mathbb{P}$ in $\mathcal{L}$ and for all assignments $\tau : Var \longrightarrow \mathbb{P}$, we have $[\![\varphi]\!]_{\mathbb{P},\tau} = [\![\psi]\!]_{\mathbb{P},\tau}$. When this happens, we say that the equation $\varphi = \psi$ *holds* on $\mathcal{L}$.

If $\vec{x}$ is a tuple of variables, we denote by $\bigwedge \vec{x}$ the greatest lower bound of the set of variables occurring in $\vec{x}$. Given a class of fixpoint algebras $\mathcal{L}$, we say that the meet operator $\wedge$ *commutes* with *Sig* on $\mathcal{L}$ if for all finite tuples $(f_1, \ldots, f_n)$ of functional symbols of $Sig \backslash \{\wedge, \vee\}$, there exists a function $g \in \mathcal{F}_{Sig}$ built without the symbol $\wedge$ such that an equation of the form:

$$\bigwedge\{f_i(\vec{x}_i) \mid 1 \leq i \leq n\} = g(\bigwedge \vec{y}_1, \ldots, \bigwedge \vec{y}_m). \tag{4.4}$$

holds on $\mathcal{L}$, where the $\vec{x}_i$s are tuples of distinct variables of the appropriate length and the $\vec{y}_j$s are tuples of distinct variables taken among those appearing in $\vec{x}_i$s.

It was shown in [AN01] that when the meet operator commutes with a signature on a class $\mathcal{L}$ of fixpoint algebras, then each fixpoint formula is equivalent to a formula that does not contain the meet operator. A close inspection of the proof shows that this last formula can be computed from the initial formula.

4.2.5. Theorem. *[Corollary 9.6.9 from[AN01]] When the meet operator $\wedge$ commutes with Sig on $\mathcal{L}$, each fixpoint formula φ is equivalent over $\mathcal{L}$ to a formula ψ built without the symbol $\wedge$.*

Moreover, if the maps gs occurring in the equations (4.4) *are computable, then* ψ *is computable.*

Now we show how to use Theorem 4.2.5 in order to obtain Theorem 4.2.4. We define the signature $Sig_{\mu\mathrm{GL}^\triangledown}$ as the set consisting of $\top$, $\bot$, $\vee$, $\wedge$ and the operators of the form $\alpha \bullet \nabla^g_{k,l}$, with $k, l \in \mathbb{N}$ and α a subset of literals over *Prop*. The arity of the operator $\alpha \bullet \nabla^g_{k,l}$ is $k + l$.

With each model $\mathcal{M} = (W, R, V)$, we associate a fixpoint algebra $\mathbb{P}_\mathcal{M}$ over $Sig_{\mu\mathrm{GL}^\triangledown}$ in the following way. The fixpoint algebra $P_\mathcal{M}$ is based on the complete distributive lattice $(\mathcal{P}(W), \subseteq)$. Hence, the interpretations of the operators $\top$, $\bot$, $\vee$ and $\wedge$ are respectively the constants W and $\emptyset$ and the operations $\cup$ and $\cap$.

Given a subset α of *Prop* and $k, l \in \mathbb{N}$, the operator $\alpha \bullet \nabla^g_{k,l}$ maps subsets $U_1, \dots, U_k, T_1, \dots, T_l$ to the set of $w \in W$ which belongs to $\bigcap\{V(p) \mid p \in \alpha\} \cap \bigcap\{W \backslash V(p) \mid \neg p \in \alpha\}$ and for which there exists a set $\{w_i \mid 1 \le i \le k\} \subseteq R[w]$ such that:

- for all $1 \le i < j \le k$, $w_i \ne w_j$,
- for all $1 \le i \le k$, w_i belongs to U_i,
- for all successors u of w such that $u \notin \{w_i \mid 1 \le i \le k\}$, u belongs to $\bigcup\{T_j \mid 1 \le j \le l\}$.

If we identify each operator $\alpha \bullet \nabla^g_{k,l}$ with $\bigwedge \alpha \bullet \nabla^g$ in the obvious way, then for all fixpoint formulas φ over $Sig_{\mu\mathrm{GL}^\triangledown}$ and for all valuations $\tau : Var \to \mathcal{P}(W)$, we have that $[\![\varphi]\!]_{\mathbb{A}_\mathcal{M},\tau} = [\![\varphi]\!]_{\mathcal{M},\tau}$.

Finally we define $\mathcal{L}_{\mathrm{ML}}$ as $\{\mathbb{P}_\mathcal{M} \mid \mathcal{M} \text{ is a Kripke model }\}$. In order to derive Theorem 4.2.4 from Theorem 4.2.5, it is sufficient to show the following result.

4.2.6. PROPOSITION. *The meet operator commutes with* $Sig_{\mu\mathrm{GL}^\triangledown}$ *on* $\mathcal{L}_{\mathrm{ML}}$.

Proof Let $(f_1, \dots, f_n)$ be a tuple of symbols in $Sig_{\mu\mathrm{GL}^\triangledown}$ and for all $1 \le i \le n$, let $\vec{x}_i$ be a vector of distinct variables, the length of which is the arity of f_i. Let also F be the set $\{f_i(\vec{x}_i) \mid 1 \le i \le n\}$. We have to prove that an equation of the form of (4.4) holds on $\mathcal{L}_{\mathrm{ML}}$.

We start by showing that without loss of generality, we can suppose that for all $1 \le i \le n$, the operator f_i is an operator of the form $\alpha_i \bullet \nabla^g_{k_i,l_i}$. This is based on the following two observations. First, if an operator f_{i_0} is equal to $\bot$, then the equation $\bigwedge F = \bot$ holds on $\mathcal{L}_{\mathrm{ML}}$. Second, if an operator f_{i_0} is $\top$, then the equation $\bigwedge F = \bigwedge\{f_i(\vec{x}_i) \mid 1 \le i \le n, i \ne i_0\}$ holds on $\mathcal{L}_{\mathrm{ML}}$.

Now we show that an equation of the form (4.4) holds if all the f_is are operators of the form $\alpha_i \bullet \nabla^g_{k_i,l_i}$. We start by giving some intuition. Let Y_i be the set of variables occurring in $\vec{y}_i$ and suppose that $\bigwedge\{\alpha_i \bullet \nabla^g(\vec{x}_i, Y_i) \mid 1 \le i \le n\}$ is true at a point w. Hence, for all $1 \le i \le n$, there exists a tuple $\vec{w}_i$ of ∇^g-witnesses for $(\vec{x}_i, Y_i)$. We define the set of witnesses as the set of points that occurs in one of the $\vec{w}_i$s. Our goal is to push the conjunctions "inside" the ∇ operators. The way we proceed depends on how the witnesses are distributed. Or to put it differently, how the witnesses for distinct pairs $(\vec{x}_i, Y_i)$ and $(\vec{x}_j, Y_j)$ overlap.

In order to encode this information, we define for each witness u a n-tuple $h(u) = (z_1, \dots, z_n)$ in the following way. Take $i_0 \in \{1, \dots, n\}$. If u is not a witness for $(\vec{x}_{i_0}, Y_{i_0})$, then we define z_{i_0} as $*$. Otherwise, there is a unique variable x occurring in $\vec{x}_{i_0}$ such that u is the witness for $(\vec{x}_{i_0}, Y_{i_0})$ associated with x. In this case, we define z_{i_0} as x.

If we define X_i as the variables occurring in $\vec{x}_i$ and X_i^* as the set $X_i \cup \{*\}$, then the set consisting of all the tuples $h(u)$ (with u being a witness) defines a relation $S \subseteq X_1^* \times \dots \times X_n^*$. The characteristic of that relation is that each variable occurring in one of the $\vec{x}_i$s occurs exactly in one tuple of S.

Now if we consider a successor v of w, which formulas are true at v? If v is not a witness, then for all $1 \leq i \leq n$, the formula $\bigvee Y_i$ is true at v. So if we define φ by:

$$\varphi = \bigwedge \{ \bigvee Y_i \mid 1 \leq i \leq n \},$$

φ is true at v. Next if v is a witness, then the formulas true at v is determined by the tuple $h(v) = (z_1, \dots, z_n)$. If $z_i = *$, then $\bigvee Y_i$ is true at v, whereas if $z_i = x$, x is true at v. So if for all $z \in X_i^*$, we define $\bar{z}$ by:

$$\bar{z} = \begin{cases} z & \text{if } z \in X_i, \\ \bigvee Y_i & \text{otherwise,} \end{cases}$$

this means that $\varphi_{h(v)} := \bigwedge \{ \bar{z}_i \mid 1 \leq i \leq n \}$ is true at v. We can think of v as being a witness for the formula $\varphi_{h(v)}$. In fact, we even have something stronger. If the set of witnesses is equal to $\{u_1, \dots, u_m\}$, we will show that the formula

$$\nabla^g ((\varphi_{h(u_1)}, \dots, \varphi_{h(u_m)}), \varphi),$$

is true at w. Note that all the conjunctions of this formula occurs "inside" the ∇ operator. Conversely, the truth of this formula at w ensures that for all $1 \leq i \leq n$, $\nabla^g(\vec{x}_i, Y_i)$ is true at w. Formally, we have the following claim.

1. Claim. With a tuple $\vec{z} = (z_1, \dots, z_n)$, we associate the formula $\varphi_{\vec{z}}$ given by

$$\varphi_{\vec{z}} = \bigwedge \{ \bar{z}_i \mid 1 \leq i \leq n \},$$

where $\bar{z}_i$ is defined as in the previous paragraph. Next, given a relation $S \subseteq X_1^* \times \dots \times X_n^*$, we write k_S for the size of S. If $S = \{\vec{z}_1, \dots, \vec{z}_m\}$, we define $\vec{\varphi}_S$ as a tuple of formulas

$$(\varphi_{\vec{z}_1}, \dots, \varphi_{\vec{z}_m}).$$

A relation $S \subseteq X_1^* \times \dots \times X_n^*$ is a *relevant distribution* if for all $1 \leq i \leq n$, for all $x \in X_i$, there is a exactly one tuple $(z_1, \dots, z_n) \in S$ such that $z_i = x$. Finally, we define α as the set $\bigcup \{ \alpha_i \mid 1 \leq i \leq n \}$.

With these definition, the equation

$$\bigwedge\{\alpha_i \bullet \nabla^g_{k_i,l_i}(\vec{x}_i,\vec{y}_i) \mid 1 \leq i \leq n\}$$
$$= \bigvee\{\alpha \bullet \nabla^g_{k_S,1}(\vec{\varphi}_S,\varphi) \mid S \text{ is a relevant distribution}\}$$

holds on $\mathcal{L}_{\mathrm{ML}}$.

Proof of Claim For all $1 \leq i \leq n$, we define γ_i as the formula $\alpha_i \bullet \nabla^g_{k_i,l_i}(\vec{x}_i,\vec{y}_i)$. We also define ψ as the formula

$$\bigvee\{\alpha \bullet \nabla^g_{k_S,1}(\vec{\varphi}_S,\varphi) \mid S \text{ is a relevant distribution}\}.$$

Fix a Kripke model $\mathcal{M} = (W, R, V)$, its associated fixpoint algebra $\mathbb{P}_\mathcal{M}$ and an assignment $\tau : Var \longrightarrow \mathcal{P}(W)$. We have to show that

$$[\![\gamma_1 \wedge \cdots \wedge \gamma_n]\!]_{\mathbb{P}_{\mathcal{M},\tau}} = [\![\psi]\!]_{\mathbb{P}_{\mathcal{M},\tau}}. \tag{4.5}$$

First we prove the inclusion from right to left of (4.5). Suppose that a point w belongs to $[\![\psi]\!]_{\mathbb{P}_{\mathcal{M},\tau}}$. Hence, there is a relevant distribution $S \subseteq X_1^* \times \cdots \times X_n^*$ such that the formula $\bigwedge \alpha \bullet \nabla^g(\vec{\varphi}_S,\{\varphi\})$ is true at w. Assume that $S = \{\vec{z}_j \mid 1 \leq j \leq k_S\}$. We have to show that for all $1 \leq i \leq n$, w belongs $[\![\gamma_i]\!]_{\mathbb{P}_{\mathcal{M},\tau}}$. Let i be an element in $\{1,\ldots,n\}$ and let us prove that $\bigwedge \alpha_i \bullet \nabla^g(\vec{x}_i, Y_i)$ is true at w.

Since $\bigwedge \alpha$ is true at w, it is immediate that $\bigwedge \alpha_i$ is true at w. So it remains to show that $\nabla^g(\vec{x}_i, Y_i)$ is true at w. Recall that for all $x \in X_i$, there is a exactly one tuple $(z_1,\ldots,z_n) \in S$ such that $z_i = x$. So we can define an injective map $c : X_i \longrightarrow S$ such that for all variable $x \in X_i$,

$$c(x) = (z_1,\ldots,z_n) \quad \text{iff} \quad z_i = x.$$

As $\nabla^g(\vec{\varphi}_S,\{\varphi\})$ is true at w, there is a tuple $\vec{w}_S = (w_1,\ldots,w_{k_S})$ such that

- for all $1 \leq j \leq k_s$, $\varphi_{\vec{z}_j}$ is true at w_j,
- for all $1 \leq j < j' \leq k_s$, $w_j \neq w_{j'}$,
- for all $u \in R[w]\backslash W_S$, φ is true at u,

where W_S is the set $\{w_1,\ldots,w_{k_S}\}$. We let $d : S \longrightarrow W_S$ be the map such that for all $j \in \{1,\ldots,k_S\}$

$$d(\vec{z}_j) = w_j.$$

It easily follows from the properties of the w_js that d is a bijection such that for all $\vec{z} \in S$, $\varphi_{\vec{z}}$ is true at $d(\vec{z})$. If $X_i = \{x_{i1},\ldots,x_{ik_i}\}$, we define $\vec{w}_i$ as the tuple

$$(d(c(x_{i1})),\ldots,d(c(x_{ik_i}))).$$

We show that $\vec{w}_i$ is a tuple of ∇^g-witness for $(\vec{x}_i, Y_i)$. That is,

(i) for all distinct variables x and x' in X_i, $d(c(x)) \neq d(c(x'))$,

(ii) for all x in X_i, x is true at $d(c(x_{ij}))$,

(iii) for all successors u of w such that u does not belong to W_i, $\bigvee Y_i$ is true at u.

where W_i is the set $\{d(c(x)) \mid x \in X_i\}$.

The first item follows from the fact that c and d are injective map. For (ii), let x be a variable in X_i and let $\vec{z} = (z_1, \dots, z_n)$ be the tuple $c(x)$. Recall that $\varphi_{\vec{z}}$ is true at $d(\vec{z})$. In particular, if $z_i \neq *$, z_i is true at $d(\vec{z})$. It follows from the definition of c that $z_i = x$. So x is true at $d(\vec{z}) = d(c(x))$.

It remains to show (iii). That is, for all successors u of w such that u does not belong to W_i, $\bigvee Y_i$ is true at u. Let u be such a successor of w. In case u does not belong to W_S, then φ is true at u. In particular, $\bigvee Y_i$ is true at u. So we may assume that u belongs to W_S. Hence, there is $\vec{z} = (z_1, \dots, z_n)$ in S such that $u = d(\vec{z})$. Now we show that $z_i = *$. Suppose for contradiction that $z_i \neq *$. Then z_i is mapped by c to the unique tuple $(z'_1, \dots, z'_n) \in S$ such that $z'_i = z_i$. Since $(z_1, \dots, z_n)$ belongs to S, $c(z_i)$ is nothing but the tuple $(z_1, \dots, z_n)$. It follows that $d(\vec{z})$ $=u$ is equal to $d(c(z_i))$, which belongs to W_i. Hence, u belongs to W_i, which is a contradiction. Thus, $z_i = *$. Recall that $\varphi_{\vec{z}}$ is true at $d(\vec{z})$. Putting this together with the fact that $z_i = *$, we obtain that $\bigvee Y_i$ is true at $d(\vec{z})$. That is, $\bigvee Y_i$ is true at u and this finishes the proof of the inclusion from right to left.

Next we show the inclusion from left to right of (4.5). Suppose that w belongs to $[\![\gamma_1 \wedge \dots \wedge \gamma_n]\!]_{\mathbb{P}_{\mathcal{M},\tau}}$. Take $i \in \{1, \dots, n\}$ and recall that $X_i = \{x_{ij} \mid 1 \leq j \leq k_i\}$. Since w belongs to $[\![\gamma_i]\!]_{\mathbb{P}_{\mathcal{M},\tau}}$, we have $\mathcal{M}, w \Vdash_\tau \nabla^g(\vec{x}_i, Y_i)$. So there exist successors $w_{i1}, \dots, w_{ik_i}$ of w such that

(a) for all $1 \leq j < j' \leq k_i$, x_{ij} is true at w_{ij}

(b) for all $1 \leq j < j' \leq k_i$, $w_{ij} \neq w_{ij'}$,

(c) for all $v \in R[w] \backslash W_i\}$, $\bigvee Y_i$ is true at v.

where W_i is the set $\{w_{ij} \mid 1 \leq j \leq k_i\}$.

We start by fixing some notation. We define U as the set $\bigcup\{W_i \mid 1 \leq i \leq n\}$. Next for each $u \in U$, we define a n-tuple $h(u) = (z_1, \dots, z_n)$ such that for all $1 \leq i \leq n$,

$$z_i := \begin{cases} x_{ij} & \text{if for some } x_{ij} \text{ in } X,\ u = w_{ij}, \\ * & \text{otherwise.} \end{cases}$$

The n-tuple $h(u)$ is well-defined since for all distinct variables x_{ij} and $x_{ij'}$ in X_i, we have $w_{ij} \neq w_{ij'}$. We define S as the relation $\{h(u) \mid u \in U\}$.

First we show that S is a relevant distribution. Take $i \in \{1, \dots, n\}$ and let x_{ij} be a variable in X_i. We have to show that there is a exactly one tuple

$(z_1, \dots, z_n) \in S$ such that $z_i = x_{ij}$. It follows from the definitions of S and h that x_{ij} occurs in the tuple $h(w_{ij})$. Next if x_{ij} occurs in a tuple $h(u) = (z_1, \dots, z_n)$ in S, then by definition of h, u is equal to w_{ij}. Hence, $h(w_{ij})$ is the unique tuple in S in which x_{ij} occurs at position i. This finishes the proof that S is a relevant distribution.

So in order to show that $\mathcal{M}, w \Vdash_\tau \psi$, we can restrict ourselves to prove that w belongs to $[\![\alpha \bullet \nabla^g_{k_S,1}(\vec{\varphi}_S,\ \varphi)]\!]_{\mathbb{P}_\mathcal{M}}$. That is,

$$\bigwedge \alpha \bullet \nabla^g(\vec{\varphi}_S, \{\varphi\}) \text{ is true at } w. \tag{4.6}$$

First we show that $\bigwedge \alpha$ is true at W. By definition of α, it is sufficient to show that for all $1 \leq i \leq n$, $\bigwedge \alpha_i$ is true at w. This follows immediately from the fact that w belongs to $[\![\gamma_i]\!]_{\mathbb{P}_{\mathcal{M},\tau}}$.

Next we prove that for all $\vec{z} \in S$, there is a unique $u \in U$ such that $h(u) = \vec{z}$. Let u, u' be distinct elements of U and let $(z_1, \dots, z_n)$ and $(z'_1, \dots, z'_n)$ be the tuples $h(u)$ and $h(u')$ respectively. We have to prove that $h(u) \neq h(u')$. Since u belongs to U, there exists $i \in \{1, \dots, n\}$ such that u belongs to W_i. So there is $x_{ij} \in X_i$ such that $u = w_{ij}$. By definition of h, z_i is equal to x_{ij}. If u' does not belong to W_i, then $z_i = *$ and in particular, $h(u) \neq h(u')$. Suppose next that u' belongs to W_i. Hence, there exists $x_{ij'} \in X_i$ such that $u' = w_{ij'}$. Again, by definition of h, this implies that $z'_i = x_{ij'}$. Since $u \neq u'$, $u = w_{ij}$ and $u' = w_{ij'}$, we have that $j \neq j'$ and hence, $x_{ij} \neq x_{ij'}$. It follows that $z_i \neq z'_i$, which implies that $h(u) \neq h(u')$. This finishes the proof that for all $\vec{z} \in S$, there is a unique u such that $h(u) = \vec{z}$.

This means that we can define a function b between the relation S and the set $R[w]$ such that for all $\vec{z} \in S$, $b(\vec{z})$ is the unique u satisfying $\vec{z} = h(u)$. It follows from the definition of b that $h(b(\vec{z})) = \vec{z}$ and $b(h(u)) = u$ (for all $\vec{z} \in S$ and $u \in U$). As a corollary, b is a bijection.

If $S = \{\vec{z}_1, \dots, \vec{z}_m\}$ and $\vec{\varphi}_S = (\varphi_{\vec{z}_1}, \dots, \varphi_{\vec{z}_m})$, we let $\vec{w}_S$ be the tuple

$$(b(\vec{z}_1), \dots, b(\vec{z}_m)).$$

We define W_S as the set $\{b(\vec{z}_i) \mid 1 \leq i \leq k_s\}$. We show that $\vec{w}_S$ is a tuple of ∇^g-witnesses for the pair $(\vec{\varphi}_S, \varphi)$. Since b is an injective map, we have that for all $i \neq j$, $b(\vec{z}_i) \neq b(\vec{z}_j)$. Hence, it remains to show the two following conditions:

(i) for all $\vec{z} \in S$, the formula $\varphi_{\vec{z}}$ is true at $b(\vec{z})$,

(ii) for all successors v of w which do not occur in $\vec{w}_S$, φ is true at v.

We start to prove that for all $\vec{z} \in S$, $\varphi_{\vec{z}}$ is true at $b(\vec{z})$. Suppose that $\vec{z} = (z_1, \dots, z_n)$. We have to show that for all $i \in \{1, \dots, n\}$, $\bar{z}_i$ is true at $b(\vec{z})$. Take $i \in \{1, \dots, n\}$. First, suppose that $z_i \neq *$. Hence, $\bar{z}_i$ is equal to z_i. Recall that for all $u \in U$, if $h(u) = (z_1, \dots, z_n)$ and $z_i \neq *$, then $u = w_{ij}$, where j is the unique natural number in $\{1, \ldots, k_i\}$ such that $z_i = x_{ij}$. Putting that together with $h(b(\vec{z})) = (z_1, \ldots, z_n)$, we obtain that $b(\vec{z}) = w_{ij}$, where j is the unique natural number in $\{1, \ldots, k_i\}$ such that $z_i = x_{ij}$. Now x_{ij} is true at w_{ij}. It follows that z_i $(= x_{ij})$ is true at $b(\vec{z})$ $(= w_{ij})$.

It remains to consider the case when $z_i = *$. This means that $\bar{z}_i = \bigvee Y_i$. So we have to show that $\bigvee Y_i$ is true at $b(\vec{z})$. We know that for all $u \in U$ such that $h(u) = (z_1, \ldots, z_n)$ and $z_i = *$, u does not belong to W_i (if u belongs to W_i, then $u = w_{ij}$ for some j, which implies that $z_i = x_{ij}$). Using this together with the fact that $h(b(\vec{z})) = (z_1, \ldots, z_n)$, we have that $b(\vec{z})$ does not belong to W_i. It follows from condition (c) that $\bigvee Y_i$ is true at u. In particular, $\bigvee Y_i$ is true at $b(\vec{z})$. This finishes the proof of (i).

Now we show that (ii) holds. Let v_0 be a successor of w that does not belong to W_S. We prove that φ is true at v_0. Recall that φ is the formula given by:

$$\varphi = \bigwedge\{\bigvee Y_i \mid 1 \leq i \leq n\}.$$

Take $i \in \{1, \ldots, n\}$. We have to show that $\bigvee Y_i$ is true at v_0. By condition (c), it is sufficient to prove that v_0 does not belong to W_i. If it is not the case, then v_0 belongs to U and hence, $h(v_0)$ belongs to S. It follows from the definition of W_S that $b(h(v_0))$ belongs to W_S. Recall that for all $u \in U$, $b(h(u)) = u$. Hence, v_0 $=b(h(v_0))$ belongs to W_S, which is a contradiction. This finishes the proof that $\alpha \bullet \nabla^g(\vec{\varphi}_S, \{\varphi\})$ is true at w. Hence, the proof of the inclusion from left to right. ◀

Now recall that the meet distributes over the join. That is, for all sets $Y_1, \ldots, Y_n$ of variables, the equation

$$\bigwedge\{\bigvee Y_i \mid 1 \leq i \leq n\} = \bigvee\{\bigwedge Y \mid \text{ for all } 1 \leq i \leq n,\, Y \cap Y_i \neq \emptyset\}$$

holds on $\mathcal{L}_{\mathrm{ML}}$. Putting this together with the claim above, we obtain that an equation of the form of (4.4) holds on $\mathcal{L}_{\mathrm{ML}}$, in case all the f_i's are operator of the form $\alpha_i \bullet \nabla_{k_i,l_i}$. This finishes the proof. □

4.3 Preservation of MSO under p-morphic images

This section corresponds to arrow (4) in Figure 4.1. We introduce the operator ∇'. As mentioned earlier, the logic associated with this operator is in between the μ-calculus and the graded μ-calculus. We show in this section that on trees, it corresponds to the fragment of the μ-calculus that is preserved under p-morphic images on trees.

The logic $\mu\mathrm{ML}^{\nabla'}$ The set $\mu\mathrm{ML}^{\nabla'}$ of ∇'-formulas of the graded μ-calculus is given by:

$$\varphi ::= x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \alpha \bullet \nabla'(\vec{\varphi}, \Psi) \mid \mu x.\varphi \mid \nu x.\varphi,$$

where $x \in \mathit{Var}$, α is a conjunction of literals over *Prop*, $\vec{\varphi}$ is a tuple of formulas and Ψ is a finite set of formulas.

Given a formula φ, a Kripke model $\mathcal{M} = (W, R, V)$, an assignment $\tau : \mathit{Var} \to \mathcal{P}(W)$ and a point $w \in W$, the relation $\mathcal{M}, w \Vdash_\tau \varphi$ is defined by induction as in the case of the μ-calculus with the extra condition:

$$\begin{array}{lll} \mathcal{M}, w \Vdash_\tau \alpha \bullet \nabla'(\vec{\varphi}, \Psi) & \text{iff} & \mathcal{M}, w \Vdash_\tau \alpha \text{ and } \mathcal{M}, w \Vdash_\tau \nabla'(\vec{\varphi}, \Psi), \\ \mathcal{M}, w \Vdash_\tau \nabla'(\vec{\varphi}, \Psi) & \text{iff} & \text{there exists a tuple } (w_1, \dots w_k) \text{ over } R[w] \text{ such that,} \\ & & \quad 1.\ \text{for all } 1 \leq i \leq k,\ \mathcal{M}, w_i \Vdash_\tau \varphi_i, \\ & & \quad 2.\ \text{for all } u \text{ in } R[w] \backslash \{w_i \mid 1 \leq i \leq k\}, \\ & & \quad \mathcal{M}, u \Vdash_\tau \bigvee \Psi. \end{array}$$

where $\vec{\varphi} = (\varphi_1, \dots, \varphi_k)$. The tuple $(w_1, \dots, w_k)$ is a tuple of ∇'-*witnesses* for the pair $(\vec{\varphi}, \Psi)$.

A map $m : \mu\mathrm{ML}^{\nabla'} \to \mathcal{P}(R[w])$ is a ∇'-*marking* for $(\vec{\varphi}, \Psi)$ if there exists a tuple $(w_1, \dots, w_k)$ such that for all $1 \leq i \leq k$, $w_i \in m(\varphi_i)$ and for all successors u of w such that $u \notin \{w_i \mid 1 \leq i \leq k\}$, there is $\psi \in \Psi$ such that $u \in m(\psi)$.

A sentence φ in $\mu\mathrm{ML}^{\nabla'}$ is *equivalent on trees* to an MSO formula ψ if for all trees over *Prop* $\mathcal{T}$, ψ is valid on $\mathcal{T}$ iff φ is true at the root of $\mathcal{T}$. A formula φ in $\mu\mathrm{ML}^{\nabla'}$ is *equivalent on trees* to a formula ψ in $\mu\mathrm{GL}^{\nabla}$ if for all trees $\mathcal{T}$ over *Prop* with root r and for all assignments $\tau : \mathit{Var} \to \mathcal{P}(\mathcal{T})$, we have $\mathcal{T}, r \Vdash_\tau \varphi$ iff $\mathcal{T}, r \Vdash_\tau \psi$.

The set of formulas in $\mu\mathrm{ML}^{\nabla'}$ in *disjunctive normal form* is defined by induction in the following way:

$$\varphi ::= x \mid \varphi \vee \varphi \mid \alpha \bullet \nabla'(\vec{\varphi}; \Psi) \mid \mu x.\varphi \mid \nu x.\varphi,$$

where $x \in \mathit{Var}$, α is a conjunction of literals over *Prop*, Ψ is a finite set of formulas and $\vec{\varphi}$ is a finite tuple of formulas.

The main difference between the semantics for the operator ∇' and the semantics for ∇^g is that the points occurring in a tuple of ∇'-witnesses for a pair $(\vec{\varphi}, \Psi)$ might not be distinct. A typical example of property that we can express with $\mu\mathrm{GL}^{\nabla}$, but not with $\mu\mathrm{ML}^{\nabla'}$, is the existence of k successors satisfying a formula φ (where k is a natural number strictly greater than 1 and φ is a given formula).

We also introduce a game semantics for the languages $\mu\mathrm{GL}^{\nabla}$ and $\mu\mathrm{ML}^{\nabla'}$. The fact that the existence of a winning strategy for a player in the game corresponds to the truth of a formula at a given point, is proved using classical methods (as in the case of μ-calculus, see for instance [EJ91]).

Game semantics Let φ be a formula in $\mu\mathrm{GL}^{\nabla} \cup \mu\mathrm{ML}^{\nabla'}$ such that each variable in φ is bound. Without loss of generality, we may assume that for all $x \in \mathit{Var}$ which occurs in φ, there is a unique subformula of φ, which is of the form $\eta x.\delta_x$, where $\eta \in \{\mu, \nu\}$. We also fix a model $\mathcal{M} = (W, R, V)$. We define the *evaluation game* $\mathcal{E}(\mathcal{M}, \varphi)$ as a graph game between two players, $\forall$ and $\exists$. The rules of the game are given in the table below.

Position	Player	Possible moves
$(w, \top)$	$\forall$	$\emptyset$
(w, x)	-	$\{(w, \delta_x)\}$
$(w, \varphi_1 \wedge \varphi_2)$	$\forall$	$\{(w, \varphi_1), (w, \varphi_2)\}$
$(w, \varphi_1 \vee \varphi_2)$	$\exists$	$\{(w, \varphi_1), (w, \varphi_2)\}$
$(w, \eta x.\psi)$	-	$\{(w, \psi)\}$
$(w, \alpha \bullet \nabla^g(\vec{\varphi}, \Psi))$	$\forall$	$\{\alpha, \nabla^g(\vec{\varphi}, \Psi)\}$
$(w, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$	$\forall$	$\{\alpha, \nabla'(\vec{\varphi}, \Psi)\}$
$(w, \nabla^g(\vec{\varphi}, \Psi))$	$\exists$	$\{m : \mu\mathrm{GL}^{\nabla} \to \mathcal{P}(R[w]) \mid m$ is a ∇^g-marking for $(\vec{\varphi}, \Psi)\}$
$(w, \nabla'(\vec{\varphi}, \Psi))$	$\exists$	$\{m : \mu\mathrm{ML}^{\nabla'} \to \mathcal{P}(R[w]) \mid m$ is a ∇'-marking for $(\vec{\varphi}, \Psi)\}$
$m : \mu\mathrm{GL}^{\nabla} \to \mathcal{P}(R[w])$	$\forall$	$\{(u, \psi) \mid u \in m(\psi)\}$
$m : \mu\mathrm{ML}^{\nabla'} \to \mathcal{P}(R[w])$	$\forall$	$\{(u, \psi) \mid u \in m(\psi)\}$

where w belongs to W, x belongs to Var, η belongs to $\{\mu, \nu\}$, φ_1, φ_2 and ψ belongs to $\mu\mathrm{GL}^{\nabla} \cup \mu\mathrm{ML}^{\nabla'}$, α is a conjunction of literals, Ψ is a finite subset of $\mu\mathrm{GL}^{\nabla} \cup \mu\mathrm{GL}^{\nabla'}$, $\vec{\varphi}$ is a tuple of formulas in $\mu\mathrm{GL}^{\nabla} \cup \mu\mathrm{ML}^{\nabla'}$.

If a match is finite, the player who gets stuck, loses. If a match ρ s infinite, we let $\mathit{Inf}(\rho)$ be the set of variables x such that there are infinitely many positions of the form (w, x) in the match. There must be a variable x_0 in $\mathit{Inf}(\rho)$ such that for all variables $x \in \mathit{Inf}$, δ_x is a subformula of δ_{x_0}. If x_0 is bound by a μ-operator, then $\forall$ wins. Otherwise $\exists$ wins.

4.3.1. Proposition. *Let φ be a formula in $\mu\mathrm{GL}^{\nabla} \cup \mu\mathrm{ML}^{\nabla'}$. For all Kripke models $\mathcal{M} = (W, R, V)$ and all $w \in W$, $\mathcal{M}, w \Vdash \varphi$ iff $\exists$ has a winning strategy in the game $\mathcal{E}(\mathcal{M}, \varphi)$ with starting position (w, φ).*

We are now ready to show that modulo equivalence on trees, the MSO formulas preserved under p-morphic images are exactly the formulas in $\mu\mathrm{ML}^{\nabla'}$. This will take care of the arrow (4) in Figure 4.1.

4.3.2. Proposition. *For all sentences φ in $\mu\mathrm{ML}^{\nabla'}$, we can compute an* MSO *formula that is equivalent on trees to φ and that is preserved under p-morphic images on trees.*

Moreover, given an MSO *formula φ, we can compute a disjunctive sentence $\psi \in \mu\mathrm{ML}^{\nabla'}$ such that φ and ψ are equivalent on trees iff φ is preserved under p-morphic images on trees.*

Proof It is routine to check that a formula in $\mu\mathrm{ML}^{\nabla'}$ is equivalent on trees to an MSO formula preserved under p-morphic images on trees. Next, we show that an MSO formula that is preserved under p-morphic images on trees is equivalent on trees to a disjunctive formula in $\mu\mathrm{ML}^{\nabla'}$. Let φ be an MSO formula that is preserved under p-morphic images on trees. By Theorem 4.2.1, there is a graded μ-formula γ such that for all trees $\mathcal{T}$ over *Prop* with root r,

$$\mathcal{T} \vDash \varphi \quad \text{iff} \quad \mathcal{T}, r \Vdash \gamma.$$

By Proposition 4.2.4, we may assume the formula γ to be in disjunctive normal form.

Now let χ be the formula γ in which we replace each operator ∇^g by ∇'. We show that under the assumption that φ is preserved under p-morphic images on trees, γ and χ are equivalent on trees. Using Proposition 4.3.1 together with the fact that a ∇^g-marking is a ∇'-marking, it is easy to check that for all trees $\mathcal{T}$ over *Prop* with root r and all assignments $\tau : \mathit{Var} \longrightarrow \mathcal{P}(\mathcal{T})$, we have

$$\mathcal{T}, r \Vdash_\tau \gamma \quad \text{implies} \quad \mathcal{T}, r \Vdash_\tau \chi.$$

For the other direction, let $\mathcal{T} = (T, R, V)$ be a tree over *Prop* with root r and suppose that χ is true at r under an assignment $\tau : \mathit{Var} \longrightarrow \mathcal{P}(T)$. We have to show that γ is true at r. Since χ is true at r, $\exists$ has a winning strategy h in the evaluation game with starting position (r, χ). We say that a position (u, φ) is *h-reachable* if there is an h-conform match during which (u, φ) occurs. A node u is *h-reachable* if there is a formula φ such that (u, φ) is h-reachable.

We start by giving some intuition. The difference between a ∇'-marking and a ∇^g-marking is that the ∇'-witnesses might not be pairwise distinct. Hence, in order to use h as a strategy for the evaluation game associated with γ, we are going to expand the tree $\mathcal{T}$. More precisely, if u is a node and if a map $m : \mu\mathrm{ML}^{\nabla'} \longrightarrow \mathcal{P}(R[u])$ is a ∇'-marking associated with a pair $(\vec{\varphi}, \Psi)$, then by "making copies" of certain successors of u, we can transform this ∇'-marking into a ∇^g-marking. Now the choice of the successors that are copied (and the number of copies) is determined by m and the pair $(\vec{\varphi}, \Psi)$. So in case there is more than one formula of the form $\alpha \bullet \nabla'(\vec{\varphi}, \Psi)$ associated with u (in the sense that the pair $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$ is h-reachable), we are stuck. However, since the formula γ is disjunctive, we may assume that for all nodes u, there is at most one such formula associated with u.

Before defining the "expansion" of $\mathcal{T}$, we fix some notation. We denote by $\mathbb{N}^*$ the set of finite sequences over $\mathbb{N}$. In particular, the empty sequence ϵ belongs to $\mathbb{N}^*$. If ψ is a formula in $\mu\mathrm{ML}^{\nabla'}$, we write ψ^g for the formula obtained by replacing ∇' by ∇^g in ψ. If $\vec{\varphi}$ is a tuple $(\varphi_1, \dots, \varphi_k)$ of formulas in $\mu\mathrm{ML}^{\nabla'}$, we write $(\vec{\varphi})^g$ for the tuple $(\varphi_1^g, \dots, \varphi_k^g)$. Similarly, if Ψ is set of formulas, we let Ψ^g be the set $\{\psi^g \mid \psi \in \Psi\}$.

Now we define a new tree $\mathcal{T}' = (T', R', V')$ over *Prop* such that

- the root of $\mathcal{T}'$ is (r, ϵ) and T' is a subset of $T \times \mathbb{N}^*$,
- the child relation is such that for all $(u_1, (n_1, \dots, n_k))$ and $(u_2, \vec{m})$ in $\mathcal{T}'$,

$$(u_1, (n_1, \dots, n_k))R'(u_2, \vec{m}) \quad \text{iff} \quad \begin{array}{l} u_1 R u_2 \text{ and there is } n_{k+1} \in \mathbb{N} \\ \text{such that } \vec{m} = (n_1, \dots, n_k, n_{k+1}). \end{array}$$

  Thus, the depth of a node $(u_1, (n_1, \dots, n_k))$ in $\mathcal{T}'$ is $k + 1$.

- for all $p \in Prop$ and for all $(u, \vec{n}) \in T'$, $(u, \vec{n})$ belongs to $V'(p)$ iff u belongs to $V(p)$.

Moreover, we define a positional strategy h' for the evaluation game $\mathcal{E}(\mathcal{T}', \gamma)$ with starting position $((r, \epsilon), \gamma)$ that satisfies the two following conditions:

(a) each point in T' is h'-reachable and h' is scattered (see Section 2.6 of Chapter 2),

(b) for all h'-conform matches $((u_0, \vec{n}_0), (\varphi_0)^g) \dots ((u_m, \vec{n}_\kappa), (\varphi_\kappa)^g)$, the match $(u_0, \varphi_0) \dots (u_\kappa, \varphi_\kappa)$ is h-conform,

where the definition of h'-reachability is a straightforward adaption of the definition of h-reachability.

The definitions of $\mathcal{T}'$ and h' are by induction. More precisely, at stage i of the induction, we specify what the nodes of $\mathcal{T}'$ of depth i are. We also define h' for all positions of the form $((u, \vec{n}), \varphi)$, where the depth of $(u, \vec{n})$ in $\mathcal{T}'$ is at most $i - 1$.

For the basic case, the only node of depth 1 in $\mathcal{T}'$ is the node (r, ϵ). For the induction step, take $i_0 \geq 1$ and suppose that we already know what the nodes in $\mathcal{T}'$ of depth at most i_0 are. We have also defined the strategy h' for all positions of the form $((u, \vec{n}), \varphi)$, where the depth of $(u, \vec{n})$ in $\mathcal{T}'$ is at most $i_0 - 1$. Let $(u, \vec{n}) = (u, (n_1, \dots, n_{i_0 - 1}))$ be a node in $\mathcal{T}'$ of depth i_0. We are going to define the set of children of this point and $\exists$'s move when a position of the form $((u, \vec{n}), \varphi)$ is reached.

By induction hypothesis, $(u, \vec{n})$ is h'-reachable. Hence, there exists an h'-conform match π' the last position of which is of the form $((u, \vec{n}), \varphi^g)$. If $\vec{n} = \epsilon$, we may assume that $\varphi = \chi$ and $\varphi^g = \gamma$. We know (by the induction hypothesis (b) if $i_0 > 1$ or trivially if $i_0 = 1$) that there is an h-conform match π the last position of which is (u, φ).

Now, there are two different possibilities depending on the shape of φ. First, suppose that φ is a disjunction $\varphi_1 \vee \varphi_2$. Then, in the h-conform match π, the position following (u, φ) is of the form (u, ψ), where ψ is either φ_1 or φ_2. We define h' such that the position following $((u, \vec{n}), \varphi^g)$ is $((u, \vec{n}), \psi^g)$.

Second, suppose that φ^g is of the form $\alpha \bullet \nabla^g((\vec{\varphi})^g, \Psi^g)$, with $\vec{\varphi} = (\varphi_1, \dots, \varphi_k)$. Then, in the h-conform match π, the position following (u, φ) is a marking

$$m : \mu\mathrm{ML}^{\nabla'} \to \mathcal{P}(R[u])$$

such that m is a ∇'-marking for $(\vec{\varphi}, \Psi)$. First, we define what the children of $(u, \vec{n})$ in $\mathcal{T}'$ are. Second, in the h'-conform match π', we give a ∇^g-marking $m^g : \mu\mathrm{GL}^\nabla \longrightarrow \mathcal{P}(R'[(u, \vec{n})])$ for the pair $((\vec{\varphi})^g, \Psi^g)$.

Since m is a ∇'-marking for $((\vec{\varphi}), \Psi)$, there exists a tuple $(u_1, \dots, u_k)$ such that the two following conditions holds. For all $i \in \{1, \dots, k\}$, u_i is a child of u at which φ_i is true. For all $v \in R[u]\backslash\{u_i \mid 1 \leq i \leq k\}$, $\bigvee \Psi$ is true at v. Hence, for each such a v, we can fix an arbitrary formula ψ_v such that $\psi_v \in \Psi$ and $v \in m(\psi_v)$. Next we fix a tuple $(u'_1, \dots, u'_{k'})$ such that

$$\{u'_i \mid 1 \leq i \leq k'\} = \{u_i \mid 1 \leq i \leq k\}$$

and for all $1 \leq i < j \leq k'$, we have $u'_i \neq u'_j$. For all $i \in \{1, \dots, k'\}$, we define $r_i + 1$ as the size of the set $\{j \mid u_j = u'_i\}$ and we fix an arbitrary bijection f_i from $\{0, \dots, r_i\}$ to $\{j \mid u_j = u'_i\}$. We define U as the subset of $T \times N^*$ given by:

$$\{(u'_i, (n_1, \dots, n_{i_0-1}, l)) \mid i \in \{1, \dots, k'\}, 0 \leq l \leq r_i\}$$

and W by:

$$\{(v, (n_1, \dots, n_{i_0-1}, 0)) \mid v \in R[u]\backslash\{u_1, \dots, u_k\}\}.$$

Finally, we define the set of children $(u, \vec{n})$ in $\mathcal{T}'$ as the set $U \cup W$. Recall that the (partially defined) strategy h' is scattered. Hence, there is a unique formula of the form $\alpha \bullet \nabla'(\vec{\varphi}, \Psi)$ such that $((u, \vec{n}), \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$ is h'-reachable. This means that the set of children $(u, \vec{n})$ in $\mathcal{T}'$ is well-defined.

We are now going to define a ∇^g-marking $m^g : \mu\mathrm{GL}^\nabla \longrightarrow \mathcal{P}(R'[(u, \vec{n})])$ for the pair $((\vec{\varphi})^g, \Psi^g)$. We start by defining a map $f : U \longrightarrow \{1, \dots, k\}$ in the following way. If a point w belongs to U, there exist $i \in \{1, \dots, k'\}$ and $l \in \{0, \dots, r_i\}$ such that

$$w = (u'_i, (n_1, \dots, n_{i_0-1}, l)).$$

We define $f(w)$ as the natural number $f_i(l)$. Next we define a map $g : U \cup W \longrightarrow \mu\mathrm{ML}^{\nabla'}$ such that

$$\begin{cases} g((v, (n_1, \dots, n_{i_0-1}, 0))) & = \psi_v, \\ g(w) & = \varphi_{f(w)}, \end{cases}$$

where $v \in R[u]\backslash\{u_i \mid 1 \leq i \leq k\}$ and $w \in U$. We let the marking $m^g : \mu\mathrm{GL}^\nabla \longrightarrow \mathcal{P}(R'[(u, \vec{n})])$ be such that for all $w \in U \cup W$ and for all formulas $\delta \in \mu\mathrm{ML}^{\nabla'}$, we have

$$w \in m^g(\delta^g) \quad \text{iff} \quad \delta = g(w).$$

We show that m^g is a ∇^g-marking. That is, we have to define points $v_1, \dots, v_k$ such that

(i) for all $1 \leq i < j \leq k$, $v_i \neq v_j$,

(ii) for all $1 \leq j \leq k$, v_j belongs to $m^g((\varphi_j)^g)$,

(iii) for all children w of $(u, \vec{n})$ such that $w \notin \{v_j \mid 1 \leq j \leq k\}$, there is $\psi \in \Psi$ such that $w \in m^g(\psi^g)$.

The idea is to show that $f : U \longrightarrow \{1, \dots, k\}$ is a bijection and to define v_j as $f^{-1}(j)$. First we prove that f is a surjective map. Take $j \in \{1, \dots, k\}$. Since $\{u_1, \dots, u_k\} = \{u'_1, \dots, u'_{k'}\}$, there is a natural number i such that $u'_i = u_j$. If we define l as the natural number $f_i^{-1}(j)$ and w as the point $(u'_i, (n_1, \dots, n_{i_0-1}, l))$, it is easy to see that $f(w) = j$.

Next we show that f is an injective map. Let w and w' be two distinct points of U. We have to prove that $f(w) \neq f(w')$. Since w and w' belong to U, we may assume that

$$w = (u'_i, (n_1, \dots, n_{i_0-1}, l)) \text{ and } w' = (u'_{i'}, (n_1, \dots, n_{i_0-1}, l')).$$

where $i, i' \in \{1, \dots, k'\}$, $l \in \{0, \dots, r_i\}$ and $l' \in \{0, \dots, k(i')\}$. Suppose first that $i = i'$. Since $w \neq w'$, this implies that $l \neq l'$. Since f_i is a bijection, we also have that $f_i(l) \neq f_i(l')$. That is, $f(w) \neq f(w')$. Next assume that $i \neq i'$ and suppose for contradiction that $f(w) = f(w')$. Since $f(w) = f_i(l)$, $f(w)$ belongs to the range of f_i which is equal to $\{j \mid u_j = u'_i\}$. Hence, $u'_i = u_{f(w)}$. For the same reason, $u'_{i'} = u_{f(w')}$. Since $f(w) = f(w)'$, u'_i is equal to $u'_{i'}$. This is a contradiction, since $i \neq i'$ and the points $u'_1, \dots, u'_{k'}$ are pairwise distinct. This finishes the proof that f is a bijection. So for all $j \in \{1, \dots, k\}$, we define v_j by

$$v_j = f^{-1}(j)$$

and we check that conditions (i), (ii) and (iii) are verified. Condition (i) follows immediately from the fact that f is a bijection. For (ii), take $j \in \{1, \dots, k\}$. We have to show that $v_j \in m^g((\varphi_j)^g)$. That is, $g(v_j) = \varphi_j$. Since w_j belongs to U, $g(v_j)$ is equal to $\varphi_{f(v_j)}$. Since $v_j = f^{-1}(j)$, $f(v_j) = j$. Hence, $g(v_j)$ is equal to φ_j, which finishes the proof that (ii) holds.

For condition (iii), let w be a children of $(u, \vec{n})$ that does not belong to $\{v_1, \dots, v_k\}$. First we show that w belongs to W. Otherwise, w belongs to U and by definition, $v_{f(w)} = f^{-1}(f(w))$; that is $v_{f(w)} = w$, which contradicts the fact that $w \notin \{v_1, \dots, v_k\}$. Since w belongs to W, there exists $v \in R[u]\backslash\{u_i \mid 1 \leq i \leq k\}$ such that

$$w = (v, (n_1, \dots, n_{i_0-1}, 0)).$$

By definition of g, $g(w) = \psi_v$ and hence, w belongs to $m^g((\psi_v)^g)$.

Now we check that the induction hypothesis remains true. That is, the conditions (a) and (b) hold. First we show that h' is scattered. By induction hypothesis, it is sufficient to show that for all children w of $(u, \vec{n})$, there is at most one formula $\delta \in \mu\text{ML}^{\nabla'}$ such that w belongs to $m^g(\delta^g)$. This follows immediately from the definition of m^g. Hence, to prove (a), it remains to prove that each point in the set $U \cup W$ is h'-reachable. Recall that the position $((u, \vec{n}), \varphi)$, where $\varphi = \alpha \bullet \nabla^g(\vec{\varphi}, \Psi)$ is h'-reachable. Hence, it is sufficient to show that all the points in $U \cup W$ belong to a set in the range of m^g, which directly follows from the definition of m^g.

Finally we show that condition (b) holds. Let

$$((u_0, \vec{n}_0), (\delta_0)^g) \dots ((u_\kappa, \vec{n}_\kappa), (\delta_\kappa)^g) \tag{4.7}$$

be an h'-conform match such that $(u_{\kappa-1}, \vec{n}_{\kappa-1}) = (u, \vec{n})$ and $\delta_{\kappa-1} = \alpha \bullet \nabla'(\vec{\varphi}, \Psi)$. We have to show that $(u_0, \delta_0) \dots (u_\kappa, \delta_\kappa)$ is an h-conform match. By induction hypothesis, $(u_0, \delta_0) \dots (u_{\kappa-1}, \delta_{\kappa-1})$ is an h-conform match. As m is the ∇'-marking provided by h at position $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$, it is sufficient to show that

$$u_\kappa \text{ belongs to } m(\delta_\kappa). \tag{4.8}$$

We abbreviate $(u_\kappa, \vec{n}_\kappa)$ by w and δ_κ by δ. By definition of h' and since (4.7) is an h'-conform match, we have that w belongs to $m^g(\delta^g)$. Hence, $g(w) = \delta$. Suppose first that w belongs to W. Then w is equal to $(v, (n_1, \dots, n_{i_0-1}, 0))$ for some $v \in R[u] \backslash \{u_i \mid 1 \leq i \leq k\}$. It follows that $u_\kappa = v$ and $g(w) = \psi_v$. Since $g(w)$ is also equal to δ, we obtain $\delta = \psi_v$. By definition of ψ_v, v $(= u_\kappa)$ belongs to $m(\psi_v)$ $(=m(\delta))$. This finishes the proof of (4.8) in the case where $w \in W$.

Assume now that w belongs to U. There exist $i \in \{1, \dots, k'\}$ and $l \in \{0, \dots, r_i\}$ such that

$$w = (u'_i, (n_1, \dots, n_{i_0-1}, l)).$$

Hence, $u_\kappa = u'_i$. It also follows that $f(w) = f_i(l)$. In particular, $f(w)$ belongs to the set $\{j \mid u_j = u'_i\}$. So there exists $j \in \{1, \dots, k\}$ such that $u_j = u'_i$ and $f(w) = j$. Since $g(w) = \varphi_{f(w)}$, this implies that $g(w) = \varphi_j$. Recall also that $g(w) = \delta$. Hence, $\delta = \varphi_j$. By definition of the u_js, we know that u_j $(= u'_i = u_\kappa)$ belongs to φ_j $(=\delta)$. This finishes the proof of (4.8). Hence, the definition of $\mathcal{T}'$ and h'.

Using condition (b) together with the fact that h is a winning strategy for $\exists$ in $\mathcal{E}(\mathcal{T}, \chi)$ with initial position (r, χ), we can easily check that the strategy h' is winning for $\exists$ in the game $\mathcal{E}(\mathcal{T}', \gamma)$ with starting position $((r, \epsilon), \gamma)$. Therefore, the formula γ is true at the root of $\mathcal{T}'$. Now the map that sends a node $(u, \vec{n})$ to u is a surjective p-morphism between $\mathcal{T}'$ and $\mathcal{T}$. Hence, $\mathcal{T}$ is a p-morphic image of $\mathcal{T}'$. Since φ is preserved under p-morphic images for trees and γ and φ are equivalent on trees, γ is also true at the root of $\mathcal{T}$ and this finishes the proof that γ and χ are equivalent on trees. It follows that γ and φ are also equivalent on trees. □

We explained in Section 4.1 that the proof of Theorem 4.1.1 is divided in two parts. The proposition we just proved finishes the first part, which was illustrated by Figure 4.1. In this first part, we investigated several logics and established relations between them. A general picture is given by Figure 4.2. All these equivalences are at the level of models. In the second part, we will show that in the context of frames, $\mu\text{ML}^{\nabla'}$ is equivalent to the μ-calculus.

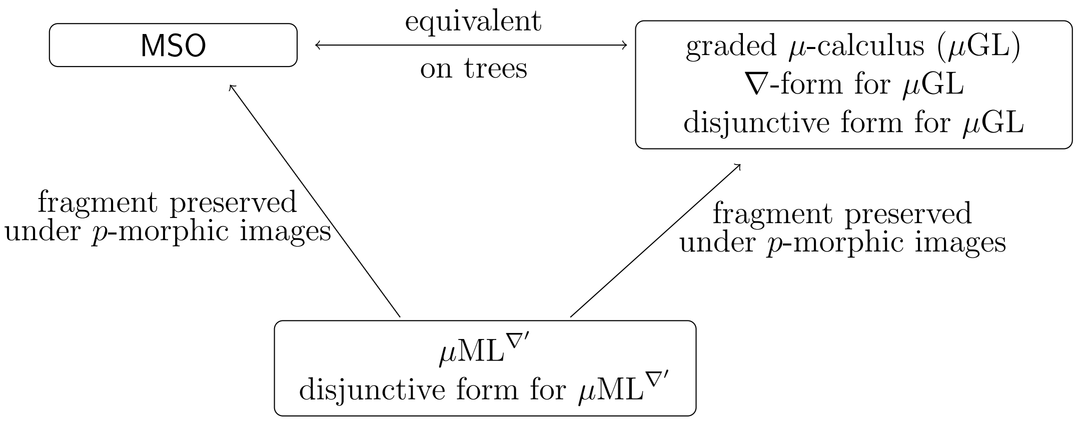


Figure 4.2: Relations between the different logics.

4.4 Local definability

The main goal of this section is to prove that given a disjunctive sentence φ in $\mu\mathrm{ML}^{\nabla'}$, we can find a set $Prop'$ of proposition letters and a μ-sentence φ^t over $Prop \cup Prop'$ such that for all trees $\mathcal{T} = (T, R, V)$ over $Prop$ with root r,

$$\mathcal{T}, r \Vdash \varphi \quad \text{iff} \quad \text{for all } V' : Prop' \to \mathcal{P}(T),\ (T, R, V, V'), r \Vdash \varphi^t. \tag{4.9}$$

Putting this result together with Proposition 4.3.2 and Lemma 4.1.2, we will be able to derive the main result of this chapter, namely Theorem 4.1.1.

In case φ is a formula of the form $\alpha \bullet \nabla'(\vec{\varphi}, \Psi)$, the truth of φ involves the existence of ∇'-witnesses for the pair $(\vec{\varphi}, \Psi)$. So we *existentially* quantify over points in the model. One problem that we may encounter when defining φ^t, is that in (4.9), we *universally* quantify over all valuations.

The idea is to define a new evaluation game for the formulas in $\mu\mathrm{ML}^{\nabla'}$. The main feature of that game is that when reaching a position of the form $(w, \nabla'(\vec{\varphi}, \Psi))$, it will be $\forall$ who has to make a move, unlike in the usual evaluation game (see definition before Proposition 4.3.1). The intuition is that by letting $\forall$ play instead of $\exists$, we replace the existential quantification corresponding to the formula $\nabla'(\vec{\varphi}, \Psi)$ by a universal quantification.

The definition of the new evaluation game is based on the following observation. Let U be a subset of a model $\mathcal{M} = (W, R, V)$, let $w \in W$ and let $\chi(w)$ be the following first-order formula:

$$\chi(w) = \exists\, w_1, \ldots, w_k \in R[w] \text{ such that } \forall\, u \in R[w] \backslash \{w_1, \ldots, w_k\}, u \in U.$$

That is, $\chi(w)$ expresses that there are at most k successors of the point w that do not belong to U. In order to check whether $\chi(w)$ holds, we can let $\exists$ pick k successors $w_1, \ldots, w_k$ of w. Next, $\forall$ chooses a successor u of w that does not belong to $\{w_1, \ldots, w_k\}$. Finally we check whether u belongs to U.

Note that if ψ is a sentence such that $[\![\psi]\!]_{\mathcal{M}} = U$, then $\chi(w)$ holds iff the formula $\nabla'((\top, \dots, \top), \{\psi\})$ is true at w, where the length of the tuple $(\top, \dots, \top)$ is k. Indeed, $\nabla'((\top, \dots, \top), \{\psi\})$ is true at w iff there exist $\{w_1, \dots, w_k\}$ such that for all $1 \leq i \leq k$, $\top$ is true at w_i and for all $u \in R[w] \backslash \{w_1, \dots, w_k\}$, ψ is true at u.

It is easy to see that $\chi(w)$ is in fact equivalent to the formula $\chi'(w)$ given by:

$$\begin{aligned}\chi'(w) \quad = \quad & \exists\, u \in R[w] \wedge \Big(\forall\, w_1, \dots, w_{k+1} \in R[w] \text{ such that } \bigwedge \{w_i \neq w_j \mid i \neq j\}, \\ & \exists\, u \in \{w_1, \dots, w_{k+1}\} \text{ such that } u \in U\Big).\end{aligned}$$

In order to check whether $\chi'(w)$ is valid, we let $\forall$ choose between options (a) and (b). If he chooses option (a), then $\exists$ has to provide a successor u of w. If he chooses option (b), he can pick $k+1$ distinct successors $w_1, \dots, w_{k+1}$ of w. Next, $\exists$ chooses a point u in $\{w_1, \dots, w_{k+1}\}$ and we check whether u belongs to U.

This illustrates how in the new evaluation game we will turn an existential quantification into an universal quantification. Of course, the formula $\nabla'(\vec{\varphi}, \Psi)$ can express more difficult conditions than the one expressed by $\chi(w)$. So it will be more difficult to transform the usual evaluation game into the new evaluation game than to construct the formula $\chi'(w)$ from the formula $\chi(w)$. But the intuition is roughly the same for both transformations.

We define now the new evaluation game $\mathcal{E}'(\mathcal{M}, \varphi)$ for a model $\mathcal{M} = (W, R, V)$ and a sentence φ in $\mu\mathrm{ML}^{\nabla'}$.

The evaluation game $\mathcal{E}'(\mathcal{M}, \varphi)$ for the formulas in $\mu\mathrm{ML}^{\nabla'}$ Let $\mathcal{M} = (W, R, V)$ be a model and let φ be a sentence in $\mu\mathrm{ML}^{\nabla'}$. The rules and the winning conditions of the game $\mathcal{E}'(\mathcal{M}, \varphi)$ are the same as the ones for the game $\mathcal{E}(\mathcal{M}, \varphi)$ (see definition before Proposition 4.3.1), except when we reach a position of the form $(w, \nabla'(\vec{\varphi}, \Psi))$.

In the usual game $\mathcal{E}(\mathcal{M}, \varphi)$, $\exists$ has to propose a ∇'-marking $m : \mu\mathrm{ML}^{\nabla'} \to \mathcal{P}(R[w])$ for the pair $(\vec{\varphi}, \Psi)$. In the game $\mathcal{E}'(\mathcal{M}, \varphi)$, when we reach a position of the form $(w, \nabla'(\vec{\varphi}, \Psi))$ and $\vec{\varphi} = (\varphi_1, \dots, \varphi_k)$, then $\forall$ makes a choice:

(a) Either $\forall$ picks a natural number $i \in \{1, \dots, k\}$. Then $\exists$ has to provide a successor v of w, moving to the position (v, φ_i).

(b) Or $\forall$ picks distinct successors $v_1, \dots, v_m$ of w, where $m \leq k+1$. Next, it is $\exists$ who makes a move. She has the following choice:

   (i) either she picks a point v in $\{v_1, \dots, v_m\}$ and a formula $\psi \in \Psi$, moving to the position (v, ψ),

   (ii) or she provides an injective map $f : \{1, \dots, m\} \to \{1, \dots k\}$. In this case, $\forall$ can choose a natural number $i \in \{1, \dots, m\}$, moving to the position $(v_i, \varphi_{f(i)})$.

We will call the two possible moves of $\forall$ at position $(w, \nabla'(\vec{\varphi}, \Psi))$, *options (a) and (b) in the game* $\mathcal{E}'(\mathcal{M}, \varphi)$. Similarly, after $\forall$ picked the successors $v_1, \ldots, v_m$, we will say that $\exists$ has a choice between *options (i) and (ii) in the game* $\mathcal{E}'(\mathcal{M}, \varphi)$.

We prove now that the new evaluation game captures the truth of a sentence in $\mu\text{ML}^{\nabla'}$.

4.4.1. Proposition. *Let* $\mathcal{M} = (W, R, V)$ *be a model, let* w_0 *be a point in* W *and let* φ *be a sentence in* $\mu\text{ML}^{\nabla'}$. *Then* $\mathcal{M}, w_0 \Vdash \varphi$ *iff* $\exists$ *has a winning strategy in the game* $\mathcal{E}'(\mathcal{M}, \varphi)$ *with initial position* (w_0, φ).

Proof Given a natural number $k \geq 1$, we abbreviate the set $\{1, \ldots, k\}$ by $[k]$. We also denote by $\mathcal{E}$ and by $\mathcal{E}'$ the games $\mathcal{E}(\mathcal{M}, \varphi)$ and $\mathcal{E}'(\mathcal{M}, \varphi)$ respectively. Given a subformula δ of φ and $w \in W$, we write $\mathcal{E}@(w, \delta)$ for the game $\mathcal{E}$ with initial position (w, δ). Similarly, we let $\mathcal{E}'@(w, \delta)$ be the game $\mathcal{E}'$ with initial position (w, δ).

By Proposition 4.3.1, it is sufficient to prove that

$$\exists \text{ has a winning strategy in } \mathcal{E}'@(w_0, \varphi) \text{ iff } \exists \text{ has a winning strategy in } \mathcal{E}@(w_0, \varphi). \tag{4.10}$$

We start by proving the direction from right to left. Suppose that $\exists$ has a winning strategy h in $\mathcal{E}@(w_0, \varphi)$. We have to define a winning strategy h' for $\exists$ in $\mathcal{E}'@(w_0, \varphi)$. The obvious way to proceed is to define h' directly from the strategy h. That is, consider a position of the board of the game $\mathcal{E}'$ that belongs to $\exists$ and define, using the map h, the move dictated by h' at that position.

It would be hard to define h' in such a way; the problem is that the correspondence between the game $\mathcal{E}$ and $\mathcal{E}'$ is not a correspondence that link a move to another move, but a correspondence linking a *sequence* of moves in one game to another *sequence* of moves in the other game. For example, if in the game $\mathcal{E}$, $\exists$ chooses a marking m and $\forall$ picks a pair (w, δ) with $w \in m(\delta)$, then those two moves correspond, in the game $\mathcal{E}'$ to the following sequence of move. First, $\forall$ chooses between option (a) and (b); if he chooses (a), he picks a natural number and $\exists$ picks a successor, whereas if $\forall$ chooses (b), he may pick a sequence of successors, etc.

The idea to define h' is to prove a claim of the type:

(†) Suppose that (w, δ) is a winning position for $\exists$ in $\mathcal{E}$ with respect to h. Then in the game $\mathcal{E}'$, $\exists$ has a strategy g' which will guarantee that, from the same position, within a finite number of steps, she will reach a position of the form (v, γ) such that in some h-conform $\mathcal{E}$-match with initial position (w, δ), position (v, γ) would have been reached at some point as well. Hence, in particular, (v, γ) is a winning position for $\exists$ in $\mathcal{E}$ with respect to h.

Using inductively (†), we can construct a strategy h' for $\exists$. This strategy will ensure that all the finite h'-conform matches are won by $\exists$. However, there is no guarantee for the infinite g'-conform matches. The problem is that during the g'-conform match ρ' from (w, δ) to (v, γ), we might unfold different variables than the ones unfolded in the h-conform match ρ from (w, δ) to (v, γ).

The solution is to strengthen (†). We will require that the only positions of the form (u, x) (with $u \in W$ and $x \in \mathit{Var}$) occurring in ρ' are either the initial position (w, δ) or the last position (v, γ). Similarly, the match ρ should be such that the only positions of the form (u, x) occurring in ρ' are either the initial position (w, δ) or the last position (v, γ). Those two requirement imply that the variables encountered in ρ and ρ' coincide. This motivates the introduction of the notion of *variable scarceness*. A match is *variable scarce* if it contains at most one position of the form (u, x) with $x \in \mathit{Var}$, and this position can only occur as either the first or the last position of the match.

We are now ready to state the claim that will be used to define h' inductively. We denote by $\sqsubseteq$ the prefix (initial segment) relation between sequences.

1. Claim. If $w \in W$ and (w, δ) is a winning position for $\exists$ in $\mathcal{E}$ with respect to h, then $\exists$ has a strategy g' in $\mathcal{E}'@(w, \delta)$ with the property that for all g'-conform matches λ', there exists a g'-conform match ρ' with last position (v, γ) satisfying ($\rho' \sqsubseteq \lambda'$ or $\lambda' \sqsubseteq \rho'$) and conditions (1) and (2) below:

(1) there is an h-conform match ρ leading from (w, δ) to (v, γ),

(2) both ρ and ρ' are variable scarce.

If Claim 1 holds, then we can define a strategy h' for $\exists$ in $\mathcal{E}'_0$ such that $\exists$ will never get stuck and for all h'-conform $\mathcal{E}'_0$-matches π', there exists an h-conform $\mathcal{E}_0$-conform matches π such that $\mathit{Inf}(\pi) = \mathit{Inf}(\pi')$, where $\mathit{Inf}(\pi)$ (resp. $\mathit{Inf}(\pi')$) is the set of variables occurring infinitely often in the match π (resp. in the match π'). It immediately follows that h' is a winning strategy for $\exists$ in $\mathcal{E}'@(w_0, \varphi)$.

Hence, in order to prove the implication from right to left of (4.10), it is sufficient to show Claim 1.

Proof of Claim We only treat the most difficult case, that is, δ is a formula of the form $\nabla'(\vec{\varphi}, \Psi)$. Suppose that $\vec{\varphi} = (\varphi_1, \dots, \varphi_k)$. At position $(w, \nabla'(\vec{\varphi}, \Psi))$ in the game $\mathcal{E}$, $\exists$ chooses, according to h, a ∇'-marking

$$m : \mu\mathrm{ML}^{\nabla'} \to \mathcal{P}(R[w])$$

for the pair $(\vec{\varphi}, \Psi)$. In the game $\mathcal{E}'$, it is $\forall$ who plays at position $(w, \nabla'(\vec{\varphi}, \Psi))$: he chooses either option (a) or option (b). Suppose first that $\forall$ chooses option (a) and picks a natural number $i \in [k]$. Since m is a ∇'-marking, there exists $v \in R[w]$ such that $v \in m(\varphi_i)$. We propose v as the next move for $\exists$ in the game $\mathcal{E}'$. This choice ensures that conditions (1) and (2) of Claim 1 are verified.

Next assume that $\forall$ chooses options (b). That is, $\forall$ picks m distinct successors $v_1, \ldots, v_m$ of w, where $m \leq k+1$. If there is $i \in [m]$ and $\psi \in \Psi$ such that $v_i \in m(\psi)$, then we let $\exists$ propose the pair (v_i, ψ) as the next position. Again, it is immediate that conditions (1) and (2) of Claim 1 hold.

Otherwise, for all $i \in [m]$ and all $\psi \in \Psi$, v_i does not belong to $m(\psi)$. Since m is a ∇'-marking, there exist a tuple $(w_1, \ldots, w_k)$ such that for all $i \in [k]$, $w_i \in m(\varphi_i)$ and for all $v \notin \{w_i \mid i \in [k]\}$, there is $\psi \in \Psi$ such that $v \in m(\psi)$. Putting that together with the fact that for all $i \in [m]$ and all $\psi \in \Psi$, v_i does not belong to $m(\psi)$, we obtain that

$$\{v_1, \ldots, v_m\} \subseteq \{w_1, \ldots, w_k\}.$$

Hence, for all $i \in [m]$, there exists $f(i) \in [k]$ such that $v_i = w_{f(j)}$. Moreover, as the successors $v_1, \ldots, v_m$ are distinct, if $i \neq j$, then $f(i) \neq f(j)$. We let $\exists$ play this injective map f in the game $\mathcal{E}'$ (she chooses possibility (ii)). Now, it is $\forall$'s turn in $\mathcal{E}'$: he picks a pair $(v_i, \varphi_{f(i)})$. Given the definition of f, this pair is equal to $(w_{f(i)}, \varphi_{f(i)})$. It follows from the definition of $(w_1, \ldots, w_k)$ that $w_{f(i)}$ belongs to $m(\varphi_{f(i)})$. Hence, the position $(w_{f(i)}, \varphi_{f(i)})$ satisfies conditions (1) and (2) of Claim 1. This finishes the proof of the implication from right to left of (4.10). ◀

We prove now implication from left to right of (4.10). Assume that $\exists$ has a winning strategy h' in the game $\mathcal{E}'@(w_0, \varphi)$. The definition of a winning strategy for $\exists$ in $\mathcal{E}@(w_0, \varphi)$ is obtained by inductively applying the following claim.

2. Claim. If $w \in W$ and (w, δ) is a winning position for $\exists$ in $\mathcal{E}$ with respect to h', then $\exists$ has a strategy g in $\mathcal{E}'@(w, \delta)$ with the property that for all g-conform matches λ, there exists a g-conform match ρ with last position (v, γ) satisfying ($\rho \sqsubseteq \lambda$ or $\lambda \sqsubseteq \rho$) and conditions (1) and (2) below:

(1) there is an h'-conform match ρ' leading from (w, δ) to (v, γ),

(2) both ρ and ρ' are variable scarce.

Using an argument similar to the one in the proof of the implication from right to left of (4.10), we can show that Claim 2 ensures the existence of a winning strategy for[1] $\exists$ in $\mathcal{E}@(w_0, \varphi)$. Hence, in order to prove the implication from left to right of (4.10), it is sufficient to prove Claim 2.

[1] We would like to mention that Claims 1 and 2 are basically showing that there is a bisimulation between the game $\mathcal{E}@(w_0, \varphi)$ and $\mathcal{E}'@(w_0, \varphi)$, in the sense of [KV09]. It immediately follows from the results of [KV09] that $\exists$ has a winning strategy in one game iff she has a winning strategy in the other. The issue of finding appropriate notions of game equivalence was first raised by Johan van Benthem [Ben02].

Proof of Claim We restrict ourselves to a proof in the case where $\delta = \nabla'(\vec{\varphi}, \Psi)$. Suppose that $\vec{\varphi} = (\varphi_1, \dots, \varphi_k)$. We define U as the set of successors u of w such that for some $\psi_u \in \Psi$ there is an h'-conform partial match with initial position $(w, \nabla'(\vec{\varphi}, \Psi))$ and last position (u, ψ_u). Next, let U' be the set of successors of w that do not belong to U.

Let m be the size of U'. First we show that m is less or equal to k. Suppose for contradiction that $m > k$. Then there exist distinct successors $v_1, \dots, v_{k+1}$ of w that belong to U'. In the game $\mathcal{E}'$, at position $(w, \nabla'(\vec{\varphi}, \Psi))$, we can let $\forall$ play the successors $v_1, \dots, v_{k+1}$. Then, using the strategy h', $\exists$ chooses either option (i) or option (ii) in the game $\mathcal{E}'$. Option (i) means that $\exists$ picks a v_i and a formula $\psi \in \Psi$, moving to the position (v_i, ψ). This means that v_i belongs to U, which is a contradiction. Option (ii) means that $\exists$ chooses an injective map $f : [k+1] \longrightarrow [k]$, which is clearly impossible. Hence, the size m of U' is less or equal to k.

Let $v_1, \dots, v_m$ be distinct successors of w such that $U' = \{v_1, \dots, v_m\}$. In the game $\mathcal{E}'$, at position $(w, \nabla'(\vec{\varphi}, \Psi))$, we let $\forall$ play the successors $v_1, \dots, v_m$. According to h', $\exists$ chooses between option (i) and option (ii). As in the previous paragraph, it is not possible that she chooses option (i). Indeed, if she chooses option (i), the game moves to a position of the form (v_i, ψ) for some $i \in [m]$ and $\psi \in \Psi$. This implies that v_i belongs to U, which is a contradiction. Hence, $\exists$ chooses option (ii) and according to h', she defines a map $f : [m] \longrightarrow [k]$.

Recall that our goal is to define $\exists$'s move at position $(w, \nabla'(\vec{\varphi}, \Psi))$ in the game $\mathcal{E}$. That is, we have to provide a ∇'-marking

$$m : \mu\mathrm{ML}^{\nabla'} \longrightarrow \mathcal{P}(R[w])$$

for the pair $(\vec{\varphi}, \Psi)$. The idea is to define the marking such that each $u \in U$ is marked with ψ_u and each $v_i \in U'$ is marked with $\varphi_{f(i)}$. All the positions reached after this marking in the game $\mathcal{E}$ would satisfy conditions (1) and (2) of Claim 2. Now the problem is that such a marking might not be a ∇'-marking because there is no guarantee that for all $j \in [k]$, there is a point marked with φ_j. Since we want to mark each v_i with the formula $\varphi_{f(i)}$, we already know that for all j in the range of f, there is a successor of w marked with φ_j.

Consider now a natural number j that does not belong to the range of f. At position $(w, \nabla'(\vec{\varphi}, \Psi))$ in the game $\mathcal{E}'$, we can let $\forall$ choose possibility (a) and pick the natural number j. Then, using h', $\exists$ provides a successor u_j of w, moving to the position (u_j, φ_j). We are now ready to define the marking m. We first define the relation R_w in the following way:

$$R_w = \{(u, \psi_u) \mid u \in U\} \cup \{(v_i, \varphi_{f(i)}) \mid i \in [m]\} \cup \{(u_j, \varphi_j) \mid j \notin \mathit{Ran}(f)\},$$

where $\mathit{Ran}(f)$ is the range of f. It follows from the definition of R_w that for all pairs $(v, \theta) \in R_w$,

$$\text{there is an } h'\text{-conform partial match with initial position } (w, \delta) \text{ and last position } (v, \theta) \text{ and no variable is unfolded during this match} \tag{4.11}$$

Finally, we define a map $m : \mu\mathrm{ML}^{\nabla'} \to \mathcal{P}(R[w])$ such that for all formulas $\theta \in \mu\mathrm{ML}^{\nabla'}$ and for all $v \in R[w]$,

$$v \in m(\theta) \quad \text{iff} \quad (v, \theta) \in R_w.$$

It follows from the definition of m and (4.11) that conditions (1) and (2) of Claim 2 hold. We check now that m is a ∇'-marking for the pair $(\vec{\varphi}, \Psi)$. Take $j \in [k]$. If j belongs to the range of f, then there exists $i \in [m]$ such that $f(i) = j$. In that case, we define w_j as the point v_i. If j does not belong to the range of f, we define w_j as the point u_j. It is easy to see that the tuple $(w_1, \dots, w_k)$ is such that for all $j \in [k]$, $w_j \in m(\varphi_j)$ and for all $v \in R[w] \backslash \{w_1, \dots, w_k\}$, there is $\psi \in \Psi$ such that $v \in m(\psi)$. ◀

This finishes the proof of Claim 2 and the proof of the proposition. □

We are now ready to prove that an MSO formula is locally μMLF-definable on trees iff it is preserved under p-morphic images on trees. Putting this result together with Lemma 4.1.2, we obtain the main result of this chapter, that is, Theorem 4.1.1.

4.4.2. PROPOSITION. *A* MSO *formula is locally* μMLF*-definable on trees iff it is preserved under p-morphic images on trees.*

Proof The only difficult direction is from right to left. Let φ_0 be an MSO formula that is preserved under p-morphic images on trees. By Proposition 4.3.2, there is a disjunctive sentence φ in $\mu\mathrm{ML}^{\nabla'}$ such that for all trees $\mathcal{T}$ over *Prop* with root r, φ is true at r iff $\mathcal{T} \vDash \varphi_0$. Our goal is to define a set $Prop_\varphi$ of proposition letters and a μ-sentence φ^t over $Prop \cup Prop_\varphi$ such that for all trees $\mathcal{T} = (T, R, V)$ over *Prop* with root r,

$$\mathcal{T}, r \Vdash \varphi \quad \text{iff} \quad \text{for all } V' : Prop_\varphi \to \mathcal{P}(T),\ (T, R, V, V'), r \Vdash \varphi^t.$$

As usual, given a natural number $k \geq 1$, we abbreviate the set $\{1, \dots, k\}$ by $[k]$. The definitions of the set $Prop_\varphi$ and the formula φ^t are by induction on the complexity of φ. If φ is a proposition letter or a variable, we simply define φ^t as φ and $Prop_\varphi$ as the empty set. If φ is a formula of the form $\eta x.\varphi_1$, then we define φ^t as the formula $\eta x.\varphi^t$ and $Prop_\varphi$ as the set $Prop_{\varphi_1}$. Next, suppose that φ is a formula of the form $\varphi_1 \vee \varphi_2$. Without loss of generality, we may assume that $Prop_{\varphi_1} \cap Prop_{\varphi_2} = \emptyset$. Then we define $Prop_\varphi$ as the set $Prop_{\varphi_1} \cup Prop_{\varphi_2}$ and φ^t as the formula $\varphi_1^t \vee \varphi_2^t$.

Finally suppose that φ is a formula of the form $\alpha \bullet \nabla'(\vec{\varphi}, \Psi)$, with $\vec{\varphi} = (\varphi_1, \ldots, \varphi_k)$ and $\Psi = \{\psi_1, \ldots, \psi_l\}$. Without loss of generality, we may assume that the sets

$$Prop_{\varphi_1}, \ldots, Prop_{\varphi_k}, Prop_{\psi_1}, \ldots, Prop_{\psi_l}$$

have pairwise empty intersection. We let $p_1, \ldots, p_{k+1}$ be fresh proposition letters; that is, for all $i \in [k+1]$, for all $j \in [k]$ and for all $j' \in [l]$, p_i does not belong $Prop_{\varphi_j}$ and p_i does not belongs to $Prop_{\psi_{j'}}$. We define $Prop_{\varphi}$ as the set

$$\{p_1, \ldots, p_{k+1}\} \cup \bigcup\{Prop_{\varphi_i} \mid i \in [k]\} \cup \bigcup\{Prop_{\psi_i} \mid i \in [l]\}.$$

We call $p_1, \ldots, p_{k+1}$ the *proposition letters associated with the formula* $\alpha \bullet \nabla'(\vec{\varphi}, \Psi)$.

In order to define the formula φ^t, we will use the structure of the new evaluation game $\mathcal{E}'(\mathcal{M}, \varphi)$. It is given by:

$$\begin{aligned}
\varphi^t &= \alpha \wedge \psi_1 \wedge (\psi_{21} \vee \psi_{22} \vee \psi_{23}),\\
\psi_1 &= \bigwedge\{\Diamond\varphi_i^t \mid 1 \leq i \leq k\},\\
\psi_{21} &= \bigvee\{\Diamond(p_i \wedge p_j) \mid i, j \in [k+1], i \neq j\},\\
\psi_{22} &= \bigvee\{\Diamond(p_i \wedge \psi^t) \mid i \in [k], \psi \in \Psi\},\\
\psi_{23} &= \bigvee\{\psi_g \mid g : [k+1] \longrightarrow \mathcal{P}([k]) \text{ such that for all } i \neq j,\ g(i) \cap g(j) = \emptyset\},\\
\psi_g &= \bigwedge\Big\{\Box\Big(\neg p_i \vee \bigvee\{\varphi_j^t \mid j \in g(i)\}\Big) \,\Big|\, i \in [k+1]\Big\}.
\end{aligned}$$

The intuition for the construction of φ^t is as follows. If at position $(w, \nabla'(\vec{\varphi}, \Psi))$ in the new evaluation game, $\forall$ chooses possibility (a), this intuitively corresponds to the fact that $\forall$ wants to check that for all $i \in [k]$, there is a successor of w at which φ_i is true. This motivates the introduction of the formula ψ_1.

If $\forall$ chooses possibility (b), then he picks distinct successors $v_1, \ldots, v_m$ of w, with $m \leq k+1$. Intuitively, we have in mind that for all $i \leq m$, the proposition letter p_i is assigned to the point v_i. This implies that the formula $\neg\psi_{21}$ is true.

Now if $\neg\psi_{21}$ is true, $\exists$ has to make a choice between options (i) and (ii). Suppose she chooses (i); that is, she picks a point in $\{v_1, \ldots, v_m\}$ and a formula ψ in Ψ, moving to the position (v_i, ψ). Intuitively, this corresponds to the fact that the formula ψ_{22} is true at w.

Finally suppose that $\exists$ chooses option (ii). Hence, she proposes an injective map $f : [m] \longrightarrow [k]$. Then for all $i \in [m]$, $\forall$ can move to the position $(v_i, \varphi_{f(i)})$. The intended meaning is that the formula $\varphi_{f(i)}$ is true at v_i. As it will become clearer further on, we can think of f as being a map $g : [k+1] \longrightarrow \mathcal{P}([k])$ such that for all $i \neq j$, $g(i) \cap g(j) = \emptyset$. Within that perspective, the fact that the formula $\varphi_{f(i)}$ is true at v_i is equivalent to the fact that the formula $\bigvee\{\varphi_j^t \mid j \in g(i)\}$ is true at v_i. Now if the valuation for the proposition letters $p_1, \ldots, p_n$ that we have in mind is such that for all $v \in R[w]$

$$v \in V(p_i) \quad \text{implies} \quad v = v_i,$$

then, for all $i \in [m]$, the formula

$$\Box\Big(\neg p_i \vee \bigvee\{\varphi^t_j \mid j \in g(i)\}\Big)$$

is true at w. This implies that the formula ψ_g is true at w. This finishes the definitions of $Prop_\varphi$ and φ^t.

We have to show that for all trees $\mathcal{T} = (T, R, V)$ over *Prop* with root r,

$$\mathcal{T}, r \Vdash \varphi \quad \text{iff} \quad \text{for all } V' : Prop_\varphi \to \mathcal{P}(T),\ (T, R, V, V'), r \Vdash \varphi^t. \tag{4.12}$$

We start by proving the implication from left to right. Let $\mathcal{T} = (T, R, V)$ be a tree over *Prop* with root r such that $\mathcal{T}, r \Vdash \varphi$. Let also $V' : Prop_\varphi \to \mathcal{P}(T)$ be a valuation. We have to show that $\mathcal{T}', r \Vdash \varphi^t$, where $\mathcal{T}' := (T, R, V, V')$. Given a game $\mathcal{E}$ and a position z of the board, we denote by $\mathcal{E}@z$ the game $\mathcal{E}$ with initial position z.

Since $\mathcal{T}, r \Vdash \varphi$, it follows from Proposition 4.4.1 that $\exists$ has winning strategy h' in the game $\mathcal{E}'(\mathcal{T}, \varphi)@(r, \varphi)$. We have to find a winning strategy for $\exists$ in the game $\mathcal{E}(\mathcal{T}', \varphi^t)@(r, \varphi^t)$. The definition of the strategy is based on the following claim. Recall that $\sqsubseteq$ is the prefix (initial segment) relation between sequences.

1. Claim. If $u \in T$ and (u, δ) is a winning position for $\exists$ in $\mathcal{E}'(\mathcal{T}, \varphi)$ with respect to h', then $\exists$ has a strategy g^t in $\mathcal{E}(\mathcal{T}', \varphi^t)@(u, \delta^t)$ with the property that for all g^t-conform matches λ, there exists a g^t-conform match ρ with last position (v, γ^t) satisfying ($\rho \sqsubseteq \lambda$ or $\lambda \sqsubseteq \rho$) and condition (1) or (2) below:

(1) either $\forall$ is stuck at position (v, γ^t),

(2) or there is an h'-conform match ρ' leading from (u, δ) to (v, γ). Moreover, both ρ and ρ' are variable scarce.

Recall that a partial match is *variable scarce* if it contains at most one position of the form (u, x) with $x \in Var$, and this position can only occur as either the first or the last position of the match.

Using inductively the claim, we can define a strategy h^t for $\exists$ in the game $\mathcal{E}(\mathcal{T}', \varphi^t)@(r, \varphi^t)$ such that $\exists$ never gets stuck and for all infinite h^t-conform matches π, there is an h'-conform match π' such that $Inf(\pi) = Inf(\pi')$. Since h' is a winning strategy for $\exists$, this implies that h^t is a winning strategy for $\exists$ and this finishes the proof of the implication from left to right of (4.12).

Proof of Claim We abbreviate by $\mathcal{E}'$ and by $\mathcal{E}^t$ the games $\mathcal{E}'(\mathcal{T}, \varphi)$ and $\mathcal{E}(\mathcal{T}', \varphi^t)$ respectively. The proof is by induction on δ. The only case that is not straightforward is the case where δ is a formula of the form $\alpha \bullet \nabla'(\vec{\varphi}, \Psi)$. Suppose that $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$ is a winning position for $\exists$ in $\mathcal{E}'$ with respect to

h' and that $\vec{\varphi} = (\varphi_1, \ldots, \varphi_k)$. We have to define a strategy for $\exists$ in the game $\mathcal{E}^t@(u, (\alpha \bullet \nabla'(\vec{\varphi}, \Psi))^t)$. Recall that $(\alpha \bullet \nabla'(\vec{\varphi}, \Psi))^t$ is given by

$$\alpha \wedge \psi_1 \wedge (\psi_{21} \vee \psi_{22} \vee \psi_{23}),$$

where $\psi_1, \psi_{21}, \psi_{22}$ and ψ_{23} are defined as previously. At position $(u, (\alpha \bullet \nabla'(\vec{\varphi}, \Psi))^t)$ in the game $\mathcal{E}^t$, it is $\forall$ who has to play: he can choose between the formulas α, ψ_1 and $(\psi_{21} \vee \psi_{22} \vee \psi_{23})$.

- Suppose first that he chooses the formula α. Then, at position $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$ in the game $\mathcal{E}'$, we can let $\forall$ move to the position (u, α). It is immediate that condition (2) of Claim 1 holds.

- Next, suppose that at position $(u, (\alpha \bullet \nabla'(\vec{\varphi}, \Psi))^t)$ in the game $\mathcal{E}^t$, $\forall$ chooses the formula ψ_1, moving to the position (u, ψ_1). Recall that ψ_1 is the formula

  $$\bigwedge \{\Diamond \varphi_i^t \mid 1 \leq i \leq k\}.$$

  Hence, at position (u, ψ_1), it is again $\forall$ who has to play and he has to pick a natural number $i_0 \in [k]$, moving to the position $(u, \Diamond \varphi_{i_0}^t)$ in $\mathcal{E}^t$. At position $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$ in the game $\mathcal{E}'$, we can let $\forall$ choose option (a) and pick the natural number i_0. It follows from the rules of the game $\mathcal{E}'$ that $\exists$'s answer (according to h') is a successor v of u and the next position is (v, φ_{i_0}). We can define the strategy g^t in the game $\mathcal{E}^t$ such that at position $(u, \Diamond \varphi_{i_0}^t)$, $\exists$ proposes the pair $(v, \varphi_{i_0}^t)$ as the next position. It is immediate that in this case, condition (2) of Claim 1 holds.

- Suppose finally that at position $(u, (\alpha \bullet \nabla'(\vec{\varphi}, \Psi))^t)$ in the game $\mathcal{E}^t$, $\forall$ chooses the formula $\psi_{21} \vee \psi_{22} \vee \psi_{23}$.

  - Suppose first that ψ_{21} is true at u. Recall that ψ_{21} is the formula

    $$\psi_{21} = \bigvee \{\Diamond (p_i \wedge p_j) \mid i, j \in [k+1], i \neq j\}.$$

    If ψ_{21} is true at u, we let g^t be such that at position $(u, \psi_{21} \vee \psi_{22} \vee \psi_{23})$, $\exists$ moves to position (u, ψ_{21}). Hence, it is clear that $\exists$ can play in such a way that after 3 moves, condition (1) of Claim 1 is verified.

  - Suppose that there is an h'-conform match ρ' that satisfies the two following conditions. The match ρ' is variable scarce. Moreover, the initial position of ρ' is $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$ and the last position of ρ' is (v_0, ψ_0), for some $\psi_0 \in \Psi$ and some $v_0 \in R[u]$ such that $\mathcal{T}', v_0 \Vdash p_1 \vee \cdots \vee p_{k+1}$.

    In this case, we let g^t be such that at position $(u, \psi_{21} \vee \psi_{22} \vee \psi_{23})$, $\exists$ moves to position (u, ψ_{22}). The formula ψ_{22} is given by:

    $$\psi_{22} = \bigvee \{\Diamond (p_i \wedge \psi^t) \mid i \in [k], \psi \in \Psi\}.$$

Hence, at position (u, ψ_{22}) in $\mathcal{E}^t$, it is $\exists$ who has to play and she has to pick a natural number $i \in [k]$ and a formula $\psi \in \Psi$. Since $\mathcal{T}', v_0 \Vdash p_1 \vee \cdots \vee p_{k+1}$, there exists $i \in [k]$ such that $\mathcal{T}'_0, v_0 \Vdash p_i$.

We can propose this natural number i and the formula ψ_0 as the next move for $\exists$ in the game $\mathcal{E}^t$ at position (u, ψ_{22}). This means that we are now at position $(u, \Diamond(p_i \wedge \psi_0^t))$ in the game $\mathcal{E}^t$. Hence, it is again $\exists$'s turn and she has to provide a successor v of u. We define g^t such that at position $(u, \Diamond(p_i \wedge \psi_0^t))$ in the game $\mathcal{E}^t$, $\exists$ picks the point v_0, moving to the next position $(v_0, p_i \wedge \psi_0^t)$.

Now it is $\forall$ who has to make a choice. If $\forall$ chooses the pair (v_0, p_i) as the next position, then condition (1) of Claim 1 is met, as $\mathcal{T}'_0, v_0 \Vdash p_i$. If $\forall$ chooses the pair (v_0, ψ_0^t) as the next position, then condition (2) of Claim 1 is verified, since there is an h'-conform variable scarce match with initial position $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$ and last position (v_0, ψ_0).

– Suppose finally that ψ_{21} is false at u and

> there is no h'-conform match ρ' satisfying the two following conditions.
>
> (*)
> - The match ρ' is variable scarce.
> - The initial position of ρ is $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$ and the last position of ρ' is (v, ψ), for some $\psi \in \Psi$ and some $v \in R[u]$ such that $\mathcal{T}', v \Vdash p_1 \vee \cdots \vee p_{k+1}$.

Let U' be the set of points $v \in R[u]$ such that $\mathcal{T}', v \Vdash p_1 \vee \cdots \vee p_{k+1}$ and let m be the size of U'. We start by showing that m is less or equal to k. Suppose otherwise. Then there exist distinct successors $v_1, \ldots, v_{k+1}$ of u such that for all $i \in [k+1]$, $v_i \in U'$. At position $(u, \nabla'(\vec{\varphi}, \Psi))$ in $\mathcal{E}'$, we can let $\forall$ play the sequence $v_1, \ldots, v_{k+1}$ (option (b) in the game $\mathcal{E}'$). Then, using h', $\exists$ chooses either (i) a successor v in $\{v_1, \ldots, v_{k+1}\}$ and a formula $\psi \in \Psi$, or (ii) an injective map $f : [k+1] \to [k]$. Case (ii) is obviously impossible. So suppose that $\exists$ chooses a successor v in $\{v_1, \ldots, v_{k+1}\}$ and a formula $\psi \in \Psi$, moving to the position (v, ψ). Since v belongs to $\{v_1, \ldots, v_{k+1}\}$, v belongs to U'. That is, $\mathcal{T}', v \Vdash p_1 \vee \cdots \vee p_{k+1}$. Hence, there is an h'-conform variable scarce match with initial position $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$ and last position (v, ψ) such that $\mathcal{T}', v \Vdash p_1 \vee \cdots \vee p_{k+1}$, which contradicts (*). This finishes the proof that m is less or equal to k.

Now let $v_1, \ldots, v_m$ be distinct successors of u such that

$$U' = \{v_1, \ldots, v_m\}.$$

At position $(u, \nabla'(\vec{\varphi}, \Psi))$ in the match $\mathcal{E}'$, we can let $\forall$ pick the successors $v_1, \ldots, v_m$ of u (option (b) in the game $\mathcal{E}'$). According to h', $\exists$

chooses either possibility (i) or possibility (ii). In case she chooses (i), she has to propose a point $v \in \{v_1, \dots, v_m\}$ and a formula $\psi \in \Psi$, moving to the position (v, ψ). As in the previous paragraph, we can show that this contradicts (*).

Hence, $\exists$ chooses possibility (ii); that is, she proposes an injective map $f : [m] \longrightarrow [k]$. Using the map f, we are now going to define a map $g : [k+1] \longrightarrow \mathcal{P}([k])$ such that for all $i \neq i'$, $g(i) \cap g(i') = \emptyset$. For each $i \in [k+1]$, we define U'_i as the set of points v in U' such that $\mathcal{T}', v \Vdash p_i$. Since the formula ψ_{21} given by:

$$\bigvee \{\Diamond(p_i \wedge p_j) \mid i, j \in [k+1], i \neq j\}.$$

is false at u, we know that for all $i \neq i'$, $U'_i \cap U'_{i'} = \emptyset$. Given $i \in [k+1]$, we define $g(i)$ by:

$$g(i) = \{f(j) \mid v_j \in U'_i\}.$$

We verify that for all $i \neq i'$, $g(i) \cap g(i') = \emptyset$. Take $i, i' \in [k+1]$ such that $i \neq i'$. Suppose for contradiction that there exists a natural number k_0 in $g(i) \cap g(i')$. Since k_0 belongs to $g(i)$, it follows from the definition of g that there exists $v_j \in U'_i$ such that $k_0 = f(j)$. Similarly, there exists $v_{j'} \in U'_{i'}$ such that $k_0 = f(j')$. Thus, $f(j) = f(j')$. Since f is an injective map, this means that $j = j'$. Hence, v_j belongs to $U'_i \cap U'_{i'}$. This contradicts the fact that for all $i \neq i'$, $U'_i \cap U'_{i'} = \emptyset$.

We are now ready to define the strategy g^t for the game $\mathcal{E}^t$ at position $(u, \psi_{21} \vee \psi_{22} \vee \psi_{23})$. First, we let $\exists$ choose the pair (u, ψ_{23}) as the next position. Recall that ψ_{23} is the formula

$$\bigvee \{\psi_g \mid g : [k+1] \longrightarrow \mathcal{P}([k]) \text{ such that for all } i \neq j,\ g(i) \cap g(j) = \emptyset\}.$$

Second, at position (u, ψ_{23}), we define $\exists$'s next move as the position (u, ψ_g), where g is the map defined in the previous paragraph and ψ_g is the formula

$$\psi_g = \bigwedge \Big\{ \Box \Big(\neg p_i \vee \bigvee \{\varphi^t_j \mid j \in g(i)\} \Big) \,\Big|\, i \in [k+1] \Big\}.$$

Hence, at position (u, ψ_g) in $\mathcal{E}^t$, it is $\forall$ who has to play: he has to choose a natural number $i \in [k+1]$, moving to the position (u, δ_i), where δ_i is given by:

$$\delta_i = \Box \Big(\neg p_i \vee \bigvee \{\varphi^t_j \mid j \in g(i)\} \Big).$$

At position (u, δ_i) in $\mathcal{E}^t$, $\forall$ has to pick a successor v of u and the next position is $(v, \neg p_i \vee \bigvee \{\varphi^t_j \mid j \in g(i)\})$. If p_i is not true at v, we let

$\exists$ propose the pair $(v, \neg p_i)$ as the next position and condition (1) of Claim 1 is verified.

Otherwise, suppose that p_i is true at v. Then we let $\exists$ choose the position $(v, \bigvee\{\varphi^t_j \mid j \in g(i)\})$ in the game $\mathcal{E}^t$. Since p_i is true at v, v belongs to U'_i. Hence, there exists a unique $j \in [m]$ such that $v_j = v$. It follows from the definition of g that $f(j)$ belongs to $g(i)$. At position $(v, \bigvee\{\varphi^t_j \mid j \in g(i)\})$ in the game $\mathcal{E}^t$, we let $\exists$'s strategy be such that the next position is $z_i := (v, \varphi^t_{f(j)})$.

We check now that for that position, condition (2) of Claim 1 is met. Recall that f was chosen by $\exists$ according to h', after $\forall$ picked the sequence $v_1, \dots, v_m$ in the game $\mathcal{E}'$. Hence, it follows from the rules of the game $\mathcal{E}'$ that after $\exists$ proposed the map f, $\forall$ can move to the position $(v_j, \varphi_{f(j)})$ $(= (v, \varphi_{f(j)}))$. This finishes the proof that condition (2) of Claim 1 is verified for the position z_i.

This also finishes the proof of the claim and the proof of the implication from left to right of (4.12). ◀

Now we turn to the proof of the implication from right to left of (4.12). Let $\mathcal{T} = (T, R, V)$ be a tree over *Prop* with root r. Suppose that for all $V' : Prop_\varphi \to \mathcal{P}(T)$, $(T, R, V, V'), r \Vdash \varphi^t$. We have to show that $\mathcal{T}, r \Vdash \varphi$. Suppose for contradiction that $\mathcal{T}, r \nVdash \varphi$. Hence, by Proposition 4.4.1, $\forall$ has a winning strategy e' in the game $\mathcal{E}'(\mathcal{T}, \varphi)@(r, \varphi)$. The idea is to use this strategy to define a valuation $V'_0 : Prop_\varphi \to \mathcal{P}(T)$.

For all proposition letters $p \in Prop_\varphi$, we define $V'_0(p) \cap \{r\}$ as the empty set. Next, for each $u \in T$ and each $p \in Prop_\varphi$, we define the set $V'_0(p) \cap R[u]$. Take a point $u \in T$ and a proposition letter $p \in Prop_\varphi$. It follows from the definition of $Prop_\varphi$ that there is a unique subformula $\alpha \bullet \nabla'(\vec{\varphi}, \Psi)$ of φ such that p is associated with $\alpha \bullet \nabla'(\vec{\varphi}, \Psi)$. Let $p_1, \dots, p_{k+1}$ be the set of all proposition letters associated with $\alpha \bullet \nabla'(\vec{\varphi}, \Psi)$. For some $i \in [k+1]$, we have $p = p_i$.

Consider the position $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$. If this position does not belong to the domain of e', we define $V'_0(p) \cap R[u]$ as the empty set. Otherwise, at position $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$, according to e', $\forall$ chooses either (a) a natural number or (b) a sequence of distinct successors $v_1, \dots, v_m$ of u, with $m \leq k+1$. In case of (a), we define $V_0(p) \cap R[u]$ as the empty set. If $\forall$ chooses possibility (b), then we define $V'_0(p_i) \cap R[u]$ by:

$$V'_0(p_i) \cap R[u] = \begin{cases} \{v_i\} & \text{if } i \leq m, \\ \emptyset & \text{otherwise.} \end{cases}$$

We observe that it immediately follows that for all $i, j \in [k+1]$,

$$i \neq j \quad \text{implies} \quad V'_0(p_i) \cap V'_0(p_j) = \emptyset. \tag{4.13}$$

This finishes the definition of V_0.

We know by assumption that $\mathcal{T}_0', r \Vdash \varphi^t$, where $\mathcal{T}_0' := (T, R, V, V_0')$. Hence, $\exists$ has a winning strategy h_0^t in the game $\mathcal{E}(\mathcal{T}_0', \varphi^t)@(r, \varphi^t)$. The idea is to use the strategy h_0^t to play against $\forall$'s strategy in the game $\mathcal{E}'(\mathcal{T}, \varphi)@(r, \varphi)$ and to obtain a contradiction. More precisely, we define an infinite e'-conform match π' such that for some h_0^t-conform match π, we have $Inf(\pi) = Inf(\pi')$. The construction of π' is based on the following claim.

2. Claim. If (u, δ) is a winning position for $\forall$ in $\mathcal{E}'(\mathcal{T}, \varphi)$ with respect to e' and (u, δ^t) is a winning position for $\exists$ in $\mathcal{E}(\mathcal{T}_0', \varphi^t)$ with respect to h_0^t, then there exists a partial e'-conform match ρ' with initial position (u, δ) and final position (v, γ) such that

(1) there is a h_0^t-conform match ρ leading from (u, δ^t) to (v, γ^t),

(2) both ρ and ρ' are variable scarce.

Using inductively the claim, we can define an infinite e'-conform match π' such that for some h_0^t-conform match π, we have $Inf(\pi) = Inf(\pi')$. Since e' is a winning strategy for $\forall$, π' is won by $\forall$. On the other hand, as h_0^t is a winning strategy for $\exists$, π is won by $\exists$. Putting that together with the fact that $Inf(\pi) = Inf(\pi')$, we obtain that π' is also won by $\exists$, which is a contradiction. Hence, to prove the implication from right to left of (4.12), it is sufficient to prove the claim.

Proof of Claim We abbreviate by $\mathcal{E}'$ and by $\mathcal{E}^t$ the games $\mathcal{E}'(\mathcal{T}, \varphi)$ and $\mathcal{E}(\mathcal{T}_0', \varphi^t)$ respectively. The proof is by induction on the complexity of δ. We concentrate on the most difficult case, where δ is a formula of the form $\alpha \bullet \nabla'(\vec{\varphi}, \Psi)$. Assume that $\vec{\varphi}$ is equal to $(\varphi_1, \dots, \varphi_k)$. Suppose also that $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$ is a winning position for $\forall$ in $\mathcal{E}'$ with respect to e' and that $(u, (\alpha \bullet \nabla'(\vec{\varphi}, \Psi))^t)$ is a winning position for $\exists$ in $\mathcal{E}^t$ with respect to h_0^t. At position $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$, depending on the strategy e', there are several types of moves that $\forall$ can make.

- Suppose first that $\forall$ moves to the position (u, α). Since e' is a winning strategy for $\forall$ and $(u, \alpha \bullet \nabla'(\vec{\varphi}, \Psi))$ is a winning position for $\forall$ with respect to e', this means that α is false at u. Now in the game $\mathcal{E}^t$ at position $(u, (\alpha \bullet \nabla'(\vec{\varphi}, \Psi))^t)$, we can also let $\forall$ propose the position (u, α). Since h_0^t is a winning strategy for $\exists$ and $(u, (\alpha \bullet \nabla'(\vec{\varphi}, \Psi))^t)$ is a winning position for $\exists$ with respect to h_0^t, this implies that α is true at u. This is a contradiction.

- Suppose next that $\forall$ moves to the position $(u, \nabla'(\vec{\varphi}, \Psi))$ and chooses option (a) in the game $\mathcal{E}'$. That is, $\forall$ picks a natural number $i \in [k]$. Recall that $\alpha \bullet \nabla'(\vec{\varphi}, \Psi))^t$ is the formula

$$\alpha \wedge \psi_1 \wedge (\psi_{21} \vee \psi_{22} \vee \psi_{23}).$$

Hence, at position $(u, (\alpha \bullet \nabla'(\vec{\varphi}, \Psi))^t)$ in the game $\mathcal{E}^t$, we can let $\forall$ move to the position (u, ψ_1), where ψ_1 is the formula

$$\bigwedge\{\Diamond\varphi_i^t \mid 1 \leq i \leq k\}.$$

So $\forall$ may play again and propose the pair $(u, \Diamond\varphi_i^t)$ as the next position. Now it is $\exists$'s turn. According to h_0^t, she chooses a successor v of u, moving to the position (v, φ_i^t). Now, we define the $\mathcal{E}'$-match ρ' such that after $\forall$ picked the natural number $i \in [k]$, $\exists$ moves to the position (v, φ_i). It is immediate that conditions (1) and (2) of Claim 2 are met.

- Suppose finally that $\forall$ moves to the position $(u, \nabla'(\vec{\varphi}, \Psi))$ and chooses option (b) in the game $\mathcal{E}'$. Hence, $\forall$ picks distinct successors $v_1, \ldots, v_m$ of u, with $m \leq k+1$. At position $(u, (\alpha \bullet \nabla'(\vec{\varphi}, \Psi))^t)$ in the game $\mathcal{E}^t$, we can decide that $\forall$ moves to the position $(u, \psi_{21} \vee \psi_{22} \vee \psi_{23})$. Now, depending on h_0^t, we make the following case distinction.

  - Suppose that the position following $(u, \psi_{21} \vee \psi_{22} \vee \psi_{23})$ in $\mathcal{E}^t$ (dictated by h_0^t) is the pair (u, ψ_{21}). Recall that ψ_{21} is the formula given by:

  $$\bigvee\{\Diamond(p_i \wedge p_j) \mid i, j \in [k+1], i \neq j\}.$$

  Since (u, ψ_{21}) is a winning position for $\exists$ in $\mathcal{E}^t$, the formula ψ_{21} is true at u. Hence, there exists a successor v of u such that $v \in V_0'(p_i) \cap V_0'(p_j)$, for some $i \neq j$. This contradicts implication (4.13).

  - Next assume that at position $(u, \psi_{21} \vee \psi_{22} \vee \psi_{23})$ in $\mathcal{E}^t$, according to h_0^t, $\exists$ chooses the position (u, ψ_{22}). The formula ψ_{22} is given by:

  $$\bigvee\{\Diamond(p_i \wedge \psi^t) \mid i \in [k], \psi \in \Psi\}.$$

  Hence, at position (u, ψ_{22}), it is again $\exists$ who has to play. Using h_0^t, she chooses a formula ψ in Ψ and a natural number $i \in [k]$, moving to the position $(u, \Diamond(p_i \wedge \psi^t))$. Next, according to h_0^t, $\exists$ picks a successor v of u, moving to the position $(v, p_i \wedge \psi^t)$. Since h_0^t is a winning strategy for $\exists$, p_i is true at v. By definition of V_0', we have $R[u] \cap V_0'(p_i) = \{v_i\}$. This implies that v is equal to v_i. So we are now at position $(v_i, p_i \wedge \psi^t)$ in the game $\mathcal{E}^t$. It is $\forall$ who has to play. We can let him move to the position (v_i, ψ^t).

  We define the $\mathcal{E}'$-match ρ' such that after $\forall$ picked the successors $v_1, \ldots, v_m$ of u, $\exists$ chooses option (i) in the game $\mathcal{E}'$ and moves to the position (v_i, ψ). It is clear that conditions (1) and (2) of Claim 2 are verified.

– Finally suppose that at position $(u, \psi_{21} \vee \psi_{22} \vee \psi_{23})$ in $\mathcal{E}^t$, according to h^t_0, $\exists$ moves to the position (u, ψ_{23}). Recall that ψ_{23} is the formula given by:

$$\bigvee\{\psi_g \mid g : [k+1] \longrightarrow \mathcal{P}([k]) \text{ such that for all } i \neq j,\ g(i) \cap g(j) = \emptyset\}.$$

Hence, the strategy h^t_0 provides $\exists$ with a map $g : [k+1] \longrightarrow \mathcal{P}([k])$ such that for all $i \neq j$, $g(i) \cap g(j) = \emptyset$. Moreover, the new position in the game $\mathcal{E}^t$ is (u, ψ_g), where ψ_g is the formula

$$\bigwedge\Big\{\Box\Big(\neg p_i \vee \bigvee\{\varphi^t_j \mid j \in g(i)\}\Big) \Bigm| i \in [k+1]\Big\}.$$

Take $i \in [m]$. We can let $\forall$ choose

$$\Big(u, \Box\Big(\neg p_i \vee \bigvee\{\varphi^t_j \mid j \in g(i)\}\Big)\Big) \text{ and } \Big(v_i, \Big(\neg p_i \vee \bigvee\{\varphi^t_j \mid j \in g(i)\}\Big)\Big)$$

as the next two positions in the game $\mathcal{E}^t$. It is now $\exists$ who has to play according to h^t_0: either (A) she moves to the position $(v_i, \neg p_i)$ or (B) she picks a natural number number $f(i) \in g(i)$, moving to the position $(v_i, \varphi^t_{f(i)})$. By definition of V'_0, we know that v_i belongs to $V'_0(p_i)$. Hence, the position $(v_i, \neg p_i)$ is not a winning position for $\exists$. This means that case (A) cannot happen, as $\exists$ played using her winning strategy h^t_0. Hence, $\exists$ chooses a natural number number $f(i) \in g(i)$, moving to the position $(v_i, \varphi^t_{f(i)})$.
We define the $\mathcal{E}'$-match ρ' such that after $\forall$ picked the successors $v_1, \dots, v_m$ of u, $\exists$ chooses option (ii) in the game $\mathcal{E}'$ and provides the map $f : [m] \longrightarrow [k+1]$ as defined in the previous paragraph. Next, in the game $\mathcal{E}'$, it is $\forall$ who makes a move according to e': he picks a pair $(v_i, \varphi_{f(i)})$, with $i \in [m]$. By definition of f, condition (1) of Claim 2 is satisfied. An inspection of the proof easily shows that condition (2) of Claim 2 is also verified.

This finishes the proof of Claim 2. ◀

As mentioned earlier, Claim 2 implies that the implication from right to left of (4.12) holds. This finishes the proof of the proposition. □

We can now derive the following decidability result.

4.4.3. Corollary. *It is decidable whether a given* MSO *formula is locally* μMLF*-definable on trees. It is decidable whether a given* MSO *formula is* μMLF*-definable on trees.*

Proof In order to derive this result from Theorem 4.1.1, it is sufficient to show it is decidable whether a given MSO formula is preserved under p-morphic images on trees and under taking subtrees. It follows from Proposition 4.3.2 and the fact that MSO is decidable on trees that it is decidable whether an MSO formula is preserved under p-morphic images on trees.

It remains to show that it is decidable whether a given formula φ is preserved under taking subtrees. Let φ be an MSO formula. The proof consists in defining an MSO formula ψ such that ψ is valid on all trees iff φ is preserved under taking subtrees. It will follow from the decidability of MSO on trees that it is decidable whether φ is preserved under taking subtrees.

First we introduce the formula $\chi_1(X)$ with one free second-order variable X by:

$$\chi_1(X) = \exists x,\, \forall y \in X(y \in X \leftrightarrow xR^*y),$$

where R^* is the reflexive transitive closure of R. Since the reflexive transitive closure of a relation can be expressed by an MSO formula, we may assume that $\chi_1(X)$ is an MSO formula. The formula $\chi_1(X)$ is such that for all trees $\mathcal{T}$ and all subsets U of $\mathcal{T}$,

$$\mathcal{T} \vDash \chi_1(U) \quad \text{iff} \quad U \text{ is the domain of a subtree of } \mathcal{T}.$$

Next, we define the formula $\chi(X)$ with one free second-order variable X by:

$$\neg\chi_1(X) \vee (\chi_1(X) \wedge \varphi_X),$$

where φ_X is the relativization of φ to X. That is, we replace each subformula of the form $(\exists x, \delta)$ by $(\exists x \in X, \delta)$ and each subformula of the form $(\exists Y, \delta)$ by $(\exists Y \subseteq X,\, \delta)$. For all trees $\mathcal{T}$ and all subsets U of $\mathcal{T}$, $\chi(U)$ holds in $\mathcal{T}$ iff U is not the domain of a subtree or U is the domain of a subtree in which φ holds. Finally, we define ψ as the formula $\forall X(\varphi \to \chi(X))$. It is easy to check that ψ has the required property. □

4.5 Negative and projective definability

In this section, we consider other notions of frame definability. The one that we used until now is the standard one and consists in a universal quantification over all valuations and over all nodes. Here, we show how our results can be adapted to the cases where we existentially quantify over all the valuations.

4.5.1 Negative definability

The first alternative notion of definability that we consider, is the *negative definability*. It has been introduced by Yde Venema in [Ven93] and investigated by Marco Hollenberg [Hol98a]. A frame (W, R) is negatively defined by a formula

φ if everywhere in (W, R), φ is false under some valuation. This notion enables us to capture interesting classes that are not definable in the standard sense. A typical example is the class of all irreflexive frames. In the special case of trees, negative definability corresponds to the dual of the usual notion of definability (as we will see from the result in this section).

Negative definability An MSO formula φ is *negatively* μMLF-*definable on trees* if there are a set *Prop*$'$ and a μ-sentence ψ over $Prop \cup Prop'$ such that for all trees $\mathcal{T} = (T, R, V)$ over *Prop*,

$$\mathcal{T} \vDash \varphi \quad \text{iff} \quad \text{for all } u \in T, \text{ there exists } V' : Prop' \to \mathcal{P}(T),\ (T, R, V, V'), u \nVdash \psi.$$

An MSO formula φ is *locally negatively* μMLF-*definable on trees* if there are a set *Prop*$'$ and a μ-sentence ψ over $Prop \cup Prop'$ such that for all trees $\mathcal{T} = (T, R, V)$ over *Prop*,

$$\mathcal{T} \vDash \varphi \quad \text{iff} \quad \text{there exists } V' : Prop' \to \mathcal{P}(T),\ (T, R, V, V'), r \nVdash \psi,$$

where r is the root of $\mathcal{T}$. When this happens, we say that φ is *locally negatively* μMLF-*definable on trees by* ψ.

For characterizing negative definability, we use the dual notion of preservation under p-morphic images.

Reflection of p-morphic images An MSO formula φ is *reflects p-morphic images on trees* if for all surjective p-morphisms f between two trees $\mathcal{T}$ and $\mathcal{T}'$ over *Prop*, then

$$\mathcal{T}' \vDash \varphi \quad \text{implies} \quad \mathcal{T} \vDash \varphi.$$

The next proposition is the dual of Proposition 4.4.2.

4.5.1. PROPOSITION. *A* MSO *formula is locally negatively* μMLF-*definable on trees iff it reflects p-morphic images on trees.*

Moreover, given an MSO *formula φ, we can compute at μ-sentence ψ such that φ is locally negatively* μMLF-*definable on trees iff φ is locally negatively* μMLF-*definable on trees by ψ.*

Using easy adaptations of proofs of Lemma 4.1.2 and Corollary 4.4.3, we obtain the following characterization for negative definability.

4.5.2. PROPOSITION. *An* MSO *formula is negatively μMLF-definable on trees iff it reflects p-morphic images on trees and is closed under taking subtrees. Moreover, it is decidable whether an* MSO *formula is negatively μMLF-definable on trees.*

4.5.2 Projective definability

Another alternative notion of definability is projective definability. A frame (W, R) is projectively defined by a formula φ if there is a valuation such that φ is true everywhere under this valuation.

Projective definability is a relevant notion in the framework of knowledge representation. The general idea is as follows. We collect all the knowledge concerning a specific subject and in order to reason about this knowledge, we encode it using some logical language, in a knowledge base. Using axioms and rules of the logical language, we can deduce consequences from this knowledge base. Something common is to add fresh new proposition letters to the logical language, that is, proposition letters which did not occur in the observational core of the knowledge base. For this reason, these proposition letters are called theoretical. The intuitive idea behind a sentence which contains a theoretical proposition letter, is that, under some interpretation of the theoretical proposition letter, the sentence holds. This corresponds exactly to an existential second order quantification over the theoretical proposition letters. This explains the name "projective definability".

Projective definability An MSO formula φ is *projective* μMLF*-definable on trees* if there are a set *Prop*$'$ and a μ-sentence ψ over $Prop \cup Prop'$ such that for all trees $\mathcal{T} = (T, R, V)$ over *Prop*,

$$\mathcal{T} \models \varphi \quad \text{iff} \quad \text{there exists } V' : Prop' \to \mathcal{P}(T), \text{ for all } u \in T,\ (T, R, V, V'), u \Vdash \psi.$$

The argument to derive a characterization of projective μ-definability from Proposition 4.5.1 above is a bit more tedious, than for the case of negative definability.

4.5.3. PROPOSITION. *An* MSO *formula φ is projective* μMLF*-definable on trees iff it reflects p-morphic images on trees and is closed under taking subtrees. Moreover, it is decidable whether an* MSO *formula φ is projective* μMLF*-definable on trees.*

Proof It is easy to check that an MSO formula φ that is projective μMLF-definable on trees, reflects p-morphic images on trees and is closed under taking subtrees.

For the other direction, let φ be an MSO formula that reflects p-morphic images on trees and is closed under taking subtrees. By Proposition 4.5.1, φ is locally projective μMLF-definable on trees. That is, there exist a set *Prop*$'$ and a μ-sentence ψ over $Prop \cup Prop'$ such that for all trees $\mathcal{T} = (T, R, V)$ over *Prop* with root r,

$$\mathcal{T} \models \varphi \quad \text{iff} \quad \text{there is } V' : Prop' \to \mathcal{P}(T) \text{ such that } (T, R, V, V'), r \Vdash \psi. \tag{4.14}$$

Now we may assume that ψ is in disjunctive normal form. A formula χ is said to be ψ*-reachable* if there exists a tree $\mathcal{T}'$ over $Prop \cup Prop'$ with root r such that $\exists$ has a winning strategy f in the evaluation game $\mathcal{E}(\mathcal{T}', \psi)$ with starting position (r, ψ), and there is a f-conform match during which a position of the form (u, χ) is reached.

In order to define the formula ψ' which will witness that φ is projective μMLF-definable on trees, we make a case distinction:

(a) Suppose first that there are a tree $\mathcal{T}'$ over $Prop \cup Prop'$ with root r, a node u in $\mathcal{T}'$ and a winning strategy f for $\exists$ in the evaluation game $\mathcal{E}(\mathcal{T}', \psi)$ with initial position (r, ψ) such that u does not occur in any f-conform match. Then we define ψ' as $\top$.

(b) Otherwise, we let ψ' be the formula $\bigvee\{e(\chi) \mid \chi \text{ is } \psi\text{-reachable}\}$. Recall that $e(\chi)$ is the expansion of χ (see Section 2.1 of Chapter 2) (roughly, this means that $e(\chi)$ is the sentence obtained from χ by replacing each free variable x of χ, by the unfolding of x in χ).

In order to show that φ is projective μMLF-definable on trees, it is sufficient to prove that for all trees $\mathcal{T} = (T, R, V)$ over $Prop$,

$$\mathcal{T} \vDash \varphi \quad \text{iff} \quad \text{there is } V' : Prop' \to \mathcal{P}(T) \text{ such that } (T, R, V, V') \Vdash \psi'. \tag{4.15}$$

First, suppose that $\mathcal{T} = (T, R, V)$ is a tree over $Prop$ with root r such that $\mathcal{T} \vDash \varphi$. Then there is a valuation $V' : Prop' \to \mathcal{P}(T)$ such that $(T, R, V, V'), r \Vdash \psi$. We show that $(T, R, V, V') \Vdash \psi'$. If $\psi' = \top$, this is trivial. So we may assume that we are in case (b). Let $\mathcal{T}'$ be the tree (T, R, V, V') over $Prop \cup Prop'$ and let u be a node in $\mathcal{T}'$.

Since ψ is true at r in $\mathcal{T}'$, $\exists$ has a winning strategy f in the evaluation game $\mathcal{E}(\mathcal{T}', \psi)$ with starting position (r, ψ). As we are in case (b), there is a formula χ such that a position of the form (u, χ) occurs in an f-conform match. It follows that the formula $e(\chi)$ is true at u in $\mathcal{T}'$ (see Section 2.3 of Chapter 2). The formula χ is also ψ-reachable. It follows from the definition of ψ' that ψ' is true at u in $\mathcal{T}'$.

For the other direction of equivalence (4.15), suppose that there is a tree $\mathcal{T} = (T, R, V)$ over $Prop$ and a valuation $V' : Prop' \to \mathcal{P}(T)$ such that for all nodes u, ψ' is true at u in (T, R, V, V'). We have to show that $\mathcal{T} \vDash \varphi$. First we suppose that we are in case (b).

Let r be the root of $\mathcal{T}$ and let $\mathcal{T}'$ be the tree (T, R, V, V') over $Prop \cup Prop'$. Since the formula ψ' is true at r in $\mathcal{T}'$, there is a formula $e(\chi)$ such that χ is ψ-reachable and $\mathcal{T}', r \Vdash e(\chi)$. This implies that $\exists$ has a winning strategy g in the evaluation game $\mathcal{E}(\mathcal{T}', \psi)$ with starting position $(r, e(\chi))$.

Moreover, since χ is ψ-reachable, there is a tree $\mathcal{S}' = (T_S, R_S, V_S, V_s')$ over $Prop \cup Prop'$ with root s such that $\exists$ has a winning strategy f in the evaluation game $\mathcal{E}(\mathcal{S}', \psi)$ with starting position (s, ψ) and a position of the form (u_S, χ) occurs during an f-conform match. Without loss of generality, we may assume that $\mathcal{S}'$ is ω-expanded and that f is scattered (see Section 2.6).

Now we define a new tree $\mathcal{T}_0'$ over $Prop \cup Prop'$, which is obtained by "replacing" the subtree $\mathcal{S}'_{u_S}$ of $\mathcal{S}'$ by the tree $\mathcal{T}'$. More formally, we let (T_1, R_1) be the subframe of (T_S, R_S), the domain of which consists of all the nodes of $\mathcal{S}'$, except the proper descendants of u_S. We define the frame (T_0, R_0) in the following way. The set T_0 is equal to $T_1 \cup T \backslash \{r\}$. The relation R_0 is $R_1 \cup (R \backslash \{(r, u) \mid u \in T\}) \cup \{(u_S, u) \mid (r, u) \in R\}$.

We also define the valuations $V_0 : Prop \longrightarrow \mathcal{P}(T_0)$ and $V_0' : Prop' \longrightarrow \mathcal{P}(T_0)$ such that

$$\begin{cases} V_0(p) & = V(p) \backslash \{r\} \cup (V_s(p) \cap T_1), \\ V_0'(p') & = V'(p') \backslash \{r\} \cup (V_s'(p') \cap T_1), \end{cases}$$

for all proposition letters $p \in Prop$ and all $p' \in Prop'$. We prove that the formula ψ is true at the root r_0 of $\mathcal{T}_0'$. It is sufficient to construct a winning strategy for $\exists$ in the evaluation game $\mathcal{E}(\mathcal{T}_0', \psi)$ with starting position (r_0, ψ). This winning strategy is defined in the following way.

As long as the match stays in T_1, $\exists$ follows the strategy f. Recall that f is scattered. So the only way for an f-conform match to get out of T_1 is to go through the position (u_S, χ). If this position is reached, we let $\exists$ follow the strategy g until the end of the game. It is easy to see that this defines a winning strategy for $\exists$ in $\mathcal{E}(\mathcal{T}_0', \varphi)$ with starting position (r_0, ψ). Thus, ψ is true at r_0 in $\mathcal{T}_0'$.

Putting this together with the fact that φ is locally projective μMLF-definable by ψ, we obtain that $\mathcal{T}_0 \Vdash \varphi$, where $\mathcal{T}_0 := (T_0, R_0, V_0)$. Using now the facts that $\mathcal{T}$ is a subtree of $\mathcal{T}_0$ and that φ is preserved under taking subtrees, we obtain $\mathcal{T} \Vdash \varphi$. This finishes the proof of the implication from right to left of (4.15) in case (b).

It remains now to consider case (a). In order to prove the implication from right to left of (4.15) in case (a), we have to show that for all trees $\mathcal{T}$ over $Prop$, $\mathcal{T} \vDash \varphi$. Let $\mathcal{T} = (T, R, V)$ be a tree over $Prop$. Since (a) holds, there is a tree $\mathcal{S}' = (T_S, R_S, V_S, V_S')$ over $Prop \cup Prop'$ with root s, a node u_S in $\mathcal{S}'$ and a strategy f in the game $\mathcal{E}(\mathcal{S}', \psi)$ with initial position (s, ψ) such u_S does not occur in any f-conform match.

It is possible to define a tree $\mathcal{T}_0'$ over $Prop \cup Prop'$ which is obtained by replacing the subtree $\mathcal{S}'_{u_S}$ of $\mathcal{S}'$ by the tree $\mathcal{T}'$. The construction is done as in case (b). Similarly to case (b), we can also show that ψ is true at the root r_0 of $\mathcal{T}_0'$ and prove that it implies that $\mathcal{T} \vDash \varphi$.

We turn now to the proof of the decidability result. By Proposition 4.5.1, given an MSO formula φ, we can compute a formula ψ such that (4.14) holds. Then we check whether ψ is equivalent to $\top$ or if there is a set Ψ of subformulas of ψ such that ψ is equivalent to $\bigvee\{e(\psi) \mid \psi \in \Psi\}$. The formula φ is projective

μMLF-definable on trees iff one of these equivalence holds. □

4.6 Conclusion

We gave a natural characterization of the MSO formulas that are μMLF-definable on trees. Precisely, we showed that an MSO formula is μMLF-definable on trees iff it is preserved under p-morphic images on trees and under taking subtrees. Using this characterization, we proved that it is decidable whether a given MSO formula is μMLF-definable on trees.

A natural further question is to investigate the μMLF-definability for classes of arbitrary frames, not only classes of trees. Unlike on trees, the graded μ-calculus does not have the same expressive power as MSO on arbitrary models: it corresponds to the fragment of MSO invariant under counting bisimulations (see [JL03] and [Wal02]). Moreover, the proof of Proposition 4.3.2 does not work for classes of arbitrary frames, as it relies on the fact that given a disjunctive formula that is true in an ω-expanded tree, there is a scattered strategy for $\exists$. As a consequence, in order to characterize μMLF-definability for classes of frames, we probably need to use different methods than the ones used here.

Another problem that we are investigating at the moment, is the characterization of classes of trees that are definable in modal logic. We define the notion of MLF-definability, exactly as we defined μMLF-definability, but we require φ to be a modal formula, instead of a μ-sentence. Our goal is to prove results similar to Theorem 4.1.1 and Corollary 4.4.3, but for MLF-definability. A key notion for obtaining the characterization of MLF-definability, is the notion of local testability. The decidability of the problem whether an MSO formula is MLF-definable on trees is showed by combining the characterization together with a decidability result from [PS09].

The notions that we use in that result are closely related the the notion of local testability.

Chapter 5

Syntactic characterizations of semantic fragments of the μ-calculus

This chapter is inspired by the model-theoretic tradition in logic of linking semantic properties of formulas to syntactic restrictions on their shape. Such correspondences abound in the model theory of classical (propositional or first-order) logic [CK73]. Well-known preservation results are the Łos-Tarski theorem stating that the models of a sentence φ are closed under taking submodels iff φ is equivalent to an universal sentence, or Lyndon's theorem stating that a sentence φ is monotone with respect to the interpretation of a relation symbol R iff φ is equivalent to a sentence in which all occurrences of R are positive. In the last example, the semantic property is monotonicity, and the syntactic restriction is positivity.

Our aim here is to establish such correspondences in the setting of the μ-calculus. Some results are known: in particular, preservation results, similar to the Łos-Tarski and Lyndon theorems, have been shown for the μ-calculus by Giovanna D'Agostino and Marco Hollenberg [DH00]. However, in the intended semantics of the μ-calculus, where models represent computational processes, and accessibility relations, bisimulations, and trees play an important role, there are some specific properties of interest that have not been studied in classical model theory.

As an example we mention the property of complete additivity with respect to a proposition letter p. Given a proposition letter p, a formula φ is completely additive in p iff it is monotone in p and in order to establish the truth of φ, we need exactly one point at which p is true. Equivalently, this corresponds to the fact that the map associated with φ (as defined in Section 2.2) distributes over countable unions.

One of the main reasons for studying complete additivity is given by its important role in the characterization of the fragment of first- and monadic second-order logic of formulas that are safe for bisimulations (see Section 2.6). Syntactic characterizations of this property were obtained for modal logic by Johan van

Benthem [Ben96], and for the μ-calculus by Marco Hollenberg [Hol98b]. As an alternative to Marco Hollenberg's result, we shall give a different (but clearly, equivalent) syntactic fragment characterizing complete additivity.

Our work continues this line of investigation by studying, next to complete additivity, a number of additional properties that are related to the notion of continuity. In [Ben06], Johan van Benthem already identified the continuous fragment (under the name ω-μ-calculus) as an important fragment for several purposes in modal logic. Intuitively, given a proposition letter p, a formula φ is said to be continuous in p if it is monotone in p and if in order to establish the truth of φ at a point, we only need finitely many points at which p is true.

We believe that this continuous fragment is of interest for a number of reasons. A first motivation concerns the relation between continuity and another property, constructivity. The constructive formulas are the formulas whose fixpoint is reached in at most ω steps. Locally, this means that a state satisfies a least fixpoint formula if it satisfies one of its finite approximations. It is folklore that if a formula is continuous, then it is constructive. While, the other implication does not strictly hold, interesting questions concerning the link between constructivity and continuity remain. In any case, given our Theorem 5.4.4, continuity can be considered as the most natural candidate to approximate constructivity syntactically.

Another reason for looking at the continuous fragment (which also explains the name) is its link with Scott continuity. A formula is continuous in p iff for all models, the map φ_p (as defined in Chapter 2) is continuous in the Scott topology on the powerset algebra. Scott continuity is of key importance in many areas of theoretical computer sciences where ordered structures play a role, such as domain theory (see, e.g., [AJ94]). For many purposes, it is sufficient to check that a construction is Scott continuous in order to show that it is computationally feasible.

Continuity can be seen as the independent combination of a "vertical" and a "horizontal" component. A subset of a tree is finite iff it does not contain any infinite path (which has a natural vertical representation in the usual picture of a tree) and is finitely branching (which would be represented in an horizontal way in the usual picture of a tree). As continuity is a restriction to finite sets, it is the combination of the two following properties: a first one that corresponds to a restriction to sets with no infinite path (the finite path property) and a second one that corresponds to a restriction to finitely branching sets (the finite width property). We give syntactic characterizations for these finite path and finite width properties and by combining these two results, we immediately obtain a syntactic characterization for the continuity property. Let us mention that in [Ben96], Johan van Benthem gave a version of this characterization in the setting of first-order logic.

Our proofs, though different in each case, follow a fairly uniform method, which goes back to the proofs of David Janin and Igor Walukiewicz [JW96] and

Giovanna D'Agostino and Marco Hollenberg [DH00]. For each semantic fragment F, we will exhibit an effective translation which given a μ-sentence φ, computes a formula $\tau_F(\varphi)$ in the desired syntactic fragment such that

$$\text{a } \mu\text{-sentence } \varphi \text{ has the } F\text{-property} \quad \text{iff} \quad \varphi \text{ is equivalent to } \tau_F(\varphi).$$

While the proofs of [JW96] and [DH00] rely on the tight link between μ-formulas and μ-automata, the definitions of our translations are based on the syntax only.

The chapter is organized as follows. In the first section, we introduce the main concepts and fix the notation. The second and third section are about the finite path and the finite width properties. The next section is about the continuity property, which is obtained as a combination of the two fragments previously studied. We finish by a characterization of the complete additive formulas.

5.1 Preliminaries

We start by recalling and introducing some definitions and terminology, that will play an important role in this chapter. We also recall the link between positivity and monotonicity (established in [DH00]) and give an alternative proof for this result, which follows the same scheme as most of the proofs further on. This link is essential to prove the main results of the chapter.

5.1.1 Structures and games

Terminology for models Let $\mathcal{M} = (W, R, V)$ be a Kripke model. If $U \subseteq W$, we write $\mathcal{M}[p \mapsto U]$ for the model $(W, R, V[p \mapsto U])$, where $V[p \mapsto U]$ is the valuation V' such that $V'(p) = U$ and $V'(p') = V(p)$ for all proposition letters $p' \neq p$. The model $\mathcal{M}[p \mapsto V(p) \cap U]$ will be denoted as $\mathcal{M}[p \restriction U]$.

A set $U \subseteq W$ is *downward closed* if for all $u \in U$, the predecessors of u belongs to U. A *path* through $\mathcal{M}$ is a sequence $(w_i)_{i<\kappa}$ such that $(w_i, w_{i+1}) \in R$ for all i with $i + 1 < \kappa$; here $\kappa \leq \omega$ is the *length* of the path. We let $\sqsubseteq$ denote the prefix (initial segment) relation between paths, and use $\sqsubset$ for the strict (irreflexive) version of $\sqsubseteq$. Given a path $(w_i)_{i<\kappa}$, we may occasionally also refer to the *set* $\{w_i \mid i < \kappa\}$ as a path.

Tree, branch and ω-expansion In this chapter, we are only interested in models and never consider frames. Hence, there is no confusion to write "tree" instead of "tree model".

A tree is ω-*expanded* if every node (apart from the root) has at least ω many bisimilar siblings. Given a pointed model $(\mathcal{M}, w)$ (with $\mathcal{M} = (W, R, W)$ and $w \in W$), its ω-*expansion* is the model $\mathcal{M}^\omega_w := (W^\omega, R^\omega, V^\omega)$, where W^ω, R^ω and V^ω are defined as follows. W^ω is the set of all finite sequences $w_0 k_1 w_1 \dots k_n w_n$ $(n \geq 0)$ such that $w_0 = w$, $k_i \in \omega$, $w_i \in W$ and $w_{i-1} R w_i$ for all $i > 0$. R^ω is the relation $\{(w_0k_1w_1 \dots k_nw_n, w_0k_1w_1 \dots k_nw_nkv) \mid k \in \omega \text{ and } w_nRv\}$. Finally, for all proposition letters p, $V^\omega(p)$ is the set $\{w_0k_1w_1 \dots k_nw_n \mid w_n \in V(p)\}$.

If w is clear from the context, we may simply write $\mathcal{M}^\omega$ for the ω-expansion of $(\mathcal{M}, w)$. As a particular case, if $\mathcal{T}$ is a tree with root r, we write $\mathcal{T}^\omega$ to denote the ω-expansion of $(\mathcal{T}, r)$.

5.1.1. Fact. Given a pointed model $(\mathcal{M}, w)$, the structure $\mathcal{M}^\omega_w$ is an ω-expanded tree with root w. Moreover, the pointed models$(\mathcal{M}^\omega_w, w)$ and $(\mathcal{M}, w)$ are bisimilar via the canonical bisimulation linking any point $w_0k_1s_1 \dots k_nw_n$ to w_n.

Recall that if there is a bisimulation B between two models $\mathcal{M}$ and $\mathcal{M}'$ such that (w, w') belongs to B, we write $\mathcal{M}, w \leftrightarroweq \mathcal{M}', w'$.

Terminology for games Let φ_0 be a μ-sentence, $\mathcal{M}$ a Kripke model, w and v points in $\mathcal{M}$ and φ a subformula of φ_0. We recall that $\mathcal{E}(\mathcal{M}, \varphi_0)$ denotes the evaluation game for the formula φ_0 in the model $\mathcal{M}$. We denote by $Win_\exists(\mathcal{E}(\mathcal{M}, \varphi_0))$ the set of winning positions for $\exists$ in this game. We also use the notation $\mathcal{E}(\mathcal{M}, \varphi_0)@(v, \varphi)$ for the evaluation game $\mathcal{E}(\mathcal{M}, \varphi_0)$ initiated at position (v, φ). In order to avoid confusion, we use letters of the form $\mathcal{E}, \mathcal{E}', \dots$ as abbreviations for evaluation games, while we reserve letters of the form $\mathcal{E}_0, \mathcal{E}'_0, \dots$ for evaluation games with an initial position.

Let $\mathcal{E}$ stands for either the game $\mathcal{E}(\mathcal{M}, \varphi_0)$ or the game $\mathcal{E}(\mathcal{M}, \varphi_0)@(v, \varphi)$. Given a strategy f for a player P in $\mathcal{E}$, we say that a position (u, ψ) is *f-reachable* (in $\mathcal{E}$) if there is an f-conform match of $\mathcal{E}$ during which (u, ψ) occurs. Finally, recall that a strategy f for player σ is a *maximal winning strategy* in $\mathcal{E}(\mathcal{M}, \varphi_0)$ if all winning positions z of σ are winning with respect to f in $\mathcal{E}(\mathcal{M}, \varphi_0)$.

5.1.2 Guarded and disjunctive formulas

As we are interested in syntactic characterizations, the precise shape of formulas will matter to us. Our proofs become easier if we assume certain restrictions on the use of certain connectives, without modifying the expressive power. We recall two results (already stated with more details in Chapter 2), which allow us to make such restrictions. Recall also that throughout this thesis we assume that the formulas are well-named.

5.1.2. Proposition ([Koz83]). *Each formula in* μML *can be effectively transformed into an equivalent guarded formula in* μML.

Each formula in μML^∇ *can be effectively transformed into an equivalent guarded formula in* μML^∇.

On a number of occasions it will be convenient to assume the formulas to be disjunctive.

5.1.3. Theorem. *Each μ-formula can be effectively transformed into an equivalent disjunctive guarded formula.*

We also recall the notion of a scattered strategy and its connection with disjunctive formulas (for more details, see Chapter 2).

Scattered strategy Given a state $w \in \mathcal{M}$, a strategy f for a player σ in the game $\mathcal{E}(\mathcal{M}, \varphi_0)$ with initial position (w, φ_0), is *scattered* [KV05] if for all states v in $\mathcal{M}$, for all f-conform matches $\pi = (z_i)_{i<\kappa}$ and $\pi' = (z'_i)_{i<\kappa'}$ and for all μ-formulas φ and φ',

$$z_{\kappa-1} = (v, \psi) \text{ and } z_{\kappa'-1} = (v, \psi') \quad \text{implies} \quad \pi \sqsubseteq \pi' \text{ or } \pi' \sqsubseteq \pi.$$

5.1.4. Proposition. *If a sentence $\varphi_0 \in \mu\mathrm{ML}^\nabla$ is disjunctive and $\mathcal{T}$ is an ω-expanded tree with root r, then $\mathcal{M}, r \Vdash \varphi_0$ iff there is a scattered winning strategy f for $\exists$ in $\mathcal{E}(\mathcal{M}, \varphi_0)@(r, \varphi_0)$.*

5.1.3 Expansion of a formula

We recall the notion of expansion of a formula, its link with the game semantics and introduce the notion of being active.

Expansion of a formula Given a well-named sentence φ_0, for each variable x occurring in φ_0, there is a unique formula of the form $\eta_x x.\delta_x$ which is a subformula of φ_0 (where $\eta_x \in \{\mu, \nu\}$). Moreover, we define the *dependency order* $<_{\varphi_0}$ on the variables of φ_0 as the least strict partial order such that $x <_{\varphi_0} y$ if δ_x is a subformula of δ_y.

If $\{x_1, \ldots, x_n\}$ is the set of variables occurring in φ_0, where we may assume that $i < j$ if $x_i <_{\varphi_0} x_j$, we define the *expansion* $e_{\varphi_0}(\varphi)$ of a subformula φ of φ_0 as:

$$e(\varphi) \ := \ \varphi[x_1/\eta_{x_1}.\delta_{x_1}] \ldots [x_n/\eta_{x_n}.\delta_{x_n}].$$

That is, we substitute first x_1 by δ_{x_1} in φ; in the obtained formula, we substitute x_2 by δ_{x_2}, etc. If no confusion is likely we write $e(\varphi)$ instead of $e_{\varphi_0}(\varphi)$.

Being active Let φ_0 be a μ-sentence. A proposition letter p is *active in a subformula* φ of φ_0 if p occurs in $e_{\varphi_0}(\varphi)$.

5.1.5. Proposition. *Let φ_0 be a μ-sentence, and let $(\mathcal{M}, w)$ be some pointed Kripke model. For all subformulas φ of φ_0, all $w \in W$, we have*

$$\mathcal{M}, w \Vdash e_{\varphi_0}(\varphi) \quad \textit{iff} \quad (w, \varphi) \in \mathit{Win}_\exists(\mathcal{E}(\mathcal{M}, \varphi_0)).$$

As mentioned in the introduction, in this paper we shall focus on the role of one specific proposition letter. It will be convenient from now on to fix this letter, and reserve the name 'p' for it. Formulas in which this proposition letter does not occur, will be called *p-free*.

We will often make use of the substitution $[\bot/p]$, and abbreviate $\varphi[\bot/p]$ as $\varphi^\bot$; similarly, we will write $e^\bot(\varphi)$ for $(e(\varphi))[\bot/p]$. For explicit referencing we give the following, easily proved result.

5.1.6. Fact. Let φ be a μ-formula. For all models $\mathcal{M} = (W, R, V)$, for all w in $\mathcal{M}$ and for all $\tau : \mathit{Var} \longrightarrow \mathcal{P}(W)$,

$$\mathcal{M}, w \Vdash_\tau \varphi[\bot/p] \quad \text{iff} \quad \mathcal{M}[p \mapsto \emptyset], w \Vdash_\tau \varphi.$$

5.1.4 Monotonicity and positivity

Since all the semantic properties that we study in this paper involve monotonicity, we need to recall some results on the associated preservation result.

Monotonicity A formula $\varphi_0 \in \mu\mathrm{ML} \cup \mu\mathrm{ML}^\nabla$ is *monotone* in $p \in \mathit{Prop}$ if for all models $\mathcal{M} = (W, R, V)$, all $w \in W$, all assignments $\tau : \mathit{Var} \longrightarrow \mathcal{P}(W)$ and all sets $U \supseteq V(p)$,

$$\mathcal{M}, w \Vdash \varphi_0 \quad \text{implies} \quad \mathcal{M}[p \mapsto U], w \Vdash \varphi_0.$$

Positivity The set of formulas $\mu\mathrm{ML}_+(p)$ in $\mu\mathrm{ML}$, *positive* in $p \in \mathit{Prop}$ is defined by induction in the following way:

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

where p' is a proposition letter distinct from p and x is a variable.

The set of formulas $\mu\mathrm{ML}^\nabla_+(p)$ in $\mu\mathrm{ML}^\nabla$, *positive* in $p \in \mathit{Prop}$ is defined by

$$\varphi \ ::= \ \top \mid x \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \alpha \bullet \nabla\Phi \mid \mu x.\varphi \mid \nu x.\varphi,$$

where x is a variable, Φ is a finite subset of $\mu\mathrm{ML}^\nabla$ and α is a conjunction of literals, in which $\neg p$ does not occur.

As we already mentioned, Giovanna D'Agostino and Marco Hollenberg [DH00] proved a Lyndon theorem for the modal μ-calculus, stating that $\mu\mathrm{ML}_+(p)$ characterizes monotonicity (in p). We give a simpler proof here, based on the same methods as used further on in this chapter.

Given a disjunctive sentence φ, let $\tau^d_m(\varphi)$ denote the formula we obtain by replacing each occurrence of $\neg p$ in φ by $\top$. Clearly the resulting formula is positive in p. The key observation is the following result.

5.1.7. Proposition. *Let $\varphi \in \mu\mathrm{ML}^{\nabla}$ be a disjunctive formula. Then*

$$\varphi \text{ is monotone in } p \quad \text{iff} \quad \varphi \equiv \tau^d_m(\varphi). \tag{5.1}$$

Proof We only give a sketch of the proof, as this result was already proved in [DH00]. Since the direction '$\Leftarrow$' of (5.1) is routine, we focus on the opposite direction. Consider an arbitrary model $\mathcal{T} = (T, R, V)$ and a point $r \in T$. We have to show

$$\mathcal{T}, r \Vdash \varphi \quad \text{iff} \quad \mathcal{T}, r \Vdash \tau^d_m(\varphi).$$

By Fact 5.1.1, we may assume that $\mathcal{T}$ is an ω-expanded tree with root r. The direction from left to right is easy to prove.

For the direction from right to left, assume that $\mathcal{T}, r \Vdash \tau^d_m(\varphi)$. By Fact 5.1.4, $\exists$ has a scattered winning strategy f in the game $\mathcal{E}_0 := \mathcal{E}(\mathcal{T}, \tau^d_m(\varphi))@(r, \tau^d_m(\varphi))$. Since φ is guarded, this means that for each state $t \in T$ there is at most one formula of the form $\alpha_t \bullet \nabla\Phi_t$ such that the position $(t, \alpha_t \bullet \nabla\Phi_t)$ is f-reachable from $(r, \tau^d_m(\varphi))$. Let U be the set of all states t for which such a position, with $\neg p$ occurring in α_t, is f-reachable. Using the scatteredness of f, one may show that f is (or may in the most obvious way be transformed into) a winning strategy for $\exists$ in the evaluation game $\mathcal{E}'_0 := \mathcal{E}(\mathbb{S}[p \mapsto V(p)\backslash U], \varphi)@(s, \varphi)$. (The key observation here is that if the $\mathcal{E}'_0$-match reaches a position of the form $(t, \alpha_t \bullet \nabla\Phi_t)$, there are two cases: Either $\neg p$ occurs in α, implying that $t \in U$ and $\mathcal{T}[p \mapsto V(p)\backslash U], t \Vdash \neg p$, or $\neg p$ does not occur in α, meaning that the situation is as in $\mathcal{E}_0$.) Then it follows from Fact 5.1.5 that $\mathcal{T}[p \mapsto V(p)\backslash U], s \Vdash \varphi$, and hence by monotonicity of φ we may conclude that $\mathcal{T}, r \Vdash \varphi$. □

As an almost immediate corollary of Proposition 5.1.7 we obtain that an arbitrary sentence is monotone in p iff it is equivalent to a formula that is positive in p.

5.1.8. Theorem ([DH00]). *The sentences in $\mu\mathrm{ML}_+(p)$ and in $\mu\mathrm{ML}^{\nabla}_+(p)$ are monotone in p. Moreover, there is an effective translation which, given a μ-sentence φ, computes a formula $\tau_m(\varphi) \in \mu\mathrm{ML}_+$ and a formula $\tau^{\nabla}_m(\varphi) \in \mu\mathrm{ML}^{\nabla}_+$ such that*

$$\begin{aligned} \varphi \text{ is monotone in } p \quad & \text{iff} \quad \varphi \equiv \tau_m(\varphi), \\ & \text{iff} \quad \varphi \equiv \tau^{\nabla}_m(\varphi). \end{aligned}$$

As a corollary, it is decidable whether a formula $\varphi \in \mu\mathrm{ML}$ is monotone in p.

Proof It is easy to check by induction on the complexity of the formulas that the sentences in $\mu\mathrm{ML}_+(p)$ and in $\mu\mathrm{ML}^{\nabla}_+(p)$ are monotone in p.

Next, let φ be a μ-sentence. By Theorem 5.1.3, we can compute a disjunctive formula ψ that is equivalent to φ. We can then compute $\tau^d_m(\psi)$ and it follows from Proposition 5.1.7 that ψ is monotone in p iff $\tau_m^d(\psi)$ belongs to $\mu\mathrm{ML}_+^\nabla$. Since φ and ψ are equivalent, φ is monotone in p iff ψ is monotone in p. Hence, φ is monotone in p iff $\tau_m^d(\psi)$ belongs to $\mu\mathrm{ML}_+^\nabla$. We can define $\tau_m^\nabla(\varphi)$ as $\tau_m^d(\psi)$.

Now using the fact that

$$\nabla\Phi \equiv \bigvee(\{\Diamond\chi \mid \chi \in \Phi\}) \wedge \Box \bigvee \Phi,$$

we can transform $\tau_m^\nabla(\varphi)$ into an equivalent formula $\tau_m(\varphi)$ in $\mu\mathrm{ML}_+$. $\Box$

5.2 Finite path property

The first property that we consider is that of the finite path property.

Finite path property A sentence φ_0 has the *finite path property* for $p \in \mathit{Prop}$ if φ_0 is monotone in p and for every tree $\mathcal{T}$ with root r,

$$\mathcal{T}, r \Vdash \varphi_0 \quad \text{iff} \quad \mathcal{T}[p{\restriction}U], r \Vdash \varphi_0, \text{ for some } U \subseteq S \text{ which is downward closed and does not contain any infinite path.} \tag{5.2}$$

Note that monotonicity needs to be specified explicitly, since it does not follow from the equivalence (5.2): a simple counterexample is given by the formula $\neg p \wedge \Diamond p$. Observe as well that we only require condition (5.2) to hold for trees, since this condition would not make much sense on any model with circular paths.

The syntactic fragment of μML that corresponds to this property is given as follows.

The fragment $\mu\mathrm{ML}_D(p)$ We define the fragment $\mu\mathrm{ML}_D(p)$ by induction in the following way:

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

where x is a variable and ψ is a sentence which does not contain p.

The following theorem states that modulo equivalence, $\mu\mathrm{ML}_D(p)$ exactly captures the fragment of the modal μ-calculus that has the finite path property. In addition, the problem, whether a sentence has this property, is decidable.

5.2.1. Theorem. *The μ-sentences in $\mu\mathrm{ML}_D(p)$ have the finite path property for p. Moreover, there is an effective translation which given a μ-sentence φ, computes a formula $\varphi^d \in \mu\mathrm{ML}_D(p)$ such that*

$$\varphi \text{ has the finite path property for } p \quad \text{iff} \quad \varphi \equiv \varphi^d. \tag{5.3}$$

As a corollary, it is decidable whether a given sentence φ has the finite path property for p.

Before we turn to the proof of this theorem, we state and prove two auxiliary results. First we show that formulas in $\mu\mathrm{ML}_D(p)$ indeed have the finite path property.

5.2.2. PROPOSITION. *All sentences in $\mu\mathrm{ML}_D(p)$ have the finite path property with respect to p.*

Proof Fix a sentence φ_0 in $\mu\mathrm{ML}_D(p)$ and a tree $\mathcal{T} = (T, R, V)$ and with root r. We have to show

$$\mathcal{T}, r \Vdash \varphi_0 \quad \text{iff} \quad \mathcal{T}[p{\restriction}U], r \Vdash \varphi_0, \text{ for some } U \subseteq T \text{ which is downward closed and does not contain any infinite path.}$$

The direction from right to left follows from Theorem 5.1.8 and the fact that φ_0 is positive in p. For the opposite direction, suppose that $\mathcal{T}, r \Vdash \varphi_0$. We need to find a subset $U \subseteq T$ which is downward closed, does not contain any infinite path and such that $\mathcal{T}[p{\restriction}U], r \Vdash \varphi_0$. Since $\mathcal{T}, r \Vdash \varphi_0$, $\exists$ has a positional winning strategy f in the game $\mathcal{E}_0 := \mathcal{E}(\mathcal{T}, \varphi_0)@(r, \varphi_0)$. We define $U \subseteq T$ such that

$$u \in U \quad \text{iff} \quad \text{there is } \varphi \text{ such that } (u, \varphi) \text{ is } f\text{-reachable in } \mathcal{E}_0 \text{ and } p \text{ is active in } \varphi.$$

It is not difficult to see that U is downward closed: If a position (u, φ) occurs in an $\mathcal{E}_0$-match π and p is not active in φ, then all the positions occurring after (u, φ) are of the form (v, ψ), where p is not active in ψ.

Hence it suffices to show that U does not contain any infinite path. Suppose for contradiction that U contains an infinite path P. We let $\mathcal{A}$ be the set of all finite f-conform $\mathcal{E}_0$-matches π such that for all positions (u, φ) occurring in π, u belongs to P and p is active in φ. Recall that $\sqsubseteq$ denotes the initial-segment relation on paths and on matches.

Clearly, the structure $(\mathcal{A}, \sqsubseteq)$ is a tree. Moreover, it is finitely branching, as P is a single path, and all the formulas occurring in matches in $\mathcal{A}$ belongs to the finite set $\mathit{Sfor}(\varphi_0)$. Next we show that the set $\mathcal{A}$ is infinite. It suffices to define an injective map h from P to $\mathcal{A}$. Fix t in P. In particular, t belongs to U and by definition of U, there is a formula φ such that (t, φ) is f-reachable in $\mathcal{E}_0$ and p is active in ψ. We let $h(t)$ be a finite f-conform $\mathcal{E}_0$-match with last position (t, φ). It is easy to check that any such map h is an injection from P to $\mathcal{A}$.

By König's lemma, since $(\mathcal{A}, \sqsubseteq)$ is infinite and finitely branching, it must contain an infinite path. This infinite path corresponds to an infinite f-conform $\mathcal{E}_0$-match π such that for all positions (t, φ) occurring in π, t belongs to P and p is active in φ. Since φ_0 belongs to the fragment $\mu\mathrm{ML}_D(p)$, this can only happen if all the variables unfolded in π are μ-variables. This implies that π is lost by $\exists$ and this contradicts the fact that f is a winning strategy for $\exists$ in $\mathcal{E}_0$.

It remains to show that $\mathcal{T}[p{\restriction}U], r \Vdash \varphi_0$. Let $\mathcal{E}_0'$ be the game $\mathcal{E}(\mathcal{T}[p{\restriction}U], \varphi_0)@(r, \varphi_0)$. We show that f itself is a winning strategy for $\exists$ in the game $\mathcal{E}_0'$. The winning conditions for $\mathcal{E}_0$ and $\mathcal{E}'_0$ are the same. Moreover, the rules of the two games are the same, except when we reach a position of the form (t, p). So to prove that f is a winning strategy for $\exists$ in $\mathcal{E}'_0$, it suffices to show that if an f-conform $\mathcal{E}'_0$-match π arrives at a position (t, p), then $\mathcal{T}[p{\restriction}U], t \Vdash p$, that is, $t \in V(p) \cap U$. Suppose that we are in this situation. Since π is also an f-conform $\mathcal{E}_0$-match and since f is a winning strategy for $\exists$ in $\mathcal{E}_0$, t belongs to $V(p)$. It remains to show that t belongs to U. That is, we have to find a formula φ such that (t, φ) is f-reachable in $\mathcal{E}_0$ and p is active in φ. Clearly, the formula p itself satisfies these two conditions. This proves that f is a winning strategy for $\exists$ in the game $\mathcal{E}'_0$ and hence shows that $\mathcal{T}[p{\restriction}U], r \Vdash \varphi_0$. □

The hard part in the proof of Theorem 5.2.1 is to show that any sentence φ_0 with the finite path property can be rewritten into an equivalent sentence $\tau_d(\varphi_0) \in \mu\mathrm{ML}_D$. First we define the translation τ_d.

The translation τ_d Fix a positive sentence φ_0. We define the map $\tau_d : \mathit{Sfor}(\varphi_0) \to \mu\mathrm{ML}$ by induction on the complexity of subformulas of φ_0:

$$
\begin{aligned}
\tau_d(l) &= l, \\
\tau_d(\varphi \vee \psi) &= \tau_d(\varphi) \vee \tau_d(\psi), \\
\tau_d(\varphi \wedge \psi) &= \tau_d(\varphi) \wedge \tau_d(\psi), \\
\tau_d(\Box\varphi) &= \Box\tau_d(\varphi), \\
\tau_d(\Diamond\varphi) &= \Diamond\tau_d(\varphi), \\
\tau_d(\mu x.\varphi) &= \mu x.\tau_d(\varphi), \\
\tau_d(\nu x.\varphi) &= \mu x.\left(\tau_d(\varphi) \vee e^{\perp}(\varphi)\right),
\end{aligned}
$$

where l is a literal, a variable, or one of the constants $\perp$ or $\top$.

The translation τ_d maps a positive sentence to a sentence in $\mu\mathrm{ML}_D(p)$. So given a positive sentence φ_0, we have to modify the subformulas of the form $\nu x.\varphi$, in which p occurs. The idea is as follows. If we play the evaluation game for φ_0 and if we assume φ_0 to be guarded and with the finite depth property for p, then after finitely many steps, we are sure that in the game we will never encounter points at which p is true. So after finitely many steps, we can simply replace $\nu x.\varphi$ by $e^{\perp}(\nu x.\varphi)$, or equivalently, by $e^{\perp}(\varphi)$. This is captured by the last clause of the definition of τ_d.

The following proposition is the key technical lemma of this section.

5.2.3. PROPOSITION. *A positive guarded sentence φ_0 has the finite path property with respect to proposition letter p iff φ_0 is equivalent to $\tau_d(\varphi_0)$.*

Proof Fix a positive guarded sentence φ_0. The direction from right to left of this Proposition is an immediate consequence of Proposition 5.2.2 and the observation that $\tau_d(\varphi_0)$ belongs to the fragment $\mu\mathrm{ML}_D(p)$.

For the opposite direction, assume that φ_0 has the finite path property with respect to p. In order to prove that φ_0 is equivalent to $\tau_d(\varphi_0)$, consider an arbitrary pointed model $(\mathcal{T}, r)$.

We have to show that

$$\mathcal{T}, r \Vdash \varphi_0 \quad \text{iff} \quad \mathcal{T}, r \Vdash \tau_d(\varphi_0). \tag{5.4}$$

By Fact 5.1.1, we may assume that $\mathcal{T}$ is a tree with root r.

For the direction '$\Rightarrow$' of (5.4), assume that $\mathcal{T}, r \Vdash \varphi_0$. There is some subset $U \subseteq S$ which is downward closed (implying that $r \in U$), has no infinite paths, and satisfies $\mathcal{T}[p{\upharpoonright}U], r \Vdash \varphi_0$.

Therefore, (r, φ_0) is a winning position for $\exists$ in the evaluation game $\mathcal{E}'_0 := \mathcal{E}(\mathcal{T}[p{\upharpoonright}U], \varphi_0)@(r, \varphi_0)$. Let f denote some positional winning strategy of $\exists$ in the game $\mathcal{E}'_0$; also fix some maximal positional winning strategy g for $\exists$ in the evaluation game $\mathcal{E}^d := \mathcal{E}(\mathcal{T}, \tau_d(\varphi_0))$. In order to prove that $\mathcal{T}, r \Vdash \tau_d(\varphi_0)$, it suffices to provide $\exists$ with a winning strategy in the game $\mathcal{E}^d$ initialized at (r, φ_0).

The key idea underlying the definition of this winning strategy h, is that $\exists$ maintains, during an initial part of the $\mathcal{E}^d$-match π, an f-conform shadow match π' of $\mathcal{E}'_0$. The relation between π and π' will be that at all times,

with $last(\pi) = (t, \psi)$ and $last(\pi') = (t', \psi')$, we have

$$t = t' \text{ and } \psi \in \{\tau_d(\psi'), \tau_d(\psi') \vee e^{\perp}(\psi')\}. \tag{†}$$

1. Claim. $\exists$ has a strategy h_0 that enables her to maintain the condition (†) for as long as she pleases (unless she wins the match at some finite stage).

Proof of Claim It is obvious that (†) holds at the beginning of the two matches, with π and π' consisting of the single positions $(r, \tau_d(\varphi_0))$ and (r, φ_0), respectively. Inductively, suppose that during the play of an $\mathcal{E}^d$-match, $\exists$ has managed to maintain an f-conform $\mathcal{E}'_0$-match, and that play has arrived at the respective partial matches π and π', satisfying the condition (†). Let t, t', ψ and ψ' be as in (†). In order to show that $\exists$ can push the condition forward, we distinguish cases.

First of all, in case ψ is of the form $\tau_d(\psi') \vee e^{\perp}(\psi')$ then in the $\mathcal{E}^d$-match, $\exists$ chooses the disjunct $\tau_d(\psi')$, making $(t, \tau_d(\psi'))$ the next position. Clearly then the two partial matches, $\pi(t, \tau_d(\psi'))$ and π', still satisfy (†).

If, on the other hand, ψ is of the form $\tau_d(\psi')$, then we make a further case distinction as to the nature of ψ'. If $\psi' = p$ or $\psi' \in \{q, \neg q\}$ for some proposition letter q different from p, then both π and π' are full matches. Now π', being f-conform, is won by $\exists$. This means that $\mathcal{T}[p{\upharpoonright}U], t \Vdash \psi'$, and so in all three different cases we also have $\mathcal{T}, t \Vdash \psi'$, which implies that $\exists$ is also the winner of π. The cases where ψ' is a conjunction, disjunction, or of the form $\Diamond\chi$ or $\Box\chi$, are routine, and so are the cases where ψ' is a variable x or a fixpoint formula $\mu x.\chi$.

This leaves the case where ψ' is of the form $\nu x.\chi$, meaning that (u, χ) is the next position in the $\mathcal{E}'_0$-match. The corresponding last position of π is

$(t, \mu x.\tau_d(\chi) \vee e^{\perp}(\chi))$, and from there an automatic move leads to position $(t, \tau_d(\chi) \vee e^{\perp}(\chi))$ which is of the required shape for (†).

Thus in each case either $\exists$ manages to win the match, or else condition (†) is maintained one more step. This suffices to prove the claim. ◀

However, $\exists$'s strategy in an $\mathcal{E}^d$-match will not be to maintain the condition (†) indefinitely. Rather, her strategy h will be as follows: She plays strategy h_0, until the $\mathcal{E}^d$-match arrives at a position of the form $(t, \tau_d(\psi') \vee e^{\perp}(\psi'))$ such that the second disjunct, $e^{\perp}(\psi')$, is true at t in $\mathcal{T}$. Should she arrive at such a position, she picks this second disjunct, moving to position $(t, e^{\perp}(\psi'))$. By Proposition 5.1.5, and since p is not active in $e^{\perp}(\psi')$, this position is winning for her in $\mathcal{E}^d$, and so from this moment on she plays her (positional) winning strategy g of $\mathcal{E}^d$.

In order to prove that this is a winning strategy for her, suppose for contradiction that π is an h-conform match which is lost by $\exists$. This means that $\exists$ never swaps to her winning strategy g but keeps playing her strategy h_0, during the entire match π. As a consequence, the shadow match π' is infinite as well, and it easily follows from (†) that the sequence of unfolded fixpoint variables is the same in π as in π'. But π', being conform to her winning strategy f in $\mathcal{E}'_0$, is won by $\exists$. In other words, the highest variable unfolded infinitely often in π', say x, is a ν-variable (with respect to φ_0).

Observe that since $\tau_d(\varphi_0)$ is guarded and U does not contain any infinite paths, at a certain moment the infinite match π will leave U. Since x is unfolded infinitely often, both in π' and in π, there is a first unfolding of x in π after the match has left U. Suppose that this unfolding happens at position (t, x), and note that the corresponding position in π' is also (t, x). In π' the next position after (t, x) is (t, δ_x), and since π' is f-conform, this is a winning position for $\exists$ in $\mathcal{E}'_0$. It follows from Proposition 5.1.5 that $\mathcal{T}[p{\restriction}U], t \Vdash e(\delta_x)$. However, since u does not belong to U, and U is downward closed, we may infer by Fact 5.1.6 that $\mathcal{T}, t \Vdash e^{\perp}(\delta_x)$. This would mean that in π, where the next position after (t, x) is $(t, \tau_d(\delta_x) \vee e^{\perp}(\delta_x))$, $\exists$ would be in the position to pick the second disjunct and jump to the winning strategy g after all. Thus we have arrived at the desired contradiction, which means that h is a winning strategy for $\exists$.

For the opposite direction '$\Leftarrow$' of (5.4), assume that $\mathcal{T}, r \Vdash \tau_d(\varphi_0)$, and let f^d be a positional winning strategy for $\exists$ in the game $\mathcal{E}^d_0 = \mathcal{E}(\mathcal{T}, \tau_d(\varphi_0))@(r, \tau_d(\varphi_0))$. In order to prove that $\mathcal{T}, r \Vdash \varphi_0$, we provide her with a winning strategy in the game $\mathcal{E} := \mathcal{E}(\mathcal{T}, \varphi_0)$ initiated ar (r, φ_0).

As before, our proof is based on $\exists$ maintaining, during an initial part of the $\mathcal{E}$-match π, an f^d-conform shadow match π^d of $\mathcal{E}^d_0$. Inductively, we will make sure that $\exists$ can keep the following constraint on the two matches:

$$\text{with } \mathit{last}(\pi) = (t, \psi), \text{ we have } \mathit{last}(\pi^d) = (t, \psi^d) \quad \text{with } \psi^d \in \{\tau_d(\psi), \tau_d(\psi) \vee e^{\perp}(\psi)\}. \quad (‡)$$

Analogous to Claim 1 above, it is straightforward to prove that $\exists$ can maintain this condition until she either wins the $\mathcal{E}$-match, or else the two matches arrive at positions (t, ψ) and $(t, \tau_d(\psi) \vee e^{\perp}(\psi))$, respectively, where in the shadow game $\exists$'s strategy f^d tells her to pick the second disjunct, $e^{\perp}(\psi)$.

Once this happens, $\exists$ has a guaranteed win in $\mathcal{E}$: Since she plays π^d based on her winning strategy f^d, the pair $(t, e^{\perp}(\psi))$ is a winning position for her in $\mathcal{E}_0^d$, and so $\mathcal{T}, t \Vdash e^{\perp}(\psi)$ by Proposition 5.1.5 (note that $e^{\perp}(\psi)$ is a sentence). Since φ_0 is positive in p, ψ is also positive in p. So $e^{\perp}(\psi)$ implies $e(\psi)$. It follows that $\mathcal{T}, t \Vdash e(\psi)$, and so by Proposition 5.1.5 again, we obtain that $(t, \psi) \in \mathit{Win}_{\exists}(\mathcal{E})$. Hence she has no further need of the shadow match, and can continue π by following any winning strategy of $\mathcal{E}$.

Should, on the other hand, $\exists$ never leave the shadow match by moving to a position of the form $(t, e^{\perp}(\psi))$, then she maintains the shadow match forever. In this case, it follows directly from (‡) that the resulting infinite matches π and π^d have the same sequence of unfolded fixpoint variables. Now observe that π^d, being f^d-conform, is won by $\exists$; in other words, the highest variable unfolded infinitely often during π^d, is a ν-variable with respect to $\tau_d(\varphi_0)$. But then this variable is also a ν-variable with respect to φ_0, and from this it is immediate that $\exists$ is also the winner of π. □

We finish the section with the proof of its main theorem.

Proof of Theorem 5.2.1. We already know by Proposition 5.2.2 that the sentences in $\mu\mathrm{ML}_D(p)$ have the finite path property for p.

Next, for an arbitrary μ-sentence φ, we define $\varphi^d := \tau_d(\tau_m(\varphi))$, where τ_m is the translation of Theorem 5.1.8. Then the equivalence (5.3) is immediate by Theorem 5.1.8 and Proposition 5.2.3.

The decidability of the finite path property follows by the observation that the construction of the formula φ^d from φ is effective, and that it is decidable whether φ and φ^d are equivalent. □

5.2.4. Remark. Since our main interest in this chapter is model-theoretic, we have not undertaken an in-depth study of the size of the formula φ^d (in Theorem 5.2.1) or of the exact complexity of the problem of deciding whether a given μ-sentence has the finite path property. However, we would like to make few remarks in that respect.

While at first sight, the clause for the greatest fixpoint operator in the definition of τ_d may seem to create exponentially long formulas, given our definition of size as the number of elements in the closure of a formula, one may show that the size of $\tau_d(\varphi_0)$ is in fact linear in the size of φ_0. This is a consequence of the two following equalities which can be proved by induction on the complexity of the formulas:

$$\begin{aligned} Cl(e^{\perp}_{\varphi_0}(\varphi)) &\subseteq \{e^{\perp}_{\varphi_0}(\psi) \mid \psi \trianglelefteq \varphi\}, \\ Cl(\tau_d(\varphi)) &\subseteq \{e_{\tau_d(\varphi)}(\tau_d(\psi)) \mid \psi \trianglelefteq \varphi\} \cup Cl(\varphi_0^{\perp}), \end{aligned}$$

where φ_0 is a μ-sentence and φ is a subformula of φ_0.

However, in order to obtain an upper bound for the size of φ^d, it follows from the proof of Theorem 5.2.1 that we also need to investigate the complexity of the translation τ_m of Theorem 5.1.8. To perform the translation τ_m, we first have to transform a μ-sentence into an equivalent disjunctive sentence. To our knowledge, the exact complexity of this transformation and the size of the obtained formula has not been studied yet in detail. Nevertheless, the complexity and the size are known to be elementary (see for instance [Jan96]). It follows that the size of φ^d is elementary in the size of φ and that it is decidable in elementary time whether a given μ-sentence has the finite path property.

Another interesting remark is that the transformations τ_m and τ_d, which are the key ingredients for proving Theorem 5.2.1, could also have been defined at the level of automata. More precisely, given a sentence φ and a non-deterministic automaton $\mathbb{A} = (Q, q_I, \delta, \Omega)$ equivalent to φ, we can define the automata $\tau_m(\mathbb{A})$ and $\tau_d(\mathbb{A})$ such that

$$\begin{array}{rcl} \varphi \text{ is monotone in } p & \text{iff} & \varphi \equiv \tau_m(\mathbb{A}), \\ \varphi \text{ has the finite path property for } p & \text{iff} & \varphi \equiv \tau_m(\tau_d(\mathbb{A})). \end{array}$$

Moreover, if we employ the "usual" procedure to transform $\tau_m(\mathbb{A})$ and $\tau_m(\tau_d(\mathbb{A}))$ into μ-sentences (see Theorem 11.6 of [GTW02]), these automata correspond to formulas in $\mu\mathrm{ML}^{\nabla}_{+}(p)$ and $\mu\mathrm{ML}^{\nabla}_{d}(p)$ respectively.

We do not want to give too many details, but let us give a brief description of the automata $\tau_m(\mathbb{A})$ and $\tau_d(\mathbb{A})$. The automaton $\tau_m(\mathbb{A})$ is simply obtained by replacing $\neg p$ by $\top$ in each formula $\delta(q)$ (with $q \in Q$). The automaton $\tau_d(\mathbb{A}) = (Q_d, q_d, \delta_d, \Omega_d)$ is given by:

$$\begin{array}{rcl} Q^d & = & Q \times \{0, 1\}, \\ q_d & = & (q_I, 0), \\ \delta(q, 0) & = & \delta(q)[q'/(q', 0)] \vee \delta(q)[q'/(q', 1)], \\ \delta(q, 1) & = & \delta(q)[p/\top][q'/(q', 1)], \\ \Omega(q, i) & = & \begin{cases} 1 & \text{if } i = 0, \\ \Omega(q) & \text{if } i = 1, \end{cases} \end{array}$$

where $\delta(q)[q'/(q', i)]$ is the formula $\delta(q)$ in which each occurrence of a state $q' \in Q$ is replaced by (q', i) and where $\delta(q)[p/\top][q'/(q', 1)]$ is the formula $\delta(q)$ in which each occurrence of p is replaced by $\top$ and each occurrence of a state $q' \in Q$ is replaced by $(q', 1)$. Intuitively, $\tau_d(\mathbb{A})$ consists of two copies of $\mathbb{A}$, corresponding respectively to the 'initial part' of the model where p still might be true, and the 'final part' where p is false. The link between these two parts is given by the clause of the definition for $\delta(q, 0)$. The disjunction of $\delta(q)[q'/(q', 0)]$ and $\delta(q)[q'/(q', 1)]$ offered here corresponds to the disjunction in the ν-clause in the definition of the translation τ_d for μ-formulas.

The definitions of $\tau_m(\mathbb{A})$ and $\tau_d(\mathbb{A})$ consist in mimicking the translations τ_m and τ_d defined in the context of formulas. Using arguments similar to the proofs of Proposition 5.1.7 and Proposition 5.2.3, we can show that $\tau_m(\mathbb{A})$ and $\tau_d(\mathbb{A})$ satisfy the required properties.

We believe that this approach might be more appropriate if we are interested in complexity issues. For example, suppose that we want to check whether a sentence is monotone. In order to apply the translation τ_m from Theorem 5.1.8, we first transform the sentence into a disjunctive sentence. The method we use to transform a sentence into a disjunctive sentence is to transform the sentence into a non-deterministic automata and then transform this automata into a formula. Hence, we have to move to the context of automata anyway. The point is that, in order to get a better complexity result, it might be a good idea to stay at the level of automata and apply the translation τ_m at that level (as described in the previous paragraphs).

5.3 Finite width property

The second property that we are interested in is that of the finite width property.

Finite width property A μ-sentence φ_0 has the *finite width property* for $p \in$ *Prop* if φ_0 is monotone in φ and for all trees $\mathcal{T} = (T, R, V)$ with root r,

$$\mathcal{T}, r \Vdash \varphi_0 \quad \text{iff} \quad \mathcal{T}[p{\restriction}U], r \Vdash \varphi_0, \text{ for some } U \subseteq T \text{ which is downward closed and finitely branching,}$$

where we call a subset $U \subseteq T$ *finitely branching* if the set $R[u] \cap U$ is finite for every $u \in U$.

The syntactic fragment associated with this property is given as follows.

The fragment $\mu\text{ML}_W(p)$ We define the fragment $\mu\text{ML}_W(p)$ by induction in the following way:

$$\varphi \ ::= \ p \mid x \mid \psi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \mu x.\varphi \mid \nu x.\varphi,$$

where x is a variable and ψ is a sentence which does not contain p.

The following theorem states that modulo equivalence, $\mu\text{ML}_W(p)$ exactly captures the fragment of the modal μ-calculus that has the finite width property.

5.3.1. THEOREM. *The μ-sentences in $\mu\text{ML}_W(p)$ have the finite width property for p. Moreover, there is an effective translation which given a μ-sentence φ, computes a formula $\varphi^w \in \mu\text{ML}_W(p)$ such that*

$$\varphi \text{ has the finite width property for } p \quad \text{iff} \quad \varphi \equiv \varphi^w. \tag{5.5}$$

As a corollary, it is decidable whether a given sentence φ has the finite width property for p.

The proof of Theorem 5.3.1 follows the same lines as that of Theorem 5.2.1. First we show that formulas in the fragment $\mu\mathrm{ML}_W(p)$ indeed have the required property.

5.3.2. Proposition. *If a sentence φ_0 belongs to $\mu\mathrm{ML}_W(p)$, then φ_0 has the finite width property with respect to p.*

Proof Let φ_0 be a sentence in $\mu\mathrm{ML}_W(p)$. Fix a tree $\mathcal{T} = (T, R, V)$ with root r. We have to prove

$$\mathcal{T}, r \Vdash \varphi_0 \quad \text{iff} \quad \mathcal{T}[p{\restriction}U], r \Vdash \varphi_0, \text{ for some } U \subseteq T \text{ which is downward closed and finitely branching.} \tag{5.6}$$

The direction from right to left follows from Theorem 5.1.8 and the fact that φ_0 is positive in p. For the direction from left to right, suppose that $\mathcal{T}, r \Vdash \varphi_0$. We need to find a finitely branching subset U of T that is downward closed and such that $\mathcal{T}[p{\restriction}U] \Vdash \varphi_0$. Let f be a positional winning strategy of $\exists$ in the game $\mathcal{E}_0 := \mathcal{E}(\mathcal{T}, \varphi_0)@(r, \varphi_0)$. We define $U \subseteq S$ such that

$$u \in U \quad \text{iff} \quad \text{there is } \varphi \text{ such that } (u, \varphi) \text{ is } f\text{-reachable in } \mathcal{E}_0 \text{ and } p \text{ is active in } \varphi.$$

The set U is downward closed. Indeed, if a position (u, φ) is reached during an $\mathcal{E}_0$-match π and p is not active in φ, then all positions occurring after (u, φ) will be of the form (v, ψ), where p is not active in ψ.

Hence it suffices to show that U is finitely branching. Fix $u \in U$ and let us show that $R[u] \cap U$ is finite. Let $v \in U$ be a successor of u. Since u is the only predecessor of v, by definition of U, there must be an f-conform match during which a move occurs from $(u, \triangle\varphi_v)$ to (t, φ_v), where $\triangle \in \{\Box, \Diamond\}$ and φ_v is a subformula of φ_0 such that p is active in φ_v. By definition of $\mu\mathrm{ML}_W(p)$, this can only happen if $\triangle = \Diamond$. But then, $(u, \Diamond\varphi_v)$ is a position which belongs to $\exists$ and so v is her choice as dictated by f. From this, it follows that for all v and v' in $R[u] \cap U$, we have $\varphi_v \neq \varphi_{v'}$ if $v \neq v'$. Putting this together with the fact that $Sfor(\varphi_0)$ is finite, we obtain that $R[u] \cap U$ is finite. This finishes the proof that U is downward closed and finitely branching.

It remains to show that $\mathcal{T}[p{\restriction}U], r \Vdash \varphi_0$. Here we omit the details since the proof is similar to the one in Proposition 5.2.2. □

As before, the hard part of the proof will consist in showing that any μ-sentence φ_0 with the finite width property can be effectively rewritten into a formula $\tau_w(\varphi_0)$ in the fragment $\mu\mathrm{ML}_W$. In order to define the translation τ_w, it will be convenient to use the ∇-syntax for the source formulas.

The translation τ_w Given a positive sentence $\varphi \in \mu\mathrm{ML}^\nabla$, we define its translation $\tau_w(\varphi)$ by the following induction:

$$\begin{aligned}
\tau_w(\top) &= \top, \\
\tau_w(x) &= x, \\
\tau_w(\varphi \vee \psi) &= \tau_w(\varphi) \vee \tau_w(\psi), \\
\tau_w(\varphi \wedge \psi) &= \tau_w(\varphi) \wedge \tau_w(\psi), \\
\tau_w(\alpha \bullet \nabla\Phi) &= \alpha \wedge \bigvee \Big\{ \bigwedge \{\Diamond \tau_w(\varphi) \mid \varphi \in \Phi_1\} \wedge \nabla \left(e^\perp[\Phi_2]\right) \mid \Phi_1 \cup \Phi_2 = \Phi \Big\}, \\
\tau_w(\eta x.\varphi) &= \eta x.\tau_w(\varphi),
\end{aligned}$$

where α is a conjunction of literals, Φ is a finite set of formulas in $\mu\mathrm{ML}^\nabla$, and $\eta \in \{\mu, \nu\}$.

The intuition for τ_w is as follows. We want τ_w to map a positive sentence to a sentence in $\mu\mathrm{ML}_W(p)$. So we need to replace the subformula of the form $\alpha \bullet \nabla\Phi$ by a formula in which p is not in the scope of a $\Box$ operator. When we reach a position of the form $(t, \alpha \bullet \nabla\Phi)$ in the evaluation game and if φ_0 has the finite width property for p, we know that there are only finitely many points $t_1, \dots, t_n$ in $R[t]$ such that p is true at a point of the model generated by t_i.

In the evaluation game, at position $(t, \alpha \bullet \nabla\Phi)$, $\exists$ has to come up with a marking $m : \Phi \longrightarrow \mathcal{P}(R[t])$. If a point in $R[t] \backslash \{t_1, \dots, t_n\}$ is marked with a formula ψ, we can simply replace ψ by $e^\perp(\psi)$. So in the definition of $\tau_w(\alpha \bullet \nabla\Phi)$, Φ_2 corresponds to the set of formulas ψ such that a point in $R[t] \backslash \{t_1, \dots, t_n\}$ is marked with ψ. Φ_1 corresponds to the set of formulas ψ such that some t_i is marked with ψ.

5.3.3. Proposition. *A positive sentence $\varphi_0 \in \mu\mathrm{ML}^\nabla$ has the finite width property with respect to proposition letter p iff φ_0 is equivalent to $\tau_w(\varphi_0)$.*

Proof The direction from right to left of this proposition is an immediate consequence of Proposition 5.3.2 and the observation that τ_w maps formulas in ∇-format to formulas in the fragment $\mu\mathrm{ML}_W(p)$. For the opposite direction, assume that φ_0 has the finite width property with respect to p. In order to prove that $\varphi_0 \equiv \tau_w(\varphi_0)$, consider an arbitrary pointed model $(\mathcal{T}, r)$. We will show that

$$\mathcal{T}, r \Vdash \varphi_0 \quad \text{iff} \quad \mathcal{T}, r \Vdash \tau_w(\varphi_0). \tag{5.7}$$

By Fact 5.1.1, we may assume that $\mathcal{T}$ is an ω-unravelled tree with root r.

For the direction '$\Rightarrow$' of (5.7), assume that $\mathcal{T}, r \Vdash \varphi_0$. Then by the finite width property there is some downward closed, finitely branching subset U of S such that $\mathcal{T}[p{\restriction}U], r \Vdash \varphi_0$. Since U is downward closed, r belongs to U. By Proposition 5.1.5 we may assume that $\exists$ has a positional winning strategy f in the evaluation game $\mathcal{E}_0' := \mathcal{E}(\mathcal{T}[p{\restriction}U], \varphi_0)@(r, \varphi_0)$. We fix also some maximal positional winning strategy g for $\exists$ in the evaluation game $\mathcal{E}^w := \mathcal{E}(\mathcal{T}, \tau_w(\varphi_0))$. In order to prove that $\mathcal{T}, r \Vdash \tau_w(\varphi_0)$, it suffices to provide $\exists$ with a winning strategy in the game $\mathcal{E}^w$ initialized at (r, φ_0).

The key idea underlying the definition of this winning strategy h, is that $\exists$ maintains, during (an initial part of) the $\mathcal{E}^w$-match, an f-conform shadow match of $\mathcal{E}'_0$. Inductively we will make sure that $\exists$ can keep the following condition (†) on the partial $\mathcal{E}^w$-match $\pi = z_0 \dots z_n$ and its shadow match $\pi' = z'_0 \dots z'_k$ (where $z_0 = (r, \tau_w(\varphi_0))$ and $z'_0 = (r, \varphi_0)$):

(†)

First of all, if $\exists$ is to move at z_n, then she will not get stuck.
Furthermore, one of the following two constraints is satisfied:
(i) there is some $m \leq n$ with $z_m \in \mathit{Win}_\exists(\mathcal{E}^w)$, and $z_m z_{m+1} \dots z_n$ is a g-conform $\mathcal{E}^w$-match;
(ii) there is an order preserving partial map $b : \{0, \dots, k\} \longrightarrow \{0, \dots, n\}$ such that
(a) $b(0) = 0$,
(b) if $b(i) = j$ then for some $u \in U$ and some φ, z_i and z'_j are of the form (u, φ) and $(u, \tau_w(\varphi))$,
(c) for all variables x, if $z_i = (t, x)$, then $i \in \mathit{Dom}(b)$, and if $z'_j = (t, x)$, then $j \in \mathit{Ran}(b)$.

Here we call a partial map b between two sets of natural numbers an *order preserving partial map* if for all $i < j$ such that $i, j \in \mathit{Dom}(b)$, we have $b(i) < b(j)$.

Two observations on (†) may be in order. First, the idea behind condition (†-i) is that $\exists$ no longer needs to maintain the shadow match if she arrives at a position z_m that is already known to be winning for her in $\mathcal{E}^w$. Once this happens, following her winning strategy g will guarantee that she wins the $\mathcal{E}^w$-match. Second, the aim of condition (†-ii.c) is to ensure that every position involving a variable is linked to some position in the other match. Condition (ii.b) then guarantees that the companion position involves the same variable. As a corollary, in case (ii) the sequence of variables that get unfolded in π is identical to the sequence of unfolded variables in π'.

Let us first show why $\exists$ is guaranteed to win any match in which she can keep the condition (†). Consider such a (full) match π. It is an immediate consequence of (†) that $\exists$ will not get stuck during π, and so if π is finite she will be its winner. Assume then that π is infinite. If at some moment the match arrived at a position z_m that is winning for $\exists$ in $\mathcal{E}^w$, then by (†-i) the tail $z_m z_{m+1} \dots$ of π is conform her winning strategy in $\mathcal{E}^w$, and so she will be the winner of π. This leaves the case where $\exists$ needs to maintain the shadow match during the entire match π. Thus in the limit she creates an infinite $\mathcal{E}'_0$-match π' that is conform f, and linked to π by an order preserving partial map $b : \omega \longrightarrow \omega$ satisfying the conditions (†-ii:a-c). We already observed that the respective sequences of variables that get unfolded in π and π' are identical, and so the winners of π and π' are identical as well. But the match π' starts at a winning position for $\exists$, and she is assumed to play according to her winning strategy f. Clearly then it is $\exists$ who wins π', and therefore, she must be the winner of π as well.

Let us now see how $\exists$ can manage to maintain the condition (†) during the match. The following claim will be the key instrument in pushing (†) forward. Here, a local strategy is simply a strategy but we use the word "local" to emphasize the fact that this strategy will help us to define, round by round, the global strategy for $\exists$ in the game $\mathcal{E}^w$ initialized at (s, φ_0). Recall that $\sqsubseteq$ is the prefix (initial segment) relation between sequences.

1. Claim. If $t \in U$ and $(t, \varphi') \in Win_\exists(\mathcal{E}'_0)$, then $\exists$ has a local strategy h in $\mathcal{E}^w@(t, \tau_w(\varphi'))$ with the property that for all h-conform matches λ, there exists an h-conform match ρ with last position z satisfying ($\rho \sqsubseteq \lambda$ or $\lambda \sqsubseteq \rho$) and condition (a) or (b) below:
(a) $z \in Win_\exists(\mathcal{E}^w)$;
(b) z is of the form $(u, \tau_w(\psi'))$ for some $u \in U$ and $\psi' \in Sfor(\varphi_0)$, and there is a f-conform partial match ρ' leading from (t, φ') to (u, ψ'); furthermore, both ρ and ρ' are variable-scarce.

Here we call a (partial) match *variable scarce* if it contains at most one position of the form (t, x) with x, and this position can only occur as either the first or the last position of the match.

Proof of Claim Fix a point $t \in U$ and a formula φ' such that (t, φ') is a winning position for f (in the game $\mathcal{E}'_0$). Clearly, $\exists$'s local strategy in $\mathcal{E}^w@(t, \tau_w(\varphi'))$ depends on the shape of φ'.

If φ' is the formula $\top$, then $\exists$ does not even need to play, since the one-position match $(t, \tau_w(\varphi')) = (t, \top)$ satisfies condition (a).

If φ' is of the form $\varphi'_1 \vee \varphi'_2$, then by definition we have $\tau_w(\varphi') = \tau_w(\varphi'_1) \vee \tau_w(\varphi'_2)$. Suppose that in the $\mathcal{E}'_0$-game, at position $(t, \varphi'_1 \vee \varphi'_2)$, the strategy f will tell $\exists$ to pick a position (t, φ'_i) (where $i \in \{1, 2\}$). Then in the $\mathcal{E}^w$-game, at position $(t, \tau_w(\varphi'_1) \vee \tau_w(\varphi'_2))$, we let $\exists$ pick the position $(t, \tau_w(\varphi'_i))$. This choice ensures that condition (b) is satisfied.

If φ' is of the form $\varphi'_1 \wedge \varphi'_2$, then $\tau_w(\varphi') = \tau_w(\varphi'_1) \wedge \tau_w(\varphi'_2)$. Suppose that in the $\mathcal{E}^w$-game, at position $(t, \tau_w(\varphi'_1) \wedge \tau_w(\varphi'_2))$, $\forall$ picks the conjunct $\tau_w(\varphi'_i)$, moving to position $(t, \tau_w(\varphi'_i))$. $\exists$ can mimic this in the shadow match by letting $\forall$ move to position (t, φ'_i). This ensures that condition (b) is satisfied.

Next, we look at the case when φ' is a variable x. We observe that $\tau_w(x) = x$. In the $\mathcal{E}'_0$-game, if we start at position (t, x), the next position is (t, δ_x), where $\eta x.\delta_x$ is the unique subformula of φ_0 starting with ηx ($\eta \in \{\mu, \nu\}$). In the $\mathcal{E}^w$-game, the position played after (t, x) is $(t, \tau_w(\delta_x))$. Thus condition (b) is satisfied — note that both ρ and ρ' are matches of length 2, in which the first position involves a variable. The case that φ' is of the form $\eta x.\psi'$ is similar and we omit the details.

The interesting case is where φ' is of the form $\alpha \bullet \nabla\Phi$. In the $\mathcal{E}^w$-match, $\exists$ is faced with the position $(t, \tau_w(\varphi'))$, where

$$\tau_w(\varphi') \;=\; \bigwedge \alpha \wedge \bigvee \Big\{ \bigwedge\{\Diamond\tau_w(\varphi) : \varphi \in \Phi_1\} \wedge \nabla \left(e^{\perp}[\Phi_2]\right) \mid \Phi_1 \cup \Phi_2 = \Phi \Big\}.$$

First suppose that $\forall$ chooses the left conjunct of $\tau_w(\varphi)$, moving to position (t, α). So we have to make sure that α is true at t in $\mathcal{T}[p{\restriction}U]$. Since $(t, \alpha \bullet \nabla\Phi)$ is winning for $\exists$ in $\mathcal{E}'_0$, α is true at t in $\mathcal{T}$. As t belongs to U, α is also true at t in $\mathcal{T}[p{\restriction}U]$.

Next suppose that in the $\mathcal{E}^w$-game, at position $(t, \tau_w(\varphi'))$, $\forall$ chooses the right conjunct of $\tau_w(\varphi')$. Assume that at the position $(t, \alpha \bullet \nabla\Phi)$ in $\mathcal{E}'_0$, $\exists$'s winning strategy f provides her with a ∇-marking $m : \Phi \longrightarrow \mathcal{P}(R[t])$. Define

$$\begin{aligned} \Psi_1 &:= \{\psi \in \Phi \mid u \in m(\psi) \text{ for some } u \in U\}, \\ \Psi_2 &:= \{\psi \in \Phi \mid v \in m(\psi) \text{ for some } v \notin U\}. \end{aligned}$$

Then $\Phi = \Psi_1 \cup \Psi_2$, because for every $\psi \in \Phi$ there is some $v \in R(t)$ with $v \in m(\psi)$. The strategy of $\exists$ will be to pick the sets Ψ_1 and Ψ_2, moving to position

$$\left(t, \bigwedge_{\psi \in \Psi_1} \Diamond\tau_w(\psi) \wedge \nabla \left(e^{\perp}[\Psi_2]\right) \right).$$

Now it is $\forall$'s turn; distinguish cases as to the conjunct of his choice:

- First assume that $\forall$ picks one of the conjuncts $\Diamond\tau_w(\psi)$ with $\psi \in \Psi_1$. By definition of Ψ_1, t has a successor $u \in U$ such that $u \in m(\psi)$, and $\exists$'s response in $\mathcal{E}^w$ to $\forall$'s move will be to pick exactly this u, making $(u, \tau_w(\psi))$ the next position in the match. It is straightforward to verify that this position satisfies condition (b).

- The other possibility is that $\forall$ picks the formula $\nabla\left(e^{\perp}[\Psi_2]\right) = \nabla\{e^{\perp}(\psi) \mid \psi \in \Psi_2\}$, choosing $(t, \nabla(e^{\perp}[\Psi_2]))$ as the next position in the $\mathcal{E}^w$-game. $\exists$ has to come up with a marking $m^w : e^{\perp}[\Psi_2] \longrightarrow \mathcal{P}(R[t])$. Let ψ be a formula in Ψ_2. We define $m^w(e^{\perp}(\psi))$ such that for all successors v of t,

  - if $v \notin U$, v belongs to $m^w(e^{\perp}(\psi))$ iff $v \in m(\psi)$,
  - if $v \in U$, v belongs to $m^w(e^{\perp}(\psi))$ iff there exists $w_v \in R[t] \backslash U$ such that $\mathcal{T}, v \leftrightarroweq \mathcal{T}, w_v$ and $w_v \in m(\psi)$.

  In order to show that this is a legal move for $\exists$, we have to prove that $m^w : e^{\perp}[\Psi_2] \longrightarrow \mathcal{P}(R[t])$ is a ∇-marking. Given a successor v of t, distinguish cases. If $v \notin U$, then since m is a ∇-marking, there is a formula $\psi \in \Phi$ such that $v \in m(\psi)$, and so by definition ψ belongs to the set Ψ_2. Hence we find $v \in m^w(e^{\perp}(\psi))$. If, on the other hand, v belongs to U, then since $\mathcal{T}$ is

ω-expanded and U is finitely branching, there is a state $w_v \in R[t]\backslash U$ such that $\mathcal{T}, v \leftrightarroows \mathcal{T}, w_v$. As m is a ∇-marking, there exists a formula $\psi \in \Phi$ such that $w_v \in m(\psi)$. Therefore, v belongs to $m^w(e\bot(\psi))$. Conversely, an arbitrary formula in $e^\bot[\Psi_2]$ is of the form $e^\bot(\psi)$ for some $\psi \in \Psi_2$. Then by definition of Ψ_2 there is some state $v \in R(t) \setminus U$ such that $v \in m(\psi)$, and thus $v \in m^w(e^\bot(\psi))$. This finishes the proof that m^w is a ∇-marking.

The game continues with $\forall$ choosing a pair in $(v, e^\bot(\psi))$ with $v \in m^w(e^\bot(\psi))$, as the next position in the $\mathcal{E}^w$-match. If such a pair does not exist, then $\forall$ gets stuck and condition (a) is met immediately. Otherwise, we will show that (a) holds in any case since we have

$$\{(v, e^\bot(\psi) \mid v \in m^w(e^\bot(\psi))\} \subseteq Win_\exists(\mathcal{E}^w). \tag{5.8}$$

For a proof of (5.8), first consider a pair $(v, e^\bot(\psi))$, with $v \in R[t]\backslash U$, and $v \in m^w(e^\bot(\psi))$. By definition of m^w, this means that v belongs to $m(\psi)$. Since m was part of $\exists$'s winning strategy f, we may conclude that (v, ψ) is a winning position for $\exists$ in $\mathcal{E}'_0$. Then by Proposition 5.1.5 it follows that $\mathcal{T}[p{\restriction}U], v \Vdash e(\psi)$, and by Fact 5.1.6 we may infer that $\mathcal{T}, v \Vdash e^\bot(\psi)$, and so clearly $(v, e^\bot(\psi)) \in Win_\exists(\mathcal{E}^w)$.

Next we consider an arbitrary element $(u, e^\bot(\psi))$, with $u \in R(t) \cap U$, and $u \in m^w(e^\bot(\psi))$. By definition of m^w, there exists $w_u \in R[t] \cap U$ such that $\mathcal{T}, u \leftrightarroows \mathcal{T}, w_u$ and $w_u \in m(\psi)$. As in the previous case it follows from Proposition 5.1.5 and Fact 5.1.6 that $\mathcal{T}, w_u \Vdash e^\bot(\psi)$, and so by $\mathcal{T}, u \leftrightarroows \mathcal{T}, w_u$ we obtain that $\mathcal{T}, u \Vdash e^\bot(\psi)$. From this again it is immediate that $(u, e^\bot(\psi)) \in Win_\exists(\mathcal{E}^w)$. This finishes the proof of (5.8).

Thus we have shown that for each type of formula φ', $\exists$ has a strategy leading to a position z satisfying condition (a) or (b). In case (b), the fact that the matches ρ and ρ' are variable scarce can be verified by a direct inspection of the proof. ◀

On the basis of Claim 1, $\exists$ can find a strategy that enables her to keep the condition (†). The basic idea is that she maintains the shadow $\mathcal{E}'_0$-match $\pi' = z'_0 \dots z'_k$ of the actual partial $\mathcal{E}^w$-match $\pi = z_0 \dots z_n$ in stages, inductively ensuring that at the end of each stage, unless π ends with a winning position for $\exists$ in $\mathcal{E}^w$, we have (†-ii.a-c) and $b(n) = k$. More precisely, we say that $\pi = z_0 \dots z_n$ and $\pi' = z'_0 \dots z'_k$ are at the *end of a stage* if π and π' satisfy (†) and in case (ii), $b(n) = k$.

It is immediate that if $z_0 = (s, \tau_w(\varphi_0))$ and $z'_0 = (s, \varphi_0)$, then $\pi := z_0$ and $\pi' := z'_0$ are at the end of a stage. Therefore, in order to prove that $\exists$ can maintain the condition (†) during any $\mathcal{E}^w@(r, \tau_w(\varphi_0))$-match, it is sufficient to show that if π and π' are at the end of a stage, then we can properly extend π and π' to partial matches $\pi \circ \rho$ and $\pi' \circ \rho'$ which are at the end of a stage.

Suppose that π and π' are at the end of a stage. Then $\pi = z_0 \dots z_n$ and $\pi' = z'_0 \dots z'_k$ are such that (†) holds, with the additional assumption that in case of condition (ii) we have $b(n) = k$. We make the obvious case distinction. First, if π satisfies condition (†-i) then $\exists$ can simply continue playing her winning strategy g (and she has no further need of the shadow match). Second, in case π and π' satisfy condition (†-ii), it follows from $b(n) = k$ that we may assume z'_k and z'_n to be of the form (t, φ') and $(t, \tau_w(\varphi'))$, respectively, with $t \in U$ and $\varphi \in \mathit{Sfor}(\varphi_0)$. Observe that since π' is conform $\exists$'s winning strategy f, we have $(t, \varphi') \in \mathit{Win}_\exists(\mathcal{E}'_0)$.

So we may assume that $\exists$ continues the match π by playing the strategy h given by Claim 1. Hence there is a partial $\mathcal{E}^w$-match $\pi \circ \rho$ such that $\rho \neq \emptyset$ and the last position (u, ψ) of ρ satisfies (a) or (b). If (a) the position (u, ψ) is winning for $\exists$ in $\mathcal{E}^w$, then the partial match $\pi \circ \rho$ satisfies condition (†-i) (and from this moment on $\exists$ can switch to the positional winning strategy of $\mathcal{E}^w$, forgetting about the shadow match). In the other case (b), there is a f-conform partial match ρ' leading from (t, φ) to some (u, ψ') with $\psi = \tau_w(\psi')$. Suppose that $\pi \circ \rho = z_0 \dots z_m$ and $\pi' \circ \rho' = z'_0 \dots z'_l$. Let also $b' : \{0, \dots, m\} \to \{0, \dots l\}$ be the partial map such that for all $i \in \{0, \dots, m\}$, $i \in \mathit{Dom}(b')$ iff $i \in \mathit{Dom}(b)$ or $i = m$. Moreover, for all $i \in \mathit{Dom}(b)$, $b'(i) = b(i)$ and $b'(m) = l$. Then the matches $\pi \circ \rho$ and $\pi' \circ \rho'$ satisfy the conditions (†-ii.a-c) and are at the end of a stage. Here condition (c) is an immediate consequence of the variable scarcity of ρ and ρ'.

This finishes to show that $\exists$ can maintain the condition (†), during any match of $\mathcal{E}^w@(s, \tau_w(\varphi_0))$. And as we have seen, this suffices to prove that $\mathcal{T}, r \Vdash \tau_w(\varphi_0)$. Thus we have finished the proof of the left-to-right direction of (5.7).

The proof of the opposite direction '$\Leftarrow$' of (5.7) will be very similar, and so we will omit some details. Assume that $\mathcal{T}, s \Vdash \tau_w(\varphi_0)$, and let f^w be a positional winning strategy for $\exists$ in the game $\mathcal{E}^w_0 = \mathcal{E}^w@(r, \tau_w(\varphi_0))$. In order to prove that $\mathcal{T}, r \Vdash \varphi_0$, we will need to provide her with a winning strategy in the game $\mathcal{E} := \mathcal{E}(\mathcal{T}, \varphi_0)$ initialized at (r, φ_0). Let g denote some maximal positional winning strategy for $\exists$ in $\mathcal{E}$.

As before, our proof is based on $\exists$ maintaining, during (an initial part of) the $\mathcal{E}$-match π, an f^w-conform shadow match π^w of $\mathcal{E}^w_0$. Inductively, we will make sure that $\exists$ can maintain the following constraint (‡) on the $\mathcal{E}$-match $\pi = z_0 \dots z_k$ and its shadow $\pi^w = z^w_0 \dots z^w_n$ (with $z_0 = (r, \varphi_0)$ and $z^w_0 = (r, \tau_w(\varphi_0))$):

First of all, if $\exists$ is to move at z_k, then she will not get stuck.
Furthermore, one of the following two constraints is satisfied:
(i) there is some $m \leq k$ with $z_m \in Win_\exists(\mathcal{E})$, and $z_m z_{m+1} \dots z_n$ is a g-conform $\mathcal{E}_0^w$-match;
(ii) there is an order preserving partial map $b : \{0, \dots, k\} \to \{0, \dots, n\}$
(‡) such that
(a) $b(0) = 0$,
(b) if $b(i) = j$ then for some $u \in T$ and some φ, z_i and z_j^w are of the form (u, φ) and $(u, \tau_w(\varphi))$,
(c) for all variables x, if $z_i = (t, x)$, then $i \in Dom(b)$, and if $z_j^w = (t, x)$, then $j \in Ran(b)$.

The proof that maintaining condition (‡) suffices for $\exists$ to win the game, is completely analogous to the proof given above for the other direction of (5.7). We omit the details, apart from stating and proving the crucial proposition replacing Claim 1.

2. Claim. Let $t \in T$ and $\varphi \in Sfor(\varphi_0)$ be such that $(t, \tau_w(\varphi)) \in Win_\exists(\mathcal{E}_0^w)$. Then $\exists$ has a local strategy h in $\mathcal{E}@(t, \varphi)$ with the property that for all h-conform matches λ, there exist an h-conform match ρ with last position z satisfying ($\rho \sqsubseteq \lambda$ or $\lambda \sqsubseteq \rho$) and one of the conditions below:
(a) $z \in Win_\exists(\mathcal{E})$;
(b) z is of the form (u, ψ) for some $u \in S$ and $\psi \in Sfor(\varphi_0)$, and there is a f-conform partial match ρ^w leading from $(t, \tau_w(\varphi))$ to $(u, \tau_w(\psi))$; furthermore, both ρ and ρ' are variable-scarce.

Proof of Claim As in the analogous claim given in the first part of the proof, the definition of h depends on the shape of the formula φ. We confine our attention to the only case of interest, viz., where φ is of the form $\alpha \bullet \nabla\Phi$. $\exists$'s strategy will consist of a marking $m : \Phi \to \mathcal{P}(R[t])$ that we define now.

It follows from the fact that $(t, \tau_w(\varphi)) \in Win_\exists(\mathcal{E}_0^w)$ that $\mathcal{T}, t \Vdash \alpha$, and that for some Ψ_1, Ψ_2 with $\Psi_1 \cup \Psi_2 = \Phi$, each position $(t, \Diamond\tau_w(\psi))$ with $\psi \in \Psi_1$, and the position $(t, \nabla e^\perp[\Psi_2])$ belong to $Win_\exists(\mathcal{E}_0^w)$. Given $\psi \in \Psi_1$, let $u_\psi \in R(t)$ be the successor of t such that $(u_\psi, \tau_w(\psi))$ is the move dictated by $\exists$'s winning strategy f at the position $(t, \Diamond\tau_w(\psi)) \in \mathcal{E}_0^w$, and let $m : \Phi \to \mathcal{P}(R[t])$ such that for all $\psi \in \Phi$, we have

- if $\psi \in \Psi_1 \backslash \Psi_2$, $m(\psi) = \{u_\psi\}$.
- if $\psi \in \Psi_2 \backslash \Psi_1$, $m(\psi) = \{v \mid v \in m^w(e^\perp(\psi))\}$,
- if $\psi \in \Psi_1 \cap \Psi_2$, $m(\psi) = \{u_\psi, v \mid v \in m^w(e^\perp(\psi))\}$.

First we show that in $\mathcal{E}$, m is a legitimate move for $\exists$ at the position (t, φ). Since $t \Vdash \alpha$, we only need to check that m is a ∇-marking. For this purpose, first consider an arbitrary successor v of t. Since m^w is a ∇-marking, there is a formula $\psi \in \Psi_2$ such that $v \in m^w(e^\perp(\psi))$, and hence, by definition of m, $v \in m(\psi)$. Conversely, take an arbitrary formula $\psi \in \Phi$. If $\psi \in \Psi_1$, then u_ψ is a successor of t with $u_\psi \in m(\psi)$. Otherwise we have $\psi \in \Psi_2$ and then since m^w is a ∇-marking, there is a state $u \in R[t]$ with $v \in m^w(e^\perp(\psi))$, and hence, $v \in m(\psi)$.

Next $\forall$ may pick a pair $z := (v, \psi)$ such that $v \in m(\psi)$. If there is no such a pair, we have arrived at case (a). Otherwise we claim that any next position z picked by $\forall$ satisfies condition (a) or (b). If this z is of the form (u_ψ, ψ) with $\psi \in \Psi_1$, then since $(u, \tau_w(\psi))$ was $\exists$'s choice at position $(t, \Diamond\tau_w(\psi)) \in \mathcal{E}_0^w$, it should clear that this position satisfies condition (b). This leaves the case where z is of the form (v, ψ) for some $\psi \in \Psi_2$ with $v \in m^w(e^\perp(\psi))$. Since m^w was chosen accordingly to f^w, we know that $(e, e^\perp(\psi))$ belongs to $Win_\exists(\mathcal{E}_0^w)$, and hence by Proposition 5.1.5 we obtain that $\mathcal{T}, v \Vdash e^\perp(\psi)$. From this it follows by Fact 5.1.6 that $\mathcal{T}[p \mapsto \emptyset], v \Vdash e(\psi)$, and since ψ is positive this implies that $\mathcal{T}, v \Vdash e(\psi)$. But then it is immediate by Proposition 5.1.5 again that $(v, \psi) \in Win_\exists(\mathcal{E})$; that is, $z = (v, \psi)$ satisfies condition (a). ◀

The proof that Claim 2 enables us to provide $\exists$ with a strategy satisfying (‡) is again very similar to the corresponding proof given above for the opposite direction of (5.7). We leave the details as an exercise for the reader. □

Proof of Theorem 5.3.1. Fix an arbitrary μ-sentence φ. We define $\varphi^w := \tau_w(\tau_m^\nabla(\varphi))$, where τ_m^∇ is the translation of Theorem 5.1.8. Then the equivalence (5.5) is immediate by that Theorem and Proposition 5.3.3.

The decidability of the finite width property follows by the observation that the construction of the formula φ^w from φ is effective, and that it is decidable whether φ and φ^w are equivalent. □

5.3.4. Remark. We would like to mention that given a monotone sentence φ_0 in $\mu\mathrm{ML}^\nabla$, the size of $\tau_w(\varphi_0)$ is exponential in the size of φ_0. Hence, using a similar argument to the one given in Remark 5.2.4, we easily obtain that it is decidable in elementary time whether a sentence has the finite width property. Moreover, the size of the sentence φ^w of Theorem 5.3.1 is elementary in the size of φ. The exact complexity and size are left for further work.

As in Remark 5.2.4, we could also provide a construction τ_w on automata such that for all sentences φ and for all non-deterministic automata $\mathbb{A} = (Q, q_I, \delta, \Omega)$ equivalent to φ,

$$\varphi \text{ has the finite width property for } p \quad \text{iff} \quad \varphi \equiv \tau_m(\tau_w(\mathbb{A})),$$

where τ_m is the translation for automata presented in Remark 5.2.4. Moreover, the automaton $\tau_m(\tau_w(\mathbb{A}))$ corresponds to a formula in $\mu\mathrm{ML}_W(p)$.

5.4 Continuity

This section is devoted to one of our main results, namely, we give a syntactic characterization of the continuous fragment of the modal μ-calculus.

Continuity A μ-sentence φ_0 is *continuous* in $p \in Prop$ if for every pointed model $(\mathcal{M}, w)$,

$$\mathcal{M}, w \Vdash \varphi_0 \quad \text{iff} \quad \mathcal{M}[p \upharpoonright U], w \Vdash \varphi_0, \text{ for some finite subset } U \subseteq S.$$

We leave it for the reader to verify that continuity implies monotonicity: Any formula that is continuous in p is also monotone in p.

Before we turn to its syntactic characterization, we first discuss the motivations for studying this property. The property of continuity is of interest for at least two reasons: its link to the well-known topological notion of Scott continuity [GHK$^+$80] and its connection with the notion of constructivity.

5.4.1 Link with Scott continuity

The name 'continuity' that we have given to this property is appropriate because of the following topological connection.

Scott topology Given a complete lattice $(P, \leq)$ (see Section 2.2), a subset $D \subseteq P$ is *directed* if for every pair $d_1, d_2 \in D$ there is a $d \in D$ such that $d_1 \leq d$ and $d_2 \leq d$. A subset $U \subseteq P$ is called *Scott open* if it upward closed (that is, if $u \in U$ and $u \leq v$ then $v \in U$), and satisfies, for any directed $D \subseteq P$, the property that $U \cap D \neq \emptyset$ whenever $\bigvee D \in U$. It is not hard to prove that the Scott open sets indeed provide a topology, the so-called *Scott topology* (see for instance [GHK$^+$80]).

Let $(P, \leq)$ and $(P', \leq')$ be two complete lattices. A map $f : P \longrightarrow P'$ is *Scott continuous* if for all Scott open sets $U' \subseteq P'$, $f^{-1}[U]$ is Scott open.

It is a standard result (see for instance [GHK$^+$80]) that a map $f : P \longrightarrow P'$ is Scott continuous iff f preserves *directed joins* (that is, if $D \subseteq P$ is directed, then $f(\bigvee D) = \bigvee' f[D]$).

Now we show that the notion of continuity that we introduced earlier, corresponds to the standard notion of Scott continuity. Recall that given a sentence φ, a proposition letter p and a model $\mathcal{M} = (W, R, V)$, the map $\varphi_p : \mathcal{P}(W) \longrightarrow \mathcal{P}(W)$ is such that for all $U \subseteq W$,

$$\varphi_p(U) = \{w \in W \mid \mathcal{T}[p \mapsto U], w \Vdash \varphi\}.$$

5.4.1. Proposition. *A sentence is continuous in p iff for all models $\mathcal{M} = (W, R, V)$, $\varphi_p : \mathcal{P}(W) \longrightarrow \mathcal{P}(W)$ is Scott continuous.*

Proof For the direction from left to right, let φ be a continuous sentence in p. Fix a model $\mathcal{M} = (W, R, V)$. We show that the map $\varphi_p : \mathcal{P}(W) \to \mathcal{P}(W)$ preserves directed joins.

Let $\mathcal{F}$ be a directed subset of $(\mathcal{P}(W), \subseteq)$. It follows from the monotonicity of φ that the set $\bigcup \varphi_p[\mathcal{F}]$ is a subset of $\varphi_p(\bigcup \mathcal{F})$. Thus, it remains to show that $\varphi_p(\bigcup \mathcal{F}) \subseteq \bigcup \varphi_p[\mathcal{F}]$. Take w in $\varphi_p(\bigcup \mathcal{F})$. That is, the formula φ is true at w in the model $\mathcal{M}[p \mapsto \bigcup \mathcal{F}]$. As φ is continuous in p, there is a finite subset F of $\bigcup \mathcal{F}$ such that φ is true at w in $\mathcal{M}[p \mapsto F]$. Now, since F is a finite subset of $\bigcup \mathcal{F}$ and since $\mathcal{F}$ is directed, there exists a set U in $\mathcal{F}$ such that F is a subset of U. Moreover, as φ is monotone, $\mathcal{M}[p \mapsto F], w \Vdash \varphi$ implies $\mathcal{M}[p \mapsto U], w \Vdash \varphi$. Therefore, w belongs to $\varphi_p(U)$ and in particular, w belongs to $\bigcup \varphi_p[\mathcal{F}]$. This finishes to show that $\varphi_p(\bigcup \mathcal{F}) \subseteq \bigcup \varphi_p[\mathcal{F}]$.

For the direction from right to left, let φ be a sentence such that for all models $\mathcal{M} = (W, R, V)$, $\varphi_p : \mathcal{P}(W) \to \mathcal{P}(W)$ is Scott continuous. First we show that φ is monotone in p. Let $\mathcal{M} = (W, R, V)$ be a model. We check that $\varphi_p(U) \subseteq \varphi_p(U')$, in case $U \subseteq U'$. Suppose $U \subseteq U'$ and let $\mathcal{F}$ be the set $\{U, U'\}$. The family $\mathcal{F}$ is clearly directed and satisfies $\bigcup \mathcal{F} = U'$. Using the fact that φ_p preserves directed joins, we get that $\varphi_p(U') = \varphi_p(\bigcup \mathcal{F}) = \bigcup \varphi_p[\mathcal{F}]$. By definition of $\mathcal{F}$, we have $\bigcup \varphi_p[\mathcal{F}] = \varphi_p(U) \cup \varphi_p(U')$. Putting everything together, we obtain that $\varphi_p(U') = \varphi_p(U) \cup \varphi_p(U')$. Thus, $\varphi_p(U) \subseteq \varphi_p(U')$.

To show that φ is continuous in p, it remains to show that if $\mathcal{M}, w \Vdash \varphi$, then there exists a finite subset U of $V(p)$ such that $\mathcal{M}[p \mapsto U], w \Vdash \varphi$. Suppose that the sentence φ is true at w in $\mathcal{M}$. That is, w belongs to $\varphi_p(V(p))$. Now let $\mathcal{F}$ be the family $\{U \subseteq V(p) \mid U \text{ finite}\}$. It is not hard to see that $\mathcal{F}$ is a directed subset of $(\mathcal{P}(W), \subseteq)$. Since φ_p preserves directed joins, we obtain $\varphi_p(V(p)) = \varphi_p(\bigcup \mathcal{F}) = \bigcup \varphi_p[\mathcal{F}]$. From $w \in \varphi_p(V(p))$, it then follows that $w \in \bigcup \varphi_p[\mathcal{F}]$. Therefore, there exists $U \in \mathcal{F}$ such that $w \in \varphi_p(F)$. That is, U is a finite subset of $V(p)$ such that $\mathcal{M}[p \mapsto U], w \Vdash \varphi$. □

5.4.2 Constructivity

Intuitively, a formula is constructive if the ordinal approximation of its least fixpoint is reached in at most ω steps.

Constructivity Given a sentence φ and a model $\mathcal{M} = (W, R, V)$, we define, by induction on $i \in \mathbb{N}$, a map $\varphi_p^i : \mathcal{P}(W) \to \mathcal{P}(W)$. We set $\varphi_p^0 := \varphi_p$ (where φ_p is as defined in Chapter 2 or as in the previous subsection) and for all $i \in \mathbb{N}$, $\varphi_p^{i+1} := \varphi_p \circ \varphi_p^i$. The sentence φ is *constructive* in p if for all models $\mathcal{M} = (W, R, V)$, the least fixpoint of the map φ_p is equal to $\bigcup \{\varphi_p^i(\emptyset) \mid i \in \mathbb{N}\}$.

In [Ott99], Martin Otto proved that it is decidable in exponential time whether a basic modal formula φ is bounded, but to the best of our knowledge, decidability of constructivity (that is, the problem whether a given μ-sentence is constructive in p) is an open problem. In passing we mention that recently, Marek Czarnecki [Cza10], found, for each ordinal $\beta < \omega^2$, a formula φ_β for which β is the least ordinal such that the least fixpoint of φ_β is always reached in β steps.

The connection between the notions of continuity and constructivity is an intriguing one. It is a routine exercise to prove that any sentence, continuous in p, is also constructive in p.

5.4.2. PROPOSITION. *A sentence φ continuous in p is constructive in p.*

Proof Let φ be a sentence continuous in p and let $\mathcal{M} = (W, R, V)$ be a model. We show that the least fixpoint of φ_p is $\bigcup\{\varphi_p^i(\emptyset) \mid i \in \mathbb{N}\}$.

Let $\mathcal{F}$ be the family $\{\varphi_p^i(\emptyset) \mid i \in \mathbb{N}\}$. It is sufficient to check that $\varphi_p(\bigcup \mathcal{F}) = \bigcup \mathcal{F}$. First remark that the subset $\mathcal{F}$ of the partial order $(\mathcal{P}(W), \subseteq)$ is directed. Therefore, by Proposition 5.4.1, $\varphi_p(\bigcup \mathcal{F}) = \bigcup \varphi_p[\mathcal{F}]$. It is also easy to prove that $\bigcup \varphi_p[\mathcal{F}] = \bigcup \mathcal{F}$. Putting everything together, we obtain that $\varphi_p(\bigcup \mathcal{F}) = \bigcup \mathcal{F}$ and this finishes the proof. □

However, not all constructive sentences are continuous. We give two examples. The formula $\varphi = \Box p \wedge \Box\Box\bot$ is true at a point w in a model if the depth of w is less or equal to 2 (that is, there are no v and v' satisfying $wRvRv'$) and all successors of w satisfy p. It is not hard to see that φ is not continuous in p. However, we have that for all models $\mathcal{M} = (W, R, V)$, $\varphi_p^2(\emptyset) = \varphi_p^3(\emptyset)$. In particular, φ is constructive in p.

The formula $\psi = \nu x.p \wedge \Diamond x$ is true at a point w if there is an infinite path starting from w and such that at each point of this path, p is true. This sentence is not continuous in p. However, it is constructive, since for all models $\mathcal{M} = (W, R, V)$, we have $\psi_p(\emptyset) = \emptyset$.

Observe that in the previous examples, we have $\mu p.\varphi \equiv \mu p.\Box\Box\bot$ and $\mu p.\psi \equiv \mu p.\bot$. Thus, there is a continuous sentence (namely $\Box\Box\bot$) that is equivalent to φ, modulo the least fixpoint operation. Similarly, there is a continuous sentence (the formula $\bot$) that is equivalent to ψ, modulo the least fixpoint operation. This suggests the following question concerning the link between continuity and constructivity: Given a constructive formula φ, can we find a continuous formula ψ satisfying $\mu p.\varphi \equiv \mu p.\psi$?

We leave this question as an open problem, as we do with the question whether constructivity can be captured by a nice syntactic fragment of the modal μ-calculus.

5.4.3 Semantic characterization of continuity

We now turn to the main result of this section, namely, our characterization result for continuity. Our approach towards continuity is based on the observation that this property can be seen as the combination of the finite path and the finite width properties.

5.4.3. PROPOSITION. *A μ-sentence φ_0 is continuous in p iff φ_0 has both the finite path and the finite width property with respect to p.*

Proof Confining our attention to the direction from right to left, assume that a μ-sentence φ_0 has both the finite path and the finite width property. Fix a model $\mathcal{T} = (T, R, V)$ and a point $r \in T$. We have to show

$$\mathcal{T}, r \Vdash \varphi_0 \quad \text{iff} \quad \mathcal{T}[p{\restriction}U], r \Vdash \varphi_0, \text{ for some finite subset } U \subseteq S.$$

The direction from right to left follows from Proposition 5.1.8 and the fact that φ_0 is positive in p. For the direction from left to right, suppose that $\mathcal{T}, r \Vdash \varphi_0$. As in the proof of Proposition 5.3.2, we may assume that $\mathcal{T}$ is a tree with root r.

Since φ_0 has the finite width property with respect to p, there is a downward closed subset $U_1 \subseteq T$ which is finitely branching and such that $\mathcal{T}[p{\restriction}U_1], r \Vdash \varphi_0$. But φ_0 also has the finite path property with respect to p. Hence there is a subset U_2 of T such that U_2 is downward closed, does not contain any infinite path and $\mathcal{T}[p{\restriction}(U_1 \cap U_2)], r \Vdash \varphi_0$. By König's Lemma, $U_1 \cap U_2$ is finite. □

The syntactic fragment corresponding to continuity can be defined as follows.

The fragment $\mu\mathrm{ML}_C(p)$ We define the fragment $\mu\mathrm{ML}_C(p)$ by induction in the following way:

$$\varphi \ ::= \ p \mid x \mid \psi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \Diamond\varphi \mid \mu x.\varphi,$$

where x is a variable and ψ is a sentence which does not contain p.

Observe that this fragment is contained in the intersection of the fragments $\mu\mathrm{ML}_D(p)$ and $\mu\mathrm{ML}_W(p)$ (defined in the previous two sections). Our characterization then is as follows.

5.4.4. THEOREM. *The μ-sentences in $\mu\mathrm{ML}_C(p)$ are continuous in p. Moreover, there is an effective translation which given a μ-sentence φ, computes a formula $\varphi^c \in \mu\mathrm{ML}_C(p)$ such that*

$$\varphi \text{ is continuous in } p \quad \text{iff} \quad \varphi \equiv \varphi^c. \tag{5.9}$$

As a corollary, it is decidable whether a given sentence φ is continuous in p.

We turn now to the proof of Theorem 5.4.4. First, it is easy to see that the fragment $\mu\mathrm{ML}_C(p)$ consists of continuous formulas.

5.4.5. PROPOSITION. *If φ_0 belongs to $\mu\mathrm{ML}_C(p)$, then φ_0 is continuous in p.*

Proof If φ_0 belongs to $\mu\mathrm{ML}_C(p)$, then since $\mu\mathrm{ML}_C(p) \subseteq \mu\mathrm{ML}_D(p) \cap \mu\mathrm{ML}_W(p)$, by the Propositions 5.2.3 and 5.3.2, it has both the finite path and the finite width property with respect to p. Hence by Proposition 5.4.3, φ_0 is continuous in p. □

We now turn to the hard part of the proof of Theorem 5.4.4. This part of the proof will be based on the following translation, which combines clauses of the translations τ_d and τ_w.

The translation τ_c Fix a positive sentence $\varphi_0 \in \mu\mathrm{ML}^\nabla$. We define the map $\tau_c : \mathit{Sfor}(\varphi_0) \longrightarrow \mu\mathrm{ML}$ as the map $\tau_d \circ \tau_w$ (where we extend the translation τ_w to ∇-formulas in the obvious way). That, τ_c is the map defined by induction on the complexity of the subformulas of φ_0 by:

$$
\begin{aligned}
\tau_c(\top) &= \top, \\
\tau_c(x) &= x, \\
\tau_c(\varphi \vee \psi) &= \tau_c(\varphi) \vee \tau_c(\psi), \\
\tau_c(\varphi \wedge \psi) &= \tau_c(\varphi) \wedge \tau_c(\psi), \\
\tau_c(\alpha \bullet \nabla\Phi)^t &= \bigwedge\alpha \wedge \bigvee\{\bigwedge\{\Diamond\tau_c(\varphi) \mid \varphi \in \Phi_1\} \wedge \nabla\left(e^\perp[\Phi_2]\right) \mid \Phi_1 \cup \Phi_2 = \Phi\}, \\
\tau_c(\mu x.\varphi) &= \mu x.\tau_c(\varphi), \\
\tau_c(\nu x.\varphi) &= \mu x.\left(\tau_c(\varphi) \vee e^\perp(\varphi)\right)
\end{aligned}
$$

where x is a variable, α is a conjunction of literals, and Φ is a finite subset of $\mu\mathrm{ML}^\nabla$.

5.4.6. PROPOSITION. *A positive guarded sentence $\varphi_0 \in \mu\mathrm{ML}^\nabla$ is continuous with respect to proposition letter p iff φ_0 is equivalent to $\tau_c(\varphi_0)$.*

Proof Fix a positive guarded sentence φ_0 in $\mu\mathrm{ML}^\nabla$. The direction from right to left is an easy consequence of Proposition 5.4.5 and the observation that $\tau_c(\varphi_0)$ belongs to $\mu\mathrm{ML}_C(p)$.

For the opposite direction, suppose that φ_0 is continuous with respect to p. In particular, φ_0 has the finite width property with respect to p. It follows from Proposition 5.3.3 that φ_0 is equivalent to $\tau_w(\varphi_0)$. Since φ_0 has the finite path property with respect to p, $\tau_w(\varphi_0)$ also has the finite path property with respect to p. Therefore, from Proposition 5.2.3, we obtain that $\tau_w(\varphi_0)$ is equivalent to $\tau_d(\tau_w(\varphi_0))$. Putting everything together, we get that that φ_0 is equivalent to $\tau_c(\varphi_0)$. $\square$

We turn to the proof of the main result of this section.

Proof of Theorem 5.4.4. Fix an arbitrary μ-sentence φ. We define $\varphi^c := \tau_c(\tau_m^\nabla(\varphi))$, where τ_m^∇ is the translation of Theorem 5.1.8. Given the fact that continuity implies monotonicity, the equivalence (5.9) is then immediate by Theorem 5.1.8 and Proposition 5.4.6.

The decidability of the associated continuity problem follows by the observation that the construction of the formula φ^c from φ is effective, and that it is decidable whether φ and φ^c are equivalent. $\square$

5.4.7. REMARK. It follows from Proposition 5.4.3 and Remarks 5.2.4 and 5.3.4 that it is decidable in elementary time whether a sentence is continuous in p.

Again, the exact complexity is left for further work. Given a positive sentence φ, the size of φ^c is exponential in the size of φ.

It also follows from Remarks 5.2.4 and 5.3.4 that we can define a construction similar to τ_c in the context of automata. This construction might be helpful if we study complexity issues related to the continuous fragment.

5.5 Complete additivity

The last property of formulas that we look at concerns the way in which the semantics of the formula depends on the proposition letter p being true at some single point.

Complete additivity A μ-sentence φ_0 is *completely additive* in $p \in Prop$ if for all models $\mathcal{M} = (W, R, V)$ and $w \in W$,

$$\mathcal{M}, w \Vdash \varphi_0 \quad \text{iff} \quad \mathcal{M}[p{\restriction}\{v\}], w \Vdash \varphi_0 \text{ for some } v \in V(p). \tag{5.10}$$

A formula φ_0 that is completely additive in a proposition letter p is monotone in p. Moreover, if $\mathcal{M} = (W, R, V)$ is a model such that $V(p) = \emptyset$, it can never happens that φ_0 is true at a point in $\mathcal{M}$.

One of the main reasons for studying complete additivity is its pivotal role in the characterization of the fragment of first- and monadic second-order logic formulas that are safe for bisimulations. A (first- or second order) formula $\varphi(x, y)$ is called safe for bisimulations if, whenever $B \subseteq W \times W'$ is a bisimulation between two models $\mathcal{M}$ and $\mathcal{M}'$, then B is also a bisimulation for the two models we obtain by considering the interpretations of φ as accessibility relations on $\mathcal{M}$ and $\mathcal{M}'$, respectively (for a precise definition, see Section 2.6).

This notion was introduced by Johan van Benthem [Ben96] who also gave a characterization of the safe fragment of first-order logic. The link with complete additivity is given by the observation that a formula $\varphi(x, y)$ is safe for bisimulation iff the formula $\exists y\, (\varphi(x, y) \wedge P(y))$ (with P is a fresh unary predicate) is bisimulation invariant. Recall that modal logic is the bisimulation invariant fragment of first-order logic (see Section 2.6) and μ-calculus is the bisimulation invariant fragment of MSO (Theorem 2.6.7). Hence, in case φ is a first-order formula, the formula $\exists y\, (\varphi(x, y) \wedge P(y))$ is safe for bisimulation iff it is equivalent to a modal formula $\varphi'(p)$. Similarly, if φ is a monadic second-order formula, $\exists y\, (\varphi(x, y) \wedge P(y))$ is safe for bisimulation iff it is equivalent to a μ-sentence $\varphi'(p)$. Moreover, using the fact that $\varphi'(p)$ is equivalent to $\exists y\, (\varphi(x, y) \wedge P(y))$, it is easy to see that $\varphi'(p)$ is completely additive in p, where p is a proposition letter corresponding to the predicate P.

Thus, a syntactic characterization of the completely additive modal formulas and modal μ-sentences, respectively, may help to obtain a syntactic characterization of the safe fragments of first- and monadic second-order logic. Such characterizations were obtained for modal logic by Johan van Benthem [Ben96], and for the μ-calculus by Marco Hollenberg [Hol98b], see Remark 5.5.2 for more details on the latter result.

The syntactic fragment that corresponds to complete additivity is defined as follows.

The fragment $\mu\mathrm{ML}_A(p)$ We define the fragment $\mu\mathrm{ML}_A(p)$ by induction in the following way:

$$\varphi ::= p \mid x \mid \varphi \vee \varphi \mid \varphi \wedge \psi \mid \Diamond\varphi \mid \mu x.\varphi,$$

where x is a variable and ψ is a sentence which does not contain p.

5.5.1. Remark. There are interesting connections between these fragments and the language PDL. Since PDL is by nature a poly-modal language, we momentarily allow the set *Act* of actions to not be a singleton. One may show that PDL has the same expressive power as the fragment of μML in which the formula construction $\mu x.\varphi$ is allowed only if φ is completely additive with respect to x. More precisely, define the set F_A of formulas by the following grammar:

$$\varphi ::= p' \mid \neg\varphi \mid \varphi \vee \varphi \mid \Diamond_a\varphi \mid \mu x.\varphi,$$

where p' is an arbitrary proposition letter, a belongs to *Act* and φ belongs to the fragment $\mu\mathrm{ML}_A(x)$. Then there are inductive, truth-preserving translations from PDL to the fragment F_A, and vice versa [Ven08b].

5.5.2. Remark. The characterization provided by Marco Hollenberg [Hol98b] of the completely additive fragment of the modal μ-calculus states that a formula φ is completely additive in p iff φ is equivalent to a formula of the form $\langle\pi\rangle p$, where π is a μ-program. We do not recall the definition of μ-program there, as we do not need it in this section (for more details, see Section 2.6.3).

Comparing the two fragments, Marco Hollenberg's fragment is more suited to prove the result on safety, while the shape of the fragment $\mu\mathrm{ML}_A(p)$ is closer to the shape of the syntactic fragments considered in the earlier sections. Hence, if we express the characterization in terms of the fragment $\mu\mathrm{ML}_A(p)$, it will be easier to adapt the proofs of the previous sections. The fragment $\mu\mathrm{ML}_A(p)$ is not only useful for our proof, but also fairly intuitive. In any case, there are direct translations between our fragment and Marco Hollenberg's.

The main result of this section is the following theorem showing that, modulo equivalence, $\mu\mathrm{ML}_A(p)$ captures the fragment of the μ-calculus that is completely additive.

5.5.3. Theorem. *The μ-sentences in $\mu\mathrm{ML}_A(p)$ are completely additive in p. Moreover, there is an effective translation which given a μ-sentence φ, computes a formula $\varphi^a \in \mu\mathrm{ML}_A(p)$ such that*

$$\varphi \text{ is completely additive in } p \quad \text{iff} \quad \varphi \equiv \varphi^a. \tag{5.11}$$

As a corollary, it is decidable whether a given sentence φ is completely additive in p.

As mentioned already, we could obtain our result as a corollary to Marco Hollenberg's characterization. The first proof of the characterization of the continuous fragment [Fon08] (Theorem 5.4.4) was in fact inspired by Marco Hollenberg's proof for the characterization of the completely additive fragment. It turns out that the proof of Theorem 5.4.4 could be simplified. The new proof (which is the one presented in Section 5.4) is not only easier: unlike the original proof, it provides a direction translation (the translation τ_C of Section 5.4).

This raised the following question: can we also simplify Marco's Hollenberg proof and find a direct translation playing the same role as τ_C? The answer is yes. However, the definition of the translation is a lot more involved than the definition of τ_C.

The existence of that new proof can also be an indication of the flexibility (and limitations) of the method used in this chapter. As witnessed by the proof of Proposition 5.1.7, the method can be adapted to prove the Lyndon's theorem for the μ-calculus [DH00]. We do not give details, but the method also works to show the Łos-Tarski theorem for the μ-calculus [DH00]. In the case of the property of complete additivity, we can also use the same method, but as we will see in a few paragraphs, the proof becomes considerably harder than for the other properties. Let us finally mention that in [FV10], we consider a property (the single point property) for which we were not able to use the same method. The characterization was obtained using an automata theoretic approach.

As in the other sections, we start by proving the easy direction of Theorem 5.5.3. That is, if a sentence belongs to $\mu\mathrm{ML}_A(p)$, then it is completely additive.

5.5.4. Proposition. *Any sentence in $\mu\mathrm{ML}_A(p)$ is completely additive in p.*

Proof Let φ_0 be a sentence in $\mu\mathrm{ML}_A(p)$. Fix a model $\mathcal{M} = (W, R, V)$ and a point $w \in W$. We have to show that

$$\mathcal{M}, w \Vdash \varphi_0 \quad \text{iff} \quad \mathcal{M}[p{\restriction}\{v\}], w \Vdash \varphi_0 \text{ for some } v \in V(p). \tag{5.12}$$

The direction from right to left follows from Theorem 5.1.8 and the fact that φ_0 is positive in p. For the opposite direction, suppose that $\mathcal{M}, w \Vdash \varphi_0$. We have to find a point $v \in V(p)$ such that $\mathcal{M}[p{\restriction}\{v\}], w \Vdash \varphi_0$.

Since $\mathcal{M}, w \Vdash \varphi_0$, $\exists$ has a winning strategy f in the game $\mathcal{E}_0 := \mathcal{E}(\mathcal{M}, \varphi_0)@(w, \varphi_0)$. We start by defining an f-conform $\mathcal{E}_0$-match π such that

(i) for all f-reachable positions (v, φ) with p active in φ, (v, φ) occurs in π,

(ii) for all positions (v, φ) occurring in π, φ belongs to $\mu\mathrm{ML}_A(p)$.

For all $i \in \mathbb{N}$, we define z_i by induction on i such that z_i is an f-reachable position of the form (v, φ) with $\varphi \in \mu\mathrm{ML}_A(p)$. If $i = 0$, we define z_i as the position (w, φ_0). For the induction step, suppose that we already defined the position $z_i = (v, \varphi)$ and φ belongs to $\mu\mathrm{ML}_A(p)$. We make the following case distinction:

- If $\varphi = p$, we stop the construction and we define π as $z_0 \dots z_i$.
- If $\varphi = x$, we define z_{i+1} as the position (v, δ_x), where $\eta x.\delta_x$ is a subformula of φ_0 ($\eta \in \{\mu, \nu\}$).
- If φ is a formula of the form $\varphi_1 \vee \varphi_2$ (with φ_2, φ_2 in $\mu\mathrm{ML}_A(p)$), then according to f, $\exists$ moves from the position (v, φ) to a position of the form (v, φ_k), where $k \in \{1, 2\}$. We define z_{i+1} as (v, φ_k).
- If φ is a formula of the form $\varphi' \wedge \psi$ (with φ in $\mu\mathrm{ML}_A(p)$ and ψ a closed sentence in which p does not occur), then we define z_{i+1} as (v, φ').
- If φ is of the form $\mu x.\varphi'$ (with φ' in $\mu\mathrm{ML}_A(p)$), we define z_{i+1} as (v, φ').
- Finally, if φ is of the form $\Diamond\varphi'$ (with φ' in $\mu\mathrm{ML}_A(p)$), then according to f, $\exists$ moves from the position (v, φ) to the position (u, φ'), for some $u \in R[v]$. We let z_{i+1} be the position (u, φ').

If at some point the construction described above stopped, then π is already defined. Otherwise, for all $i \in \mathbb{N}$, z_i is defined and we let π be the match $z_0 z_1 \dots$. It is easy to see that conditions (i) and (ii) are verified.

It follows from (ii) that the only positions of the form $(\eta x.\varphi)$ (where $\eta \in \{\mu, \nu\}$) occurring in π are such that $\eta = \mu$. Hence, if π is infinite, π is lost by $\exists$. This is impossible since π is an f-conform match and f is a winning strategy for $\exists$.

So π is a finite match of the form $z_0 \dots z_n$, which is won by $\exists$. Moreover, it follows from the definition of π that z_n is a position of the form (v, p). Putting this together with the fact that π is won by $\exists$, we have $p \in V(p)$. Therefore, in order to show that the left-to-right direction of (5.12) holds, it is sufficient to prove that $\mathcal{M}[p{\restriction}\{v\}], w \Vdash \varphi_0$.

For that purpose, we show that f is a winning strategy for $\exists$ in $\mathcal{E}_0' := \mathcal{E}(\mathcal{M}[p{\restriction}\{v\}, \varphi_0)@(w, \varphi_0)$. The games $\mathcal{E}_0$ and $\mathcal{E}_0'$ are exactly the same, except when we reach a position of the form (u, p). In $\mathcal{E}_0'$, such a position is winning for $\exists$ only if $u = v$. So in order to prove that f is winning for $\exists$ in $\mathcal{E}_0'$, it is sufficient to show that for all f-reachable positions (u, p) in $\mathcal{E}_0$, we have $u = v$.

Let (u, p) be an f-reachable position in $\mathcal{E}_0$. In particular, (u, p) is a position of the form (u, φ) with p active in φ. It follows from (ii) that (u, p) occurs in π. This implies that $u = v$ and this finishes the proof. $\square$

We turn now to the hard part of the proof, which consists in showing that a formula that has the single point property can be rewritten as a formula in the right syntactic fragment.

The translation τ_a Let φ_0 be a positive sentence in $\mu\mathrm{ML}^\nabla$, let Φ be a finite set of subformulas of φ_0, and let $\mathcal{F}$ be a family of finite sets of formulas in $Sfor(\varphi_0)$. The formula $\tau_\mathcal{F}(\Phi)$ is defined by the list of instructions below, which has to be read as follows: We apply the first instruction of the list which is applicable (there will always be at least one instruction applicable). There are two possibilities. Either the process stops or we end up with a new formula $\tau_{\mathcal{F}'}(\Phi')$ to be defined. In the second case, we go trough the list again and apply the first instruction that is applicable and so on. We show later that at some point, the process stops.

(1) If Φ is empty, we define $\tau_\mathcal{F}(\Phi)$ as $\top$.

(2) If Φ contains the formula $\top$, then we define $\tau_\mathcal{F}(\Phi)$ as $\tau_\mathcal{F}(\Phi \backslash \{\top\})$.

(3) If Φ contains a formula φ of the form $\varphi_1 \wedge \varphi_2$, we let Φ' be the set $(\Phi \backslash \{\varphi\}) \cup \{\varphi_1, \varphi_2\}$ and we define $\tau_\mathcal{F}(\Phi)$ as $\tau_\mathcal{F}(\Phi')$.

(4) If Φ contains a formula φ of the form $\mu x.\psi$ or $\nu x.\psi$, then we let Φ' be the set Φ where we replace each formula of the form $\mu x.\chi$ or $\nu x.\chi$ by the formula χ. We define $\tau_\mathcal{F}(\Phi)$ as $\tau_\mathcal{F}(\Phi')$.

(5) If Φ contains a formula φ of the form $\varphi_1 \vee \varphi_2$, then we let Φ_1 be the set Φ where we replace φ by φ_1 and we let Φ_2 be the set Φ where we replace φ by φ_2. We define $\tau_\mathcal{F}(\Phi)$ as $\tau_\mathcal{F}(\Phi_1) \vee \tau_\mathcal{F}(\Phi_2)$.

(6) If Φ contains a formula which is a variable, all formulas in Φ that are not variables are of the form $\alpha \bullet \nabla\Psi$ and Φ does not belong to $\mathcal{F}$, then we let X_Φ be the biggest set of variables such that $X_\Phi \subseteq \Phi$. We also let Φ' be the set Φ where we replace each formula $x \in X_\Phi$ by δ_x. We define $\tau_\mathcal{F}(\Phi)$ as $\mu x_\Phi.\tau_{\mathcal{F}\cup\{\Phi\}}(\Phi')$.

(7) If Φ contains a formula which is a variable, all formulas in Φ that are not variables are of the form $\alpha \bullet \nabla\Psi$ and Φ belongs to $\mathcal{F}$, then we define $\tau_\mathcal{F}(\Phi)$ as x_Φ.

(8) If Φ is a set of the form $\{\alpha_1 \bullet \nabla\Phi_1, \dots, \alpha_k \bullet \nabla\Phi_k\}$ and p does not occur in $\alpha_1 \wedge \dots \wedge \alpha_k$, we define $\tau_\mathcal{F}(\Phi)$ as

$$\bigvee \Big\{ \Diamond\tau_{\mathcal{F}}\Big(\bigcup_{i=1}^{k} \Phi'_i\Big) \wedge \Big(\bigwedge_{i=1}^{k} \alpha_i \wedge \nabla e^{\perp}[\Phi''_i]\Big) \;\Big|\; \text{for all } 1 \leq i \leq k, \\ \Phi'_i \neq \emptyset, \Phi'_i \cup \Phi''_i = \Phi_i \Big\}.$$

(9) If Φ' is a set of the form $\{\alpha_1 \bullet \nabla\Phi_1, \dots, \alpha_k \bullet \nabla\Phi_k\}$ and p does occur in $\alpha_1 \wedge \cdots \wedge \alpha_k$, we define $\tau_{\mathcal{F}}(\Phi)$ as

$$(\alpha_1 \wedge \cdots \wedge \alpha_k) \wedge \bigwedge\{\nabla e^{\perp}[\Phi_i] \mid 1 \leq i \leq k\}.$$

We will show in the next proposition that $\tau_{\mathcal{F}}(\Phi)$ is well-defined. Finally, we define $\tau_a(\varphi_0)$ as $\tau_{\emptyset}(\varphi_0)$.

The intuition behind the definition of $\tau_{\mathcal{F}}(\Phi)$ is as follows. Our goal is to translate the formula $\bigwedge \Phi$ into a formula in $\mu\text{ML}_A(p)$. This is done by induction on the complexity of the formulas in Φ. If Φ contains a disjunction or the symbol $\top$, then it is easy how to proceed. If all the formulas in Φ are ∇-formulas, we use a translation that is a variant of the one defined for the width property. Suppose that the formula $\bigwedge \Phi$ is completely additive and is true at a point t in a model. Assume that $\Phi = \{\alpha_1 \bullet \nabla\Phi_1, \dots, \alpha_k \bullet \nabla\Phi_k\}$. There are two possibilities: either p occurs in $\alpha_1 \wedge \dots \alpha_k$ or not. In the first case, p has to be true at t. Since $\bigwedge \Phi$ is completely additive, we may assume that p is not true at all proper descendants of t. Hence, we can replace p by $\perp$ in the formulas of $\Phi_1 \cup \dots \cup \Phi_k$. This corresponds to case (9).

Otherwise, p does not occur in $\alpha_1 \wedge \dots \alpha_k$. Since $\bigwedge \Phi$ is completely additive, there is a unique successor t' such that p is true at one of the points of the model generated by t'. In the evaluation game, at position $(t, \alpha_i \bullet \nabla_i)$, $\exists$ has to provide a marking $m_i : \Phi_i \longrightarrow \mathcal{P}(R[t])$. If a point t'' in $R[t]\backslash\{t'\}$ is marked with a formula ψ, then we can replace ψ by $e^{\perp}(\psi)$, since p is not true in the model generated by t''. Hence, if we define Φ'_i as the set of formulas ψ such that $t' \in m_i(\psi)$ and Φ''_i as the set of formulas ψ such that $m_i(\psi) \cap (R[t]\backslash\{t'\}) \neq \emptyset$, then the formula given in (8) is true.

Next, suppose that one formula in Φ starts with a fixpoint operator. We simply forget about it (we know that in the translated formula, p can only be in the scope of least fixpoint operators by definition of $\mu\text{ML}_A(p)$). If Φ contains a variable, we unfold the variables in Φ. The problem is that by doing so, the process will never stop and we will keep on unfolding variables.

The solution is to use the set $\mathcal{F}$. Basically, in $\mathcal{F}$, we keep track of all the sets of formulas Φ that were encountered during the process. So when Φ contains variables, instead of unfolding the variables, we first check, by looking at $\mathcal{F}$, whether the set Φ appeared earlier. If not, we unfold the variables and we let Φ'

be the set obtained after unfolding the variables in Φ. We also put a "marker" for Φ, by writing the operator μx_Φ before the formula corresponding to Φ'.

Otherwise, if Φ does belong to $\mathcal{F}$, we stop the process and we simply write x_φ. Note that the unfolding of x_Φ in the translated formula, is the unique formula ψ such that $\mu x_\Phi.\psi$ is a subformula of the translated formula. It follows from the construction of the translation, that ψ corresponds to the stage of the process when we encountered Φ and unfolded its variables.

5.5.5. Proposition. *For all positive guarded sentences φ_0 in $\mu\mathrm{ML}^\nabla$, $\tau_a(\varphi_0)$ is well-defined.*

Proof We give a sketch of the proof that for all sentences φ_0 in $\mu\mathrm{ML}^\nabla$, the translation $\tau_a(\varphi_0)$ is well-defined. Given a formula φ, we write $s(\varphi)$ for the number of subformulas of φ. We let m be the size of the set $Sfor(\varphi_0)$, n the number $2^{(2^m)}$ and l the number $m \cdot s(\varphi_0)$. Given a set of formulas $\Phi \subseteq Sfor(\varphi_0)$ and given a family $\mathcal{F}$ of subsets of $Sfor(\varphi_0)$, we define the weight of $(\Phi, \mathcal{F})$, notation: $w(\Phi, \mathcal{F})$, by the following:

$$w(\Phi, \mathcal{F}) := (n - |\mathcal{F}|) \cdot (l + 1) + \Sigma_{\varphi \in \Phi} s(\varphi),$$

where $|\mathcal{F}|$ is the cardinal of $\mathcal{F}$. We can look at the definition of τ_a as a process which starts with the pair $(\{\varphi_0\}, \emptyset)$ and associates with each pair $(\Phi, \mathcal{F})$ in $\mathcal{P}(Sfor(\varphi_0)) \times \mathcal{PP}(Sfor(\varphi_0))$ finitely many new pairs in $\mathcal{P}(Sfor(\varphi_0)) \times \mathcal{PP}(Sfor(\varphi_0))$, until the process finishes. For example, if we apply the rule "If Φ contains a formula φ of the form $\mu x.\psi$", then there is a unique pair associated to $(\Phi, \mathcal{F})$, namely the pair $(\Phi', \mathcal{F})$, where Φ' is the set Φ in which we replaced each formula of the form $\eta x.\chi$ by χ (where $\eta \in \{\mu, \nu\}$).

To show that $\tau_a(\varphi_0)$ is well-defined, it is enough to show that the following two properties hold. The weight of each pair is positive. The weight of each new pair in $\mathcal{P}(Sfor(\varphi_0)) \times \mathcal{PP}(Sfor(\varphi_0))$ associated with a pair $(\Phi, \mathcal{F})$ is strictly smaller than the weight of $(\Phi, \mathcal{F})$.

It follows easily from the definition of n that for all $(\Phi, \mathcal{F})$ in $\mathcal{P}(Sfor(\varphi_0)) \times \mathcal{PP}(Sfor(\varphi_0))$, $w(\Phi, \mathcal{F}) > 0$. It is routine to verify the second property in most cases and we only treat the most difficult case: case (6). That is, Φ contains a formula that is a variable, all formulas in Φ that are not variables are of the form $\alpha \bullet \nabla\Psi$ and Φ does not belong to $\mathcal{F}$. Recall that Φ' is the set Φ where we replace each variable $x \in X_\Phi$ by δ_x and that $\tau_{\mathcal{F}}(\Phi)$ is $\mu x_\Phi.\tau_{\mathcal{F}\cup\{\Phi\}}(\Phi')$. So we have to check that the weight of $(\Phi', \mathcal{F} \cup \{\Phi\})$ is strictly smaller than the weight of $(\Phi, \mathcal{F})$.

This follows from the following chain of inequalities:

$$\begin{aligned} w(\Phi', \mathcal{F} \cup \{\Phi\}) &= (n - |\mathcal{F}| - 1) \cdot (l+1) + \Sigma_{\varphi' \in \Phi'} s(\varphi'), && (\Phi \notin \mathcal{F}) \\ &\leq (n - |\mathcal{F}| - 1) \cdot (l+1) + l, && (\dagger) \\ &= (n - |\mathcal{F}|) \cdot (l+1) - 1, \\ &< (n - |\mathcal{F}|) \cdot (l+1) + \Sigma_{\varphi \in \Phi} s(\varphi), \\ &= w(\Phi, \mathcal{F}). \end{aligned}$$

Here (†) follows from the fact that Φ' is a subset of $Sfor(\varphi_0)$, the s of $Sfor(\varphi_0)$ is m and for all $\varphi' \in \Phi'$, $s(\varphi') \leq s(\varphi_0)$. □

5.5.6. Proposition. *A positive guarded sentence φ_0 in $\mu\mathrm{ML}^\nabla$ is completely additive p iff φ_0 is equivalent to $\tau_a(\varphi_0)$.*

Proof For the direction from right to left, suppose that φ_0 is equivalent to $\tau_a(\varphi_0)$. Looking at the definition of τ_a, we see that the formula $\tau_a(\varphi_0)$ belongs to the fragment $\mu\mathrm{ML}_A(p)$. By Proposition 5.5.4, this implies that $\tau_a(\varphi_0)$ is completely additive in p. Since φ_0 is equivalent to $\tau_a(\varphi_0)$, φ_0 is completely additive in p.

For the direction from left to right, fix a positive sentence $\varphi_0 \in \mu\mathrm{ML}^\nabla$ that is completely additive in p. In order to prove that φ_0 and $\tau_a(\varphi_0)$ are equivlent, consider an arbitrary pointed model $(\mathcal{T}, r)$. We show that

$$\mathcal{T}, r \Vdash \varphi_0 \quad \text{iff} \quad \mathcal{T}, r \Vdash \tau_a(\varphi_0). \tag{5.13}$$

Without loss of generality (see Proposition 5.1.1) we may assume that $\mathcal{T} = (T, R, V)$ is an ω-unravelled tree with root r. Recall we use abbreviations of the form $\mathcal{E}_0, \mathcal{E}_0', \ldots$ for initialized games, while we use abbreviations $\mathcal{E}, \mathcal{E}', \ldots$ for non-initialized games.

For the direction '⇒' of (5.13), assume that $\mathcal{T}, r \Vdash \varphi_0$. Since φ_0 is completely additive, there is a point $t_0 \in V(p)$ such that $\mathcal{T}[p \mapsto \{t_0\}], r \Vdash \varphi_0$. By Proposition 5.1.5 we may assume that $\exists$ has a positional winning strategy f in the evaluation game $\mathcal{E}_0' := \mathcal{E}(\mathcal{T}[p \mapsto \{t_0\}], \varphi_0)@(r, \varphi_0)$. We observe that any finite full f-conform $\mathcal{E}_0'$-match ends with a position of the form $(t, \top)$ or with a ∇-marking $m : \Phi \longrightarrow R[t]$ for which there is no pair (φ, u) satisfying $u \in m(\psi)$, since the match is won by $\exists$. In the later case, this means that the point t has no successor. Let π be any finite full f-conform $\mathcal{E}_0'$-match with last position $(t, \top)$. In order to simplify our proof later, we will assume that for all paths $u_0 u_1 \ldots$ such that $t = u_0$, we have that $\pi(u_1, \top)(u_2, \top)\ldots$ is an f-conform $\mathcal{E}_0'$-match.

We also let g be some maximal positional winning strategy $\exists$ in the evaluation game $\mathcal{E}^s := \mathcal{E}(\mathcal{T}, \tau_a(\varphi_0))$. In order to prove that $\mathcal{T}, r \Vdash \tau_a(\varphi_0)$, it suffices to provide $\exists$ with a winning strategy in the game $\mathcal{E}^s$ initialized at $(r, \tau_a(\varphi_0))$.

The winning strategy h will be defined by stages. After finitely many stages, $\forall$ will get stuck or we will reach a position that is winning for $\exists$ in $\mathcal{E}^s$ and she will then use the strategy g. This will guarantee that all h-conform $\mathcal{E}^s$-matches are won by $\exists$, as g is a winning strategy for $\exists$ in $\mathcal{E}^s$.

Before defining h, we introduce some notation and we prove a first claim that will correspond to a local construction of h (that is, during one stage).

Let P be the unique finite path from r to t_0. There are finitely many partial f-conform $\mathcal{E}'_0$-matches with first position (r, φ_0) and last position of the form (t_0, ψ), for some formula ψ. We denote by M the set of all these partial matches. If $\pi = (t_0, \varphi_0) \dots (t_n, \varphi_n)$ is a full finite f-conform $\mathcal{E}'_0$-match such that $t_n \in P$ and $t_n \neq t_0$, then $\varphi_n = \top$ since t has at least one successor. In particular, such a match π appears in disguise in M, namely as the match $\pi(u_1, \top) \dots (u_n, \top)$, where $u_0 \dots u_n$ is the path between t_n and t_0.

We say that a position (t, φ) in the game $\mathcal{E}^s$ *M-corresponds* to a set Φ of formulas if the two following conditions hold. First, for all $\chi \in \Phi$, (t, χ) occurs in a match in M and for all matches $\pi \in M$, there is a formula $\chi \in \Phi$ such that (t, χ) occurs in π. Second, for some family $\mathcal{F}$ of finite sets of formulas, we have $\varphi = \tau_{\mathcal{F}}(\Phi)$.

1. Claim. If (t, φ) M-corresponds to Φ and $\varphi \neq \top$, then $\exists$ has a strategy h in $\mathcal{E}^s@(t, \varphi)$ with the property that every h-conform partial match ρ leads to a position z (distinct from (t, φ)) satisfying condition (a) or (b) below:
(a) $z \in Win_{\exists}(\mathcal{E}^s)$;
(b) z M-corresponds to some set Ψ.

Proof of Claim Fix a point $t \in T$ and assume that (t, φ) M-corresponds to Φ. So for some family $\mathcal{F}$ of finite sets of formulas, we have $\varphi = \tau_{\mathcal{F}}(\Phi)$.

First, we show that we may assume that no $\chi \in \Phi$ is equal to $\top$, a variable, a conjunction or a formula of the form $\mu x.\psi$ or $\nu x.\psi$. Take a formula χ in Φ.

If $\chi = \top$, we may delete χ in Φ and it will still be the case that (t, φ) M-corresponds to Φ. If χ is a variable x, then we may replace x by δ_x in Φ. If χ is a conjunction of the form $\varphi_2 \wedge \varphi_3$, we can replace Φ by $(\Phi \backslash \{\chi\}) \cup \{\varphi_2, \varphi_3\}$. If χ is of the form $\mu x.\psi$ or $\nu x.\psi$, we may replace χ by ψ in Φ. In all the cases, it will still be true that (t, φ) M-corresponds to Φ.

Now suppose that a formula $\chi \in \Phi$ is of the form $\varphi_1 \vee \varphi_2$. Then, in the $\mathcal{E}'_0$-game, it is $\exists$'s turn at position $(t, \varphi_1 \vee \varphi_2)$. Following her strategy f, she moves to position (t, φ_l), where $l \in \{1, 2\}$.

Now, let Φ_i be the set Φ where we replace φ by φ_i ($i \in \{1, 2\}$). Remember that from our definition of τ_a, it follows that

$$\tau_{\mathcal{F}}(\Phi) = \tau_{\mathcal{F}}(\Phi_1) \vee \tau_{\mathcal{F}}(\Phi_2).$$

Since $\varphi = \tau_{\mathcal{F}}(\Phi)$, it is $\exists$'s turn in the game $\mathcal{E}^s$, at position (t, φ). We let the strategy h be such that $\exists$ chooses the position $z = (t, \tau_{\mathcal{F}}(\Phi_l))$. The position z M-corresponds to Ψ, where Ψ is the set Φ in which we replaced χ by φ_l. Therefore, condition (b) holds.

Next suppose that each formula in Φ is of the form $\alpha \bullet \nabla\Psi$. We let Φ be the set $\{\varphi_1, \dots, \varphi_k\}$, where for all $i \in \{1, \dots, k\}$, the formula φ_i is of the form $\alpha_i \bullet \nabla\Phi_i$. Since (t, φ) M-corresponds to Φ, there exists a match $\pi_i \in M$ such that (t, φ_i) occurs in π_i. In the partial $\mathcal{E}'_0$-match π_i, the position $(t, \alpha_i \bullet \nabla\Phi_i)$ belongs to $\exists$. Since π_i is won by $\exists$, this implies that α_i is true at t and according to f, $\exists$ picks a ∇-marking $m_i : \Phi_i \to \mathcal{P}(R[t])$.

First we show that

$$p \text{ occurs in } \alpha_1 \wedge \dots \wedge \alpha_k \quad \text{iff} \quad t = t_0. \tag{5.14}$$

For the direction from left to right, suppose that p occurs in $\alpha_1 \wedge \dots \wedge \alpha_k$. Then there is an $i \in \{1, \dots, k\}$ such that α_i contains p. Recall that by definition of M-correspondence, (t, φ_i) is a winning position for $\exists$ in $\mathcal{E}'_0$. In particular, α_i is true at t, which implies that p is true at t. In the game $\mathcal{E}'_0$, there is a unique point in T making p true, namely the point t_0. So $t = t_0$.

For the direction from right to left of equivalence (5.14), assume that $t = t_0$. Since φ_0 is completely additive in p, φ_0 is not true at r in the model $\mathcal{T}[p \mapsto \emptyset]$. So this means that $\exists$ does not have a winning strategy in the game $\mathcal{E}_0^{\perp} = \mathcal{E}(\mathcal{T}[p \mapsto \emptyset], \varphi_0)@(r, \varphi_0)$. In particular, f is not a winning strategy for $\exists$ in $\mathcal{E}_0^{\perp}$. The winning conditions of $\mathcal{E}'_0$ and $\mathcal{E}_0^{\perp}$ are the same. The rules are the same, except when we reach a position of the form $(t_0, \alpha \bullet \nabla\Psi)$. The difference between the two games is that in $\mathcal{E}'_0$, α may contain p and $\exists$ could still win. In $\mathcal{E}_0^{\perp}$, if α contains p, then $\exists$ looses. Hence, in order to show that f is a a winning strategy for $\exists$ in $\mathcal{E}_0^{\perp}$, it is sufficient to prove that no position of the form $(t_0, (\alpha \wedge p) \bullet \nabla\Psi)$ is reached in an f-conform $\mathcal{E}'_0$-match with starting position (r, φ_0). It follows from the definition of M that there is a partial match $\pi_i \in M$ such that a position of the form $(t_0, (\alpha \wedge p) \bullet \nabla\Psi)$ occurs in π_i. This implies that $\alpha \wedge p$ is equal to α_i, so p occurs in $\alpha_1 \wedge \dots \wedge \alpha_k$ and this finishes the proof of equivalence (5.14).

Next, we distinguish two cases:

- First, assume that p does not occur in $\alpha_1 \wedge \dots \wedge \alpha_k$. By (5.14), $t \neq t_0$. So t has a successor u on P.

  Recall that φ is equal to $\tau_{\mathcal{F}}(\Phi)$ and $\Phi = \{\varphi_1, \dots, \varphi_k\}$, where for all $i \in \{1, \dots, k\}$, $\varphi_i = \alpha_i \bullet \nabla\Phi_i$. By definition of τ_a, since p does not occur in $\alpha_1 \wedge \dots \wedge \alpha_k$, φ is equal to

  $$\bigvee \Big\{ \Diamond\tau_{\mathcal{F}}\Big(\bigcup_{i=1}^{k}\Phi'_i\Big) \wedge \Big(\bigwedge_{i=1}^{k}\alpha_i \wedge \nabla e^{\perp}[\Phi''_i]\Big) \;\Big|\; \text{for all } 1 \leq i \leq k, \; \Phi'_i \neq \emptyset, \Phi'_i \cup \Phi''_i = \Phi_i \Big\}. \tag{5.15}$$

  At position (t, φ) in the game $\mathcal{E}^s$, it is $\exists$'s turn and in order to pick one of the disjuncts in (5.15), she has to choose for all $i \in \{1, \dots, k\}$, sets Φ'_i and Φ''_i such that $\Phi'_i \neq \emptyset$ and $\Phi'_i \cup \Phi''_i = \Phi_i$.

Take $i \in \{1, \ldots, k\}$. Recall that we denote by m_i the ∇-marking chosen by $\exists$ at position $(t_i, \alpha_i \bullet \Phi_i)$, according to f. We define Φ'_i as the set $\{\psi \in \Phi_i \mid u \in m_i(\psi)\}$ and Φ''_i as the set $\{\psi \in \Phi_i \mid \text{ for some } u' \neq u, u' \in m_i(\psi)\}$. Since m_i is a ∇-marking, the conditions $\Phi'_i \neq \emptyset$ and $\Phi'_i \cup \Phi''_i = \Phi_i$ are easily verified. Recall that by definition of the match π_i, after $\exists$ chose the marking m_i in π_i, $\forall$ picks a formula ψ_i such that $u \in m_i(\psi_i)$. .

So the next position in the game $\mathcal{E}^s$ is the position $z_0 = (t, \Diamond\tau_{\mathcal{F}}\left(\bigcup_{i=1}^k \Phi'_i\right) \wedge \left(\bigwedge_{i=1}^k \alpha_i \wedge \nabla e^{\perp}[\Phi''_i)]\right))$. It is $\forall$'s turn and there are two possibilities.

- Suppose first that at position z_0, $\forall$ chooses the conjunct $\Diamond\tau_{\mathcal{F}}\left(\bigcup_{i=1}^k \Phi'_i\right)$. Then $\exists$ has to play and we let her choose the successor u of t, moving to position $z := (u, \tau_{\mathcal{F}}\left(\bigcup_{i=1}^k \Phi'_i\}\right)$. To show that condition (b) is satisfied, recall that for all $i \in \{1, \ldots, k\}$, ψ_i is the formula chosen by $\forall$ in π_i, after $\exists$ played the marking m_i. We let Ψ be the set $\{\psi_1, \ldots, \psi_n\}$ and we prove that z M-corresponds to Φ.
  It is sufficient to show that $\tau_{\mathcal{F}}\left(\bigcup_{i=1}^k \Phi'_i\}\right) = \tau_{\mathcal{F}}(\{\psi_1, \ldots, \psi_k\})$. So we can restrict ourselves to prove that $\{\psi_1, \ldots, \psi_k\} = \bigcup_{i=1}^k \Phi'_i$. By definition of Φ'_i and ψ_i, we have that for all $i \in \{1, \ldots, k\}$, ψ_i belongs to Φ'_i. So we only have to show that for all $i \in \{1, \ldots, k\}$, Φ'_i is a subset of $\{\psi_1, \ldots, \psi_k\}$.
  Take $i \in \{1, \ldots, k\}$ and take a formula $\psi \in \Phi'_i$. Let π'_i be the partial f-conform $\mathcal{E}'_0$-match leading from (r, φ_0) to (t, m_i) and such that π'_i is a prefix of π_i. Then there is a partial f-conform $\mathcal{E}'_0$-match π which extends $\pi'_i \circ (u, \psi)$ and the last position of which is of the form (t_0, χ). Since M is the collection of all partial f-conform $\mathcal{E}'_0$-matches of which the last position is of the form (t_0, χ), π belongs to M and there is a natural number j such that $\pi = \pi_j$. It follows that $\psi = \psi_j$. So $\Phi'_i \subseteq \{\psi_1, \ldots, \psi_k\}$ and this finishes the proof that the position z M-corresponds to the set Ψ.
- Suppose next that $\forall$ chooses the conjunct $\bigwedge_{i=1}^k \left(\alpha_i \wedge \nabla e^{\perp}[\Phi''_i]\right)$. Then it is again $\forall$ who has to make a move and he chooses a natural number $i \in \{1, \ldots, k\}$. Then the next position is $z_1 := (t, \alpha_i \wedge \nabla e^{\perp}[\Phi''])$. Suppose that $\forall$ chooses the first conjunct α_i. Then we have to check that α_i is true at t. This follows from the facts that π_i is a partial f-conform $\mathcal{E}$-match and that the position $(t, \alpha_i \bullet \nabla\Phi_i)$ occurs in π_i.
  Suppose next that at position z_1, $\forall$ chooses the second conjunct $\nabla e^{\perp}[\Phi''_i]$, leading to the position $(t, \nabla e^{\perp}[\Phi''_i])$. Then $\exists$ has to come up with a ∇-marking $m : e^{\perp}[\Phi''_i] \longrightarrow \mathcal{P}(R[t])$.
  Since $\mathcal{T}$ is ω-unravelled, there is a state $v_u \in R(t) \setminus \{u\}$ such that $\mathcal{T}, u \leftrightarroweq \mathcal{T}, v_u$. Now we define $m : e^{\perp}\Phi''_i \longrightarrow \mathcal{P}(R[t])$ such that for all

$\psi \in \Phi''_i$, we have

$$m(e^{\perp}(\psi)) = \{v \mid v \in m_i(\psi), v \neq u\} \cup \{u \mid v_u \in m_i(\psi)\}.$$

In order to show that $\exists$ is allowed to make this move, we will prove that

$$m : e^{\perp}[\Phi''_i] \longrightarrow \mathcal{P}(R[t]) \text{ is a } \nabla\text{-marking.} \tag{5.16}$$

Given a successor v of t, we make the following case distinction. If $v \neq u$, then since m_i is a ∇-marking, there is a formula $\psi \in \Phi$ such that $v \in m_i(\psi)$. So by definition of Φ''_i, ψ belongs to the set Φ''_i. Hence we get that $v \in m(e^{\perp}(\psi))$. If, on the other hand, v is equal to u, then there exists a formula $\psi \in \Phi$ such that $v_u \in m_i(\psi)$. It follows from the definitions of v_u and Φ''_i that $\psi \in \Phi''_i$. Putting this together with the definition of m, we get $u \in m(e^{\perp}(\psi))$. Conversely, an arbitrary formula in $e^{\perp}[\Phi''_i]$ is of the form $e^{\perp}(\psi)$ for some $\psi \in \Phi''_i$. Then by definition of Φ''_i there is some state $v \in R(t) \setminus \{u\}$ such that $v \in m_i(\psi)$, and thus $v \in m(e^{\perp}(\psi))$. This proves (5.16).

The game continues with $\forall$ choosing a pair $(v, e^{\perp}(\psi))$ such that $v \in m(e^{\perp}(\psi))$, as the next position in the $\mathcal{E}^s$-match. If there is no such a pair, then $\forall$ gets stuck and condition (a) is met immediately. Otherwise, we will show that (a) holds in any case since we have

$$\{(v, e^{\perp}(\psi)) \mid v \in m(e^{\perp}(\psi)\} \subseteq \mathit{Win}_{\exists}(\mathcal{E}^s). \tag{5.17}$$

For a proof of (5.17), take an arbitrary pair $(v, e^{\perp}(\psi))$ such that $v \in m(e^{\perp}(\psi))$. Suppose first that $v \neq u$. Hence, v belongs to $m_i(\psi)$. Since m_i was part of $\exists$'s winning strategy f, we may conclude that (v, ψ) is a winning position for $\exists$ in $\mathcal{E}'_0$. Then by Proposition 5.1.5 it follows that $\mathcal{T}[p \mapsto \{t_0\}], v \Vdash e(\psi)$. Since φ is positive in p and ψ is a subformula of φ, it implies that $\mathcal{T}[p \mapsto \emptyset], v \Vdash e(\psi)$. By Fact 5.1.6 we may infer that $\mathcal{T}, v \Vdash e^{\perp}(\psi)$, and so $(v, e^{\perp}(\psi)) \in \mathit{Win}_{\exists}(\mathcal{E}^s)$.

Next suppose that $v = u$. By definition of m, v_u belongs to $m(\psi)$ (where v_u is a sibling of u such that $\mathcal{T}, u \leftrightarroweq \mathcal{T}, v_u$). As in the previous case it follows from Proposition 5.1.5 and Fact 5.1.6 that $\mathcal{T}, v_u \Vdash e^{\perp}(\psi)$, and so by $\mathcal{T}, u \leftrightarroweq \mathcal{T}, v_u$ we obtain that $\mathcal{T}, u \Vdash e^{\perp}(\psi)$. From this again it is immediate that $(u, e^{\perp}(\psi)) \in \mathit{Win}_{\exists}(\mathcal{E}^w)$. This finishes the proof of (5.17), and shows that condition (a) holds for any pair chosen by $\forall$.

- Next assume that p does occur in $\alpha_1 \wedge \dots \wedge \alpha_k$. Then by (5.14), $t = t_0$. The formula φ is equal to $\tau_{\mathcal{F}}(\Phi)$ and $\Phi = \{\varphi_0, \dots, \varphi_k\}$, where for all $i \in \{1, \dots, k\}$, $\varphi_i = \alpha_i \bullet \nabla\Phi_i$. By definition of τ_a and since p occurs in $\alpha_1 \wedge \dots \wedge \alpha_{k'}$, φ is equal to

$$(\alpha_1 \wedge \dots \wedge \alpha_k) \wedge \bigwedge\{\nabla e^{\perp}[\Phi_i]) : 1 \leq i \leq k'\}.$$

At position (t_0, φ), in the game $\mathcal{E}^s$, it is $\forall$'s turn. First suppose that he chooses the first conjunct $\alpha_1 \wedge \cdots \wedge \alpha_k$. Then, for all $i \in \{1, \ldots, k\}$, we have to verify that α_i is true at t_0. This follows from the facts that π_i is an f-conform $\mathcal{E}'_0$-match and that the position $(t, \alpha_i \bullet \nabla\Phi_i)$ occurs in π_i.

Suppose next that $\forall$ chooses the second conjunct $\bigwedge\{\nabla e^{\perp}(\Phi_i)) : 1 \leq i \leq k'\}$. Then it is again $\forall$ who has to play and he chooses a natural number $i \in \{1, \ldots, k\}$, leading to the position $(t_0, \nabla e^{\perp}[\Phi''_i])$. $\exists$ has to come up with a ∇-marking $m : e^{\perp}[\Phi_i] \longrightarrow \mathcal{P}(R[t_0])$. We define m such that for all $\psi \in \Phi_i$,

$$m(e^{\perp}(\psi)) = m_i(\psi).$$

Recall that m_i is the ∇-marking chosen by $\exists$ (according to the winning strategy f) at position $(t, \alpha_i \bullet \nabla\Phi_i)$ in the $\mathcal{E}'_0$-match π_i. Since m_i is a ∇-marking, we also have that $m : e^{\perp}[\Phi_i] \longrightarrow \mathcal{P}(R[t_0])$ is a ∇-marking. Then $\forall$ has to play and he may choose a position $(u, e^{\perp}(\psi))$ such that $u \in m(e^{\perp}(\psi))$. If there is no such a position, then $\forall$ gets stuck and condition (a) is met. Otherwise, we can show that condition (a) holds in any case. The proof is similar to the one of (5.17), so we leave the details to the reader.

This finishes the proof of the claim. ◀

Now we define a winning strategy h for $\exists$ in $\mathcal{E}^s@(r, \tau_a(\varphi_0))$. It is immediate by Claim 1 that we can define a strategy h for $\exists$ in $\mathcal{E}^s$ such that for all h-conform full $\mathcal{E}^s$-matches π with initial position $(r, \tau_a(\varphi_0))$, we have:

(†) If $\forall$ does not get stuck, then $\pi = z_0 z_1 \ldots$ is infinite and
(*i*) either there is $i \in \mathbb{N}$ such that $z_i \in Win_{\exists}(\mathcal{E}^s)$ and $z_i z_{i+1} \ldots$ is a g-conform $\mathcal{E}^s$-match,
(*ii*) or for all $i \in \mathbb{N}$, there exists $j > i$ such that z_j M-corresponds to some set of formulas.

So we let h be a strategy for $\exists$ in $\mathcal{E}^s$ such that (†) holds. In order to prove that h is a winning strategy h for $\exists$ in $\mathcal{E}^s@(r, \tau_a(\varphi_0))$, let π be a full h-conform match. First, if π is finite, then, by (†), π is won by $\exists$. Suppose now that $\pi = z_0 z_1 \ldots$ is infinite. If (*i*) holds, then it follows immediately from the fact that g is a winning strategy for $\exists$, that π is won by $\exists$.

Next we prove that (*ii*) can never happen. Recall that if a position (t, φ) M-corresponds to some set of formulas, then t belongs to P. So if there are infinitely many positions in π that $\mathcal{M}$-correspond to some positions in $\pi_1, \ldots, \pi_k$, this means that the match π never leaves the path P. Since φ_0 is guarded, $\tau_a(\varphi_0)$ is guarded and so every infinite $\mathcal{E}^s$-match corresponds to an infinite path, which contradicts the fact that P is finite. This finishes the proof that h is a winning strategy for $\exists$ in $\mathcal{E}^s@(r, \tau_a(\varphi_0))$ and the proof of the direction '$\Rightarrow$' of (5.13).

For the other direction of (5.13), suppose that $\mathcal{T}, r \Vdash \tau_a(\varphi_0)$. So $\exists$ has a positional winning strategy f^s in the game $\mathcal{E}^s_0 := \mathcal{E}(\mathcal{T}, \tau_a(\varphi_0))@(r, \tau_a(\varphi_0))$. If x

is a variable of $\tau_a(\varphi_0)$, we denote by δ^s_x the unique subformula of $\tau_a(\varphi_0)$ such that $\mu x.\delta^s_x$ or $\nu x.\delta^s_x$ is a subformula of $\tau_a(\varphi_0)$. We also let g be some maximal positional strategy for $\exists$ in $\mathcal{E} := \mathcal{E}(\mathcal{T}, \varphi_0)$.

We need to define a winning strategy h for $\exists$ in the game $\mathcal{E}$ initiated at (r, φ_0). As before, the strategy h will be defined in stages. After finitely many stages, we will reach a position that is winning for $\exists$ in $\mathcal{E}$ and we will then use the strategy g. This will guarantee that all h-conform $\mathcal{E}$-matches are won by $\exists$, as g is a winning strategy for $\exists$ in $\mathcal{E}$. For the definition of the first stages of h, we make use of a specific partial f^s-conform $\mathcal{E}^s_0$-match π that is defined independently of h.

Intuitively, the match π is the longest partial f^s-conform $\mathcal{E}^s_0$-match such that each of its position contains a formula in which p is active. The construction of π is done by stages. More precisely, we define π as the final stage of an inductively defined sequence of partial f^s-conform $\mathcal{E}^s_0$-matches $\pi_0, \dots, \pi_n$ such that for all $i < n$, $\pi_i \sqsubseteq \pi_{i+1}$ (where $\sqsubseteq$ denotes the initial-segment relation). Moreover, the last position of each π_i will be of the form $(t, \tau_{\mathcal{F}}(\Phi))$, for some $t \in T$, $\Phi \subseteq Sfor(\varphi_0)$ and $\mathcal{F} \subseteq \mathcal{P}(Sfor(\varphi_0))$.

The match π_0 is the single position match $(r, \tau_a(\varphi_0))$. Next suppose that we already defined the match π_i and that the last position reached in π_i is (t, φ) where $\varphi = \tau_{\mathcal{F}}(\Phi)$, for some $t \in T$, $\Phi \subseteq Sub(\varphi_0)$ and $\mathcal{F} \subseteq \mathcal{P}(Sfor(\varphi_0))$.

First we show that we may assume that no formula in Φ is a conjunction or a formula of the form $\mu x.\chi$ or $\nu x.\chi$. If $\psi \in \Phi$ is a conjunction $\varphi_1 \wedge \varphi_2$, we can replace Φ by $(\Phi \backslash \{\psi\}) \cup \{\varphi_1, \varphi_2\}$ and it is still be the case that $\varphi = \tau_{\mathcal{F}}(\Phi)$. If $\psi \in \Phi$ is a formula of the form $\mu x.\chi$ or $\nu x.\chi$, we may replace ψ by χ in Φ and we still have $\varphi = \tau_{\mathcal{F}}(\Phi)$. Without loss of generality, we can also suppose that if $\Phi \neq \{\top\}$, then $\top \notin \Phi$ (if $\Phi \neq \{\top\}$ and $\top \in \Phi$, we may delete $\top$ from Φ and we still have $\varphi = \tau_{\mathcal{F}}(\Phi)$).

To define the match π_{i+1}, we make the following case distinction:

- Suppose that $\Phi = \{\top\}$. Then $\varphi = \top$. We stop the construction and we define π as π_i.

- Suppose that Φ contains a formula of the form $\varphi_1 \vee \varphi_2$. Then the formula φ is equal to $\tau_{\mathcal{F}}(\Phi_1) \vee \tau_{\mathcal{F}}(\Phi_2)$, where Φ_i is the set Φ in which we replaced $\varphi_1 \vee \varphi_2$ by φ_i ($i \in \{1, 2\}$). So using f^s, at position (t, φ), $\exists$ chooses a position z of the form $(t, \tau_{\mathcal{F}}(\Phi_l))$, where $l \in \{1, 2\}$. The match π_{i+1} is defined by $\pi_{i+1} := \pi_i \circ z$.

- Suppose that Φ contains a variable but no formula of the form $\varphi_1 \vee \varphi_2$. Assume also that Φ belongs to $\mathcal{F}$. Then φ is equal to x_Φ. The position in $\mathcal{E}^s_0$ following (t, φ) is $z = (t, \delta^s_{x_\Phi})$. The match π_{i+1} is defined by $\pi_{i+1} := \pi_i \circ z$.

- Suppose that Φ contains a variable but no formula of the form $\varphi_1 \vee \varphi_2$. Assume also that Φ does not belong to $\mathcal{F}$. Then φ is equal to $\mu x_\Phi.\tau_{\mathcal{F} \cup \{\Phi\}}(\Phi')$, where Φ' is the set Φ in which we replace each variable $x \in \Phi$ by δ_x. The position in $\mathcal{E}_0^s$ following (t, φ) is $z = (t, \tau_{\mathcal{F} \cup \{\Phi\}}(\Phi'))$. The match π_{i+1} is defined by $\pi_{i+1} := \pi_i \circ z$.

- Suppose that Φ is a set of the form $\{\alpha_i \bullet \nabla \Phi_i \mid i \in \{1, \dots, k\}\}$. If $\alpha_1 \wedge \dots \wedge \alpha_k$ contains p, then we stop the construction and we define π as π_i. Otherwise, the formula φ is equal to

$$\bigvee \left\{ \Diamond \tau_{\mathcal{F}} \left(\bigcup_{i=1}^{k} \Phi_i' \right) \wedge \bigwedge_{i=1}^{k} \alpha_i \wedge \nabla e^{\perp}[\Phi_i''] \;\middle|\; \Phi_i' \neq \emptyset, \Phi_i' \cup \Phi_i'' = \Phi_i \right\}.$$

  So, according to f^s, $\exists$ chooses for all $i \in \{1, \dots, k\}$, sets Φ_i' and Φ_i'' such that $\Phi_i' \neq \emptyset$ and $\Phi_i' \cup \Phi_i'' = \Phi_i$. Then, $\forall$ has to make a move and in order to continue our definition of π, we can let him choose the conjunct $\Diamond \tau_{\mathcal{F}} \left(\bigcup_{i=1}^{k} \Phi_i' \right)$. So using f^s, $\exists$ has to pick a successor u of t and the next position z is $\left(u, \tau_{\mathcal{F}} \left(\bigcup_{i=1}^{k} \Phi_i' \right)\right)$. We define π_{i+1} as an f^s-conform such that $\pi_i \sqsubseteq \pi_{i+1}$ and the last position of π_{i+1} is z.

Now we show that the above process terminates, so that we always end up with a correctly defined match $\pi = \pi_n$, for some n. Suppose for contradiction that it is not the case. So for all $i \in \mathbb{N}$, the partial match π_i is well-defined. We let π' be the unique infinite match such that for all $i \in \mathbb{N}$, $\pi_i \sqsubseteq \pi'$. It follows from the definition of the π_i's, that π' is an f^s-conform $\mathcal{E}_0^s$-match. Moreover, if we look at the definition of the π_i's, we can see that if a position of the form (t, x) occurs in π', then x is equal to a formula of the form $\tau_{\mathcal{F}}(\Phi)$. Given the definition of τ_a, this means that all the unfolded variables in π' are μ-variables. So $\exists$ looses π'. This contradicts the fact that π' is an f^s-conform $\mathcal{E}_0^s$-match and that f^s is a winning strategy for $\exists$.

Now that we defined the finite match π, we are ready to provide $\exists$ with a winning strategy h in the game $\mathcal{E}'$. As in the proof of the direction '$\Rightarrow$' of (5.13), we first prove a preliminary claim that allow us to define h for the first few moves. We say that a position (t, ψ) in the game $\mathcal{E}$ π-*corresponds* to (t, φ) if (t, φ) occurs in π and for some set $\Psi \subseteq Sub(\varphi_0)$ containing ψ and some family $\mathcal{F}$ of subsets of $Sub(\varphi_0)$, we have $\varphi = \tau_{\mathcal{F}}(\Psi)$.

2. Claim. If (t, ψ) π-corresponds to (t, φ) and $\psi \neq \top$, then $\exists$ has a strategy h in $\mathcal{E}@(t, \psi)$ with the property that every h-conform partial match ρ leads to a position z (distinct from (t, ψ)) satisfying condition (a) or (b) below:
(a) $z \in Win_{\exists}(\mathcal{E})$;
(b) z π-corresponds to a position (u, φ') .

Proof of Claim Fix a position (t, ψ) which π-corresponds to a position (t, φ). So for some set $\Psi \subseteq Sub(\varphi_0)$ containing ψ and some family $\mathcal{F}$ of subsets of

$Sub(\varphi_0)$, we have

$$\varphi = \tau_{\mathcal{F}}(\Psi)\,. \tag{5.18}$$

We may assume that $\Psi = \{\psi_1, \ldots, \psi_k\}$, $\psi = \psi_1$ and for all $i \in \{2, \ldots, k\}$, ψ_i is distinct from ψ.

We will define the strategy h depending on the shape of ψ_1. Suppose first that ψ_1 is a formula of the form $\mu x.\chi$ or $\nu x.\chi$. Then the position following (t, ψ_1) in the game $\mathcal{E}$ is $z := (t, \chi)$. Let Ψ' be the set Ψ in which we replace ψ_1 by χ. Since $\varphi = \tau_{\mathcal{F}}(\Psi')$, the position z satisfies condition (b).

Next assume that ψ_1 is a conjunction $\chi_1 \wedge \chi_2$. Then in the game $\mathcal{E}$, it is $\forall$'s turn and he can choose between the positions (t, χ_1) and (t, χ_2). Suppose that $\forall$ chooses the position $z := (t, \chi_l)$, where $l \in \{1, 2\}$. Let Ψ' be the set $(\Phi \backslash \{\psi_1\}) \cup \{\chi_1, \chi_2\}$. We have that $\tau_{\mathcal{F}}(\Psi) = \tau_{\mathcal{F}}(\Psi')$. So z π-corresponds to (t, φ), since χ_l occurs in Ψ' and $\varphi = \tau_{\mathcal{F}}(\Psi')$.

Now suppose that ψ_1 is a variable x. Then the position following (t, ψ_1) in the game $\mathcal{E}$ is $z = (t, \delta_x)$. Let X_Ψ be the set of variables y such that $y \in \Psi$ and let Ψ' be the set Ψ in which each variable $y \in \Psi$ is replaced by δ_y. If Ψ does not belong to $\mathcal{F}$, then φ is equal to $\mu x_\Psi.\tau_{\mathcal{F} \cup \{\Psi\}}(\Psi')$. So in $\mathcal{E}_0^s$, the position following the position (t, φ) is $z := (t, \tau_{\mathcal{F} \cup \{\Psi\}}(\Psi')$. The position z satisfies condition (b).

On the other hand, if Ψ belongs to $\mathcal{F}$, then φ is equal to x_Ψ. So in $\mathcal{E}_0^s$, the positions following the position $(t, \varphi) = (t, x_\Psi)$ are of the form $(t, \mu x_\Psi.\tau_{\mathcal{F}'}(\Psi''))$ and $z := (t, \tau_{\mathcal{F}'}(\Psi''))$ respectively, for some $\mathcal{F}'$ and Ψ''. So the position z satisfies condition (b).

Suppose now that ψ_1 is a formula of the form $\chi_1 \vee \chi_2$. Let Ψ_1 be the set Ψ in which we replace ψ_1 by χ_1 and let Ψ_2 be the set Ψ in which we replace ψ_1 by χ_2. Then we have $\varphi = \tau_{\mathcal{F}}(\Psi_1) \vee \tau_{\mathcal{F}}(\Psi_2)$. So at position (t, φ), in the $\mathcal{E}_0^s$-game, it is $\exists$ who has to play and suppose that according to f^s, she chooses a disjunct $\tau_{\mathcal{F}}(\Psi_l)$, where l belongs to $\{1, 2\}$. We define h such that in the game $\mathcal{E}$, at position $(t, \chi_1 \vee \chi_2)$, $\exists$ chooses the position $z := (t, \chi_l)$. Then condition (b) is met.

It remains to consider the case when ψ_1 is a formula of the form $\alpha_1 \bullet \nabla \Phi_1$. So we have to verify that α_1 is true at t and we have to provide $\exists$ with a ∇-marking $m : \Phi_1 \longrightarrow \mathcal{P}(R[t])$. First, we show that we may assume that for all $i \in \{2, \ldots, l\}$, the formula ψ_i is of the form $\alpha_i \bullet \nabla \Phi_i$.

Take $i \in \{2, \ldots, l\}$. If ψ_i is the formula $\top$, we delete it from the set Ψ. Suppose next that ψ_i is a variable. We let X_Ψ be the set of variables y such that $y \in \Psi$ and we let Ψ' be the set Ψ in which each variable $y \in \Psi$ is replaced by δ_y. We observe that $\psi_1 \notin X_\Psi$ and $\psi_1 \in \Psi'$. If Ψ does not belong to $\mathcal{F}$, then φ is equal to $\mu x_\Psi.\tau_{\mathcal{F} \cup \{\Psi\}}(\Psi')$. So if we replace Ψ by Ψ' and φ by $\tau_{\mathcal{F} \cup \{\Psi\}}(\Psi')$, equality (5.18) remains true. On the other hand, if Ψ belongs to $\mathcal{F}$, then φ is equal to x_Ψ. It follows from the definition of τ_a that $\delta^s_{x_\Psi}$ is equal to $\tau_{\mathcal{F}}(\Psi')$. So if we replace Ψ by Ψ' and φ by $\tau_{\mathcal{F}}(\Psi')$, equality (5.18) remains true.

If ψ_i is of the form $\mu x.\chi_i$ or $\nu x.\chi_i$, then we can replace ψ_i by χ_i in Ψ and equality (5.18) still holds. Finally suppose that ψ_i is a formula of the form $\chi_1 \vee \chi_2$. Let Ψ_1 be the set Ψ in which we replace ψ_i by χ_1 and let Ψ_2 be the set Ψ in which we replace ψ_i by χ_2. Then we have $\varphi = \tau_{\mathcal{F}}(\Psi_1) \vee \tau_{\mathcal{F}}(\Psi_2)$. So at position (t, φ), in the $\mathcal{E}_0^s$-game, it is $\exists$ who has to play and she chooses a disjunct $\tau_{\mathcal{F}}(\Psi_l)$, where l belongs to $\{1, 2\}$. So if we replace Ψ by Ψ_l and φ by $\tau_{\mathcal{F}}(\Psi_l)$, equality (5.18) remains true.

We observe that in order to assume that the formula ψ_i is of the form $\alpha_i \bullet \nabla\Phi_i$, we may have to use one after each other the transformations described in the last two paragraphs. We leave out the proof that this process will finish at some point, but this basically follow from the fact that the formula ψ_i is guarded.

This finishes the proof that without loss of generality, we may assume that for all $i \in \{2, \dots, l\}$, the formula ψ_i is of the form $\alpha_i \bullet \nabla\Phi_i$. Now to define h, we make the following case distinction:

- Suppose that $\alpha_1 \wedge \dots \wedge \alpha_k$ does not contain p. Then φ is equal to the formula

$$\bigvee \Big\{ \Diamond\tau_{\mathcal{F}} \Big(\bigcup_{i=1}^{k} \Phi'_i \Big) \wedge \Big(\bigwedge_{i=1}^{k} \alpha_i \wedge \nabla e^{\perp}[\Phi''_i] \Big) \;\Big|\; \text{for all } 1 \leq i \leq k, \\ \Phi'_i \neq \emptyset, \Phi'_i \cup \Phi''_i = \Phi_i \Big\}.$$

  So in the game $\mathcal{E}_0^s$, at position (t, φ), it is $\exists$'s turn and assume that according to f^s, for all $i \in \{1, \dots, k\}$, she chooses sets Φ'_i and Φ''_i such that $\Phi'_i \neq \emptyset$ and $\Phi'_i \cup \Phi''_i = \Phi_i$, leading the $\mathcal{E}_0^s$-match to a position z'. Then it is $\forall$ who makes a move in $\mathcal{E}_0^s$:

  - Suppose that $\forall$ chooses the conjunct $\bigwedge_{i=1}^{k} \alpha_i \wedge \nabla e^{\perp}[\Phi''_i]$. Then it is again $\forall$ who has to play in $\mathcal{E}_0^s$. If $\forall$ picks the conjunct α_1, the match is finite and α_1 has to be true at t, since f^s is a winning strategy for $\exists$. If $\forall$ chooses the conjunct $\nabla e^{\perp}[\Phi''_1]$, then, according to f^s, $\exists$ has to come up with a ∇-marking $m_1 : e^{\perp}[\Phi''_1]) \longrightarrow \mathcal{P}(R[t])$.
  - If at position z' in $\mathcal{E}_0^s$, $\forall$ chooses the conjunct $\Diamond\tau_{\mathcal{F}}\left(\bigcup_{i=1}^{k} \Phi'_i\right)$, then according to f^s, $\exists$ picks a point $u \in R[t]$. The next position is (u, φ'), where $\varphi' = \tau_{\mathcal{F}}\left(\bigcup_{i=1}^{l} \Phi'_i\right)$. Recall that by definition of π, this position occurs in π.

  We now go back to the game $\mathcal{E}$. Recall that our goal is to prove that α_1 is true at t and to provide a ∇-marking $m : \Phi_1 \longrightarrow \mathcal{P}(R[t])$. It follows from the previous paragraphs that α_1 is true at t. We define m as the map such that for all $\chi \in \Phi_1$,

  - if χ belongs to $\Phi'_1 \backslash \Phi''_1$, $m(\chi) := \{u\}$,
  - if χ belongs to Φ''_1, $m(\chi) := \{v \mid v \in m_1(e^{\perp}(\chi))\} \cup \{u \mid \chi \in \Phi'_1\}$.

We verify that $m : \Phi_1 \longrightarrow \mathcal{P}(R[t])$ is a ∇-marking. Fix a successor v of t. Since m_1 is a ∇-marking, there is a formula χ such that $v \in m_1(\chi)$; hence, $v \in m(\chi)$. Next, fix a formula γ in Φ. If γ belongs to Φ''_1, then there is a successor v of t such that $v \in m_1(e^{\perp}(\gamma))$, since m_1 is a ∇-marking. So v belongs to $m(\gamma)$. Otherwise, γ belongs to Φ'_1 and then, u belongs to $m(\gamma)$.

After $\exists$ chose m, it is $\forall$ who has to move in $\mathcal{E}$. Suppose that he picks a position $z = (v, \chi)$, such that $v \in m(\chi)$. There are two possibilities: either $v \in m_1(e^{\perp}(\chi))$ or $v = u$ and χ belongs to Φ'_1. First suppose that v belongs to $m_1(e^{\perp}(\chi))$. Then we show that at position z, condition (a) is met. Since m_1 was part of $\exists$'s winning strategy f^s, we may conclude that $(v, e^{\perp}(\chi))$ is a winning position for $\exists$ in $\mathcal{E}^s_0$. Then by Proposition 5.1.5 it follows that $\mathcal{T}, v \Vdash e^{\perp}(\chi)$, and by Fact 5.1.6 we know that $\mathcal{T}[p \mapsto \emptyset], v \Vdash \chi$. Since χ is positive in p, this gives $\mathcal{T}, v \Vdash e(\chi)$. So (v, χ) belongs to $Win_{\exists}(\mathcal{E})$.

Next assume that $v = u$ and χ belongs to Φ'_1. Then we claim that at position z, condition (b) is satisfied. It is sufficient to prove that (u, χ) π-corresponds to (u, φ'). But this follows from the facts that $\varphi' = \tau_{\mathcal{F}}\left(\bigcup_{i=1}^{l} \Phi'_i\right)$ and $\chi \in \Phi'_1$.

- Suppose that $\alpha_1 \wedge \cdots \wedge \alpha_k$ contains p. Then φ is equal to the formula

$$(\alpha_1 \wedge \cdots \wedge \alpha_k) \wedge \bigwedge \{\nabla e^{\perp}[\Phi_i] \mid 1 \leq i \leq k\} \wedge \nabla e^{\perp}[\Phi].$$

In the game $\mathcal{E}^s_0$, at position (t, φ), it is $\forall$'s turn. If he chooses the conjunct α_1, then the game is finite and α_1 has to be true at t, since f^s is a winning strategy for $\exists$. Otherwise, he may also choose the conjunct $\nabla e^{\perp}[\Phi_1]$. Then using f^s, $\exists$ defines a ∇-marking $m_1 : e^{\perp}[\Phi'_1] \longrightarrow \mathcal{P}(R[t])$.

We now go back to the game $\mathcal{E}$. Recall that we have to show that α_1 is true at t and to provide a ∇-marking $m : \Phi_1 \longrightarrow \mathcal{P}(R[r])$. Since we already established that α_1 is true at t, we are left with the definition of m. We define m such that for all $\chi \in \Phi'_1$, we have

$$m(\chi) = \{v \mid v \in m_1(e^{\perp}(\chi))\}.$$

Since m_1 is a ∇-marking, we also have that $m : \Phi_1 \longrightarrow \mathcal{P}(R[t])$ is a ∇-marking.

After $\exists$ chose m in $\mathcal{E}$, it is $\forall$ who has to move in $\mathcal{E}$: he has to pick a position $z = (v, \chi)$ such that $v \in m(\chi)$. We skip the proof that condition (a) is met, as it is similar to the one for the previous case (where p does not occur in $\alpha_1 \wedge \cdots \wedge \alpha_k$).

◀

As in the proof of the opposite direction of 5.13, it is immediate by Claim 2 and the fact that (r, φ_0) π-corresponds to $(r, \tau_a(\varphi_0))$, that we can define a strategy h for $\exists$ in $\mathcal{E}$ such that for all h-conform full $\mathcal{E}$-matches π', we have:

(††) If $\forall$ does not get stuck, then $\pi' = z_0 z_1 \dots$ is infinite and
(i) either there is $i \in \mathbb{N}$ such that $z_i \in Win_\exists(\mathcal{E})$ and $z_i z_{i+1} \dots$ is a g-conform $\mathcal{E}$-match.
(ii) or for all $i \in \mathbb{N}$, there exists $j > i$ such that z_j Π-corresponds to some position.

Now we let h be a strategy for $\exists$ in $\mathcal{E}$ satisfying (††). In order to prove that h is a winning strategy for $\exists$ in $\mathcal{E}^s_0$, suppose that π' is a full h-conform match. If π is finite, then, by (††), π is won by $\exists$. Suppose that $\pi' = z_0 z_1 \dots$ is infinite. If (i) holds, then π is won by $\exists$, as g is a winning strategy for $\exists$.

Next we prove that (ii) is never satisfied. If a position (t, φ) π-corresponds to some position, then t belongs to the path P. So as a corollary, if (ii) holds, then the match π' never leaves the path P. This is impossible as π' is infinite and φ_0 is guarded. □

Proof of Theorem 5.5.3. Fix an arbitrary μ-sentence φ. We may assume φ to be guarded. We define $\varphi^a := \tau_a(\tau^\nabla_m(\varphi))$, where τ^∇_m is the translation of Theorem 5.1.8. Equivalence (5.11) is a consequence that Theorem and Proposition 5.5.5 (together with the observation that the range of the translation τ_a is a subset of $\mu\mathrm{ML}_A(p)$).

The decidability of the finite width property follows by the observation that the construction of the formula φ^a from φ is effective, and that it is decidable whether φ and φ^a are equivalent. □

5.5.7. Remark. Given a monotone guarded μ-sentence φ_0, the size of $\tau_a(\varphi_0)$ is triply exponential in the size of φ_0. As in the other sections, we did not investigate the complexity of the problem of deciding whether a sentence is completely additive, but it is easily seen to be elementary.

Similarly to the other sections, we could also have proved Theorem 5.5.3 using translations at the level of automata. In the particular case of complete additivity, the automata theoretic approach would not only have been useful for complexity issues, but also to simplify the proof. We believe that part of the complexity of the proof given earlier is due to the inductive definition of formulas. It was a bit of a challenge to see how far we could go by staying at the level of formulas. This fits into the perspective (mentioned in the general introduction) of this thesis being a case study of the different methods to approach the μ-calculus.

In that respect, it is interesting to mention that if we drop the requirement "$v \in V(p)$" in equivalence 5.10 (in the definition of complete additivity), then we obtain another semantic property: the single point property, studied in a paper co-authored by Yde Venema [FV10]. For the characterization of that fragment, we were not able to find a proof on the level of formulas and had to involve automata (for more details, see [FV10]).

5.6 Conclusions

We gave syntactic characterizations of four semantic properties. The first three properties (finite depth, finite width and continuity) are related to the continuous fragment, in the sense that the combination of the finite depth and the finite width fragments corresponds to the continuity property. The study of the continuous fragment is mainly motivated by its links with Scott continuity and constructivity.

We also investigated complete additivity, which is an essential property in order to characterize safety for bisimulations. Complete additivity was already characterized by Marco Hollenberg [Hol98b]. We gave here an alternative proof of this characterization.

It is not hard to prove some variations of our results. In particular, one may show that the characterizations of the finite path property and complete additivity still hold if we restrict our attention to the class of finitely branching trees.

Putting our main results together with Theorem 5.1.8 and the complexity of the satisfiability problem for μ-calculus, we easily obtained, for each semantic property, the decidability of the question whether a given formula has that property. It is interesting to show the exact complexity of this question. One difficulty is that the procedures presented in this chapter rely on the transformation of a μ-sentence into an equivalent disjunctive sentence. Hence, a first natural step would be to understand the complexity of that latter transformation.

All the characterizations (and their proofs) presented here can be easily adapted to the setting of modal logic. An interesting question would concern the adaptations of these characterizations to logics lying in between modal logic and μ-calculus, such as PDL or CTL.

Finally, as mentioned in the section about continuity, it is interesting to clarify the link between continuity and constructivity. In particular, we could try to answer the following question: given a constructive formula φ, can we find a continuous formula ψ satisfying $\mu p.\varphi \equiv \mu p.\psi$?

Chapter 6

Expressive power of CoreXPath restricted to the descendant relation

XML is a standard for storage and exchange of data on the internet. Its basic data structure is that of finite sibling-ordered tree: a finite tree in which the children of each node are linearly ordered. Several languages were introduced to describe XML documents and among them, the language XPath, which is particularly convenient for selecting nodes and describing paths. In order to be able to study XPath from a logical point of view, Georg Gottlob, Christoph Koch and Reinhard Pichler isolated an essential navigational core of Xpath [GKP05], called CoreXPath.

The logic CoreXPath is essentially a modal logic and the XML documents are nothing but Kripke models with two basic modalities (one for the child relation and the other one for the relation between the siblings of a node). The main difference between CoreXPath and modal logic is that CoreXPath is a two-sorted language: it contains both nodes expressions (which would be similar to formulas, in the sense that they select points in a tree) and path expressions (which are like PDL programs, as they select paths in a tree).

In this chapter, we exploit the connection between CoreXPath and modal logic. The goal is not so much to prove very technical theorems, but to illustrate how, by combining well-chosen results of modal logic, we can easily obtain results for CoreXPath. Moreover, one of the results of the modal logic area that we use, is an easy consequence of Theorem 5.5.3, established in the last chapter.

The results that we present, concerns the expressive power of CoreXPath. It is easy to prove that CoreXPath is a fragment of first-order logic (using a variant of the standard translation, presented in Chapter 2). However, not all first order formulas (over the appropriate signature) are expressible in CoreXPath. In fact, it was shown by Maarten Marx and Maarten de Rijke [MdR05] that the CoreXPath node expressions capture exactly the two-variable first order formulas with one and two free variables. A characterization in the same fashion for CoreXPath path expression was also obtained.

In this chapter, we focus on CoreXPath($\downarrow^+$); that is the fragment of CoreXPath for which the only axis allowed corresponds to the descendant relation (or to put it in terms of modal logic, the only modality considered is associated with the descendant relation). In determining the expressive power of this language, there are at least two natural yardsticks. One is first-order logic, which is probably the most well known logical language. The second, even more attractive one, is monadic second order logic, which is a very well-behaved language on trees (see Chapter 2). As mentioned earlier, CoreXPath is a fragment of first-order logic. However, if we can characterize a fragments of CoreXPath in terms of MSO, this means that we have a stronger result (in the sense that we can immediately derive a characterization of this fragment in terms of FO).

Our two main results are a characterization of CoreXPath($\downarrow^+$) node expressions and a characterization of CoreXPath($\downarrow^+$) path expressions, both in terms of MSO. Each characterization is expressed in two different ways: using bisimulations and in terms of simple operations on trees. Moreover, we can derive from these results a decision procedure for establishing whether a given MSO formula is equivalent to a CoreXPath($\downarrow^+$) node expression or path expression.

The proofs of both characterizations follow the same scheme. Each of them consists in combining two results concerning the μ-calculus. In the case of the characterization for node expressions, the first result is the Janin-Walukiewicz theorem (which characterizes the μ-calculus as a fragment of MSO) and the second result is a consequence of the de Jongh-fixpoint theorem, mentioned in [Ben06] (which says that on conversely well-founded transitive models, the μ-calculus and modal logic have the same expressive power). In the case of the characterization for path expressions, we use adaptations of these two results for μ-programs. The adaptation of the Janin-Walukiewicz theorem for μ-programs is a direct consequence of Theorem 5.5.3.

For CoreXPath($\downarrow^+$) node expressions, similar characterizations have already been proved by Mikołaj Bojańczyk and Igor Walukiewicz [BW06, BW07], using the framework of forest algebras (and without reference to modal logic). The logic EF in their work corresponds to the node expressions of CoreXPath($\downarrow^+$). Mikołaj Bojańczyk and Igor Walukiewicz [BW06] also established a similar characterization for the fragment of CoreXPath where the only axis, or modalities, allowed corresponds to the child relation and the descendant relation. Let us finally mention that Mikołaj Bojańczyk [Boj07] found a characterization of the fragment of CoreXPath using the axis (modalities) associated to the ancestor and the descendant relations.

Other alternative proofs for the characterization of CoreXPath($\downarrow^+$) node expressions have also appeared, see e.g. [ÉI08], [Wu07] and [DO09]. The proof presented here has been found independently and uses different methods (combining well-known results from modal logic and μ-calculus). The advantage of this new proof is that it can easily be extended to a similar characterization for path expressions.

In the first section, we introduce CoreXPath and emphasize its link with modal logic. The second and the third sections contain respectively the characterizations of the node and path expressions of CoreXPath($\downarrow^+$). We gratefully acknowledge a contribution of Mikołaj Bojańczyk in a discussion about Theorem 6.3.1.

6.1 Preliminaries

6.1.1 XML trees

The language XPath is based on a tree representation of the XML documents. Formally, given an infinite set *Prop* of proposition letters, we define an *XML tree* as a structure $\mathcal{T} = (W, R, R_\rightarrow, V)$, where

- (W, R) is a finite tree (with W the set of nodes and R the child relation),
- $R_\rightarrow$ is the successor relation of some linear ordering between siblings in the tree,
- $V : Prop \rightarrow \mathcal{P}(W)$ labels the nodes with elements of *Prop*.

So an XML tree is nothing but a Kripke model for a modal language with two modalities: one corresponding to the child relation and the other one to the next sibling relation. In this particular setting, the elements of *Prop* correspond to XML tags, such as, in the case of HTML, `body`, `p`, `it`, ... It is customary to require that each node satisfies precisely one tag. In order to simplify the presentation, it will be convenient for us not to make this requirement from the start. However, all results can be adapted to the setting with unique node labels.

6.1.2 CoreXPath, the navigational core of XPath 1.0

There are two main types of expressions in CoreXPath: path expressions and node expressions. Path expressions describe ways of traveling through the tree and they are interpreted as binary relations, while node expressions are used to describe properties of nodes and are interpreted as subsets. More precisely, the syntax of CoreXPath is defined as follows:

$$\begin{aligned} \mathsf{Step} &:= \downarrow \mid \leftarrow \mid \uparrow \mid \rightarrow, \\ \mathsf{Axis} &:= \mathsf{Step} \mid \mathsf{Step}^+, \\ \mathsf{PathEx} &:= . \mid \mathsf{Axis} \mid \mathsf{PathEx}\,[\mathsf{NodeEx}] \mid \mathsf{PathEx}/\mathsf{PathEx} \mid \mathsf{PathEx} \cup \mathsf{PathEx}, \\ \mathsf{NodeEx} &:= p \mid \langle \mathsf{PathEx} \rangle \mid \neg \mathsf{NodeEx} \mid \mathsf{NodeEx} \vee \mathsf{NodeEx}, \end{aligned}$$

where p belongs to *Prop*.

The axes correspond to basic moves one can make in the tree. The axe "." corresponds to staying at the current node. The axes $\downarrow$, $\leftarrow$, $\uparrow$ and $\rightarrow$ correspond

Table 6.1: Comparison with Official XPath Notation [W3C]

our notation	official notation	our notation	official notation
$\downarrow$	`children` :: $*$	$\uparrow$	`parent` :: $*$
$\leftarrow$	`preceding-sibling` :: $*[1]$	$\rightarrow$	`following-sibling` :: $*[1]$
$\downarrow^+$	`descendant` :: $*$	$\uparrow^+$	`ancestor` :: $*$
$\leftarrow^+$	`preceding-sibling` :: $*$	$\rightarrow^+$	`following-sibling` :: $*$
p	`self` :: p	$\langle \mathsf{PathEx} \rangle$	PathEx
$\neg$NodeEx	*not*(NodeEx)	NodeEx $\vee$ NodeEx	NodeEx `or` NodeEx

respectively to the child relation, the next sibling relation, the parent relation and the previous sibling relation. Moreover, given one of these four axes A, the axe A^+ corresponds to the transitive closure of the relation associated to A. The axes can be composed into path expressions by using *composition* (;), *union* ($\cup$), and *node tests* ($\cdot[\cdot]$). The node expression $\langle \mathsf{PathEx} \rangle$ expresses that the current node belongs to the domain of the binary relation defined by PathEx.

The reader familiar with original XPath notation will notice that we included a number of abbreviations and alterations. Table 6.1 provides a comparison of our notation with that of [W3C].

The semantics of CoreXPath expressions is given by two functions, $[\![\cdot]\!]^{\mathsf{PExpr}}$ and $[\![\cdot]\!]^{\mathsf{NExpr}}$. For any path expression A and XML tree $\mathcal{T}$, $[\![A]\!]^{\mathsf{PExpr}}_{\mathcal{T}}$ denotes a binary relation over the domain of $\mathcal{T}$, and for any node expression φ and XML tree $\mathcal{T}$, $[\![\varphi]\!]^{\mathsf{NExpr}}_{\mathcal{T}}$ denotes a subset of the domain of $\mathcal{T}$. Given an XML tree $\mathcal{T} = (T, R, R_{\rightarrow}, V)$, the binary relation $[\![\cdot]\!]^{\mathsf{PExpr}}_{\mathcal{T}}$ and the subset $[\![\cdot]\!]^{\mathsf{NExpr}}_{\mathcal{T}}$ are defined by induction in the following way:

$$\begin{aligned}
[\![.]\!]^{\mathsf{PExpr}} &= \{(u,u) \mid u \in T\}, \\
[\![\mathsf{a}]\!]^{\mathsf{PExpr}} &= R_{\mathsf{a}} \text{ for all } \mathsf{a} \in \mathsf{Step}, \\
[\![\mathsf{a}^+]\!]^{\mathsf{PExpr}} &= (R_{\mathsf{a}})^+ \text{ for all } \mathsf{a} \in \mathsf{Step}, \\
[\![A/B]\!]^{\mathsf{PExpr}} &= \{(u,v) \mid \exists w \text{ such that } (u,w) \in [\![A]\!]^{\mathsf{PExpr}} \text{ and } (w,u) \in [\![B]\!]^{\mathsf{PExpr}}\}, \\
[\![A \cup B]\!]^{\mathsf{PExpr}} &= [\![A]\!]^{\mathsf{PExpr}} \cup [\![B]\!]^{\mathsf{PExpr}}, \\
[\![A\,[\varphi]]\!]^{\mathsf{PExpr}} &= \{(u,v) \mid (u,v) \in [\![A]\!]^{\mathsf{PExpr}} \text{ and } v \in [\![\varphi]\!]^{\mathsf{NExpr}}\},
\end{aligned}$$

$$\begin{aligned}
[\![p]\!]^{\mathsf{NExpr}} &= \{u \mid u \in V(p)\}, \\
[\![\langle \mathsf{PathEx} \rangle]\!]^{\mathsf{NExpr}} &= \{u \mid \exists v \text{ such that } (u,v) \in [\![\mathsf{PathEx}]\!]^{\mathsf{PExpr}}\}, \\
[\![\neg\varphi]\!]^{\mathsf{NExpr}} &= \{u \mid u \notin [\![\varphi]\!]^{\mathsf{NExpr}}\}, \\
[\![\varphi \vee \psi]\!]^{\mathsf{NExpr}} &= [\![\varphi]\!]^{\mathsf{NExpr}} \cup [\![\psi]\!]^{\mathsf{NExpr}},
\end{aligned}$$

where $R_{\downarrow}$ is the relation R, $R_{\leftarrow}$ is the converse of the relation $R_{\rightarrow}$, $R_{\uparrow}$ is the converse of the relation R and given a binary relation R_{a}, $(R_{\mathsf{a}})^+$ is the transitive closure of R_{a}. For readability, the superscript $\mathcal{T}$ is left out.

For $\mathsf{A} \subseteq \mathsf{Axis}$, we will denote by CoreXPath(A) the fragment of CoreXPath in which the only allowed axes are those listed in A.

6.1.3 Connections with modal logic

The are two main differences between modal logic and CoreXPath. First, the semantics for CoreXPath is more restrictive (finite sibling-ordered trees as opposed to arbitrary Kripke structure). Next, the syntax for CoreXPath is two-sorted (node and path expressions, the interpretations of which are respectively subsets and binary relations), whereas in modal logic, only formulas (which are interpreted as subsets) are considered. However, we can easily obtain a two-sorted syntax for modal logic, by introducing modal programs. The definition of modal program is a simplified version of the notion of program for PDL.

Modal programs Given a set A of actions, we define the set of *modal programs* over the set A of actions by induction in the following way:

$$\theta ::= \; R_a \mid \varphi? \mid \theta;\theta \mid \theta \cup \theta,$$

where $a \in \mathsf{A}$ and φ is a modal formula over the set A of actions.

Given a Kripke model $\mathcal{M}$, the interpretation of a modal program θ is a binary relation $[\![\theta]\!]_{\mathcal{M}}$ over the domain of the model. This interpretation is defined by induction on the complexity of the program. We do not give more details, as this definition is a particular case of the semantics for PDL (see Chapter 2) and the semantics for the μ-programs (see Chapter 2). We only recall that the interpretation of $\varphi?$ is the relation $\{(u, u) \mid \varphi \text{ is true at } u\}$.

We would like to observe that the syntax for modal programs is the same as the one for PDL, except that we do not use the Kleene star and that we can only test with modal formulas (instead of PDL formulas). Moreover, in PDL, we also allow formulas of the form $\langle\theta\rangle\varphi$ (where θ is a program and φ a formula) and it is not the case for modal formulas. However, it is possible to show that for all modal programs φ and for all modal programs, $\langle\theta\rangle\varphi$ is equivalent to a modal formula (in fact, this can be proved easily using the translation τ_2 from the proof of the next proposition).

Equivalence between CoreXPath **and modal logic** Let A be a subset of Axis. Given a CoreXPath(A) node expression φ and a modal formula over A, we say that φ and ψ are *equivalent (on finite trees)* if for all XML trees $\mathcal{T} = (T, R, R_{\rightarrow})$, we have $[\![\varphi]\!]^{\mathsf{NExpr}}_{\mathcal{T}} = [\![\psi]\!]_{\mathcal{T}}$.

Similarly, given a CoreXPath(A) path expression A and a modal program θ over A, A and θ are *equivalent (on finite trees)* if for all XML trees $\mathcal{T} = (T, R, R_{\rightarrow})$, we have $[\![A]\!]^{\mathsf{PExpr}}_{\mathcal{T}} = [\![\theta]\!]_{\mathcal{T}}$.

6.1.1. PROPOSITION. *Let* A *be a subset of* Axis*. There is an effective truth-preserving translation from the set of* CoreXPath(A) *node and path expressions to the set of modal formulas and programs over* A*, and vice-versa.*

Proof Let A be a subset of Axis. Both translations are very similar and not difficult to define; so we only give details for the translation which maps a CoreXPath(A) node expression to an equivalent modal formula (over *Prop* and A) and maps a CoreXPath(A) path expression to an equivalent modal program (over *Prop* and A). This translation τ is defined as the composition of two translations τ_1 and τ_2, which we define below.

The translation τ_1 is defined by induction on the complexity of the path and node expressions as follows:

$$
\begin{array}{rcl|rcl}
\tau_1(\mathsf{a}) & = & R_{\mathsf{a}}, & \tau_1(.) & = & \top?, \\
\tau_1(A[\varphi]) & = & \tau_1(A);\tau_1(\varphi)?, & \tau_1(A;B) & = & \tau_1(A);\tau_1(B), \\
\tau_1(A\cup B) & = & \tau_1(A)\cup\tau_1(B), & \tau_1(p) & = & p, \\
\tau_1(\langle A\rangle) & = & \langle\tau_1(A)\rangle\top, & \tau_1(\neg\varphi) & = & \neg\tau_1(\varphi), \\
\tau_1(\varphi\vee\psi) & = & \tau_1(\varphi)\vee\tau_1(\psi), & & &
\end{array}
$$

where a belongs to A, A and B are CoreXPath(A) path expressions, φ and ψ are CoreXPath(A) node expressions and p is a proposition letter. Note that τ_1 does not necessarily map a node expression to a modal formula: formulas of the form $\langle\theta\rangle\varphi$ might occur in the range of τ_1. Such formulas are not modal formulas, as defined in Chapter 2.

To fix this problem, we introduce a translation τ_2 which is defined by induction on the complexity of the formulas and programs in the range of τ_1:

$$
\begin{array}{rcl|rcl}
\tau_2(R_a) & = & R_a, & \tau_2(\varphi\vee\psi) & = & \tau_2(\varphi)\vee\tau_2(\psi), \\
\tau_2(\varphi?) & = & \tau_2(\varphi)?, & \tau_2(\varphi\wedge\psi) & = & \tau_2(\varphi)\wedge\tau_2(\psi), \\
\tau_2(\theta;\lambda) & = & \tau_2(\theta);\tau_2(\lambda), & \tau_2(\langle R_a\rangle\varphi) & = & \Diamond_a\tau_2(\varphi), \\
\tau_2(\theta\cup\lambda) & = & \tau_2(\theta)\cup\tau_2(\lambda), & \tau_2(\langle\varphi?\rangle\psi) & = & \tau_2(\varphi)\wedge\tau_2(\psi), \\
\tau_2(p) & = & p & \tau_2(\langle\theta;\lambda\rangle\varphi) & = & \tau_2(\langle\theta\rangle\tau_2(\langle\lambda\rangle\varphi)), \\
\tau_2(\neg p) & = & \neg p & \tau_2(\langle\theta\cup\lambda\rangle\varphi) & = & \tau_2(\langle\theta\rangle\varphi)\vee\tau_2(\langle\lambda\rangle\varphi),
\end{array}
$$

where a belongs to A, p is a proposition letter, φ and ψ are PDL formulas in the range of τ_1, θ and λ are PDL programs in the range of τ_1.

Finally we define τ as $\tau_2\circ\tau_1$. It is easy to check that τ has the required properties. □

As mentioned in the introduction, throughout the chapter, this connection will help us to transfer well-known results of the modal logic area into the framework of CoreXPath.

6.2 CoreXPath($\downarrow^+$) node expressions

We start by characterizing the CoreXPath($\downarrow^+$) node expressions as a fragment of monadic second order logic. Two characteristic features of CoreXPath($\downarrow^+$) are that (i) whether a node expression holds at a node depends only on the subtree below it, and (ii) CoreXPath($\downarrow^+$) expressions cannot see the difference between children and descendants. It turns out that, in some sense, these two properties characterize CoreXPath($\downarrow^+$) as a fragment of monadic second-order logic. We formalize these two features in two ways: using transitive bisimulations and in terms of simple operations on trees.

Before we state the characterization, we fix some notation and introduce some terminology.

Convention As the only axe that we consider is $\downarrow^+$, we can forget about the sibling order. More precisely, instead of evaluating node and path expressions on XML trees, we interpret these expressions on finite tree models (as defined in Section 2.6). Moreover, in this chapter, we never consider frames. So there is no confusion if we use the word "tree" instead of "tree model" and we will do so throughout this chapter.

Finally, recall that when we talk about MSO formulas on models, we always have the same signature in mind, which consists of a binary relation and a unary predicate for each proposition letter (for more details, see Section 2.6). In particular, the binary relation corresponds to the child relation, when we interpret an MSO formula on a tree.

Equivalence An MSO formula $\varphi(x)$ is equivalent to a CoreXPath($\downarrow^+$) node expression ψ if for all finite trees $\mathcal{T}$ and all nodes $u \in \mathcal{T}$, we have $\mathcal{T}, u \models \varphi(x)$ iff u belongs to $[\![\psi]\!]_{\mathcal{T}}$. When this happens, we write $\varphi(x) \equiv \psi$.

Transitive bisimulation Let $\mathcal{M} = (W, R, V)$ and $\mathcal{M}' = (W', R', V')$ be two Kripke models. A relation $B \subseteq W \times W'$ is a *transitive bisimulation* if for all (w, w') in B, we have

- the same proposition letters hold at w and w',
- if wR^+v, there exists $v' \in W'$ such that $w'(R')^+v'$ and $(v, v') \in B$,
- if $w'(R')^+v'$, there exists $v \in W$ such that wR^+v and $(v, v') \in B$.

A transitive bisimulation B between two models $\mathcal{M} = (W, R, V)$ and $\mathcal{M}' = (W', R', V')$ is *total* if the domain of B is W and the range of B is W'.

A MSO formula $\varphi(x)$ with one free first order variable is said to be *invariant under (transitive) bisimulation* on a class $\mathcal{C}$ of models if for all models $\mathcal{M}, \mathcal{M}'$ in

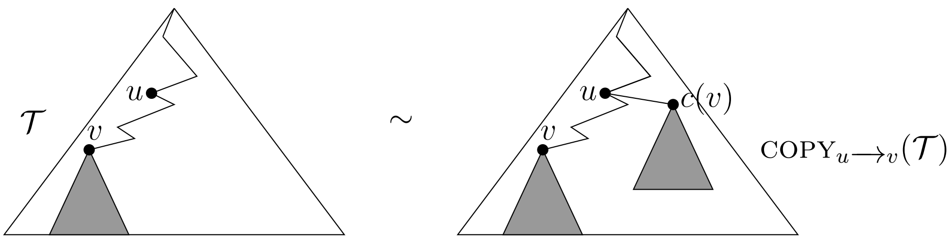


Figure 6.1: The trees $\mathcal{T}$ and $\textsc{copy}_{u \longrightarrow v}(\mathcal{T})$.

$\mathcal{C}$, all (transitive) bisimulations $B \subseteq \mathcal{M} \times \mathcal{M}'$, and pairs $(w, w') \in B$, we have $\mathcal{M}, w \vDash \varphi(x)$ iff $\mathcal{M}', w' \vDash \varphi(x)$.

Intuitively, a transitive bisimulation is nothing but a regular bisimulation, except that instead of considering the successor relation R, we focus on the transitive closure of R.

The operation copy Let $\mathcal{T} = (T, R, V)$ be a finite tree and let u, v be nodes such that v is a descendant of u. Recall that $\mathcal{T}_v$ is the submodel of $\mathcal{T}$ generated by v. We write $\textsc{copy}(\mathcal{T}_v)$ for a tree that is an isomorphic copy of $\mathcal{T}_v$. In order to make a distinction between a point w in $\mathcal{T}_v$ and the copy of w in $\textsc{copy}(\mathcal{T}_v)$, we denote by $c(w)$ the copy of w. We define $\textsc{copy}_{u \longrightarrow v}(\mathcal{T})$ as the tree that is obtained by adding the isomorphic copy $\textsc{copy}(\mathcal{T}_v)$ to the tree $\mathcal{T}$, plus an edge from u to the copy $c(v)$ of v, see Figure 6.1.

This definition allows us to make precise what it means not to distinguish children from descendants: it means that $\mathcal{T}$ and $\textsc{copy}_{u \longrightarrow v}(\mathcal{T})$ are indistinguishable.

Invariance under the subtree and the copy operations Let $\varphi(x)$ be an MSO formula. We say that $\varphi(x)$ is *invariant under the subtree operation (on finite trees)* if for all finite trees $\mathcal{T}$ and nodes u,

$$\mathcal{T}, u \vDash \varphi(x) \quad \text{iff} \quad \mathcal{T}_u, u \vDash \varphi(x).$$

The formula $\varphi(x)$ is *invariant under the copy operation (on finite trees)* if for all finite trees $\mathcal{T}$ with root r, and with nodes u, v such that v is a descendant of u,

$$\mathcal{T}, r \vDash \varphi(x) \quad \text{iff} \quad \textsc{copy}_{u \longrightarrow v}(\mathcal{T}), r \vDash \varphi(x).$$

We can now state the characterization of CoreXPath($\downarrow^+$) precisely.

6.2.1. THEOREM. *Let $\varphi(x)$ be an* MSO *formula. The following are equivalent:*

(i) $\varphi(x)$ is equivalent to a CoreXPath($\downarrow^+$) *node expression,*

(ii) $\varphi(x)$ *is invariant under transitive bisimulation on finite trees,*

(iii) $\varphi(x)$ *is invariant under the subtree and the copy operations.*

Moreover, for all MSO *formulas* $\varphi(x)$*, we can compute a* CoreXPath($\downarrow^+$) *node expression* ψ *such that* $\varphi(x) \equiv \psi$ *iff* $\varphi(x)$ *is equivalent to a* CoreXPath($\downarrow^+$) *node expression.*

The proof will be based on two known expressivity results. The first theorem we use is the bisimulation characterization of the modal μ-calculus, due to David Janin and Igor Walukiewicz (see Section 2.6 and [JW96]). This characterization works on arbitrary Kripke models, but also on the restricted class of finite trees, which is important for us.

Moreover, in the case of finite trees, the characterization is effective: it is decidable whether an MSO formula is invariant under bisimulation, and for bisimulation invariant MSO formulas an equivalent formula of the modal μ-calculus can be effectively computed. Recall that an MSO formula $\varphi(x)$ is equivalent on finite trees, to a μ-sentence ψ if for all finite trees $\mathcal{T}$ and nodes $u \in \mathcal{T}$, $\mathcal{T}, u \vDash \varphi(x)$ iff $\mathcal{T}, u \Vdash \psi$.

6.2.2. Theorem (from [JW96]). *An* MSO *formula* $\varphi(x)$ *is equivalent on finite trees to a* μ*-sentence iff* $\varphi(x)$ *is invariant under bisimulation on finite trees.*

Moreover, for all MSO *formulas* $\varphi(x)$*, we can compute a* μ*-sentence* ψ *such that* $\varphi(x)$ *and* ψ *are equivalent on finite trees iff* $\varphi(x)$ *is equivalent on finite trees to a* μ*-sentence.*

Proof Let $\varphi(x)$ be an MSO formula. It is immediate that if $\varphi(x)$ is equivalent on finite trees to a μ-sentence, then $\varphi(x)$ is invariant under bisimulation on finite trees. For the other direction of the implication, suppose that $\varphi(x)$ is invariant under bisimulation on finite trees.

It follows from the proof of the main result of [JW96] (Theorem 11) that we can compute a μ-sentence ψ such that for all pointed models $(\mathcal{M}, w)$, we have

$$\mathcal{M}, w \Vdash \psi \quad \text{iff} \quad \mathcal{M}^{\omega}_{w}, w \vDash \varphi(x).$$

Recall that $\mathcal{M}^{\omega}_{w}$ is the ω-expansion of the pointed model $(\mathcal{M}, w)$ (see Section 2.6).

A careful inspection of the proof shows that we can even get a stronger result: we can compute $n \in \mathbb{N}$ and μ-sentence ψ such that for all pointed models $(\mathcal{M}, w)$, we have

$$\mathcal{M}, w \Vdash \psi \quad \text{iff} \quad (\mathcal{M}, w)^n, w \vDash \varphi(x).$$

Recall that $(\mathcal{M}, w)^n$ is the n-expansion of the pointed model $(\mathcal{M}, w)$ (see Section 2.6). In particular, for all finite trees $\mathcal{T}$ and for all $u \in \mathcal{T}$, we have

$$\mathcal{T}, u \Vdash \psi \quad \text{iff} \quad (\mathcal{T}, u)^n, u \vDash \varphi(x). \tag{6.1}$$

Now recall that there is a bisimulation B between $(\mathcal{T}, u)^n$ and $\mathcal{T}$ such that $(u, u) \in B$ (see Section 2.6). Since $\varphi(x)$ is invariant under bisimulation on finite trees, this implies that

$$(\mathcal{T}, u)^n, u \vDash \varphi(x) \quad \text{iff} \quad \mathcal{T}, u \vDash \varphi(x).$$

Putting this together with (6.1), we obtain that

$$\mathcal{T}, u \Vdash \psi \quad \text{iff} \quad \mathcal{T}, u \vDash \varphi(x).$$

Therefore, $\varphi(x)$ is equivalent on finite trees to a μ-sentence.

Moreover, it also easily follows from our proof that given an MSO formula $\varphi(x)$, we can compute a μ-sentence ψ such that $\varphi(x)$ and ψ are equivalent on finite trees iff $\varphi(x)$ is equivalent on finite trees to a μ-sentence. □

The second result we use is a consequence of the de Jongh-fixpoint theorem which was proved independently by Dick de Jongh and Giovanni Sambin (see [Smo85]). More specifically, we use the fact that the μ-calculus over models for Gödel-Löb logic (or equivalently, evaluated over transitive Kripke models, that do not contain any infinite path) collapses to its modal fragment, as was first observed by Johan van Benthem in [Ben06].

6.2.3. Theorem ([Ben06]). *For all μ-sentences φ, we can compute a modal formula ψ satisfying the following: For all Kripke models $\mathcal{M} = (W, R, V)$ such that R is transitive and $\mathcal{M}$ does not contain any infinite path, for all $w \in W$, we have $\mathcal{M}, w \Vdash \varphi$ iff $\mathcal{M}, w \Vdash \psi$.*

In particular, for all finite transitive trees $\mathcal{T}^+$ and all nodes $u \in \mathcal{T}^+$, we have $\mathcal{T}^+, u \Vdash \varphi$ iff $\mathcal{T}^+, u \Vdash \psi$.

We are now ready to prove that CoreXPath($\downarrow^+$) is the transitive bisimulation invariant fragment of MSO, by putting together Theorem 6.2.2 and Theorem 6.2.3.

6.2.4. Proposition. *An* MSO *formula $\varphi(x)$ is equivalent to a* CoreXPath($\downarrow^+$) *node expression iff $\varphi(x)$ is invariant under transitive bisimulation on finite trees.*

Moreover, for all MSO *formulas $\varphi(x)$, we can compute a* CoreXPath($\downarrow^+$) *node expression ψ such that $\varphi(x) \equiv \psi$ iff $\varphi(x)$ is equivalent to a* CoreXPath($\downarrow^+$) *node expression.*

Proof First we show that an MSO formula $\varphi(x)$ is equivalent to a CoreXPath($\downarrow^+$) node expression iff $\varphi(x)$ is invariant under transitive bisimulation on finite trees. We restrict ourselves to prove the difficult direction (the other one is a standard induction on the complexity of CoreXPath($\downarrow^+$) node expressions). Let $\varphi(x)$ be a MSO formula that is invariant under transitive bisimulation on finite trees. We need to find a node expression χ of CoreXPath($\downarrow^+$) such that for all finite trees $\mathcal{T}$ and all nodes u, χ holds at u iff $\varphi(u)$ is true.

By Theorem 6.2.2, we can compute a μ-sentence ψ such that $\varphi(x)$ and ψ are equivalent on finite trees iff $\varphi(x)$ is invariant under bisimulation. Since $\varphi(x)$ is invariant under transitive bisimulation on finite trees, in particular $\varphi(x)$ is invariant under ordinary bisimulation on finite trees. Hence, $\varphi(x)$ and ψ are equivalent on finite trees.

Now we show that for all finite trees $\mathcal{T} = (T, R, V)$ and all nodes u in $\mathcal{T}$, we have

$$\mathcal{T}, u \Vdash \psi \quad \text{iff} \quad \mathcal{T}^+, u \Vdash \psi, \tag{6.2}$$

where $\mathcal{T}^+ = (T, R^+, V)$. Take a finite tree $\mathcal{T} = (T, R, V)$ and a node u in $\mathcal{T}$. Let $\mathcal{T}^+$ be the model (T, R^+, V) and let $\mathcal{S}$ be the unraveling of the pointed model $(\mathcal{T}^+, u)$ (see Section 2.6).

The canonical bisimulation between $\mathcal{S}$ and $\mathcal{T}^+$ links u in $\mathcal{S}$ with u in $\mathcal{T}^+$. It follows that $\mathcal{T}^+, u \Vdash \psi$ iff $\mathcal{S}, u \Vdash \psi$. Moreover, the canonical bisimulation between $\mathcal{S}$ and $\mathcal{T}^+$ constitutes a transitive bisimulation between $\mathcal{S}$ and $\mathcal{T}$, which links the node u in $\mathcal{S}$ to the node u in $\mathcal{T}$. Since ψ is equivalent to $\varphi(x)$ on finite trees and $\varphi(x)$ is invariant under transitive bisimulation on finite trees, we have that

$$\mathcal{S}, u \Vdash \psi \quad \text{iff} \quad \mathcal{T}, u \Vdash \psi.$$

This finishes the proof of (6.2).

Next it follows from Theorem 6.2.3 that we can compute a modal formula χ such that for all finite transitive trees $\mathcal{T}^+$ and for all nodes u in $\mathcal{T}^+$,

$$\mathcal{T}^+, u \Vdash \psi \quad \text{iff} \quad \mathcal{T}^+, u \Vdash \chi.$$

Given the connection between CoreXPath($\downarrow^+$) and modal logic (see Section 6.1.3), we can compute a node expression ξ of CoreXPath($\downarrow^+$) such that for all finite trees $\mathcal{T} = (T, R, V)$ and all nodes u in $\mathcal{T}$, we have

$$\mathcal{T}^+, u \Vdash \chi \quad \text{iff} \quad u \text{ belongs to } [\![\xi]\!]_{\mathcal{T}},$$

where $\mathcal{T}^+ = (T, R^+, V)$. Putting everything together, we obtain that for all finite trees $\mathcal{T}$ and all nodes u in $\mathcal{T}$, $\mathcal{T}, u \vDash \varphi(x)$ iff u belongs to $[\![\xi]\!]_{\mathcal{T}}$. This finishes the proof that an MSO formula $\varphi(x)$ is equivalent to a CoreXPath($\downarrow^+$) node expression iff $\varphi(x)$ is invariant under transitive bisimulation on finite trees.

Now it is easy to see that the fact that ξ is computable from $\varphi(x)$ does not depend on the fact that $\varphi(x)$ was invariant under transitive bisimulation on finite trees. The second statement of the proposition immediately follows.

□

To prove Theorem 6.2.1, it remains to show that (*ii*) and (*iii*) are equivalent, by exploiting the tight link between transitive bisimulations and the operation $\text{COPY}_{u \longrightarrow v}(\mathcal{T})$.

The hardest direction is to show that (*iii*) implies (*ii*). Given a tree $\mathcal{T}$ and its copy $\text{COPY}_{u \longrightarrow v}(\mathcal{T})$, there is an obvious transitive bisimulation linking the two models. We call such a transitive bisimulation, a $\sim$-transitive bisimulation. Now the idea is to show that each transitive bisimulation B, can be represented as the composition of $\sim$-transitive bisimulations. It will then be easy that we can derive (*ii*) from (*iii*). In order to make these intuitions more precise, we introduce the following terminology.

The relation $\sim$ and its associated bisimulation Let $\mathcal{T} = (T, R, V)$ and $\mathcal{S} = (S, Q, U)$ be finite trees. We write $\mathcal{T} \Rightarrow \mathcal{S}$ if there are nodes u and v of $\mathcal{T}$ such that v is a descendant of u and $\mathcal{S}$ is isomorphic to $\text{COPY}_{u \longrightarrow v}(\mathcal{T})$. We use the notation $\mathcal{T} \sim \mathcal{S}$if $\mathcal{T} \Rightarrow \mathcal{S}$ or $\mathcal{S} \Rightarrow \mathcal{T}$.

We say that a relation $B \subseteq \mathcal{T} \times \mathcal{S}$ is a *$\sim$-transitive bisimulation* for $\mathcal{T}$ and $\mathcal{S}$ if one of the two following conditions holds. Either there exist u and v in $\mathcal{T}$ such that $\mathcal{S}$ is isomorphic to $\text{COPY}_{u \longrightarrow v}(\mathcal{T})$ and B is the relation

$$\{(w, w) \mid w \in \mathcal{T}\} \cup \{(w, c(w)) \mid (v, w) \in R^+\}.$$

Or there exist u and v in $\mathcal{S}$ such that $\mathcal{T}$ is isomorphic to $\text{COPY}_{u \longrightarrow v}(\mathcal{S})$ and B is the relation

$$\{(w, w) \mid w \in \mathcal{S}\} \cup \{(c(w), w) \mid (v, w) \in Q^+\}.$$

If $\mathcal{T}_1, \ldots, \mathcal{T}_n$ is a sequence of finite trees such that $\mathcal{T}_i \sim \mathcal{T}_{i+1}$, for all $i \in \{1, \ldots, n-1\}$, we say that $\mathcal{T}_1, \ldots, \mathcal{T}_n$ is a *$\sim$-sequence* between $\mathcal{T}_1$ and $\mathcal{T}_n$. A relation $B \subseteq \mathcal{T}_1 \times \mathcal{T}_n$ is a *$\sim$-transitive bisimulation* for $\mathcal{T}_1, \ldots, \mathcal{T}_n$ if either $n = 1$ and B is the identity or for all $i \in \{1, \ldots, n-1\}$, there exists a relation B_i which is a $\sim$-bisimulation for $\mathcal{T}_i$ and $\mathcal{T}_{i+1}$ and $B = B_1 \circ \cdots \circ B_{n-1}$. A relation B between two trees $\mathcal{T}$ and $\mathcal{S}$ is a *$\sim$-transitive bisimulation* if there is a $\sim$-sequence $\mathcal{T}_1, \ldots, \mathcal{T}_n$ between $\mathcal{T}$ and $\mathcal{S}$ such that B is a $\sim$-transitive bisimulation for $\mathcal{T}_1, \ldots, \mathcal{T}_n$.

6.2.5. Lemma. *Let B be a total transitive bisimulation between two finite trees $\mathcal{T}$ and $\mathcal{S}$. Then there exists a $\sim$-transitive bisimulation between $\mathcal{T}$ and $\mathcal{S}$ that is included in B.*

Proof The proof is by induction on the depth of $\mathcal{T}$. If the depth of $\mathcal{T}$ is 1, then $\mathcal{T}$ and $\mathcal{S}$ are isomorphic and the lemma trivially holds.

For the induction step, suppose that $\mathcal{T}$ has depth $n+1$. Let r_0 be the root of $\mathcal{T}$ and s_0 the root of $\mathcal{S}$. Let also $u_0, \ldots, u_k$ be the children of r_0 and $v_0, \ldots, v_m$ the children of s_0. Note that since $\mathcal{T}$ and $\mathcal{S}$ are linked by a total bisimulation, the labels of r_0 and s_0 are the same and the depth of $\mathcal{S}$ is $n+1$.

First, we define $\mathcal{Q}$ as the tree obtained by taking the disjoint union of the trees $\{\mathcal{T}_u \mid u \text{ child of } r_0\}$ and $\{\mathcal{S}_v \mid v \text{ child of } s_0\}$ and by adding a root r to it, the label of which is the label of r_0. We show that there exist a $\sim$-sequence between $\mathcal{T}$ and $\mathcal{Q}$ and a $\sim$-transitive bisimulation for this sequence that is a subset of $\{(r_0, r)\} \cup \{(u, u) \mid u \in \mathcal{T}\} \cup B$.

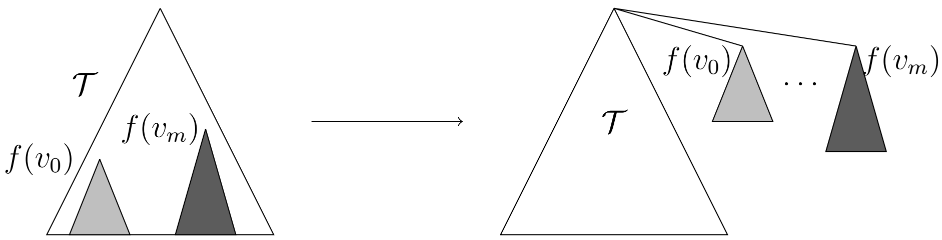


Figure 6.2: From $\mathcal{T}$ to $\mathcal{T}'$.

For each child v of s_0, there exists a node $f(v)$ in $\mathcal{T}$ such that $f(v)$ and v are linked by B. It follows that $B \cap (\mathcal{T}_{f(v)} \times \mathcal{S}_v)$ is a total transitive bisimulation between $\mathcal{T}_{f(v)}$ and $\mathcal{S}_v$. By induction hypothesis, there exist a $\sim$-sequence of finite trees between $\mathcal{T}_{f(v)}$ and $\mathcal{S}_v$ and a $\sim$-transitive bisimulation B^v associated with this sequence and included in B.

Next, let $\mathcal{T}'$ be the tree obtained by taking the disjoint union of $\{\mathcal{T}_u \mid u \text{ child of } r_0\}$ and copies of the finite trees $\{\mathcal{T}_{f(v)} \mid v \text{ child of } s_0\}$ and by adding a root to it, the label of which is the label of r_0. If w belongs to a tree $\mathcal{T}_{f(v)}$ (where v is a child of v_0), we denote by $c_v(w)$ the copy of w that belongs to the copy of $\mathcal{T}_{f(v)}$ in $\mathcal{T}'$. By definitions of $\sim$ and $\mathcal{T}'$, it is easy to see that there exists a $\sim$-sequence $\mathcal{T}_0, \ldots, \mathcal{T}_m$ between $\mathcal{T}$ and $\mathcal{T}'$. Moreover, the relation B given by:

$$\begin{aligned} B = \{(w, w) \mid w \in \mathcal{T}\} \\ \cup \{(w, c_v(w)) \mid v \text{ child of } s_0, w \text{ descendant of } f(v) \text{ or } w = f(v)\}, \end{aligned}$$

is a $\sim$-transitive bisimulation for this sequence.

Now we define a $\sim$-sequence of trees $\mathcal{T}_m, \ldots, \mathcal{T}_{2m}$ and $\sim$-transitive bisimulations $B_m, \ldots, B_{2m}$ such that $\mathcal{T}_{2m} = \mathcal{Q}$ and for all $m + 1 \leq i \leq 2m$, B_i is a $\sim$-transitive bisimulation for a $\sim$-sequence between $\mathcal{T}_{i-1}$ and $\mathcal{T}_i$ and $B_m \circ \cdots \circ B_{m+i}$ is a subset of

$$\begin{aligned} \{(w, w) \mid w \in \mathcal{T}\} \cup \{(c_{v_j}(w), c_{v_j}(w)) \mid j > i, w \in \mathcal{T}_{f(v_j)}\} \\ \cup \{(c_{v_j}(w), t) \mid j \leq i, (w, t) \in B^{v_j}\}. \end{aligned}$$

The definitions of $\mathcal{T}_m, \ldots, \mathcal{T}_{2m}$ and $B_m, \ldots, B_{2m}$ are by induction. The tree $\mathcal{T}_m$ has been previously defined and is equal to $\mathcal{T}'$. The relation B_m is defined as $\{(w, w) \mid w \in \mathcal{T}_m\}$. For the induction step, take $0 \leq i \leq m - 1$. We define $\mathcal{T}_{m+i+1}$ as the tree obtained by replacing in $\mathcal{T}_{m+i}$, the subtree with root $c_{v_{i+1}}(f(v_{i+1}))$ by the tree $\mathcal{S}_{v_{i+1}}$. That is, we replace the copy of $\mathcal{T}_{f(v_{i+1})}$ by the tree $\mathcal{S}_{v_{i+1}}$. Since there is a $\sim$-sequence between $\mathcal{T}_{f(v_{i+1})}$ and $\mathcal{S}_{v_{i+1}}$, we can easily construct from it a $\sim$-sequence between the trees $\mathcal{T}_{m+i}$ and $\mathcal{T}_{m+i+1}$. Moreover, we may assume that

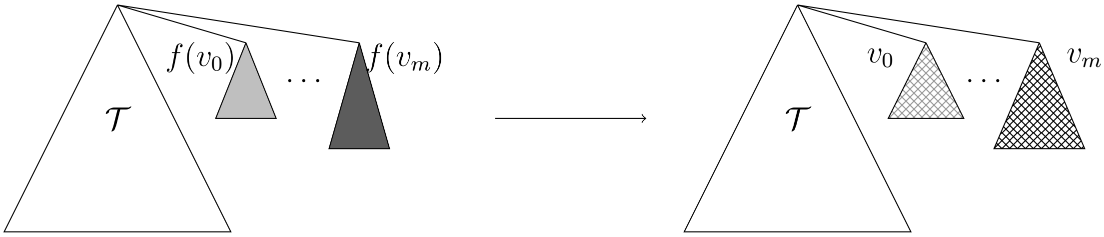


Figure 6.3: From $\mathcal{T}'$ to $\mathcal{Q}$.

there is a $\sim$-transitive bisimulation B_{m+i+1} associated to this sequence, such that

$$B_{m+i+1} = \{(w,w) \mid w \neq f(v_{i+1}) \text{ and } w \text{ is not a descendant of } c_{v_{i+1}}(f(v_{i+1}))\} \\ \cup \{(c_{v_{i+1}}(w), t) \mid (w,t) \in B^{v_{i+1}}\}.$$

It is routine to check that $\mathcal{T}_{m+i+1}$ and B_{m+i+1} satisfy the required properties.

Putting everything together, we obtain a sequence $\mathcal{T}_0, \dots, \mathcal{T}_{2m}$ such that $\mathcal{T}_0 = \mathcal{T}$, $\mathcal{T}_{2m} = \mathcal{Q}$ and for all $0 \leq i < 2m$, there is a $\sim$-sequence between $\mathcal{T}_i$ and $\mathcal{T}_{i+1}$. Moreover, the $\sim$-transitive bisimulation $B \circ B_m \circ \dots \circ B_{2m}$ between $\mathcal{T}_0$ and $\mathcal{T}_{2m}$ is equal to

$$\{(w,w) \mid w \in \mathcal{T}\} \cup \{(w,t) \mid (w,t) \in B^v, \text{ for some child } v \text{ of } v_0\}.$$

It follows from the fact that $B^v \subseteq B$ for all children v of v_0, that $B_1 \circ \dots \circ B_{2m}$ is a subset of $\{(r_0, r)\} \cup \{(u,u) \mid u \in \mathcal{T}\} \cup B$. This finishes the proof that there exist a $\sim$-sequence between $\mathcal{T}$ and $\mathcal{Q}$ and a $\sim$-transitive bisimulation for this sequence that is a subset of $\{(r_0, r)\} \cup \{(u,u) \mid u \in \mathcal{T}\} \cup B$.

Similarly we also obtain a $\sim$-sequence $\mathcal{S}_1, \dots, \mathcal{S}_l$ between $\mathcal{Q}$ and $\mathcal{S}$ and a $\sim$-transitive bisimulation for this sequence that is a subset of $\{(r, s_0)\} \cup \{(v,v) \mid v \in \mathcal{S}\} \cup B$. We can deduce that the sequence of finite trees $\mathcal{T}_1, \dots, \mathcal{T}_n, \mathcal{S}_2, \dots, \mathcal{S}_l$ is a $\sim$-sequence between $\mathcal{T}$ and $\mathcal{S}$ and there is a $\sim$-transitive bisimulation associated with this sequence and included in B. □

6.2.6. Proposition. *An* MSO *formula $\varphi(x)$ is invariant under transitive bisimulation on finite trees iff $\varphi(x)$ is invariant under the subtree and copy operations.*

Proof The direction from left to right follows easily from the facts that $\varphi(x)$ is invariant under transitive bisimulation on finite trees and that the relation $\{(w,w) \mid w \in \mathcal{T}\} \cup \{(w, c(w)) \mid vR^+w\}$ is a transitive bisimulation between $\mathcal{T}$ and $\text{COPY}_{u \longrightarrow v}(\mathcal{T})$.

For the direction from left to right, suppose that an MSO formula $\varphi(x)$ is invariant under the subtree and copy operations. We have to prove that $\varphi(x)$ is invariant under transitive bisimulation on finite trees. Let B be a transitive bisimulation between two finite trees $\mathcal{T}$ and $\mathcal{T}'$ and suppose that (u, u') belongs to B. Then, $B \cap (\mathcal{T}_u \times \mathcal{T}'_{u'})$ is a total transitive bisimulation between $\mathcal{T}_u$ and $\mathcal{T}'_{u'}$. It follows from Lemma 6.2.5 that there exists a $\sim$-sequence between $\mathcal{T}_u$ and $\mathcal{T}'_{u'}$. Using the fact that is invariant under the copy operation, we can check by induction on the length of the $\sim$-sequence that $\mathcal{T}_u, u \vDash \varphi(x)$ iff $\mathcal{T}'_{u'}, u' \vDash \varphi(x)$. Putting this together with the fact that is invariant under the subtree operation, we obtain that $\mathcal{T}, u \vDash \varphi(x)$ iff $\mathcal{T}', u' \vDash \varphi(x)$. □

Using Proposition 6.2.4 and Proposition 6.2.6, we obtain Theorem 6.2.1. Putting the second statement of Theorem 6.2.1 together with the decidability of MSO on finite trees, we obtain the following result.

6.2.7. Corollary. *It is decidable whether an* MSO *formula is equivalent to a* CoreXPath($\downarrow^+$) *node expression.*

6.2.8. Remark. We would like to mention that although the equivalence between (*i*) and (*iii*) is specific to the setting of finite trees, the equivalence between (*i*) and (*ii*) can be adapted the case of trees. Note that strictly speaking, a CoreXPath($\downarrow^+$) node expression cannot be, by definition, evaluated on an infinite tree. However, there is an obvious way to extend the interpretation of CoreXPath($\downarrow^+$) node expressions to the setting of infinite trees. Using a similar proof to the one we gave for Theorem 6.2.1, we can show that an MSO formula $\varphi(x)$ is equivalent to CoreXPath($\downarrow^+$) node expression on trees iff $\varphi(x)$ is invariant under transitive bisimulation.

6.3 CoreXPath($\downarrow^+$) path expressions

Now we will adapt this method in order to obtain a similar characterization for path expressions. As before, the characterization is twofold. We provide a first characterization that is formulated in terms of transitive bisimulations. We also give another characterization based on the link between CoreXPath($\downarrow^+$) path expressions and the operation copy. We start by introducing some terminology.

Equivalence An MSO formula $\varphi(x, y)$ is *equivalent* to a CoreXPath($\downarrow^+$) path expression A on finite trees if for all finite trees $\mathcal{T}$ and for all nodes u, v in $\mathcal{T}$, $\mathcal{T}, (u, v) \vDash \varphi(x, y)$ iff (u, v) belongs to $[\![A]\!]_{\mathcal{T}}$. When this happens, we write $\varphi(x, y) \equiv A$.

In the setting of programs, the notion corresponding to invariance under bisimulation, is the notion of safety for bisimulations, which was introduced by Johan van Benthem [Ben98].

Safety for bisimulations An MSO formula $\varphi(x, y)$ is *safe for (transitive) bisimulations* on finite trees if for all finite trees $\mathcal{T}, \mathcal{T}'$, (transitive) bisimulations $B \subseteq \mathcal{T} \times \mathcal{T}'$, pairs $(u, u') \in B$, and nodes $v \in \mathcal{T}$, if $\mathcal{T}, (u, v) \vDash \varphi(x, y)$, then there exists a node $v' \in \mathcal{T}'$ such that $(v, v') \in B$ and $\mathcal{T}', (u', v') \vDash \varphi(x, y)$.

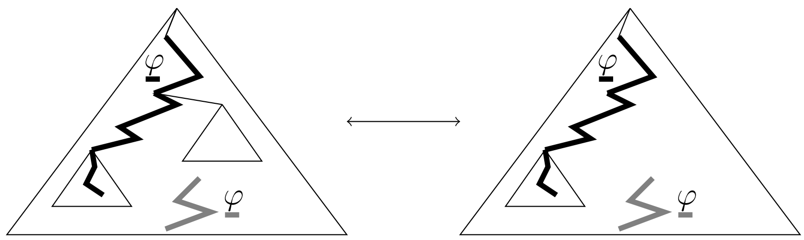


Figure 6.4: Condition (6.3).

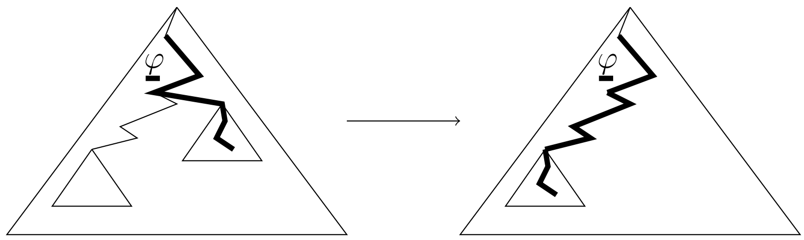


Figure 6.5: Condition (6.4).

Invariance under the subtree and the copy operations Let $\varphi(x, y)$ be an MSO formula. We say that $\varphi(x, y)$ is *invariant under the subtree operation (on finite trees)* if for all finite trees $\mathcal{T}$ and for all nodes u, v in $\mathcal{T}$ such that v is a descendant of u,

$$\mathcal{T}, (u, v) \models \varphi(x, y) \quad \text{iff} \quad \mathcal{T}_u, (u, v) \models \varphi(x, y).$$

The formula $\varphi(x, y)$ is *invariant under the copy operation (on finite trees)* if for all finite trees $\mathcal{T}$, for all nodes u, v, w, t in $\mathcal{T}$ such that v is a descendant of u and t a descendant of v, we have

$$\mathcal{T}, (w, t) \models \varphi(x, y) \quad \text{iff} \quad \textsc{copy}_{u \longrightarrow v}(\mathcal{T}), (w, t) \models \varphi(x, y), \tag{6.3}$$

$$\textsc{copy}_{u \longrightarrow v}(\mathcal{T}), (w, c(t)) \models \varphi(x, y) \quad \text{implies} \quad \mathcal{T}, (w, t) \models \varphi(x, y). \tag{6.4}$$

6.3.1. Theorem. *Let $\varphi(x, y)$ be an* MSO *formula. The following are equivalent:*

(i) $\varphi(x, y)$ is equivalent to a CoreXPath($\downarrow^+$) *path expression,*

(ii) $\varphi(x, y)$ is safe for transitive bisimulations on finite trees,

(iii) it is the case that $\varphi(x, y)$ is invariant under the subtree and copy operations.

Moreover, given an MSO *formula $\varphi(x, y)$, we can compute a* CoreXPath($\downarrow^+$) *path expression A such that $\varphi(x, y)$ is equivalent to a* CoreXPath($\downarrow^+$) *path expression iff $\varphi(x, y) \equiv A$.*

The structure of the proof is the same as the one for node expressions. It is based on versions of the Janin-Walukiewicz theorem and the de Jongh fixpoint theorem, adapted to the setting of μ-programs (instead of μ-formulas). We recall the syntax for the μ-programs. The *μ-programs* are given by

$$\theta \quad ::= \quad R \mid \varphi? \mid \theta;\theta \mid \theta \cup \theta \mid \theta^*,$$

where φ is a μ-sentence.

Given a Kripke model, these μ-programs are interpreted as binary relations over the model. These binary relations are defined by induction on the complexity of the programs. We only recall that the interpretation of θ^* is the reflexive transitive closure of the relation corresponding to θ. For more details, see Section 2.6.

There exists an expressivity result for μ-programs, which is the equivalent of the Janin-Walukiewicz theorem. It was proved by Marco Hollenberg [Hol98b] (for more details about this result, see Section 5.5 in Chapter 5). We show here how to derive this result from Theorem 5.5.3, in the special case where we restrict the class of structures to finite trees. Recall that an MSO formula $\varphi(x, y)$ is equivalent on finite trees to a μ-program θ if for all finite trees $\mathcal{T}$ and nodes u, v in $\mathcal{T}$, $\mathcal{T}, (u, v) \vDash \varphi(x, y)$ iff $(u, v) \in [\![\theta]\!]_{\mathcal{T}}$.

6.3.2. Theorem. *[from [Hol98b]] An* MSO *formula $\varphi(x, y)$ is safe for bisimulations on finite trees iff it is equivalent on finite trees to a μ-program.*

Moreover, given an MSO *formula $\varphi(x, y)$, we can compute a μ-program θ such that $\varphi(x, y)$ is equivalent on finite trees to a μ-program iff $\varphi(x, y)$ is equivalent to θ on finite trees.*

Proof The proof mainly relies on a variant of Theorem 5.5.3. The method used to derive the result from Theorem 5.5.3 is the one used by Johan van Benthem in [Ben98].

It is easy to show by induction on the complexity of μ-programs (and using the fact that μ-formulas are invariant under bisimulation) that a μ-program is safe for bisimulations. Hence, it is sufficient to show that given an MSO formula $\varphi(x, y)$, we can compute a μ-program θ such that if $\varphi(x, y)$ is safe for bisimulations on finite trees, then $\varphi(x, y)$ and θ are equivalent on finite trees.

Let $\varphi(x, y)$ be an MSO formula. Let $\psi(x)$ be the MSO formula $\exists y(\varphi(x, y) \wedge P(y))$, where P is a unary predicate corresponding to a fresh proposition letter p (i.e. p does not occur in $\varphi(x, y)$). By Theorem 6.2.2, we can compute a μ-sentence χ such that $\psi(x)$ is invariant under bisimulation on finite trees iff $\psi(x)$ and χ are equivalent on finite trees. It is easy to see that if $\varphi(x, y)$ is safe for bisimulations, then $\psi(x)$ is invariant under bisimulation on finite trees and in particular, $\psi(x)$ is equivalent to χ on finite trees.

Now the formula χ is completely additive with respect to p on finite trees. That is, for all finite trees $\mathcal{T} = (T, R, V)$ and all nodes u in $\mathcal{T}$,

$$\mathcal{T}, u \Vdash \psi \quad \text{iff} \quad \text{there is a node } v \in V(p) \text{ such that } \mathcal{T}[p \mapsto \{v\}], u \Vdash \psi.$$

It follows from Theorem 5.5.3 that for each formula that is completely additive with respect to p, we can compute an equivalent formula in the syntactic fragment $\mu\mathrm{ML}_A(p)$. The proof of Theorem 5.5.3 could be easily adapted to show that for each formula χ that is completely additive in p on finite trees, we can compute a formula in $\mu\mathrm{ML}_A(p)$, that is equivalent to χ on finite trees.

Moreover, as observed in Section 5.5 of Chapter 5, a formula belongs to $\mu\mathrm{ML}_S(p)$ iff it is equivalent to a formula of the form $\langle\theta\rangle p$, where p does not occur in the program θ. In fact, given a formula in $\mu\mathrm{ML}_S(p)$, we can compute a μ-program θ such that the formula is equivalent to $\langle\theta\rangle p$. Putting everything together, we can compute a μ-program θ in which p does not occur, such that that χ is equivalent on finite trees to $\langle\theta\rangle p$.

Recall that if $\varphi(x, y)$ is safe for bisimulations, then $\psi(x)$ is equivalent to χ on finite trees. Hence, if $\varphi(x, y)$ is safe for bisimulations, $\psi(x)$ is equivalent to $\langle\theta\rangle p$ on finite trees. It follows from the definition of $\psi(x)$ that if $\psi(x)$ is equivalent to $\langle\theta\rangle p$ on finite trees, then $\varphi(x, y)$ is equivalent to θ on finite trees. This finishes the proof that given an MSO formula $\varphi(x, y)$, we can compute a μ-program θ such that if $\varphi(x, y)$ is safe for bisimulations on finite trees, then $\varphi(x, y)$ and θ are equivalent on finite trees. □

We can also prove a variant of Theorem 6.2.3, which applies to programs. Recall that a modal program is a μ-program which does not contain any Kleene star * and all its subprograms of the form $\varphi?$ are such that φ is modal (a precise definition was given in Section 6.1).

6.3.3. Theorem. *For all μ-programs θ, we can compute a modal program λ such that for all finite transitive trees $\mathcal{T}^+$, we have $[\![\theta]\!]_{\mathcal{T}^+} = [\![\lambda]\!]_{\mathcal{T}^+}$.*

Proof The proof consists in showing that on finite transitive trees, each μ-program is equivalent to a finite disjunction of modal programs which are in a special shape. We call these modal programs basic. More precisely, we say that a modal program is a *basic modal program* if it belongs to the language defined by the following grammar

$$\theta \ ::= \ \varphi? \mid (\theta; R; \varphi?),$$

where φ is a modal formula. We show that each μ-program is equivalent to a finite disjunction of basic modal programs on finite transitive trees. The proof is by induction on the complexity of the μ-programs.

First, assume that θ a program of the form $\varphi?$ (where φ is a μ-sentence). Then it follows from Theorem 6.2.3 that φ is equivalent to a modal formula ψ on finite transitive trees. Therefore, θ is equivalent to the basic modal program $\psi?$ on finite transitive trees.

The cases where θ is the program R or a program of the form $\theta_1 \cup \theta_2$, are immediate. Next assume that θ is of the form $\theta_1; \theta_2$. By induction hypothesis, there exist sets Γ_1 and Γ_2 of basic modal programs such that $\theta_1 = \bigvee \Gamma_1$ and $\theta_2 = \bigvee \Gamma_2$. It follows that θ is equivalent to the modal program $\bigvee\{\gamma_1; \gamma_2 \mid \gamma_1 \in \Gamma_1, \gamma_2 \in \Gamma_2\}$. It remains to check that for all γ_1 in Γ_1 and all γ_2 in Γ_2, $\gamma_1; \gamma_2$ is equivalent to a basic modal program. Since γ_1 and γ_2 are basic programs, there are modal formulas $\varphi_1, \dots, \varphi_n$ and $\psi_1, \dots, \psi_k$ such that $\gamma_1 = \varphi_1?; R; \dots; R; \varphi_n?$ and $\gamma_2 = \psi_1?; R; \dots; R; \psi_k?$. Therefore, $\gamma_1; \gamma_2$ is equivalent to the basic program

$$\varphi_1?; R; \dots; R; (\varphi_n \wedge \psi_1)?; R; \psi_2; R; \dots; R; \psi_k?.$$

The only case left is where θ is a μ-program of the form λ^*. By induction hypothesis, λ is equivalent to the modal program $\bigvee \Gamma$, for some set Γ of basic modal programs. We prove that θ is equivalent to the disjunction of the set Δ given by:

$$\Delta = \{\gamma_1; \dots; \gamma_n \mid n \in \mathbb{N}, \gamma_1, \dots, \gamma_n \in \Gamma \text{ and } \gamma_1, \dots, \gamma_n \text{ are pairwise distinct}\}.$$

First, it is routine to check that $[\![\bigvee \Delta]\!]_{\mathcal{T}^+}$ is a subset of $[\![\lambda^*]\!]_{\mathcal{T}^+}$, on all finite transitive trees $\mathcal{T}^+$. So it remains to show that if $\mathcal{T}^+$ is a finite transitive tree and if the pair (u, v) belongs to $[\![\lambda^*]\!]_{\mathcal{T}^+}$, then there is a program δ in Δ such that (u, v) belongs to $[\![\delta]\!]_{\mathcal{T}^+}$.

Let (u, v) be such a pair. Since θ is equivalent to $(\bigvee \Gamma)^*$ on finite transitive trees, there are programs $\gamma_1, \dots, \gamma_n$ in Γ such that (u, v) belongs to $[\![\gamma_1; \dots; \gamma_n]\!]$. The problem is that $\gamma_1, \dots, \gamma_n$ might not be pairwise distinct. First, suppose that for all $i \in \{1, \dots, n\}$, the program γ_i is of the form $\varphi_i?$ (for some modal formula φ_i). Then, it is easy to see that if there are i and j such that $\varphi_i = \varphi_j$, we can delete the program γ_j from the list $\gamma_1, \dots, \gamma_n$, without modifying the fact that (u, v) belongs to $[\![\gamma_1; \dots; \gamma_n]\!]$.

Next suppose that there is at least one program γ_{i_0} which is not of the form $\varphi?$ (for some modal formula φ). First, we may assume that in fact no program γ_i is of the form $\psi?$. If there were such a program, say $\gamma_i = \psi?$, then we could remove γ_i from the list $\gamma_1, \dots, \gamma_n$ and it will still be the case that (u, v) belongs to the relation associated to the program $\gamma_1; \dots; \gamma_n$.

Now suppose that there are i and j such that $i < j$ and $\gamma_i = \gamma_j$. Since γ_i is not a program of the form $\psi?$, it follows from the induction hypothesis that there exists a formula ψ and a basic modal program γ such that $\gamma_i = \psi?; R; \gamma$. Since (u, v) belongs to $[\![\gamma_1; \dots; \gamma_n]\!]$ and $\gamma_i = \gamma_j = \psi?; R; \gamma$, there are nodes u_1 and u_2 such that

- (u, u_1) belongs to $[\![\gamma_1; \dots; \gamma_{i-1}; \varphi?]\!]$,
- (u_1, u_2) belongs to $[\![R; \gamma; \gamma_{i+1}; \dots; \gamma_{j-1}; \varphi?; R]\!]$,
- (u_2, v) belongs to $[\![\gamma; \gamma_{j+1}; \dots; \gamma_n]\!]$.

Note that if (u_1, u_2) belongs to $[\![R; \gamma; \gamma_{i+1}; \dots; \gamma_{j-1}; \varphi?; R]\!]$, then in particular, u_2 is a descendant of u_2. That is, (u_1, u_2) belongs to $[\![R]\!]$. So we obtain that

- (u, u_1) belongs to $[\![\gamma_1; \dots; \gamma_{i-1}; \varphi?]\!]$,
- (u_1, u_2) belongs to $[\![R]\!]$,
- (u_2, v) belongs to $[\![\gamma; \gamma_{j+1}; \dots; \gamma_n]\!]$.

That is, (u, v) belongs to $[\![\gamma_1; \dots; \gamma_i; \gamma_{j+1}; \dots; \gamma_n]\!]$. This means that whenever there are i and j such that $i < j$ and $\gamma_i = \gamma_j$, we can remove the programs $\gamma_{i+1}, \dots, \gamma_j$ from the list $\gamma_1, \dots, \gamma_j$ and it is still the case that (u, v) belongs to the relation associated to the program $\gamma_1; \dots; \gamma_n$. By repeating this operation, we may assume that (u, v) belongs to $[\![\gamma_1; \dots; \gamma_n]\!]$, where the programs $\gamma_1, \dots, \gamma_n$ are pairwise distinct. Therefore, (u, v) belongs to $[\![\bigvee \Delta]\!]$ and this finishes the proof that for all μ-programs θ, there exists a modal program λ which is equivalent to θ on finite transitive trees. A careful inspection of the proof shows that λ can be effectively computed from θ. □

We can now prove that CoreXPath($\downarrow^+$) path expressions correspond to the MSO[$\downarrow$] formulas that are safe for bisimulations.

6.3.4. Proposition. *An* MSO *formula $\varphi(x, y)$ is equivalent on finite trees to a* CoreXPath($\downarrow^+$) *path expression iff $\varphi(x, y)$ is safe for transitive bisimulations on finite trees.*

Moreover, given an MSO *formula $\varphi(x, y)$, we can compute a* CoreXPath($\downarrow^+$) *path expression A such that $\varphi(x, y)$ is equivalent to a* CoreXPath($\downarrow^+$) *path expression iff $\varphi(x, y) \equiv A$.*

Proof First, we show that an MSO formula $\varphi(x, y)$ is equivalent on finite trees to a CoreXPath($\downarrow^+$) path expression iff $\varphi(x, y)$ is safe for transitive bisimulations on finite trees. For the direction from left to right, the proof is a standard induction on the complexity of CoreXPath($\downarrow^+$) path expressions.

For the direction from right to left, let $\varphi(x, y)$ be an MSO formula that is safe for transitive bisimulations on finite trees. By Theorem 6.3.2, we can compute a μ-program θ such that $\varphi(x, y)$ and θ are equivalent on finite trees iff $\varphi(x, y)$ is safe for bisimulations on finite trees. Since $\varphi(x, y)$ is safe for transitive bisimulations on finite trees, $\varphi(x, y)$ is safe for bisimulations on finite trees. Hence, $\varphi(x, y)$ and θ are equivalent on finite trees.

Now we show that for all finite trees $\mathcal{T} = (T, R, V)$, we have that $[\![\theta]\!]_{\mathcal{T}}$ is equal to $[\![\theta]\!]_{\mathcal{T}^+}$. Take a finite tree $\mathcal{T} = (T, R, V)$ and nodes u, v in $\mathcal{T}$. We have to show that

$$(u, v) \in [\![\theta]\!]_{\mathcal{T}} \quad \text{iff} \quad (u, v) \in [\![\theta]\!]_{\mathcal{T}^+}. \tag{6.5}$$

Let $\mathcal{T}^+$ be the transitive tree (T, R^+, V) and let $\mathcal{S}$ be the unraveling of the pointed model $(\mathcal{T}^+, u)$ (see Section 2.6). Recall that $\mathcal{S}$ is a finite tree the root of which is u. Its domain is the set of paths of $\mathcal{T}^+$ with starting point u.

For the direction from left to right of (6.5), suppose that (u, v) belongs to $[\![\theta]\!]_{\mathcal{T}}$. The canonical bisimulation B between $\mathcal{S}$ and $\mathcal{T}^+$ is a transitive bisimulation between $\mathcal{S}$ and $\mathcal{T}$ such that (u, u) belongs to B. Since θ and $\varphi(x, y)$ are equivalent on finite trees and $\varphi(x, y)$ is safe for transitive bisimulations on finite trees, it follows from $(u, v) \in [\![\theta]\!]_{\mathcal{T}}$ and $(u, u) \in B$ that for some $s = (u_i)_{i \leq n}$ in $\mathcal{S}$, we have $(v, s) \in B$ and $(u, s) \in [\![\theta]\!]_{\mathcal{S}}$. By definition of B, $(v, s) \in B$ implies that $u_n = v$. Now as θ is a μ-program, it is safe for bisimulations. Since $(u, s) \in [\![\theta]\!]_{\mathcal{S}}$ and the pair (u, u) belongs to the canonical bisimulation B between $\mathcal{S}$ and $\mathcal{T}^+$, we have $(u, v') \in [\![\theta]\!]_{\mathcal{T}^+}$ and $(v', s) \in B^c$, for some v' in $\mathcal{T}^+$. By definition of the canonical bisimulation B, this can only be the case if $u_n = v'$. Since $u_n = v$, it follows that $v' = v$. Putting everything together, we obtain that (u, v) belongs to $[\![\theta]\!]_{\mathcal{T}^+}$.

For the direction from right to left of (6.5), suppose that (u, v) belongs to $[\![\theta]\!]_{\mathcal{T}^+}$. Since (u, u) belongs to the bisimulation B and α is safe for bisimulation, there exists $s = (u_i)_{i \leq n}$ such that $(u, s) \in [\![\theta]\!]_{\mathcal{S}}$ and $(v, s) \in B$. By definition of B, this can only happen if $u_n = v$. Now we have that $(u, s) \in [\![\theta]\!]_{\mathcal{S}}$, (u, u) belongs to the transitive bisimulation B and α is equivalent to $\varphi(x, y)$, which is safe for transitive bisimulations. Therefore, there exists $v' \in \mathcal{T}$ such that $(u, v') \in [\![\theta]\!]_{\mathcal{T}}$ and $(s, v') \in B$. Since (s, v') belongs to B, we have $u_n = v'$. We established earlier that $u_n = v$. Hence, $v = v'$. Putting everything together, we have $(u, v) \in [\![\theta]\!]_{\mathcal{T}}$ and this finishes the proof that $[\![\theta]\!]_{\mathcal{T}} = [\![\theta]\!]_{\mathcal{T}^+}$.

It also follows from Theorem 6.3.3 that θ is equivalent on finite transitive trees to a modal program λ. Given the connection between CoreXPath($\downarrow^+$) and modal logic (see Section 6.1.3), there exists a CoreXPath($\downarrow^+$) path expression A such that for all finite trees $\mathcal{T} = (T, R, V)$ and for all nodes $u, v \in T$.

$$(u, v) \in [\![\lambda]\!]_{\mathcal{T}^+} \quad \text{iff} \quad [\![A]\!]_{\mathcal{T}},$$

where $\mathcal{T}^+ = (T, R^+, V)$. Putting everything together, we found a CoreXPath($\downarrow^+$) path expression A such that for all finite trees $\mathcal{T}$ and all nodes u, v in $\mathcal{T}$, $\mathcal{T}, (u, v) \models \varphi(x, y)$ iff (u, v) belongs to $[\![A]\!]_{\mathcal{T}}$. This finishes the proof of the firsts statement.

Now it is easy to see that the fact that A is computable from $\varphi(x, y)$ does not depend on the fact that $\varphi(x, y)$ was safe for transitive bisimulations on finite trees. The second statement of the proposition immediately follows.

□

6.3.5. Proposition. *An* MSO *formula $\varphi(x, y)$ is safe for transitive bisimulations on finite trees iff $\varphi(x, y)$ is invariant under the subtree and copy operations.*

Proof The direction from left to right follows easily from the facts that $\varphi(x, y)$ is safe for transitive bisimulations on finite trees and that the relation $\{(w, w) \mid w \in \mathcal{T}\} \cup \{(w, c(w)) \mid vR^+w\}$ is a transitive bisimulation between $\mathcal{T}$ and $\text{COPY}_{u \longrightarrow v}(\mathcal{T})$.

For the direction from right to left, suppose that $\varphi(x, y)$ is an MSO formula that is invariant under the subtree and copy operations. In order to show that $\varphi(x, y)$ is safe for transitive bisimulation on finite trees, let B be a transitive bisimulation between two finite trees $\mathcal{T}$ and $\mathcal{S}$. Suppose that (u_0, v_0) belongs to B and that there is a node u_1 such that $\mathcal{T}, (u_0, u_1) \vDash \varphi(x, y)$. We have to prove that there exits a node $v_1 \in \mathcal{S}$ such that (u_1, v_1) belongs to B and $\mathcal{S}, (v_0, v_1) \vDash \varphi(x, y)$.

First observe that $B \cap (\mathcal{T}_{u_0} \times \mathcal{S}_{v_0})$ is a total transitive bisimulation between $\mathcal{T}_{u_0}$ and $\mathcal{S}_{v_0}$. By Lemma 6.2.5, there exist finite trees $\mathcal{T}_1, \dots, \mathcal{T}_n$, relations $B_1, \dots, B_{n_1}$ such that $\mathcal{T}_1 = \mathcal{T}_{u_0}$, $\mathcal{T}_n = \mathcal{S}_{v_0}$, B_i is a $\sim$ -transitive bisimulation between $\mathcal{T}_i$ and $\mathcal{T}_{i+1}$ (for all $i \in \{1, \dots, n-1\}$) and $B_1 \circ \dots \circ B_{n-1}$ is included in B. We denote by B_0 the transitive bisimulation $\{(w, w) \mid w \in \mathcal{T}_1\}$ between $\mathcal{T}_1$ and $\mathcal{T}_1$.

Now we prove by induction on i that for all $1 \leq i \leq n$,

$$\text{there exists } (u_1, w_i) \in B_0 \circ \dots \circ B_{i-1} \text{ such that } \mathcal{T}_i, (r_i, w_i) \vDash \varphi(x, y), \tag{6.6}$$

where r_i is the root of $\mathcal{T}_i$. The case where $i = 1$ is immediate as $\mathcal{T}, (u_0, u_1) \vDash \varphi(x, y)$ and $\mathcal{T}_1 = \mathcal{T}_{u_0}$. Let us turn to the induction step $i + 1$. By induction hypothesis, there exists $(u_1, w_i) \in B_0 \circ \dots \circ B_{i-1}$ such that $\mathcal{T}_i, (r_i, w_i) \vDash \varphi(x, y)$. Since B_i is a $\sim$-transitive bisimulation between $\mathcal{T}_i$ and $\mathcal{T}_{i+1}$, either $\mathcal{T}_i \Rightarrow \mathcal{T}_{i+1}$ or $\mathcal{T}_{i+1} \Rightarrow \mathcal{T}_i$. We suppose that $\mathcal{T}_{i+1} \Rightarrow \mathcal{T}_i$. The other case is in fact easier. So there are nodes u, v of $\mathcal{T}_{i+1}$ such that $\mathcal{T}_i$ is equal to $\text{COPY}_{u \longrightarrow v}(\mathcal{T}_{i+1})$. Since B_i is a $\sim$-transitive bisimulation and $(w_i, w_{i+1}) \in B_i$, either w_i belongs to $\mathcal{T}_{i+1}$ or $w_i = c(w)$ for some descendant w of v in $\mathcal{T}_{i+1}$.

If w_i belongs to $\mathcal{T}_{i+1}$, we can define w_{i+1} as w_i and by the fact that $\varphi(x, t)$ is invariant under the copy operation (see condition (6.3)), it is the case that $\mathcal{T}_{i+1}, (r_{i+1}, w_{i+1}) \vDash \varphi(x, y)$. If $w_i = c(w)$ for some descendant w of v in $\mathcal{T}_{i+1}$, we can define w_{i+1} as w and using the fact that $\varphi(x, y)$ is invariant under the copy operation (see condition (6.4)), we have $\mathcal{T}_{i+1}, (r_{i+1}, w_{i+1}) \vDash \varphi(x, y)$. This finishes the proof of (6.6).

Next let v_1 be a node such that (u_1, v_1) belongs to $B_1 \circ \dots \circ B_{n-1}$ and $\mathcal{T}_n, (r_n, v_1) \vDash \varphi(x, y)$. That is, $\mathcal{S}_{v_0}, (v_0, v_1) \vDash \varphi(x, y)$. Using the fact that $\varphi(x, y)$ is invariant under the subtree operation, we get that $\mathcal{S}, (v_0, v_1) \vDash \varphi(x, y)$. Finally, since $B_1 \circ \dots \circ B_{n-1} \subseteq B$, (u_1, v_1) belongs to B. Putting everything together, we found a node v_1 in $\mathcal{S}$ such that (u_1, v_1) belongs to B and $\mathcal{S}, (v_0, v_1) \vDash \varphi(x, y)$. □

Putting Proposition 6.3.5 and Proposition 6.3.4 together, we obtain a proof of Theorem 6.3.1. Putting the second statement of Theorem 6.3.1 together with the decidability of MSO on finite trees, we obtain the following result.

6.3.6. Corollary. *It is decidable whether an* MSO *formula is equivalent to a* CoreXPath($\downarrow^+$) *path expression.*

6.3.7. Remark. Similarly to the observation at the end of the previous section, we can extend the equivalence between (i) and (ii) in Theorem 6.3.1 to the setting of infinite trees.

6.4 Conclusions

In this chapter, we gave a characterization of the MSO formulas that are equivalent to a CoreXPath($\downarrow^+$) node expression. First we formulated the characterization in terms of bisimulations, using classical results about fixpoints. We derived from this result another characterization which involves closure properties under some simple operations on finite trees. We gave a similar characterization for the MSO formulas that are equivalent to a CoreXPath($\downarrow^+$) path expression. From both characterizations, we could derive a decision procedure.

We could ask what is the complexity of the procedure (in Theorem 6.2.1) that, given an MSO formula $\varphi(x)$, determines whether $\varphi(x)$ is equivalent to a CoreXPath($\downarrow^+$) node expression. We did not look in details at this question but we suspect that this procedure may be non-elementary. A possible proof would be to reduce our problem to the satisfiability problem for MSO on finite trees, which is non-elementary. The same comment holds for the complexity of the procedure in Theorem 6.3.1.

It is an important open question whether there is a similar decidable characterization for full CoreXPath in terms of MSO. Thomas Place and Luc Segoufin [PS10] recently characterized the node expressions of an important fragment of CoreXPath, namely CoreXPath($\downarrow^+, \uparrow^+, \leftarrow^+, \rightarrow^+$).

Chapter 7

Automata for coalgebras: an approach using predicate liftings

Automata theory is intimately connected to the μ-calculus, as illustrated in the preliminaries, in Chapter 5 and even in Chapter 4 (an important tool is the fact that MSO and the graded μ-calculus have the same expressive power on trees, which relies on an automata theoretic proof). Inspired by these examples, we introduce a notion of automata corresponding to coalgebraic fixpoint logic, which is a generalization of the μ-calculus. We also present one application: similarly to the case of the μ-calculus, we show that the satisfiability problem for coalgebraic fixpoint logic reduces to the non-emptiness problem for automata and derive a finite model property and a complexity result for the satisfiability problem for coalgebraic fixpoint logic.

Universal coalgebra provides a general and abstract way to define state-based evolving systems such as ω-words, trees, Kripke models, probabilistic transition systems, and many others. This general and abstract way is based on category theory. Formally, given a functor F on the category of sets, a coalgebra is a pair $\mathcal{S} = (S, \sigma)$, where S is the carrier or state space of the coalgebra, and $\sigma : S \to FS$ is its unfolding or transition map.

Logic enters the picture when one wants to reason about coalgebras. There are two main different approaches to introducing coalgebraic logic, but they both have the same starting point.

Since coalgebras are meant to represent dynamical systems, one would expect that the logic should not be able to distinguish structures describing the same process, or the same behavior. This is exactly what happens when we describe Kripke models with modal logic: the notion of bisimulation is used to express that two pointed models have the same behavior and modal logic is easily seen to be invariant under bisimulation. The idea is to use the Kripke models as a key example of coalgebra and to extend the definition of modal logic to the setting of universal coalgebra.

There were two main proposals for extending the definition of modal logic.

In [Mos99], Larry Moss introduced a modality ∇_F generalizing the ∇ modality for Kripke models to coalgebras of arbitrary type. This approach is uniform in the functor F, but as a drawback only works properly if F satisfies a certain category-theoretic property (viz., it should preserve weak pullbacks); also the nabla modality is syntactically rather nonstandard.

As an alternative, Dirk Pattinson [Pat03] and others developed coalgebraic modal formalisms, based on a completely standard syntax, that work for coalgebras of arbitrary type. In this approach, for each set Λ of predicate liftings (the definition of which is recalled in Section 7.2) for a functor F, we introduce a coalgebraic logic ML_Λ. The main idea is that each predicate lifting in Λ corresponds to a modality of ML_Λ. Many well-known variations of modal logic in fact arise as the coalgebraic logic ML_Λ associated with a set Λ of such predicate liftings; examples include both standard and (monotone) neighborhood modal logic, graded and probabilistic modal logic, coalition logic, and conditional logic.

Following the example of modal logic, we can increase the expressive power of these coalgebraic logics by adding fixpoint operators. The coalgebraic fixpoint logic obtained by adding fixpoint operators to the logic based on Moss' modality was introduced by Yde Venema [Ven06b]. Recently, Corina Cîrstea, Clemens Kupke and Dirk Pattinson [CKP09] introduced the coalgebraic μ-calculus $\mu\mathrm{ML}_\Lambda$ (where Λ is a set of predicate liftings), which is the extension of ML_Λ with fixpoint operators.

Since automata theory has played a fundamental role in the understanding of the μ-calculus, it seems natural to develop notions of automata for the broader setting of fixpoint coalgebraic logic. As there are two approaches for coalgebraic logic, one may expect two kinds of automata (one for each approach). So far only automata corresponding to fixpoint languages based on Moss' modality have been introduced (see [Ven06b]). Moreover, in [KV08], Clemens Kupke and Yde Venema generalized many results in automata theory, such as closure properties of recognizable languages, to this class of automata.

Our contribution is to define automata corresponding to the coalgebraic fixpoint logic based on the idea that modalities are predicate liftings (as defined in [CKP09]). More precisely, given a set Λ of monotone predicate liftings for a functor F, we introduce Λ-automata as devices that accept or reject pointed F-coalgebras (that is, coalgebras with an explicitly specified starting point). We emphasize that unlike the coalgebra automata introduced in [Ven06b], these automata are defined for all functors (and not only the one preserving weak pullbacks).

As announced, Λ-automata provide the counterpart to the coalgebraic fixpoint logic $\mu\mathrm{ML}_\Lambda$ and there is a construction transforming a $\mu\mathrm{ML}_\Lambda$-formula into an equivalent Λ-automaton. Hence we may use the theory of Λ-automata in order to obtain results about coalgebraic fixpoint logic.

We give here an example: We use Λ-automata in order to prove a finite model property for coalgebraic fixpoint logic. We also derive from the proof a complexity bound for the satisfiability problem for $\mu\mathrm{ML}_\Lambda$.

More precisely, we reduce the satisfiability problem for the coalgebraic fixpoint logic ML_Λ to the non-emptiness problem for a Λ-automaton. Then we show that any Λ-automaton $\mathbb{A}$ with a non-empty language recognizes a coalgebra $\mathcal{S}$ that can be obtained from $\mathbb{A}$ via some uniform construction. The size of $\mathcal{S}$ is finite (in fact, exponential in the size of $\mathbb{A}$). On the basis of our proof, in Theorem 7.3.4, we give a doubly exponential bound on the complexity of the satisfiability problem of $\mu\mathrm{ML}_\Lambda$-formulas (provided that the one-step satisfiability problem of Λ has a reasonable complexity).

Compared to the work of Corina Cîrstea, Clemens Kupke and Dirk Pattinson [CKP09], our results are more general in the sense that they do not depend on the existence of a complete tableau calculus. On the other hand, the cited authors obtain a much better complexity result. Under some mild conditions on the efficiency of their complete tableau calculus (conditions that are met by e.g. the modal μ-calculus and the graded μ-calculus), they establish an EXPTIME upper bound for the satisfiability problem of the μ-calculus for Λ. However, in Section 7.4 below we shall make a connection between our satisfiability game and their tableau game, and on the basis of this connection one may obtain the same complexity bound as in [CKP09] (if one assumes the same conditions on the existence and nature of the tableau system).

7.1 Preliminaries

7.1.1 Coalgebras

As mentioned in the introduction, universal coalgebra provides an abstract way of defining dynamical systems, which is based on category theory. However, very little knowledge about category theory is required for this chapter. We only need the notions of category, functor and natural transformation. We start by recalling these definitions.

Category A *category* consists of a class of *objects*, a class of *arrows* and two operations on arrows that are called the *identity* and the *composition*. Each arrow has a domain and a codomain which belong to the class of objects. If an arrow f has a domain S and codomain T, we write $f : S \longrightarrow T$ and we say that f is an arrow between S and T. The identity assigns to each object S an arrow 1_S with domain S and codomain S. The composition assigns to each pair of arrows (f, g) a new arrow, notation: $g \circ f$, the domain of which is the domain of f and the codomain of which is the codomain of g.

Moreover, the identity and the composition must satisfy the following two axioms. If f, g and h are arrows such that the codomain of h is equal to the domain of g and the codomain of g is equal to the domain of h, then $h \circ (g \circ f) = (h \circ g) \circ f$

(the *associativity axiom*). If S is an object, then for all arrows f with domain S and for all arrows g with codomain g, we have $f \circ 1_S = f$ and $1_S \circ g = g$ (the *identity axiom*).

The only two categories that we consider in this chapter are the category Set and the category Set^{op}. The objects of the categories Set and Set^{op} are the sets. Given two sets S and T, an arrow from S to T in Set is a map from S to T. An arrow from S to T in Set^{op} is a map from T to S. The identity and the composition operations on Set and Set^{op} are defined as usual.

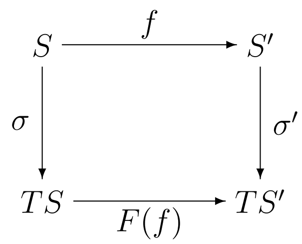
Functor Let C and D be two categories. A *functor* $F : \mathsf{C} \longrightarrow \mathsf{D}$ consists of two functions. The *object function* which assigns to each object S of C an object FS of D and the *arrow function* which assigns to each arrow $f : S \longrightarrow T$ of C, an arrow $Ff : FS \longrightarrow FT$ such that the two following conditions hold. For all objects S of C, $F(1_S) = 1_{FS}$. For all arrows $f : S \longrightarrow T$ and $g : T \longrightarrow U$ of C, we have $F(g \circ f) = Fg \circ Ff$.

Examples of functors that are crucial in this chapter are the n-contravariant powerset functors. For each $n \in \mathbb{N}_0$, we define the *n-contravariant powerset functor* $\mathcal{Q}^n : \mathsf{Set} \longrightarrow \mathsf{Set}^{op}$. $\mathcal{Q}^n$ maps a set S to the set $\mathcal{Q}^n(S) = \{(U_1, \dots, U_n) \mid \text{for all } 1 \leq i \leq n,\, U_i \subseteq S\}$. Moreover, for all arrows $f : S \longrightarrow T$ of Set^{op}, the action of $\mathcal{Q}^n$ on f is such that for all subsets $U_1, \dots, U_n$ of T, $\mathcal{Q}f(U_1, \dots, U_n) = (f^{-1}[U_1], \dots, f^{-1}[U_n])$.

Coalgebra Let $F : \mathsf{Set} \longrightarrow \mathsf{Set}$ be a functor. A *F-coalgebra* is a pair (S, σ) where S is a set and σ is a function $\sigma : S \longrightarrow FS$. S is called the *carrier set* of $\mathcal{S}$. A *pointed coalgebra* is a pair $(\mathcal{S}, s)$, where $\mathcal{S}$ is a coalgebra and s belongs to the carrier set of $\mathcal{S}$. The *size of a coalgebra* $\mathcal{S}$ is the cardinality of the set S.

For convenience, we assume the functors $F : \mathsf{Set} \longrightarrow \mathsf{Set}$ to be *standard* . That is, for all subsets U of a set S, FU is a subset of FS and the inclusion map i from U to S is mapped by F to the inclusion map from TU to TS. Note that we can make this restriction without loss of generality: For every non-standard functor F', there exists a standard functor F such that the class of F-coalgebras is equivalent to the class of F'-coalgebras (for more details, see for instance [Bar93]).

A *morphism of F-coalgebras* from $\mathcal{S}$ to $\mathcal{S}'$, written $f : \mathcal{S} \longrightarrow \mathcal{S}'$, is a function $f : S \longrightarrow S'$ such that the following diagram



commutes.

As mentioned in the introduction, many mathematical structures can be seen as coalgebras. We give here a few examples, that we use as running examples through this chapter. The examples are only here as illustrations and are not essential for understanding of this chapter.

7.1.1. EXAMPLE. (1) We write $\mathcal{P}$ for the covariant powerset functor. It maps a set to its powerset and its action on arrows is such that for all maps $f : S \to T$, $\mathcal{P}f(U) = f[U]$, for all subsets U of S. A $\mathcal{P}$-coalgebra consists of a set S and a map from S to its powerset $\mathcal{P}(S)$.

In fact, $\mathcal{P}$-coalgebras correspond to Kripke frames. The key observation is that given a set W, a binary relation $R \subseteq W \times W$ can be represented by a function $R[\cdot] : W \to \mathcal{P}(W)$ that maps a point to its set of successors. So a Kripke frame (W, R) corresponds to the $\mathcal{P}$-coalgebra $(W, R[\cdot])$.

(2) We denote by B the bags, or multiset functor that maps a set S to the set $B(S)$ that contains the maps from S to $\mathbb{N}$ with finite support (that is, there only finitely many points in S that are not mapped to 0). Given a map $f : S \to T$, the map $B(f)$ is defined such that for all maps $g : S \to \mathbb{N}$ with finite support, $((B(f))g)(t) = \sum_{f(s)=t} g(s)$.

B-coalgebras can be seen as directed graphs with $\mathbb{N}$-weighted edges, known as multigraphs [DV02]. We might assume that the edges with a weight equal to 0 are not represented. In this case, the B-coalgebras correspond exactly the directed graphs with $\mathbb{N}$-weighted edges and such that there only finitely many edges going out of each point.

(3) The monotone neighborhood functor M maps a set S to $M(S) = \{U \in \mathcal{QQ}(X) \mid U \text{ is upwards closed}\}$, and a function f to $M(f) = \mathcal{QQ}(f) = (f^{-1})^{-1}$. Coalgebras for this functor are monotone neighborhood frames [HK04].

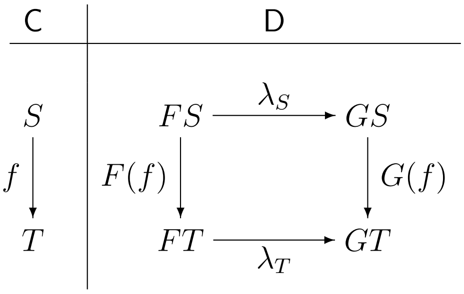
(4) We write D for the distribution functor that maps a set S to $D(S) = \{g : S \to [0, 1] \mid \sum_{s \in S} g(s) = 1\}$ and a function $f : S \to T$ to the function $D(f) : D(S) \to D(T)$ that maps a probability distribution g to a probability distribution $(D(f))(g)$ such that for all $t \in T$, $(D(f))(g)(t) = \sum_{f(s)=t} \mu(s)$. In this case coalgebras correspond to Markov chains [BSd04].

(5) Given a functor F from Set to Set, we define a new functor $\mathcal{P}(Prop) \times F$ from Set to Set. This new functor maps a set S to the set $\mathcal{P}(Prop) \times FS$ and maps a function $f : S \to T$ to a function $g : \mathcal{P}(Prop) \times FS \to \mathcal{P}(Prop) \times FT$ such that $g(P, \eta) = (P, Ff(\eta))$, for all $P \subseteq Prop$ and $\eta \in FS$.

For example, in case $F = \mathcal{P}$, a $\mathcal{P}(Prop) \times F$-coalgebra is nothing but a Kripke model. Indeed, a Kripke model (W, R, V) can be seen as the coalgebra $(W, V^{-1}[\cdot] \times R[.])$, where $R[.]$ is defined as in the first example and $V^{-1}[\cdot]$ is the map from W to $\mathcal{P}(Prop)$ such that for all $w \in W$, $V^{-1}[w] = \{p \in Prop \mid w \in V(p)\}$.

The last notion related to category theory that we use in this chapter is the notion of a natural transformation. It plays a central role in the definition of coalgebraic logic.

Natural transformation Let C and D be two categories and let $F, G : \mathsf{C} \longrightarrow \mathsf{D}$ be functors. A *natural transformation* λ from F to G, written $\lambda : F \longrightarrow G$, is a family of D-arrows $\lambda_S : FS \longrightarrow GS$, indexed by the objects of C, such that for all C-arrows $f : S \longrightarrow T$, the following diagram



commutes.

7.1.2 Graph games

The proofs of our main results are based on a game characterizing the coalgebraic automata the associated language of which is not empty. So we will be dealing often with vocabulary related to game theory. Basic notions of game theory can be found in Chapter 2. We introduce the notion of finite memory strategy, which plays a crucial role in this chapter. We also mention some sufficient conditions for games such that their winning strategies (if existing) can be assumed to be based on a finite memory.

Finite memory strategy Recall that given a set G, we write G^* for the set of *finite* sequences of elements in G; we write G^ω for the set *infinite* sequences of elements in G. If $\pi \in G^* + G^\omega$, we write $Inf(\pi)$ for the set of elements in G that appear infinitely often in π.

Recall also that given a game $\mathbb{G} = (G_\exists, G_\forall, E, Win)$, we write G for the board $G_\exists \cup G_\forall$ of the game. Similarly, we write G' for the board of a game $\mathbb{G}' = (G'_\exists, G'_\forall, E', Win')$.

Given a partial map f, we denote by $Dom(f)$ the domain of f. A strategy f for a player σ in a game $\mathbb{G} = (G_\exists, G_\forall, E, Win)$ is a *finite memory strategy* if there exist a finite set M, called the *memory set*, an element $m_I \in M$, a partial map $f_1 : G \times M \longrightarrow G$ and a partial map $f_2 : G \times M \longrightarrow M$ such that the following holds. For all $\mathbb{G}$-match $z_0 \dots z_k \in Dom(f)$, there exists a sequence $m_0 \dots m_k \in M^*$ such that

$$m_0 = m_I, m_{i+1} = f_2(z_i, m_i) \text{ (for all } i < k) \text{ and } f(z_0 \dots z_k) = f_1(z_k, m_k).$$

Next, we introduce regular games and show that their winning strategies can be assumed to have a finite memory. The regular games are games the winning condition of which is specified by an automaton.

Regular games A graph game $\mathbb{G} = (G_\exists, G_\forall, E, Win)$ is called *regular* if there exist a finite alphabet Σ, a coloring $col : G \to \Sigma$, and an ω-regular language L over Σ, such that

$$Win = \{z_0 z_1 \ldots \in G^\omega \mid col(z_0)col(z_1)\ldots \in L\}.$$

An initialized graph game $(G_\exists, G_\forall, E, Win, z_0)$ is *regular* if $\mathbb{G} = (G_\exists, G_\forall, E, Win)$ is regular.

The following result was proved by J. Richard Büchi and Lawrence Landweber [BL69]. We recall here the proof since further on, we use it to prove some complexity result.

7.1.2. Proposition. *If a $\mathbb{G} = (G_\exists, G_\forall, E, Win)$ is a regular game, then there exist finite memory strategies $f_\exists$ and $f_\forall$ for $\exists$ and $\forall$ respectively such that for all positions z on the board of $\mathbb{G}$, z is winning either with respect to $f_\exists$ or with respect to $f_\forall$.*

Moreover, the size of the memory set is bounded by the size of the smallest deterministic parity automaton recognizing an ω-regular language associated to the regular game.

Proof Let $\mathbb{G} = (G_\exists, G_\forall, E, Win)$ be a regular game. By definition, there exists a finite alphabet Σ, a coloring $col : V \to \Sigma$, and an ω-regular language L over Σ, such that

$$Win = \{z_0 z_1 \cdots \in G^\omega \mid col(z_0)col(z_1)\cdots \in L\}.$$

Moreover, since L is an ω-regular language, there is a deterministic parity automaton $\mathbb{B} = (Q_B, q_B, \delta_B, \Omega_B)$ recognizing L.

First we define a new parity game $\mathbb{G}' = (G'_\exists, G'_\forall, E', \Omega')$ in the following way: The sets $G'_\exists$ and $G'_\forall$ are respectively the sets $G_\exists \times Q_B$ and $G_\forall \times Q_B$ and the relation E' is defined by

$$E' = \{((z,q),(z',q')) \in G' \times G' \mid (z,z') \in E, \delta(q, col(z)) = q'\}.$$

The map $\Omega' : G \times Q_B \to \mathbb{N}$ is the map such that $\Omega'(z,q) = \Omega_B(q)$, for all $(z,q) \in G \times Q_B$.

Since $\mathbb{G}'$ is a parity game, there exist finite memory strategies $f'_\exists$ and $f'_\forall$ for $\exists$ and $\forall$ respectively such that for all positions (z,q) on the board, (z,q) is winning either with respect to $f'_\exists$ or with respect to $f'_\forall$. (see Theorem 2.3.1).

We are now ready to define a finite memory strategy $f_\exists$ for $\exists$ in $\mathbb{G}$. We let the memory set M be the set Q_B and we let m_I be the initial state q_B. The strategy $f_\exists$ is uniquely determined by maps $f_1 : G \times Q_B \to G$ and $f_2 : G \times Q_B \to Q_B$. We define the partial map $f_1 : G \times Q_B \to G$ as the partial map with domain $Dom(f'_\exists)$ and such that for all $(z,q) \in Dom(f'_\exists)$ with $f'_\exists(z,q) = (z',q')$, we have

$$f_1(z,q) = z'.$$

The map $f_2 : G \times Q_B \longrightarrow Q_B$ is such that for all (z, b) in $G \times Q_B$, we have

$$f_2(z, q) = \delta(q, col(z)).$$

Similarly we define the finite memory strategy $f_\forall$. It remains to check that for all positions z on the board of $\mathbb{G}$, z is winning either with respect to $f_\exists$ or with respect to $f_\forall$. Let z_0 be a position on the board G.

It follows from the definitions of $f'_\exists$ and $f'_\forall$ that (z_0, q_b) is winning either with respect to $f'_\exists$ or with respect to $f'_\forall$. Without loss of generality, we may assume that (z_0, q_B) is winning with respect to $f'_\exists$. Let $\pi = z_0 z_1 \dots$ be an $f_\exists$-conform $\mathbb{G}$-match. Let $q_0 q_1 \dots$ be the sequence such that $q_0 = m_I = q_B$ and $q_{i+1} = f_2(z_i, q_i)$, for all i. It is easy to check that the $\mathbb{G}'$-match $\pi' := (z_0, q_0), (z_1, q_1) \dots$ is an $f'_\exists$-conform match. Since $(z_0, q_B) = (z_0, q_0)$ is winning with respect to $f'_\exists$, π is won by $\exists$.

First suppose that the match π is finite. Then the $\mathbb{G}'$-match π' is also a finite match. Since this match is won by $\exists$, the last position of the match belongs to $\forall$. Therefore, the last position of the match π is also a position for $\forall$. Thus, this finite match is won by $\exists$.

Finally suppose that the match $z_0 z_1 \dots$ is infinite. Since the $\mathbb{G}'$-match (z_0, q_0) $(z_1, q_1) \dots$ is won by $\exists$, the maximum of the set $\{\Omega_B(q) \mid q \in \mathit{Inf}(q_0 q_1 \dots)\}$ is even. It follows from the definition of f_2 and m_I that the sequence $q_0 q_1 \dots$ is the unique run of $\mathbb{B}$ on $col(z_0)col(z_1) \dots$. Putting everything together, we obtain that the word $col(z_0)col(z_1) \dots$ is accepted by $\mathbb{B}$. Therefore, the match $z_0 z_1 \dots$ is won by $\exists$ and this finishes the proof. □

By putting the proof of the previous proposition together with the next theorem (see Chapter 2), we can obtain a complexity result for the problem whether a given position in a regular game is winning.

7.1.3. THEOREM. *[Jur00] Let $\mathbb{G} = (G_\exists, G_\forall, E, \Omega)$ be a parity game and let n, m and d be the size of G, E and $ran(\Omega)$, respectively. Then for each player σ, the problem, whether a given position $z \in G$ is winning for σ, is decidable in time $\mathcal{O}\left(d \cdot m \cdot \left(\frac{n}{\lfloor d/2 \rfloor}\right)^{\lfloor d/2 \rfloor}\right)$.*

7.1.4. THEOREM. *Let $\mathbb{G} = (G_\exists, G_\forall, E, \mathit{Win})$ be a regular game, let $col : G \longrightarrow \Sigma$ be a coloring of G, and let $\mathbb{B} = (Q_B, q_B, \delta_B, \Omega_B)$ be a deterministic parity ω-automaton such that $\mathit{Win} = \{z_0 z_1 \dots \in G^\omega \mid col(z_0)col(z_1) \dots \in L(\mathbb{B})\}$. Let n, m and b be the size of G, E and $\mathbb{B}$, respectively, and let d be the index of $\mathbb{B}$. The problem, whether a given position $z \in G$ is winning for a player σ, is decidable in time $\mathcal{O}\left(d \cdot m \cdot b \cdot \left(\frac{b \cdot n}{\lfloor d/2 \rfloor}\right)^{\lfloor d/2 \rfloor}\right)$.*

Proof Let $\mathbb{G} = (G_\exists, G_\forall, E, Win)$ be a regular game, let $col : G \to \Sigma$ be a coloring of G, and let $\mathbb{B}$ be a deterministic parity stream automaton such that $Win = \{z_0z_1 \ldots \in G^\omega \mid col(z_0)col(z_1) \ldots \in L(\mathbb{B})\}$. By the proof of the previous proposition, we know that there is a parity game $\mathbb{G}' = (G'_\exists, G'_\forall, E', \Omega')$ which satisfies the following conditions. The sets $G'_\exists$ and $G'_\forall$ are respectively the sets $G_\exists \times Q_B$ and $G_\forall \times Q_B$ and the relation E' is defined by

$$E' = \{((z, q), (z', q')) \in G' \times Q_B \mid (z, z') \in E, \delta(q, col(z)) = q'\}.$$

The map $\Omega' : G \times Q_B \to \mathbb{N}$ is the map such that $\Omega'(z, q) = \Omega_B(q)$, for all $(z, q) \in G \times Q_B$. Moreover, a position z is winning for a player σ in $\mathbb{G}$ iff the position (z, q_B) is winning for σ in $\mathbb{G}'$.

First we observe that we can compute the game $\mathbb{G}'$ in time linear in the size of the input. The board of $\mathbb{G}'$, the relation E' and the map Ω' are directly given by the board of $\mathbb{G}$, the set B, the relation E, the transition map δ, the coloring col and the map Ω. Moreover, the sizes of the board of $\mathbb{G}'$, the relation E' and the map Ω' are quadratic in the size of the input.

Now we show that it is decidable in time $\mathcal{O}\left(d \cdot m \cdot b \cdot \left(\frac{b \cdot d}{\lfloor d/2 \rfloor}\right)^{\lfloor d/2 \rfloor}\right)$ whether a position (z, q_B) is winning for a player σ in $\mathbb{G}'$. The size of the board of $\mathbb{G}'$ is equal to $|G| \cdot |Q_B| = n \cdot b$. The size of the relation E' is less or equal to $m \cdot b$. The size of the range of Ω' is the size of the range of Ω. That is, is equal to d. It follows from Theorem 7.1.3 that it is decidable in time $\mathcal{O}\left(d \cdot m \cdot b \cdot \left(\frac{b \cdot n}{\lfloor d/2 \rfloor}\right)^{\lfloor d/2 \rfloor}\right)$ whether a position (z, q_V) is winning for a player σ in $\mathbb{G}'$. Putting this together with the fact that a position z is winning for a player σ in $\mathbb{G}$ iff the position (z, q_B) is winning for σ in $\mathbb{G}'$, we conclude. □

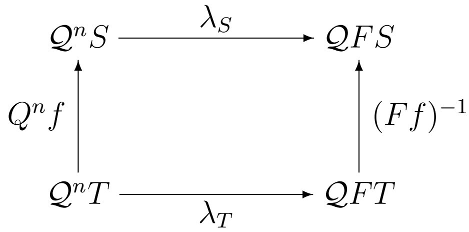
7.2 Automata for the coalgebraic μ-calculus

As mentioned in the introduction, the notion of coalgebraic logic that we consider in this chapter, is the one introduced in [Pat03]. The idea is to extend modal logic to a coalgebraic logic, by viewing the modal operators as predicate liftings.

Predicate lifting An n-ary *predicate lifting* for a functor F is a natural transformation $\lambda : \mathcal{Q}^n \to \mathcal{Q}F$.

Such a predicate lifting is *monotone* if for each set S, the operation $\lambda_S : \mathcal{Q}^n S \to \mathcal{Q}S$ preserves the subset order in each coordinate. The (Boolean) *dual* of a predicate lifting $\lambda : \mathcal{Q}^n \to \mathcal{Q}F$ is the lifting $\overline{\lambda} : \mathcal{Q}^n \to \mathcal{Q}F$ such that for all sets S and for all subsets $U_1, \ldots, U_n$ of S, $\overline{\lambda}_S(U_1, \ldots, U_n) = S \setminus \lambda(S \setminus U_1, \ldots, S \setminus U_n)$.

In other words, a predicate lifting λ is a family of maps $\lambda_S : \mathcal{Q}^n S \to \mathcal{Q}FS$ (for each set S) such that for all maps $f : S \to T$ between two sets S and T, the following diagram commutes:



Convention In the remainder of this chapter, we fix a standard functor F on Set and a set Λ of monotone predicate liftings that we assume to be closed under taking Boolean duals.

In a few paragraphs, we will some examples of predicate liftings, but let us start by introducing the notion of coalgebraic modal logic associated with the set Λ of predicate liftings.

Coalgebraic modal logic We define the set ML_Λ of *coalgebraic modal* Λ-*formulas* as follows:

$$\varphi ::= \bot \mid \top \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \heartsuit_\lambda(\varphi_1, \dots, \varphi_n)$$

where the arity of the operator $\heartsuit_\lambda$ is n if λ is a n-ary predicate. This language is the same as the one for modal logic, except that instead of having the modal operators $\Diamond$ and $\Box$, we associate with each n-ary predicate lifting λ an n-ary modality $\heartsuit_\lambda$. Another difference is that we assume the set of proposition letters to be empty. We will see later that it is possible to encode a proposition letter (or the negation of a proposition letter) using a 0-ary predicate lifting.

On a given coalgebra, a coalgebraic formula has a meaning, which corresponds to a subset of the carrier set of the coalgebra. Formally, we define the following semantics for ML_Λ.

Let $\mathcal{S} = (S, \sigma)$ be a F-coalgebra. The *meaning* $[\![\varphi]\!]_\mathcal{S}$ of a formula φ is defined by induction on the complexity of φ as follows:

$$\begin{array}{rcl} [\![\top]\!]_\mathcal{S} & = & S, \\ [\![\bot]\!]_\mathcal{S} & = & \emptyset, \\ [\![\varphi \vee \psi]\!]_\mathcal{S} & = & [\![\varphi]\!]_\mathcal{S} \cup [\![\psi]\!]_\mathcal{S}, \\ [\![\varphi \wedge \psi]\!]_\mathcal{S} & = & [\![\varphi]\!]_\mathcal{S} \cap [\![\psi]\!]_\mathcal{S}, \\ [\![\heartsuit_\lambda(\varphi_1, \dots, \varphi_n)]\!]_\mathcal{S} & = & \sigma^{-1}\lambda_S([\![\varphi_1]\!]_\mathcal{S}, \dots, [\![\varphi_n]\!]_\mathcal{S}), \end{array}$$

where λ belongs to Λ. The only non immediate step of the definition is the one for the meaning of a formula of the form $\heartsuit_\lambda(\varphi_1, \dots, \varphi_n)$. In words, $\heartsuit_\lambda(\varphi_1, \dots, \varphi_n)$ is true at a state s iff the unfolding $\sigma(s)$ belongs to the set $\lambda_S([\![\varphi_1]\!]_\mathcal{S}, \dots, [\![\varphi_n]\!]_\mathcal{S})$. As usual, if a state s belongs to $[\![\varphi]\!]_\mathcal{S}$, we write $\mathcal{S}, s \Vdash \varphi$.

As mentioned in the introduction, many well-known variations of modal logic arise as the coalgebraic logic ML_Λ, associated to a well chosen set Λ of predicate liftings. We give here few examples for the functors defined in the preliminaries of this chapter. We do not recall the definition of neighborhood modal logic and probabilistic modal logic. These logics are only mentioned for the people familiar with these notions.

7.2.1. Example. (1) In case of the covariant power set functor $\mathcal{P}$, the unary predicate lifting given by $\lambda_S(U) = \{V \in \mathcal{P}S \mid V \subseteq U\}$ induces the usual universal modality $\Box$.

Indeed, we can show by induction on the complexity of φ that for all formulas φ in $ML_{\{\lambda,\bar{\lambda}\}}$ and for all Kripke frames $\mathbb{F} = (W, R)$, seen as coalgebras (W, σ) for the covariant power set functor (that is, for all $w \in W$, $\sigma(w) = R[s]$), we have $[\![\varphi]\!]_\mathbb{F} = [\![\varphi']\!]_\mathbb{F}$, where φ' is the formula obtained from φ by replacing each occurrence of $\heartsuit_\lambda$ by $\Box$ and each occurrence of $\heartsuit_{\bar{\lambda}}$ by $\Diamond$.

The only non immediate case for the induction is when φ is a formula of the form $\heartsuit_\lambda(\psi)$. In this case, we have

$$\begin{aligned}
[\![\heartsuit_\lambda \psi]\!]_\mathbb{F} &= \sigma^{-1}\lambda_W([\![\psi]\!]_\mathbb{F}), \\
&= \sigma^{-1}\{U \subseteq W \mid U \subseteq [\![\psi]\!]_\mathbb{F}\}, \\
&= \{w \in W \mid \sigma(w) \subseteq [\![\psi]\!]_\mathbb{F}\}, \\
&= \{w \in W \mid R[w] \subseteq [\![\psi]\!]_\mathbb{F}\}, \\
&= [\![\Box\psi']\!]_\mathbb{F},
\end{aligned}$$

where the last equality follows from the induction hypothesis.

(2) Consider the multiset functor B. For all $k \in \mathbb{N}$, we define the unary predicate lifting λ^k for B such that for all sets S and subsets U of S, we have $\lambda^k_S(U) = \{B \mid S \to \mathbb{N} \mid \sum_{u \in U} B(u) \geq k\}$. In this case, $\mathcal{S}, s \Vdash \heartsuit^k_\lambda \varphi$ holds iff s has at least k many successors satisfying φ, taking into account the weight of the edges (that is, if there is an edge from s to a point t with label n, the point t "generates" n successors of s satisfying φ).

This is similar to the semantics of the graded modality $\Diamond^k$. However, graded modal logic is interpreted over Kripke frames, not multisets. Obviously, each Kripke frame $\mathbb{F}$ can be seen as a multigraph $\mathcal{S}_\mathbb{F}$, with all edges with weight 1. In this case, for all formulas φ of graded modal logic and for all $w \in \mathbb{F}$, we have $\mathbb{F}, w \Vdash \varphi$ iff $\mathcal{S}_\mathbb{F}, w \Vdash \varphi_\lambda$, where φ_λ is the formula obtained by replacing each occurrence of $\Diamond^k$ by $\heartsuit_\lambda$ and each occurrence of $\Box^k$ by $\heartsuit_{\bar{\lambda}}$. Conversely, to each multigraph $\mathcal{S}$, we can associate a Kripke frame $\mathbb{F}_\mathcal{S} = (W, R)$ and a map $f : \mathbb{F}_\mathcal{S} \to \mathcal{S}$ in the following way. The set W is the set of triples $(s, i, t) \in \mathcal{S} \times \mathbb{N} \times \mathcal{S}$ such that there is an edge from s to t with weight $> i$. The map f sends a triple (s, i, t) to the point t. Then, for all graded modal formulas φ and all $w \in \mathbb{F}_\mathcal{S}$, we have $\mathbb{F}_\mathcal{S}, w \Vdash \varphi$ iff $\mathcal{S}, f(w) \Vdash \varphi_\lambda$.

(3) The standard modalities of neighborhood modal logic can be obtained as predicate liftings for the monotone neighborhood functor M. For example, for the universal modality, we can define a unary predicate lifting λ such that for all sets S and subsets U of S, we have $\lambda_S(U) = \{N \in M(S) \mid U \in N\}$. It is then possible to show that $\mathcal{S}, s \Vdash \heartsuit_\lambda \varphi$ iff $[\![\varphi]\!]_\mathcal{S}$ belongs to $M(s)$. In case the neighborhood model happens to be a topological model, this coincides with the usual interpretation of the universal modality on topological models.

(4) Probabilistic modalities can also be viewed as predicate liftings for the distribution functors. For all p in the closed interval $[0,1]$, we introduce a unary predicate lifting λ^p such that for all sets S and subsets U of S, we have $\lambda^p_S(U) = \{g : S \to [0,1] \mid \sum_{u \in U} g(u) \geq p\}$. With such a definition of λ^p, we have that $\mathcal{S}, s \Vdash_V \heartsuit^p_\lambda \varphi$ holds iff the probability that s has a successor satisfying φ is at least p.

(5) We can encode proposition letters from a set *Prop* by using predicate liftings for the functor $\mathcal{P}(Prop \times F)$. Given a proposition letter p in *Prop*, we define a 0-ary predicate lifting λ^p such that for all sets S, we have $\lambda^p_S = \{(P, \eta) \in \mathcal{P}(Prop) \times F(S) \mid p \in P\}$. If $\mathcal{S} = (S, (\sigma_1, \sigma_2))$ is a coalgebra for the functor $\mathcal{P}(Prop \times F)$ (where σ_1 is a map from S to $\mathcal{P}(Prop)$ and σ_2 is a map from S to FS), then we have $[\![\heartsuit_{\lambda^p}]\!]_\mathcal{S} = \{s \in S \mid p \in \sigma_1(s)\}$ and $[\![\heartsuit_{\overline{\lambda^p}}]\!]_\mathcal{S} = \{s \in S \mid p \notin \sigma_1(s)\}$. So instead of using the notation $\heartsuit_{\lambda^p}$, we simply write p. Similarly, we write $\neg p$ instead of $\heartsuit_{\overline{\lambda^p}}$.

In case we are dealing with a language containing proposition letters, these are supposed to be encoded using appropriate predicate liftings, as in the last example. Observe that by convention, the set Λ is closed under taking Boolean duals. So if a language ML_Λ contains a proposition letter, it also contains its negation.

Just like for modal logic, we can extend coalgebraic logic by enriching the language with fixpoint operators. Recall that we assume the predicate liftings in Λ to be monotone, which is crucial for the definition of the coalgebraic μ-calculus.

Coalgebraic μ-calculus We fix a set *Var* of variables, and define the set $\mu\mathrm{ML}_\Lambda$ of *coalgebraic fixpoint* Λ*-formulas* φ as follows:

$$\varphi ::= x \mid \bot \mid \top \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \heartsuit_\lambda(\varphi, \dots, \varphi) \mid \mu x.\varphi \mid \nu x.\varphi$$

where x belongs to *Var* $\lambda \in \Lambda$ and the arity of the operator $\heartsuit_\lambda$ is n if λ is a n-ary predicate.

Just like coalgebraic formulas, the interpretation of a coalgebraic fixpoint formula on a given coalgebra is a subset of the carrier set of the coalgebra. We define the following semantics for $\mu\mathrm{ML}_\Lambda$.

Let $\mathcal{S} = (S, \sigma)$ be a F-coalgebra. Given an assignment $\tau : Var \to \mathcal{P}(S)$, we define the *meaning* $[\![\varphi]\!]_{\mathcal{S},\tau}$ of a formula φ by induction. The induction is the same as the one for the semantics for coalgebraic logic, with the extra clauses:

$$
\begin{aligned}
[\![x]\!]_{\mathcal{S},\tau} &= \tau(x),\\
[\![\mu x.\varphi]\!]_{\mathcal{S},\tau} &= \bigcap\{U \subseteq S \mid U \subseteq [\![\varphi]\!]_{\mathcal{S},\tau[x\mapsto U]}\},\\
[\![\nu x.\varphi]\!]_{\mathcal{S},\tau} &= \bigcup\{U \subseteq S \mid [\![\varphi]\!]_{\mathcal{S},\tau[x\mapsto U]} \subseteq U\},
\end{aligned}
$$

where x belongs to Var, λ belongs to Λ and $\tau[x \mapsto U]$ is the valuation τ' such that $\tau'(x) = U$ and $\tau'(y) = \tau(y)$, for all variables $y \neq x$. As in the case of μ-calculus, it follows from the Knaster-Tarski theorem (see Section 2.2) and the fact that each $\lambda \in \Lambda$ is monotone, that the set $[\![\mu x.\varphi]\!]_{\mathcal{S},\tau}$ is nothing but the least fixpoint of the monotone map $\varphi_x^{\mathcal{S},\tau} : \mathcal{P}(S) \to \mathcal{P}(S)$ such that $\varphi_x^{\mathcal{S},\tau}(U) = [\![\varphi]\!]_{\mathcal{S},\tau[x\mapsto U]}$, for all subsets U of V. Similarly, $[\![\nu x.\varphi]\!]_{\mathcal{S},\tau}$ is the greatest fixpoint of the map $\varphi_x^{\mathcal{S},\tau}$.

As usual, we write $\mathcal{S}, s \Vdash_\tau \varphi$ if s belongs to $[\![\varphi]\!]_{\mathcal{S},\tau}$. A sentence φ is *satisfiable in a coalgebra* $\mathcal{S}$ if there is a state s in the carrier set of $\mathcal{S}$ such that $\mathcal{S}, s \Vdash \varphi$.

A *coalgebraic fixpoint sentence* is a coalgebraic fixpoint formulas such that all its variables are bound by a fixpoint operator. A coalgebraic fixpoint sentence φ is *satisfiable* if there exists a pointed coalgebra $(\mathcal{S}, s)$ such that $\mathcal{S}, s \Vdash \varphi$.

The *subformulas*, the *alternation depth* and the *closure* $Cl(\varphi)$ of a coalgebraic fixpoint formula φ are defined similarly to the case of the μ-calculus. To extend the notion of size of a μ-formula, we assume that the set Λ is equipped with a size measure $size : \Lambda \to \mathbb{N}$ assigning a size to each symbol in Λ. We assume that if the arity of a predicate lifting λ is n, then $size(\lambda) \geq n$.

For each coalgebraic fixpoint formula φ, we define the *weight* of φ, notation: $w(\varphi)$, such that if φ is a formula of the form $\heartsuit_\lambda(\varphi_n, \dots, \varphi_n)$, $w(\varphi) := size(\lambda)$ and $w(\varphi) := 1$, otherwise. The *size* of a coalgebraic fixpoint sentence φ is equal to $\Sigma_{\psi \in Cl(\varphi)} w(\psi)$. We denote it by $size(\varphi)$. Finally, if Γ is a set coalgebraic fixpoint formulas, we write $size(\Gamma)$ for $\Sigma_{\varphi\in\Gamma} size(\varphi)$.

Before we can turn to the definition of our automata we need some preliminary notions.

Transition conditions and one-step semantics Given a finite set Q, we define the set $TC^n_\Lambda(Q)$ of *(normalized) transition conditions* as the set of formulas φ given by:

$$
\begin{aligned}
\psi &::= \top \mid \bot \mid \mid \heartsuit_\lambda(q_1, \dots, q_n) \mid \psi \wedge \psi,\\
\varphi &::= \psi \mid \varphi \vee \varphi,
\end{aligned}
$$

where $\lambda \in \Lambda$ and $q_1, \dots, q_n \in Q$. A (normalized) transition condition is a disjunction of conjunctions of formulas of the form $\heartsuit_\lambda(q_1, \dots, q_n)$.

Given a set S and an assignment $\tau : Q \to \mathcal{P}(S)$, we define a map $[\![-]\!]^1_\tau :$ $TC^n_\Lambda(Q) \to \mathcal{P}(FS)$ interpreting formulas in $TC^n_\Lambda(Q)$ as subsets of S. The map is defined by induction by

$$\begin{aligned}
[\![\top]\!]^1_\tau &= FS,\\
[\![\bot]\!]^1_\tau &= \emptyset,\\
[\![\varphi \vee \psi]\!]^1_\tau &= [\![\varphi]\!]^1_\tau \cup [\![\psi]\!]^1_\tau,\\
[\![\varphi \wedge \psi]\!]^1_\tau &= [\![\varphi]\!]^1_\tau \cap [\![\psi]\!]^1_\tau,\\
[\![\heartsuit_\lambda(q_1,\dots,q_n)]\!]^1_\tau &= \lambda_S(\tau(q_1),\dots,\tau(q_n)),
\end{aligned}$$

where λ belongs to Λ and $q_1,\dots,q_n \in Q$. We write $FS,\eta \Vdash^1_\tau \varphi$ to indicate $\eta \in [\![\varphi]\!]^1_\tau$, and refer to this relation as the *one-step semantics*. Observe that if $\mathcal{S} = (S,\sigma)$ is a coalgebra, then

$$\mathcal{S}, s \Vdash_\tau \heartsuit_\lambda(q_1,\dots,q_n) \quad \text{iff} \quad \sigma(s) \in [\![\heartsuit_\lambda(q_1,\dots,q_n)]\!]^1_\tau.$$

We are now ready for the definition of the key structures of this chapter, viz., Λ-automata, and their semantics.

Λ-automata A Λ-*automaton* $\mathbb{A}$ is a quadruple $\mathbb{A} = (Q, q_I, \delta, \Omega)$, where Q is a finite set of states, $q_I \in Q$ is the initial state, $\delta : Q \longrightarrow TC^n_\Lambda(Q)$ is the transition map, and $\Omega : Q \longrightarrow \mathbb{N}$ is a parity map. The *size* of $\mathbb{A}$ is defined as its number of states, and its *index* as the size of the range of Ω.

Acceptance game Let $\mathcal{S} = (S,\sigma)$ be a F-coalgebra and let $\mathbb{A} = (Q, q_I, \delta, \Omega)$ be a Λ-automaton. The associated *acceptance game* $\mathcal{A}(\mathcal{S},\mathbb{A})$ is the parity game given by the table below.

Position	Player	Admissible moves	Priority
$(s,q) \in S \times Q$	$\exists$	$\{\tau : Q \longrightarrow \mathcal{P}(S) \mid FS, \sigma(s) \Vdash^1_\tau \delta(q)\}$	$\Omega(q)$
$\tau \in \mathcal{P}(S)^Q$	$\forall$	$\{(s',q') \mid s' \in \tau(q')\}$	0

A pointed coalgebra $(\mathcal{S}, s_0)$ is *accepted* by the automaton $\mathbb{A}$ if the pair (s_0, q_I) is a winning position for player $\exists$ in $\mathcal{A}(\mathcal{S},\mathbb{A})$.

The acceptance game of Λ-automata proceeds in rounds, moving from one basic position in $S \times Q$ to another. In each round, at position (s,q) first $\exists$ picks an assignment τ that makes the depth-one formula $\delta(q)$ true at $\sigma(s)$. Looking at this $\tau : Q \longrightarrow \mathcal{P}(S)$ as a binary relation $\{(s',q') \mid s' \in \tau(q')\}$ between S and Q, $\forall$ closes the round by picking an element of this relation.

This game is a natural generalization of the acceptance game for normalized alternating μ-automata presented in Section 2.4. So it does not come as a surprise that Λ-automata are the counterpart of the coalgebraic μ-calculus associated with Λ. As a formalization of this we give the following proposition the proof of which is an immediate adaptation of the proofs of Theorem 6.5 and Lemma 6.7 in [VW07]. Here we say that a Λ-automaton $\mathbb{A}$ is *equivalent* to a sentence $\varphi \in \mu\mathrm{ML}_\Lambda$ if for all pointed F-coalgebra $(\mathcal{S}, s)$, $(\mathcal{S}, s)$ is accepted by $\mathbb{A}$ iff $\mathcal{S}, s \Vdash \varphi$.

7.2.2. Proposition. *There is a procedure transforming a sentence φ in $\mu\mathrm{ML}_\Lambda$ into an equivalent Λ-automaton $\mathbb{A}_\varphi$ of size dn and index d, where n is the size and d is the alternation depth of φ. Moreover, the procedure is exponential time computable in the size of φ.*

7.3 Finite model property

In this section we show that $\mu\mathrm{ML}_\Lambda$ has the finite model property. That is, if a sentence φ in $\mu\mathrm{ML}_\Lambda$ is satisfiable in a coalgebra, then it is satisfiable in a finite coalgebra. The key tool in our proof is a satisfiability game that characterizes whether the class of pointed coalgebras accepted by a given Λ-automaton, is empty or not.

Traces Let Q be a finite set and Ω a map from Q to $\mathbb{N}$. Given a relation $R \subseteq Q \times Q$, we denote by $Ran(R)$ the range of R which is the set $\{q \in Q \mid \text{for some } q' \in Q, (q', q) \in R\}$. A finite sequence $q_0 \ldots q_{k+1}$ in Q^* is a *trace* through a sequence $R_0 \ldots R_k$ in $(\mathcal{P}(Q \times Q))^*$ if for all $i \in \{0, \ldots k\}$, (q_i, q_{i+1}) belongs to R_i. We denote by $Tr(R_0 \ldots R_k)$ the set of traces through $R_0 \ldots R_k$. A sequence $q_0 q_1 \ldots$ in Q^ω is a *trace* through a sequence $R_0 R_1 \ldots$ in $(\mathcal{P}(Q \times Q))^\omega$ if for all $i \in \mathbb{N}$, $(q_i, q_{i+1}) \in R_i$. We write $Tr(R_0 R_1 \ldots)$ for the set of traces through $R_0 R_1 \ldots$.

A sequence $R_0 R_1 \ldots$ in $(\mathcal{P}(Q \times Q))^\omega$ contains a *bad trace* if there exists a trace $q_0 q_1 \ldots$ through $R_0 R_1 \ldots$ such that the maximum of the set $\{\Omega(q) \mid q \in Inf(q_0 q_1 \ldots)\}$ is odd. We denote by $NBT(Q, \Omega)$ the set of sequences $R_0 R_1 \ldots$ in $(\mathcal{P}(Q \times Q))^\omega$ that do not contain any bad trace.

We are now going to define the satisfiability game for an automaton $\mathbb{A} = (Q, q_I, \delta, \Omega)$. The satisfiability game is an initialized graph game. We want this game to be such that $\exists$ has a winning strategy in the satisfiability game iff there is a coalgebra $\mathcal{S} = (S, \sigma)$ that is accepted by $\mathbb{A}$. The idea behind the satisfiability game is to make a simultaneous projection of all the acceptance matches. So intuitively, a basic position of the satisfiability games should be a subset Q' of Q. We may associate with such a macro-state Q' a point $s \in S$ such that $\exists$ has to deal with positions (s, q) in the acceptance game, for all $q \in Q'$.

For each $t \in S$ and for each $q \in Q'$, we can define the set Q^q_t as the collection of states $q' \in Q$ such that (t, q') is a possible basic position in the acceptance game following the basic position (s, q). Since Q' is a macro-state, we define Q_t as $\bigcup\{Q^q_t \mid q \in Q'\}$. Hence, each such a set is a potential next combination of states in Q that $\exists$ has to be able to handle simultaneously. The total collection of those sets is $\{Q_t \mid t \in S\}$; this is $\exists$'s move in the satisfiability game. 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 full match corresponds to a sequence $Q'_0 Q'_1 \ldots$ of basic positions, which are subsets of A. Intuitively, Q'_i contains all the states that occur inside the i-th basic position of some acceptance match. Now we have to say whether the match $Q'_0Q'_1\dots$ is won by $\exists$. Naively, we could say that $\exists$ wins iff there is no bad trace $q_0q_1\dots$ such that for all $i \in \mathbb{N}$, $q_i \in Q'_i$. However, if there is such a bad trace $q_0q_1\dots$, this would only be a problem if $q_0q_1\dots$ actually corresponds to an acceptance match. Given the way we defined the game in the last few paragraphs, we only know that each q_i occurs in *some* acceptance match, but there is no possibility to guess whether $q_0q_1\dots$ corresponds to an acceptance match.

A solution to avoid this problem is to replace the subset Q' by a relation $R \subseteq (Q \times Q)$. The range of R would play the same role as Q' (that is, the formula $\bigwedge\{\delta(q') \mid q' \in Ran(R)\}$ is true at the point corresponding to R). Moreover, for each $q' \in Ran(R)$, the set $\{q \in Q \mid (q, q') \in R\}$ is the set of states q in Q such that there is an acceptance match that moves (after one round) from q to q'. This helps us to remember which traces are relevant, when defining the winning condition.

The satisfiability game Let (Q, q_I, δ, Ω) be a Λ-automaton. The satisfiability game $Sat(\mathbb{A})$ for $\mathbb{A}$ is an initialized graph game. The game is played following the rules given by the tableau below, where for an element $q \in Q$ and for a collection $\mathcal{R} \subseteq \mathcal{P}(Q \times Q)$, $\rho_q : Q \to Q \times Q$ is the substitution given by

$$\rho_q : q' \mapsto (q, q')$$

and $\upsilon_{\mathcal{R}} : Q \times Q \to \mathcal{P}(\mathcal{R})$ denotes the valuation given by

$$\upsilon_{\mathcal{R}} : (q, q') \mapsto \{R \in \mathcal{R} \mid (q, q') \in R\}.$$

Position	Player	Admissible moves
$R \subseteq Q \times Q$	$\exists$	$\{\mathcal{R} \subseteq \mathcal{P}(Q \times Q) \mid [\![\bigwedge\{\rho_q\delta(q) \mid q \in Ran(R)\}]\!]^1_{\upsilon_{\mathcal{R}}} \neq \emptyset\}$
$\mathcal{R} \subseteq \mathcal{P}(Q \times Q)$	$\forall$	$\{R \mid R \in \mathcal{R}\}$

The starting position is $\{(q_I, q_I)\}$. Concerning the winning conditions, finite matches are lost by the player that gets stuck. An infinite match is won by $\exists$ if the unique sequence of relations determined by the match does not contain any bad trace.

We start by showing that the satisfiability game is a game the strategies of which may assumed to have finite memory.

7.3.1. PROPOSITION. *Let $\mathbb{A}$ be a Λ-automaton. If $\exists$ has a winning strategy in the game $Sat(\mathbb{A})$, then $\exists$ has a finite memory strategy. The size of the memory is at most exponential in the size of $\mathbb{A}$.*

Proof Let (Q, q_I, δ, Ω) be a Λ-automaton. By Proposition 7.1.2, it suffices to show that the satisfiability game for the automaton $\mathbb{A}$ is an initialized regular game. Abbreviate $G := \mathcal{P}(Q \times Q) \cup \mathcal{PP}(Q \times Q)$. So we have to find a finite alphabet Σ, a coloring $col : G \longrightarrow \Sigma$ and an ω-regular language L over Σ, such that for all infinite sequences $z_0z_1\dots$ in G^ω, $col(z_0)col(z_1)\dots$ belongs to L iff for all $i \in \mathbb{N}$, z_{2i} belongs $\mathcal{P}(Q \times Q)$, z_{2i+1} belongs to $\mathcal{PP}(Q \times Q)$ and $z_0z_2z_4\dots$ does not contain any bad trace.

We let Σ be the set $\mathcal{P}(Q \times Q) \cup \{*\}$. We define col such that for all $R \in \mathcal{P}(Q \times Q)$, $col(R) = R$ and for all $\mathcal{R} \subseteq \mathcal{P}(Q \times Q)$, $col(\mathcal{R}) = *$. We define L as the set $\{R_0 * R_1 * \cdots \mid R_0R_1 \in NBT(Q, \Omega)\}$.

We only show that L is ω-regular, which is the most difficult part of the proof. It is sufficient to prove that the complement of L is ω-regular. That is, we have to find a non-deterministic parity ω-automaton $\mathbb{B}$ that accepts exactly those sequences in Σ^ω, which do not belong to L. We define a non-deterministic ω-automaton $\mathbb{B} = (Q_B, q_b, \delta_B, \Omega_B)$ as follows. The set Q_B is equal to $Q \cup \{q_* \mid q \in Q\} \cup \{q_\top\}$. The initial state q_B is q_I. The transition map $\delta_B : Q_B \times \Sigma \longrightarrow \mathcal{P}(Q_B)$ is given by putting

$$
\begin{array}{ll}
\delta(q, R) & = \{q'_* \mid (q, q') \in R\}, \\
\delta(q_*, *) & = \{q\}, \\
\delta(q, *) = \delta(q_*, *) = \delta(q_*, R) & = \{q_\top\}, \\
\delta(q_\top, *) = \delta(q_\top, R) & = \{q_\top\},
\end{array}
$$

for all $q \in Q$ and $R \in \mathcal{P}(Q \times Q)$. Finally the parity map Ω_B is defined by $\Omega(q_\top) := 0$ and $\Omega_B(q) = \Omega_B(q_*) := \Omega(q) + 1$, for all $q \in Q$.

The intuition is that as soon as we see that a word over Σ is not of the form $R_0 * R_1 * \dots$ (where for all $i \in \mathbb{N}$, $R_i \subseteq (Q \times Q)$), we move to the state $q_\top$ and the word is accepted. If the word is of the form $R_0 * R_1 * \dots$, in each match of the parity game corresponding to $\mathbb{B}$, $\forall$ constructs a trace trough $R_0R_1\dots$ and the match is winning for $\exists$ if the trace is bad.

Clearly the size of $\mathbb{B}$ is linear the size of $\mathbb{A}$. By Theorem 2.4.1, there is a deterministic parity ω-automaton the size of which is exponential in the size of $\mathbb{A}$ and which recognizes L. By Proposition 7.1.2, we can conclude that the size of the memory is at most exponential in the size of $\mathbb{A}$.

7.3.2. THEOREM. *Let $\mathbb{A} = (Q, q_I, \delta, \Omega)$ be a Λ-automaton. The following are equivalent.*

1. *$L(\mathbb{A})$ is not empty.*
2. *$\exists$ has a winning strategy in the satisfiability game associated with $\mathbb{A}$.*
3. *$L(\mathbb{A})$ contains a finite pointed coalgebra of size at most exponential in the size of $\mathbb{A}$.*

Proof We first prove that (1) implies (2). Let $\mathcal{S} = (S, \sigma, s_0)$ be a pointed coalgebra accepted by $\mathbb{A}$. We show that $\exists$ has a winning strategy in the satisfiability game $Sat(\mathbb{A})$.

Before we go into the details of this definition, we need some terminology and notation. By assumption, player $\exists$ has a winning strategy f in the acceptance game for the automaton $\mathbb{A}$ with starting position (s_0, q_I). Since the acceptance game is a parity game, we may assume this strategy to be positional. Given two finite sequences $\vec{s} = s_0 \dots s_k \in S^*$ and $\vec{q} = q_0 \dots q_k \in Q^*$, we say that $\vec{q}$ *f-corresponds to* $\vec{s}$ if there is an f-conform partial match which has basic positions $(s_0, q_0) \dots (s_k, q_k)$. The set of all sequences in Q^* that f-correspond to $\vec{s}$ is denoted as $Corr_f(\vec{s})$. Intuitively, this set represents the collection of all f-conform matches passing through $\vec{s}$.

The definition of the winning strategy for $\exists$ in the satisfiability game $Sat(\mathbb{A})$ is given by induction on the length of partial matches. Simultaneously we will select, through the coalgebra $\mathcal{S}$, a path $s_1 s_2 \dots$, which is related to the $Sat(\mathbb{A})$-match π as follows: At each finite stage $R_0 \mathcal{R}_0 R_1 \dots R_k$ of π, $R_0 = R_I$,

$$\begin{array}{l} Tr(R_1 \dots R_k) \subseteq Corr_f(s_0 \dots s_k) \\ \text{and each } q \in Ran(R_k) \text{ occurs in some trace through } R_0 \dots R_k. \end{array} \qquad (*)$$

This implies in particular that for each element $q \in Ran(R_k)$, the pair (q, s_k) is a winning position for $\exists$ in the acceptance game.

First, we check that when the satisfiability game starts, condition (*) is satisfied. In this case, we have $R_0 = \{(q_I, q_I)\}$ and $\vec{s}$ is the one element sequence s_0. It is immediate that (*) holds.

For the induction step, assume that in the satisfiability game, the partial match $\vec{R} = R_0 \mathcal{R}_0 R_1 \dots R_k$ has been played. First we will provide $\exists$ with an appropriate response $\mathcal{R}_k \subseteq \mathcal{P}(Q \times Q)$.

Inductively, we have selected a sequence $\vec{s} = s_0 s_1 \dots s_k$ satisfying condition (*). Since f is by assumption a winning strategy for $\exists$ in the acceptance game, the pair (q, s_k) is a winning position for $\exists$ for each $q \in Ran(R_k)$. This means that $\exists$'s strategy f will provide her with a collection of valuations $\{\tau_q : Q \longrightarrow \mathcal{P}(S) \mid q \in Ran(R_k)\}$ such that

$$FS, \sigma s_k \Vdash^1_{\tau_q} \delta(q). \qquad (7.1)$$

for all $q \in Ran(R_k)$. The collection $\{\tau_q \mid q \in Ran(R_k)\}$ induces a map g_τ from S to $\mathcal{P}(Q \times Q)$ such that for all s in S, we have

$$g_\tau(s) = \{(q, q') \in Q \times Q \mid q \in Ran(R_k) \text{ and } s \in \tau_q(q')\}.$$

Intuitively, $g_\tau(s)$ contains all the pairs (q, q') such that in some f-conform match, we move from the basic position (q, s_k) to the basic position (q', s).

Next, we define $\mathcal{R}_k$ as the image of S under g_τ. That is, $\mathcal{R}_k$ is equal to $g_\tau[S]$. Thus we may and will see g_τ as a surjective map from S to $\mathcal{R}_k$.

1. Claim. $\mathcal{R}_k$ is a legitimate move for $\exists$ in $Sat(\mathbb{A})$ at position R_k.

Proof of Claim To see this, first observe that Fg_τ is a map with domain FS and codomain $F\mathcal{R}_k$. Thus, the object $(Ff_V)\sigma s_k$ is a member of the set $F\mathcal{R}_k$.

Now, in order to prove that $\exists$ may legitimately play $\mathcal{R}_k$ at R_k, it suffices to prove that for all $q \in Ran(R_k)$,

$$F\mathcal{R}_k, (Fg_\tau)\sigma s_k \Vdash^1_{\upsilon_{\mathcal{R}_k}} \rho_q\delta(q), \tag{7.2}$$

where $\upsilon_{\mathcal{R}_k}$ is defined as in the definition of the satisfiability game. Fix $q \in Ran(R_k)$, and abbreviate $\upsilon := \upsilon_{\mathcal{R}_k}$. Given (7.1), it clearly suffices to prove that

$$F\mathcal{R}_k, (Fg_\tau)\sigma s_k \Vdash^1_\upsilon \rho_q\varphi \quad \text{iff} \quad FS, \sigma s_k \Vdash^1_{\tau_q} \varphi \tag{7.3}$$

for all formulas φ in $TC^n_\Lambda(Q)$. We will prove (7.3) by induction on the complexity of φ.

In the base case we are dealing with a formula $\varphi = \heartsuit_\lambda(q'_1, \ldots, q'_n)$. For simplicity however we confine ourselves to the (representative) special case where $n = 1$ and write $q' = q'_1$. In this setting, (7.3) follows from the following chain of equivalences:

$$\begin{array}{rll}
F\mathcal{R}_k, (Fg_\tau)\sigma s_k \Vdash^1_\upsilon \rho_q\varphi, & \iff F\mathcal{R}_k, (Fg_\tau)\sigma s_k \Vdash^1_\upsilon \heartsuit_\lambda(q, q') & \\
& & \text{(definition of } \rho_q \text{ and } \varphi) \\
& \iff (Fg_\tau)(\sigma s_k) \in \lambda_{\mathcal{R}_k}[\![(q, q')]\!]_\upsilon & \text{(definition of } \Vdash) \\
& \iff \sigma s_k \in (Fg_\tau)^{-1}(\lambda_{\mathcal{R}_k}[\![(q, q')]\!]_\upsilon) & \text{(definition of } (\cdot)^{-1}) \\
& \iff \sigma s_k \in \lambda_S g_\tau^{-1}([\![(q, q')]\!]_\upsilon) & (\dagger) \\
& \iff \sigma s_k \in \lambda_S([\![q']\!]_{\tau_q}) & (\ddagger) \\
& \iff FS, \sigma s_k \Vdash^1_{\tau_q} \heartsuit_\lambda q' & \text{(definition of } \Vdash)
\end{array}$$

Here the step marked (†) follows from λ being a natural transformation, which implies that the following diagram commutes:

$$\begin{array}{ccc}
\mathcal{Q}S & \xrightarrow{\lambda_S} & \mathcal{Q}FS \\
\uparrow g_\tau^{-1} & & \uparrow (Fg_\tau)^{-1} \\
\mathcal{Q}\mathcal{R}_k & \xrightarrow[\lambda_{\mathcal{R}_k}]{} & \mathcal{Q}F\mathcal{R}_k
\end{array}$$

The step marked (‡) follows from the identity $[\![q']\!]_{\tau_q} = g_\tau^{-1}([\![(q, q')]\!]_\upsilon)$, which in its turn follows from the following chain of equivalences, all applying to an arbitrary $s \in S$:

$$\begin{array}{rll}
s \in [\![q']\!]_{\tau_q} & \iff s \in \tau_q(q') & \text{(definition of } [\![\cdot]\!]) \\
& \iff (q, q') \in g_\tau(s) & \text{(definition of } g_\tau) \\
& \iff g_\tau(s) \in \upsilon(q, q') & \text{(definition of } \upsilon = \upsilon_{\mathcal{R}_k}) \\
& \iff g_\tau(s) \in [\![(q, q')]\!]_\upsilon & \text{(definition of } [\![\cdot]\!]) \\
& \iff s \in g_\tau^{-1}([\![(q, q')]\!]_\upsilon) &
\end{array}$$

Since the inductive steps in the proof of (7.3) are completely routine, and therefore, omitted, this finishes the proof of (7.3), and thus also the proof of the claim. ◀

Given the legitimacy of $\mathcal{R}_k$ as a move for $\exists$ at position R_k, we may propose it as her move in the satisfiability game. Note that this yields the definition of a strategy.

Playing this strategy enables $\exists$ to maintain the inductive condition (*). Indeed, by definition of $\mathcal{R}_k$, for every $R \in \mathcal{R}_k$ there is an $s_R \in S$ such that $R = g_\tau(s_R)$. Hence if $\forall$ picks such a relation R, that is putting $R_{k+1} := R$, $\exists$ adds the state s_R to her sequence $\vec{s}$, putting $s_{k+1} := s_R$.

To verify that the sequences $R_0 \ldots R_{k+1}$ and $s_0 \ldots s_{k+1}$ satisfy (*), let $q_0 \ldots q_{k+1}$ be a trace through $R_1 \ldots R_{k+1}$. Since $R_0 \ldots R_k$ and $s_0 \ldots s_k$ satisfy (*), there is an f-conform match of the form $(s_0, q_0) \ldots (s_k, q_k)$. In this match, when the position (s_k, q_k) is reached, on the basis of f, $\exists$ chooses a marking $\tau_{q_k} : Q \to \mathcal{P}(S)$ such that $\mathcal{S}, s_k \Vdash^1_{\tau_{q_k}} \delta(q_k)$. Then, $\forall$ may pick any pair (s, q) such that $s \in \tau_{q_k}(q)$. So in order to show that there is a partial f-conform match of the form $(s_0, q_0) \ldots (s_{k+1}, q_{k+1})$, it suffices to prove that $s_{k+1} \in \tau_{q_k}(q_{k+1})$. Recall that $(q_k, q_{k+1}) \in R_{k+1}$. Since $R_{k+1} = g_\tau(s_{k+1})$, q_{k+1} belongs to $Ran(R_k)$ and $s_{k+1} \in \tau_{q_k}(q_{k+1})$. This finishes the proof that the first part of (*) holds for $R_0 \ldots R_{k+1}$ and $s_0 \ldots s_{k+1}$.

It remains to show that for all $q \in Ran(R_{k+1})$, q occurs in a trace through $R_0 \ldots R_{k+1}$. Fix $q \in Ran(R_{k+1})$. So there exists $q_k \in Q$ such that (q_k, q) belongs to R_{k+1}. Since $R_{k+1} = g_\tau(s_{k+1})$ and by definition of g_τ, q_k belongs to $Ran(R_k)$. Moreover, it follows from the induction hypothesis that if $q \in Ran(R_k)$, there is a sequence $q_{-1} \ldots q_{k-1}$ such that $q_{-1} R_0 q_0 \ldots R_k q_k$. Putting this together with $(q_k, q) \in R_{k+1}$, this finishes to prove that q occurs in a trace through $R_0 \ldots R_{k+1}$.

Finally we show why this strategy is winning for $\exists$ in the game $Sat(\mathbb{A})$, initiated at $\{(q_I, q_I)\}$. Consider an arbitrary match of this game, where $\exists$ plays the strategy as defined above. First, suppose that this match is finite. It should be clear from our definition of $\exists$'s strategy in $Sat(\mathbb{A})$ that she never gets stuck. So if the match is finite, $\forall$ got stuck and $\exists$ wins.

In case the match is infinite, $\exists$ has constructed an infinite sequence $\vec{s} = s_0 s_1 s_2 \ldots$ corresponding to the infinite sequence $\vec{R} = R_0 R_1 R_2 \ldots$ induced by the $Sat(\mathbb{A})$-match. It is easy to see that since the relation (*) holds at each finite level, for every infinite trace $q_0 q_1 q_2 \ldots$ through $\vec{R}$, there is an f-conform infinite match of the acceptance game on $\mathcal{S}$ with basic positions $(s_0, q_0)(s_1, q_1) \ldots$ Since f was assumed to be a winning strategy, none of these traces is bad. In other words, the sequence $\vec{R}$ satisfies the winning condition of $Sat(\mathbb{A})$ for $\exists$, and thus she is declared to be the winner of the $Sat(\mathbb{A})$-match. Since we considered an arbitrary match in which she is playing the given strategy, this shows that this strategy is winning, and thus finishes the proof of the implication $(1 \Rightarrow 2)$.

We now turn to the implication ($2 \Rightarrow 3$), which is the hard part of the proof. Suppose that $\exists$ has a winning strategy f in the satisfiability game $(G_\exists, G_\forall, E, Win, \{(q_I, q_I)\})$ associated with $\mathbb{A}$. We need to show that $\mathbb{A}$ accepts some finite pointed coalgebra of size at most exponential in the size of $\mathbb{A}$.

By Proposition 7.3.1, we may assume $\exists$'s winning strategy to use finite memory only. This means that there is a finite set M, an element $m_I \in M$ and partial maps $f_1 : G_\exists \times M \longrightarrow G$ and $f_2 : G_\exists \times M \longrightarrow M$ satisfying the conditions of the definition of a finite memory strategy (see Section 7.1.2).

Moreover, the size of M is at most exponential in the size of $\mathbb{A}$. We can extend the partial maps f_1 and f_2 to maps with domain $G_\exists \times M$, by assigning dummy values to the elements that do not belongs to the domain. Without loss of generality, we may assume that for all $(\mathcal{R}, m) \in G_\forall \times M$, $f_2(\mathcal{R}, m) = m$. If this is not initially the case, we can replace f_2 by a map f_2' satisfying $f_2'(\mathcal{R}, m) = m$ and $f_2'(R, m) = f_2(f_1(R, m), f_2(R, m))$, for all $R \in G_\exists$, $\mathcal{R} \in G_\forall$ and $m \in M$.

We denote by $W_\exists$ the subset of $G_\exists \times M$ that contains exactly the pairs (R, m) satisfying the following condition. For all $Sat(\mathbb{A})$-matches $R_0\mathcal{R}_0R_1\mathcal{R}_1\dots$ and for all sequences $m_0m_1\dots$ such that

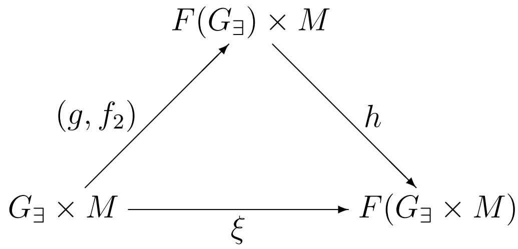
$$R_0 = R, m_0 = m_I \text{ and for all } i \in \mathbb{N},\ \mathcal{R}_i = f_1(R_i, m_i),\ m_{i+1} = f_2(R_i, m_i),$$

we have that $R_0\mathcal{R}_0R_1\mathcal{R}_1\dots$ is won by $\exists$. Such a pair (R, m) is said to be *winning* for $\exists$ with respect to f in the acceptance game.

The finite coalgebra in $L(\mathbb{A})$ that we are looking for will have the set $G_\exists \times M$ as its carrier. Therefore we first define a coalgebra map $\xi : G_\exists \times M \longrightarrow F(G_\exists \times M)$. We base this construction on two observations.

First, let (R, m) be an element of $W_\exists$ and write $\mathcal{R}$ for the set $f_1(R, m)$. It follows from the rules of the satisfiability game that there is an object $g(R, m) \in F\mathcal{R}$ such that for every $q \in Ran(R)$, the formula $\rho_q\delta(q)$ is true at $g(R, m)$ under the valuation $\upsilon_\mathcal{R}$. Note that $\mathcal{R} \subseteq G_\exists$, and thus we may think of the above as defining a function g with domain $W_\exists$ and codomain $FG_\exists$. Choosing some dummy values for elements $(R, m) \in (G_\exists \times M) \setminus W_\exists$, the domain of this function can be extended to the full set $G_\exists \times M$. To simplify our notation we will also let g denote the resulting map, with domain $G_\exists \times M$ and codomain $FG_\exists$. Second, consider the map $add_m : G_\exists \longrightarrow G_\exists \times M$, given by $add_m(R) = (R, m)$. Based on this map we define the function $h : FG_\exists \times M \longrightarrow F(G_\exists \times M)$ such that $h(\eta, m) = F(add_m)(\eta)$, for all $(\eta, m) \in FG_\exists \times M$.

We let $\mathcal{S}$ be the coalgebra $(G_\exists \times M, \xi)$, where $\xi : G_\exists \times M \longrightarrow F(G_\exists \times M)$ is the map $h \circ (g, f_2)$.



Observe that the size of $\mathcal{S}$ is at most exponential in the size of $\mathbb{A}$, since $G_\exists$ is the set $\mathcal{P}(Q \times Q)$ and M is at most exponential in the size of Q. As the designated point of $\mathcal{S}$ we take the pair (R_I, m_I), where $R_I := \{(q_I, q_I)\}$ and q_I is the initial state of the automaton $\mathbb{A}$.

It is left to prove that the resulting pointed coalgebra $(\mathcal{S}, (R_I, m_I))$ is accepted by $\mathbb{A}$. That is, using $\exists$'s winning strategy f in the satisfiability game we need to find a winning strategy for $\exists$ in the acceptance game for the automaton $\mathbb{A}$ with starting position $((R_I, m_I), q_I)$. We will define this strategy by induction on the length of a partial match, simultaneously setting up a shadow match of the satisfiability game. Inductively we maintain the following relation between the two matches: If $((R_0, m_0), q_0), \ldots, ((R_k, m_k), q_k)$ is a partial match of the acceptance game (during which $\exists$ plays the inductively defined strategy), then

(*) $q_I q_0 \ldots q_k$ is a trace through $R_0 \ldots R_k$,
for all $i \in \{0, \ldots, k-1\}$, $R_{i+1} \in f_1(R_i, m_i)$ and $m_{i+1} = f_2(R_i, m_i)$.

Setting up the induction, it is easy to see that the above condition is met at the start of the acceptance match. In this case, $((R_0, m_0), q_0)$ is equal to $((R_I, m_I), q_I)$ and $q_I q_I$ is the (unique) trace through the one element sequence R_I.

Inductively assume that, with $\exists$ playing as prescribed, the play of the acceptance game has reached position $((R_k, m_k), q_k)$. By the induction hypothesis, we have $q_k \in Ran(R_k)$ and the position (R_k, m_k) is a winning position for $\exists$ with respect to f in the satisfiability game. Abbreviate $\mathcal{R} := f_1(R_k, m_k)$ and $n := f_2(R_k, m_k)$. As the next move for $\exists$ we propose the valuation $\tau : Q \to \mathcal{P}(G_\exists \times M)$ such that $\tau(q) := \{(R, n) \mid (q_k, q) \in R \text{ and } R \in \mathcal{R}\}$, for all $q \in Q$.

2. Claim. *τ is a legitimate move at position $((R_k, m_k), q_k)$.*

Proof of Claim In order to prove the claim, we need to show that

$$\mathcal{S}, (R_k, m_k) \Vdash^1_\tau \delta(q_k) \tag{7.4}$$

For a proof of (7.4), recall that (R_k, m_k) is a winning position for $\exists$ in the satisfiability game. Hence, the element $\gamma := g(R_k, m_k)$ of $F\mathcal{R}$ satisfies the formula $\rho_{q_k}\delta(q_k)$ under the valuation $\upsilon := \upsilon_{\mathcal{R}}$ (where $\upsilon_{\mathcal{R}}$ is defined as in the definition of the satisfiability game). That is,

$$F\mathcal{R}, \gamma \Vdash^1_{\upsilon_{\mathcal{R}}} \rho_{q_k}\delta(q_k) \tag{7.5}$$

Thus in order to prove the claim it clearly suffices to prove that

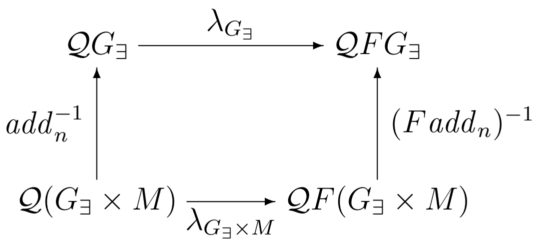
$$\mathcal{S}, (R_k, m_k) \Vdash^1_\tau \varphi \quad \text{iff} \quad F\mathcal{R}, \gamma \Vdash^1_\upsilon \rho_{q_k}\varphi \tag{7.6}$$

for all formulas φ in $\mathcal{L}_0(\Lambda(Q))$. The proof of (7.6) proceeds by induction on the complexity of φ.

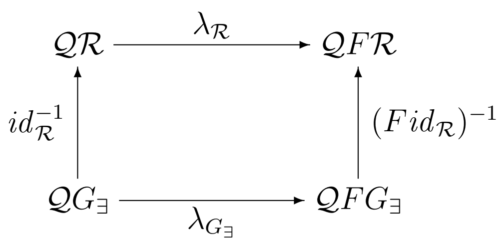
First consider formulas of the form $\heartsuit_\lambda(q'_1, \ldots, q'_l)$; in order to keep notation simple we assume $l = 1$ and write q' instead of q'_1. Then we can prove (7.6) as follows:

$$\begin{aligned}
\mathcal{S}, (R_k, m_k) \Vdash^1_\tau \heartsuit_\lambda q' &\iff \xi(R_k, m_k) \in \lambda_{G_\exists \times M}([\![q']\!]_\tau). && \text{(definition of } \Vdash) \\
&\iff (F\,add_n)(\gamma) \in \lambda_{G_\exists \times M}([\![q']\!]_\tau). && \text{(definition of } \xi) \\
&\iff \gamma \in (F\,add_n)^{-1}(\lambda_{G_\exists \times M}[\![q']\!]_\tau) && \text{(definition of } (\cdot)^{-1}) \\
&\iff \gamma \in \lambda_{G_\exists}(add_n^{-1}([\![q']\!]_\tau) && (\dagger) \\
&\iff \gamma \in \lambda_{\mathcal{R}}([\![(q_k, q')]\!]_\upsilon) && (\ddagger) \\
&\iff F\mathcal{R}, \gamma \Vdash^1_\upsilon \heartsuit_\lambda(q_k, q') && \text{(definition of } \Vdash) \\
&\iff F\mathcal{R}, \gamma \Vdash^1_\upsilon \rho_{q_k}\heartsuit_\lambda q' && \text{(definition of } \rho_{q_k})
\end{aligned}$$

Here ($\dagger$) follows from λ being a natural transformation:



For the remaining step ($\ddagger$), using again the fact that λ is a natural transformation, the following diagram commutes:



where $id_{\mathcal{R}}$ is the identity map with domain $\mathcal{R}$ and codomain $G_\exists$. Moreover, since we assume that F is standard, we have that $F id_{\mathcal{R}} = id_{\mathcal{R}}$. Putting these two observations together with the fact that $[\![q]\!]_\upsilon$ is a subset of $\mathcal{R}$ (which is itself a subset of $G_\exists$), we have $\lambda_{\mathcal{R}}[\![q]\!]_\upsilon = \lambda_{G_\exists}[\![q]\!]_\upsilon$.

But then ($\ddagger$) is immediate by the following claim:

$$add_n^{-1}([\![q']\!]_\tau) = [\![(q_k, q')]\!]_\upsilon, \tag{7.7}$$

which can be proved via the following chain of equivalences, which hold for an arbitrary $R \subseteq Q \times Q$:

$$\begin{aligned}
R \in [\![(q_k, q')]\!]_\upsilon &\iff R \in \upsilon(q_k, q') \\
&\iff R \in \mathcal{R} \text{ and } (q_k, q') \in R && \text{(definition of } \upsilon = \upsilon_{\mathcal{R}}) \\
&\iff (R, n) \in \tau(q') && \text{(definition of } \tau) \\
&\iff add_n(R) = (R, n) \in [\![q']\!]_\tau && \text{(definition of } [\![\cdot]\!]_\tau) \\
&\iff R \in add_n^{-1}([\![q']\!]_\tau) && \text{(definition of } (\cdot)^{-1})
\end{aligned}$$

Since the inductive steps of the proof of (7.6) are trivial, we omit the details. This finishes the proof of the Claim. ◀

Now that we showed that τ is a valid move for $\exists$, we verify that the induction hypothesis (*) remains true. In the acceptance game, after $\exists$ played the valuation $\tau : Q \to \mathcal{P}(G_\exists \times M)$, $\forall$ picks a pair $((R_{k+1}, m_{k+1}), q_{k+1})$ such that (R_{k+1}, m_{k+1}) belongs to $\tau(q_{k+1})$. By definition of τ, we have $(q_k, q_{k+1}) \in R_{k+1}$, $m_{k+1} = f_2(R_k, m_k)$ and $R_{k+1} \in f_1(R_k, m_k)$, which implies that the induction hypothesis holds.

To finish the proof of the implication $(2 \Rightarrow 3)$, it remains to show that the strategy defined for $\exists$ is winning in the acceptance game $\mathcal{A}(, \mathcal{S}, \mathbb{A})$ with starting position $((R_I, m_I), q_I)$. First, if we look at the strategy, we see that $\exists$ will never get stuck. So all the finite matches are won by $\exists$. So let $\pi = ((R_0, m_0), q_0)((R_1, m_1), q_1)\dots$ be an infinite match of the acceptance game during which $\exists$ played accordingly to the strategy defined above. It follows that $q_I q_0 q_1 \dots$ is a trace through $R_0 R_1 \dots$ and for all $i \in \mathbb{N}$, $R_{i+1} \in f_1(R_i, m_i)$ and $m_{i+1} = f_2(R_i, m_i)$. Since the pair (f_1, f_2) is a winning strategy for $\exists$, the satisfiability match $R_0 R_1 \dots$ is won by $\exists$, that is, $R_0 R_1 \dots$ does not contain any bad trace. In particular, $q_I q_0 q_1 \dots$ is not a bad trace, which means that π is won by $\exists$ and this finish the proof of implication $(2 \Rightarrow 3)$.

Since the implication $(3 \Rightarrow 1)$ is trivial, this finishes the proof of the theorem. □

Putting this theorem together with Proposition 7.2.2, we obtain a finite model property for the coalgebraic μ-calculus, for every set of predicate liftings.

7.3.3. Corollary. *If a sentence φ in $\mu\mathrm{ML}_\Lambda$ is satisfiable in a F-coalgebra, it is satisfiable in a F-coalgebra of size exponential in the size of φ.*

Moreover, given some mild condition on Λ and F, we obtain the following complexity result.

Exponential one-step satisfiability property Let F be a functor and Λ a set of predicate liftings for F. We say that the pair (F, Λ) has the *exponential one-step satisfiability property* if for all sets Q and S, assignments $\tau : Q \to \mathcal{P}(S)$ and formulas φ in $TC^n_\Lambda(Q)$, the relation $[\![\varphi]\!]^1_\tau \neq \emptyset$ is decidable in time exponential in the size of φ.

7.3.4. Theorem. *Let F be a functor and Λ a set of predicate liftings for F such that (F, Λ) has the exponential one-step satisfiability property. It is decidable in time doubly exponential in the size of φ, whether a fixpoint Λ-sentence φ is satisfiable in the class of F-coalgebras.*

Proof Fix a fixpoint Λ-sentence φ. By Proposition 7.2.2, we can compute in time exponential in the size of φ, a Λ-automata $\mathbb{A} = (Q, q_I, \delta, \Omega)$ in such that for all pointed coalgebras $(\mathcal{S}, s_0)$,

$$\mathcal{S}, s_0 \Vdash \varphi \quad \text{iff} \quad (\mathcal{S}, s_0) \text{ is accepted by } \mathbb{A}.$$

The automaton $\mathbb{A}$ is of size dn and index d, where n is the size of φ and d is the alternation depth of φ.

So to decide whether φ is satisfiable, it is sufficient to decide whether $\mathbb{A}$ accepts a pointed coalgebra. By Theorem 7.3.2, deciding whether $\mathbb{A}$ accepts a pointed coalgebra is equivalent to decide whether $\exists$ has a winning strategy in the game $Sat(\mathbb{A}) = (G_\exists, G_\forall, E, Win, \{(q_I, q_I)\})$. Recall that $Sat(\mathbb{A})$ is an initialized regular game. Moreover, its associated regular language is the complement of the language recognized by the automaton $\mathbb{B}$, as defined in Proposition 7.3.1.

First, we show that in time doubly exponential in the size of $\mathbb{A}$, we can compute the game $Sat(\mathbb{A})$. Since the set $G_\exists$ is equal to $\mathcal{P}(Q \times Q)$ and the set $G_\forall$ is equal to $\mathcal{P}\mathcal{P}(Q \times Q)$, each position of the board of $Sat(\mathbb{A})$ can be represented by a string exponential in the size of $\mathbb{A}$. Moreover, the size of G is doubly exponential in the size of $\mathbb{A}$.

Given a pair $(\mathcal{R}, R)$ in $\mathcal{P}\mathcal{P}(Q \times Q) \times \mathcal{P}(Q \times Q)$, we have that $(\mathcal{R}, R)$ belongs to E iff $R \in \mathcal{R}$. So it is decidable in time exponential in the size of $\mathbb{A}$ whether $(\mathcal{R}, R)$ belongs to E. Given a pair $(R, \mathcal{R})$ in $\mathcal{P}(Q \times Q) \times \mathcal{P}\mathcal{P}(Q \times Q)$, we have that $(R, \mathcal{R})$ belongs to E iff

$$[\![\bigwedge\{\rho_q \delta(q) \mid q \in Ran(R)\}]\!]^1_{\upsilon_\mathcal{R}} \neq \emptyset,$$

where $\upsilon_\mathcal{R}$ is defined as in the definition of the satisfiability game. Since $\mathbb{A}$ is computable in time exponential in the size of φ, the size of $\delta(q)$ (for each $q \in Q$) is at most exponential in the size of φ. Hence, the size of the formula $\bigwedge\{\rho_q \delta(q) \mid q \in Ran(R)\}$ is at most exponential in the size of φ. Putting this together with the fact that (F, Λ) has the exponential one-step satisfiability property, we get that it is decidable in time double exponential in the size of φ whether $(R, \mathcal{R})$ belongs to E. Moreover, as $E \subseteq G \times G$ and $G = \mathcal{P}(Q \times Q) \times \mathcal{P}\mathcal{P}(Q \times Q)$, the number of pairs $(R, \mathcal{R})$ for which we have to check whether it belongs to E, is doubly exponential in the size of $\mathbb{A}$.

The winning condition of $Sat(\mathbb{A})$ is completely determined by the coloring col and the automaton $\mathbb{B} = (Q_B, q_B, \delta_B, \Omega_B)$ (as defined in the proof of Proposition 7.3.1). We can compute the graph of col in time doubly exponential in the size of $\mathbb{A}$. Moreover, the automaton $\mathbb{B}$ is computable in time exponential in the size of $\mathbb{A}$. This finishes the proof that in time doubly exponential in the size of A, we can compute the game $Sat(\mathbb{A})$.

Finally we show that it is decidable in time doubly exponential in the size of A whether $\exists$ has a winning strategy in $Sat(\mathbb{A})$. We use Theorem 7.1.4. The size n of G is the size of the set $\mathcal{P}\mathcal{P}(Q \times Q) \cup \mathcal{P}(Q \times Q)$. So n is doubly exponential in the size of $\mathbb{A}$. The size of E is also doubly exponential in the size of $\mathbb{A}$. The size and the index of $\mathbb{B}$ are linear in the size of $\mathbb{A}$. Therefore, by Theorem 2.4.1, there exists a deterministic ω-automaton that recognizes the same language as $\mathbb{B}$, the size of which is exponential in the size of $\mathbb{A}$ and the index of which is linear in the size of $\mathbb{A}$. Putting everything together with Theorem 7.1.4, we obtain that it is decidable in time doubly exponential in the size of A whether $\exists$ has a winning strategy in $Sat(\mathbb{A})$. □

7.4 One-step tableau completeness

In this section we show how our satisfiability game relates to the work of Corina Cîrstea, Clemens Kupke and Dirk Pattinson [CKP09]. The authors of this paper proved that when assuming the existence of a certain derivation system satisfying some properties for Λ, the satisfiability problem for fixpoint coalgebraic logic in decidable in EXPTIME. In the previous section, we established a 2EXPTIME bound for the same problem, but without any assumption.

We show that using the same assumptions as in [CKP09], we can modify the satisfiability game from the previous section and obtain a new game: the tableau game. The two games are equivalent, in the sense that $\exists$ has a winning strategy in one of them iff she has a winning strategy in the other.

Moreover, it is clear from the definition of the tableau game that the tableau game is rather similar to the game defined by Corina Cîrstea et alii [CKP09]. The main difference is that our tableau game is defined with respect to a Λ-automaton, whereas their game is defined with respect to a coalgebraic fixpoint formula (recall from Proposition 7.2.2 that such a formula can be translated into an equivalent Λ-automaton). It is then not surprising that under the same assumptions as in [CKP09], we can derive from the tableau game an EXPTIME bound for the satisfiability problem.

We start by recalling the definitions of one step rule and one-step completeness, as introduced in [CKP09]. We also define the tableau game.

One-step rule Fix an infinite set X. We let $\Lambda(X)$ denote the set $\{\heartsuit_\lambda(x_1, \cdots, x_n) \mid \lambda \in \Lambda, x_i \in X\}$. A *(monotone) one-step rule* d for Λ is of the form

$$\frac{\Gamma_0}{\Delta_1 \cdots \Delta_n}$$

where Γ_0 is a finite subset of $\Lambda(X)$ and $\Delta_1, \ldots, \Delta_n$ are finite subsets of X, every propositional variable occurs at most once in Γ_0 and all variables occurring in some of the Δ_i's ($i > 0$) also occur in Γ_0. We write $\mathit{Conc}(\mathsf{d})$ for the set Γ_0 and $\mathit{Prem}(\mathsf{d})$ for the set $\{\Delta_i \mid 1 \leq i \leq n\}$.

Let φ be a formula in $TC^n_\Lambda(X)$. Recall that a formula in $TC^n_\Lambda(X)$ is a disjunction of conjunctions of formulas in $\Lambda(X)$. Hence, there exist subsets $\Gamma_1, \ldots, \Gamma_n$ of $\Lambda(X)$ such that

$$\varphi = \bigvee\{\bigwedge \Gamma_i \mid 1 \leq i \leq n\}$$

Given a finite subset Γ of $\Lambda(X)$, we say that Γ *strongly entails* φ, notation: $\Gamma \vdash_s \varphi$, if there exists $i_0 \in \{1, \ldots, n\}$ such that $\Gamma_{i_0} \subseteq \Gamma$.

For a set of such rules, with an automaton $\mathbb{A}$ we associate a so-called tableau game, in which the rules themselves are part of the game board.

Tableau game Let (Q, q_I, δ, Ω) be a Λ-automaton, let $\Lambda_{\mathbb{A}}$ be the set of predicate liftings $\lambda \in \Lambda$ such that λ or its dual occurs in the range of δ and let D be a set of one-step rules for Λ. The game $\mathit{Tab}(\mathbb{A}, \mathsf{D})$ is the initialized graph game given by the table below.

Position	Player	Admissible moves
$R \in \mathcal{P}(Q \times Q)$	$\exists$	$\{\Gamma \subseteq \Lambda_{\mathbb{A}}(Q \times Q) \mid (\forall q \in \mathit{Ran}(R))(\Gamma \vdash_s \rho_q\delta(q)\}$
$\Gamma \subseteq \Lambda_{\mathbb{A}}(Q \times Q)$	$\forall$	$\left\{(\mathsf{d}, \theta) \in \mathsf{D} \times (Q \times Q)^X \mid \theta[\mathit{Conc}(\mathsf{d})] \subseteq \Gamma\right\}$
$(\mathsf{d}, \theta) \in \mathsf{D} \times (Q \times Q)^X$	$\exists$	$\{\theta[\Delta] \mid \Delta \in \mathit{Prem}(\mathsf{d})\}$

The starting position is $\{(q_I, q_I)\}$. An infinite match $R_0\Gamma_0(\mathsf{d}_0, \theta_0)R_1\Gamma_1(\mathsf{d}_1, \theta_1)\ldots$ is won by $\exists$ if $R_0R_1\ldots$ belongs to $\mathit{NBT}(Q, \Omega)$.

Our tableau game $\mathit{Tab}(\mathbb{A}, \mathsf{D})$ is (in some natural sense) equivalent to the satisfiability game for $\mathbb{A}$ if we assume the set D to be one-step sound and complete with respect to F.

One-step completeness A set D of one-step rules is *one-step sound and complete* with respect to Λ if for all sets Y and S, all finite subset Γ of $\Lambda(Y)$ and all assignments $\tau : Y \to \mathcal{P}(S)$ the following are equivalent:
(a) $[\![\bigwedge \Gamma]\!]^1_V \neq \emptyset$
(b) for all rules $\mathsf{d} \in \mathsf{D}$ and all substitutions $\theta : X \to Y$ with $\theta[\mathit{Conc}(\mathsf{d})] \subseteq \Gamma$, there exists $\Delta_i \in \mathit{Prem}(\mathsf{d})$ such that $[\![\bigwedge \theta[\Delta_i]]\!]^1_V \neq \emptyset$.

Intuitively, one-step soundness and completeness means that the conclusion of an instance of a rule is satisfiable iff one of the premisses of the instance of the rule is satisfiable. Note that the implication (a) $\Rightarrow$ (b) corresponds to the completeness of D and the implication (b) $\Rightarrow$ (a) corresponds to the soundness of D.

7.4.1. Example. (1) For all $n \in \mathbb{N}$, let d_n be the following one-step rule:

$$\frac{\{\Diamond x_0, \Box x_1, \dots, \Box x_n\}}{\{x_0, \dots, x_n\}}.$$

The set $\{\mathsf{d}_n \mid n \in \mathbb{N}\}$ is one-step sound and complete with respect to $\{\lambda, \overline{\lambda}\}$, where λ is the predicate lifting corresponding the modality $\Box$ (as defined in Example 7.2.1(1)).

(2) The d defined by

$$\frac{\{\Diamond x, \Box y\}}{\{x, y\}},$$

is one-step sound and complete with respect to o $\{\lambda, \overline{\lambda}\}$, where λ is the predicate lifting corresponding the universal modality for neighborhood modal logic (as defined in Example 7.2.1(3)).

7.4.2. Theorem. *Let $\mathbb{A} = (Q, q_I, \delta, \Omega)$ be a Λ-automaton and let D be a set of one-step rules for Λ. If D is one-step sound and complete with respect to F, then $\exists$ has a winning strategy in $Sat(\mathbb{A})$ iff $\exists$ has a winning strategy in $Tab(\mathbb{A}, \mathsf{D})$.*

Proof For the direction from left to right, suppose $\exists$ has a winning strategy f in $Sat(\mathbb{A})$. We define a winning strategy g for $\exists$ in $Tab(\mathbb{A}, \mathsf{D})$. The idea is that during a g-conform match $R_0\Gamma_0(\mathsf{d}_0, \theta_0) \dots R_{k-1}\Gamma_{k-1}(\mathsf{d}_{k-1}, \theta_{k-1})R_k$, $\exists$ will maintain an f-conform shadow match $R'_0\mathcal{R}_0 \dots R'_{k-1}\mathcal{R}_{k-1}R'_k$ such that for all $i \leq k$, $R_i \subseteq R'_i$.

The first position of any g-conform match is $R_0 = \{(q_I, q_I)\}$. The first position of its f-conform shadow match is $R'_0 = \{(q_I, q_I)\}$, hence $R_0 \subseteq R'_0$.

For the induction step, let $\pi = R_0\Gamma_0(\mathsf{d}_0, \theta_0) \dots R_{k-1}\Gamma_{k-1}(\mathsf{d}_{k-1}, \theta_{k-1})R_k$ be a g-conform match and let $\pi' = R'_0\mathcal{R}_0 \dots R'_{k-1}\mathcal{R}_{k-1}R'_k$ be its f-conform shadow match such that for all $i \leq k$, $R_i \subseteq R'_i$. At position R'_k in the f-conform match, suppose that $\exists$'s choice is the set $\mathcal{R}_k \subseteq \mathcal{P}(Q \times Q)$. It follows from the rules of $Sat(\mathbb{A})$ that $[\![\bigwedge\{\rho_q\delta(q) \mid q \in Ran(R_k)\}]\!]^1_{\upsilon_{\mathcal{R}_k}} \neq \emptyset$. Hence, there is $\eta \in F\mathcal{R}_k$ such that

$$F\mathcal{R}_k, \eta \Vdash^1_{\upsilon_{\mathcal{R}_k}} \rho_q\delta(q), \tag{7.8}$$

for all $q \in Ran(R_k)$.

Take $q \in Ran(R_k)$. Recall that $\delta(q)$ is a disjunction of conjunctions of formulas in $\Lambda(Q)$. It follows from (7.8) that there exists a disjunct $\bigwedge \Gamma'_q$ of the formula $\delta(q)$ sucht that $F\mathcal{R}_k, \eta \Vdash^1_{\upsilon_{\mathcal{R}_k}} \rho_q \left(\bigwedge \Gamma'_q\right)$. Now we define Γ_k as the set $\{\Gamma'_q \mid q \in$

$Ran(R_k)\}$. It follows from the definition of Γ_k that $F\mathcal{R}_k, \eta \Vdash^1_{\upsilon_{\mathcal{R}_k}} \rho_q (\bigwedge \Gamma_k)$ and that $\Gamma_k \vdash_s \rho_q\delta(q)$. We define the strategy g such that $\exists$'s next move in π is the set $\Gamma_k \subseteq \Lambda_{\mathbb{A}}(Q \times Q)$.

Next, in the $Tab(\mathbb{A}, \mathsf{D})$-match, suppose that $\forall$ plays and chooses a pair (d_k, θ_k), where $\mathsf{d}_k \in \mathsf{D}$ and $\theta_k : X \longrightarrow Q \times Q$ are such that $\theta_k[Conc(\mathsf{d}_k)] \subseteq \Gamma_k$. Now $\exists$ has to find a set $\Delta \in Prem(\mathsf{d}_k)$. Since $[\![\bigwedge \Gamma_k]\!]^1_{\upsilon_{\mathcal{R}_k}} \neq \emptyset$ and D is one-step sound and complete for F, there exists $\Delta \in Prem(\mathsf{d}_k)$ such that $[\![\bigwedge \theta_k[\Delta]]\!]_{\upsilon_{\mathcal{R}_k}} \neq \emptyset$. We continue the definition of g by letting $\exists$ choose the set $\theta_k[\Delta]$ as the next position in the $Tab(\mathbb{A}, \mathsf{D})$-match, i.e. $R_{k+1} = \theta_k[\Delta]$.

Since $[\![\bigwedge \theta_k[\Delta]]\!]_{\upsilon_{\mathcal{R}_k}} \neq \emptyset$, there exists $R'_{k+1} \in \mathcal{R}_k$ such that

$$\mathcal{R}_k, R'_{k+1} \Vdash_{\upsilon_{\mathcal{R}_k}} \bigwedge \theta_k[\Delta]. \tag{7.9}$$

We define the next move for $\forall$ in the f-conform match π' as the relation R'_{k+1}.

Now we check that the induction hypothesis remains true, i.e. $R_{k+1} \subseteq R'_{k+1}$. Take (q, q') in R_{k+1}. By definition of R_{k+1}, (q, q') belongs to $\theta_k[\Delta]$. It follows from (7.9) that $R'_{k+1} \in \upsilon_{\mathcal{R}_k}(q, q')$. Recalling the definition of $\upsilon_{\mathcal{R}_k}$ (see the definition of the satisfiability game), we see that (q, q') belongs to R'_{k+1}.

It remains to show that such a strategy g is winning for $\exists$ in $Tab(\mathbb{A}, \mathsf{D})$. It follows from the definition of g that $\exists$ will never get stuck. Hence we may confine our attention to infinite g-conform matches. Let $\pi = R_0\Gamma_0(\mathsf{d}_0, \theta_0)R_1\Gamma_1(\mathsf{d}_1, \theta_1)\ldots$ be such a match. By definition of g, there exists an f-conform shadow match $\pi' = R'_0\mathcal{R}_0R'_1\mathcal{R}_1\ldots$ such that for all $i \in \mathbb{N}$, $R_i \subseteq R'_i$. Since f is a winning strategy for $\exists$, the sequence $R'_0R'_1\ldots$ does not contain any bad trace. Putting this together with the fact that $R_i \subseteq R'_i$ (for all $i \in \mathbb{N}$), we obtain that $R_0R_1\ldots$ does not contain any bad trace; that is, π is won by $\exists$.

For the direction from right to left, suppose $\exists$ has a winning strategy g in $Tab(\mathbb{A}, \mathsf{D})$. We need to provide a winning strategy f for $\exists$ in $Sat(\mathbb{A})$. During a f-conform match $R_0\mathcal{R}_0\ldots R_{k-1}\mathcal{R}_{k-1}R_k$, $\exists$ will maintain a g-conform shadow match $R_0\Gamma_0(\mathsf{d}_0, \theta_0)\ldots R_{k-1}\Gamma_{k-1}(\mathsf{d}_{k-1}, \theta_{k-1})R_k$. The first position of any f-conform match is $R_0 = \{(q_I, q_I)\}$ and the first position of its g-conform shadow match is also R_0. For the induction step, let $\pi = R_0\mathcal{R}_0\ldots R_{k-1}\mathcal{R}_{k-1}R_k$ be an f-conform match and let $\pi' = R_0\Gamma_0(\mathsf{d}_0, \theta_0)\ldots R_{k-1}\Gamma_{k-1}(\mathsf{d}_{k-1}, \theta_{k-1})R_k$ be its g-conform shadow match. In the f-conform match, $\exists$ has to define a set $\mathcal{R}_k \subseteq \mathcal{P}(Q \times Q)$ such that

$$[\![\bigwedge\{\rho_q\delta(q) \mid q \in Ran(R_k)\}]\!]^1_{\upsilon_{\mathcal{R}_k}} \neq \emptyset. \tag{7.10}$$

Assume that at position R_k in the g-conform shadow match, $\exists$ chooses a set $\Gamma_k \subseteq \Lambda_{\mathbb{A}}(Q \times Q)$ such that for all $q \in Ran(R_k)$, $\Gamma_k \vdash_s \rho_q\delta(q)$.

We say that a set $R \subseteq Q \times Q$ is *g-reachable from* R_k if there is a g-conform match of the form $R_0\Gamma_0(\mathsf{d}_0, \theta_0)\ldots R_{k-1}\Gamma_{k-1}(\mathsf{d}_{k-1}, \theta_{k-1})R_k\Gamma_k(\mathsf{d}, \theta)R$. We define $\mathcal{R}_k$ as the set $\{R \subseteq Q \times Q \mid R$ is g-reachable from R_k $\}$.

1. Claim. $\mathcal{R}_k$ is a legitimate move for $\exists$ at position R_k in $Sat(\mathbb{A})$.

Proof of Claim To show that such a move is a legitimate move for $\exists$ in $Sat(\mathbb{A})$, we have to check that (7.10) holds. Since for all $q \in Ran(R_k)$, $\Gamma_k \vdash_s \rho_q\delta(q)$, it is sufficient to prove that $[\![\bigwedge \Gamma_k]\!]^1_{\upsilon_{\mathcal{R}_k}} \neq \emptyset$. As D is one-step tableau-complete for F, it suffices to verify that for all rules d in D and for all maps $\theta : X \longrightarrow Q \times Q$ such that $\theta[Conc(\mathsf{d})] \subseteq \Gamma_k$, there exists $\Delta \in Prem(\mathsf{d})$ such that $[\![\bigwedge \theta[\Delta]]\!]_{\upsilon_{\mathcal{R}_k}} \neq \emptyset$.

Fix a rule d in D and a map $\theta : X \longrightarrow Q \times Q$ such that $\theta[Conc(\mathsf{d})] \subseteq \Gamma_k$. Now consider the g-conform match $R_0\Gamma_0(\mathsf{d}_0, \theta_0) \dots R_{k-1}\Gamma_{k-1}(\mathsf{d}_{k-1}, \theta_{k-1})R_k\Gamma_k(\mathsf{d}, \theta)$. In this match, it is $\exists$'s turn and according to g, she chooses a set $\Delta \in Prem(\mathsf{d})$, making the relation $\theta[\Delta]$ as the new position. So the relation $\theta[\Delta]$ is g-reachable from R_k. Thus it belongs to $\mathcal{R}_k$. To show $[\![\bigwedge \theta[\Delta]]\!]_{\upsilon_{\mathcal{R}_k}} \neq \emptyset$, it enough to prove that $\mathcal{R}_k, \theta[\Delta] \Vdash_{\upsilon_{\mathcal{R}_k}} \bigwedge \theta[\Delta]$. This follows from the definition of $\upsilon_{\mathcal{R}_k}$ and finishes the proof of the claim. ◀

Now that we showed that $\mathcal{R}_k$ is a valid move for $\exists$ at position R_k in $Sat(\mathbb{A})$, we may define f such that the next move for $\exists$ in π is $\mathcal{R}_k$. Next, it is $\forall$ who has to play in the f-conform match and he picks a relation $R_{k+1} \in \mathcal{R}_k$. By definition of $\mathcal{R}_k$, R_{k+1} is g-reachable from R_k. So there is a g-conform shadow match of the form $R_0\Gamma_0(\mathsf{d}_0, \theta_0) \dots R_{k-1}\Gamma_{k-1}(\mathsf{d}_{k-1}, \theta_{k-1})R_k\Gamma_k(\mathsf{d}_k, \theta_k)R_{k+1}$. This finishes the definition of f.

We still have to check that the strategy f is winning for $\exists$ in $Sat(\mathbb{A})$. First we observe that using f as a strategy $\exists$ will never get stuck. So it is sufficient to consider an infinite f-conform match $\pi = R_0\mathcal{R}_0R_1\mathcal{R}_1 \dots$. By construction of f, there exists a g-conform shadow match $\pi' = R_0\Gamma_0(\mathsf{d}_0, \theta_0)R_1\Gamma_1(\mathsf{d}_1, \theta_1) \dots$. Since π' is won by $\exists$, $R_0R_1 \dots$ does not contain any bad trace. Hence π is won by $\exists$.

For the purpose of obtaining good complexity results for the coalgebraic μ-calculus, in case we have a nice set D of derivation rules at our disposal, then the tableau game has considerable advantages over the satisfiability game. More specifically, if we follow exactly the ideas of [CKP09] and introduce the notions of contraction closure and exponential tractability for a set of derivation rules, we can show that the satisfiability problem for fixpoint coalgebraic formulas can be solved in exponential time.

Contraction closure A set D is *closed under contraction* if for all sets Y, for all rules d in D and for all substitutions $\theta : X \longrightarrow Y$, there exists d' in D and a substitution $\theta' : X \longrightarrow Y$ such that the following conditions hold. For all φ and ψ in $Conc(\mathsf{d}')$ such that $\theta'(\varphi) = \theta'(\psi)$, we have $\varphi = \psi$. Moreover, $\theta[Conc(\mathsf{d})] \subseteq \theta'[Conc(\mathsf{d}')]$ and for all Δ in $Prem(\mathsf{d})$, there exists Δ' in $Prem(\mathsf{d}')$ such that $\theta[\Delta] \subseteq \theta'[\Delta']$.

Being contraction closed means that whenever we use a rule and the conclusion contains twice the same formula, then we may replace the rule by a new one, where the formula occurs only once in the conclusion.

Exponential tractability A set D is *exponentially tractable* if for all sets Y, there is an alphabet Σ and a polynomial $p : \mathbb{N} \longrightarrow \mathbb{N}$ such that every (d, θ) (with $\mathsf{d} \in \mathsf{D}$ and $\theta : X \longrightarrow Y$) can be encoded as a string of length $\leq p(size(\theta[Conc(\mathsf{d})]))$ and the relations

$$\{(\Gamma, p(\mathsf{d}, \theta)) \mid \theta[Conc(\mathsf{d})] \subseteq \Gamma\}$$

and

$$\{(p(\mathsf{d}, \theta), \theta[\Delta]) \mid \Delta \text{ is the } i\text{-th premisse of } Prem(\mathsf{d})\}$$

are decidable in time exponential in the size of Y, for all $i \in \mathbb{N}$.

Exponential tractability will help us to encode the positions of the board of the tableau game, reduce the size of the board (its upper bound will be exponential in the size of the automaton, whereas in the previous section, the upper bound was doubly exponential) and give us a nice complexity for the relation determining legal moves in the tableau game.

The following theorem was proved in [CKP09], but here, we show how to derive it from the tableau game, which is in some sense, a particular case of the satisfiability game when the set Λ satisfies some nice properties. The key results are Proposition 7.2.2, Theorem 7.3.2 and Theorem 7.4.2. How to derive from them the next theorem is similar to what is done in [CKP09]. For clarity, we spell out the details.

7.4.3. Theorem. *[CKP09] Let F be a functor, Λ a set of predicate liftings for F, D a set of rules which is exponentially tractable, contraction closed and one-step sound and complete for Λ. It is decidable whether a fixpoint Λ-sentence φ is satisfiable in the class of F-coalgebras, in time exponential in the size of φ.*

Proof Let F, Λ and D be as in the statement of the theorem. Since D is exponentially tractable, there exist an alphabet Σ and a polynomial p which satisfy the conditions of the definition of exponential tractability. Fix a fixpoint sentence φ in ML_Λ. By Proposition 7.2.2, we can compute in time exponential in the size of φ, a Λ-automaton $\mathbb{A} = (Q, q_I, \delta, \Omega)$ such that φ and $\mathbb{A}$ are equivalent. Moreover, the size of $\mathbb{A}$ is dn, where d is the alternation depth of φ and n is the size of φ. The index of $\mathbb{A}$ is equal to d.

To check whether φ is satisfiable, it is sufficient to check whether there exists a pointed coalgebra accepted by $\mathbb{A}$. By Theorem 7.3.2 and Theorem 7.4.2, this boils down to determine whether $\exists$ has a winning strategy in the tableau game $Tab(\mathbb{A}, \mathsf{D}) = (G_\exists, G_\forall, E, Win, \{(q_I, q_I)\})$.

We start by modifying the game $Tab(\mathbb{A}, \mathsf{D})$ such that the modified game is easier to computer and is equivalent to the original game (in the sense that $\exists$ has a winning strategy in the original game iff $\exists$ has a winning strategy in the modified game). To do this modification, we use the fact D is contraction closed. We know that for all rules d and for all substitutions $\theta' : X \longrightarrow Q \times Q$, there exist a rule $\mathsf{d}' \in \mathsf{D}$ and a substitution $\theta' : X \longrightarrow Q \times Q$ such that

(i) for all φ and ψ in $Conc(\mathsf{d}')$ satisfying $\theta'(\varphi) = \theta'(\psi)$, we have $\varphi = \psi$.

(ii) $\theta[Conc(\mathsf{d})] \subseteq \theta'[Conc(\mathsf{d}')]$ and for all Δ in $Prem(\mathsf{d})$, there exists Δ' in $Prem(\mathsf{d}')$ such that $\theta[\Delta] \subseteq \theta'[\Delta']$.

It follows from (i) that θ' induces an injection from $Conc(\mathsf{d}')$ to $\Lambda_{\mathbb{A}}(Q \times Q)$. In particular, the number of formulas in $Conc(\mathsf{d}')$ is less or equal to $k := |\Lambda_{\mathbb{A}}(Q \times Q)|$. Moreover, it follows from (ii) and the rules of the tableau game that if the pair $((\mathsf{d}, \theta), R)$ belongs to the edge relation, we may remove this pair from the edge relation and replace it by the pair $((\mathsf{d}', \theta'), R)$. So in the remaining of the proof, we assume that the number of formulas in the conclusion Γ of a rule d occurring in the game is at most k.

Now we show that we can compute the game $Tab(\mathbb{A}, \mathsf{D})$ in time exponential in the size of $\mathbb{A}$. Recall that $Tab(\mathbb{A}, \mathsf{D})$ has three kinds of positions: (a) positions in $\mathcal{P}(Q \times Q)$, (b) positions in $\mathcal{P}(\Lambda_{\mathbb{A}}(Q \times Q))$ and (c) positions in $\mathsf{D} \times (Q \times Q)^X$.

1. Claim. Every position in the tableau game can be represented by a string of polynomial length in the size of $\mathbb{A}$.

Proof of Claim The claim immediate for the positions which belong to $\mathcal{P}(Q \times Q)$. Next we consider the positions in $\mathcal{P}(\Lambda_{\mathbb{A}}(Q \times Q))$. Since all the predicate liftings of $\Lambda_{\mathbb{A}}$ occur in φ, the size of $\Lambda_{\mathbb{A}}$ is smaller than the size of φ and thus, smaller than the size of $\mathbb{A}$. Hence, the size of $\Lambda_{\mathbb{A}}(Q \times Q)$ is polynomial in the size of $\mathbb{A}$. It follows that the positions of the form $\Gamma \subseteq \Lambda_{\mathbb{A}}(Q \times Q)$ can be encoded by a string polynomial in the size of $\mathbb{A}$.

Now we consider the positions of type (c). Using exponential tractability, we know that each position of the form (d, θ) can be encoded by a string (over Σ) of length $\leq p(size(\theta[Conc(\mathsf{d})]))$. Now $\theta[Conc(\mathsf{d})])$ is a subset of $\Lambda_{\mathbb{A}}(Q \times Q)$. We already observed that the size of $\Lambda_{\mathbb{A}}(Q \times Q)$ is polynomial in the size of $\mathbb{A}$. Hence, each position of the form (d, θ) can be encoded by a string of polynomial length in the size of $\mathbb{A}$ and this finishes the proof of the claim. ◀

Next we show that the size of the board is exponential in the size of $\mathbb{A}$. The number of positions of type (a) is obviously exponential in the size of $\mathbb{A}$. The size of $\mathcal{P}(\Lambda_{\mathbb{A}}(Q \times Q))$ (that is, the positions of type (b)) is also exponential in the size of $\mathbb{A}$, as the size of $\Lambda_{\mathbb{A}}$ is smaller than the size of $\mathbb{A}$. Finally we consider positions of type (c). Looking at the proof of the previous claim, we see that there is a polynomial p' such that each position of the form (d, θ) is encoded by a string over Σ of length $\leq p'(|A|)$. There are at most $(|\Sigma| + 1)^{p'(|A|)}$ such strings. Therefore, the number of positions of the form $(\mathsf{d}, \theta) \in \mathsf{D} \times (A \times A)^X$ is at most exponential in the size of $\mathbb{A}$. This finishes the proof that the size of the board is exponential in the size of $\mathbb{A}$.

Now we prove that we can compute the edge relation of the tableau game in time at most exponential in the size of $\mathbb{A}$.

2. CLAIM. *Given a pair of positions of the board, we can decide in time exponential in the size of* $\mathbb{A}$ *whether the pair belongs to the edge relation.*

PROOF OF CLAIM First, given a relation $R \in \mathcal{P}(Q \times Q)$ and a subset Γ of $\Lambda_{\mathbb{A}}(Q \times Q)$, deciding whether (R, Γ) belongs to the edge relation is equivalent to decide whether for all $q \in Ran(R)$, the relation $\Gamma \vdash_s \rho_q\delta(q)$ holds. Since we can compute $\mathbb{A}$ in time exponential in the size of φ, the size of $\delta(q)$ is at most exponential in the size of φ. Hence, the question whether $\Gamma \vdash_s \rho_q\delta(q)$ holds is decidable in exponential time.

Second, given a subset Γ of $\Lambda_{\mathbb{A}}(Q \times Q)$ and a pair (d, θ), we can check in time exponential in the size of $\mathbb{A}$ whether the pair $(\Gamma, (\mathsf{d}, \theta))$ belongs to the edge relation (that is, whether $\theta[Conc(\mathsf{d})] \subseteq \Gamma$), by using exponential tractability.

Finally, given a pair (d, θ) in $\mathsf{D} \times (Q \times Q)^X$ and a relation $R \in \mathcal{P}(Q \times Q)$, we want to decide whether the pair $((\mathsf{d}, \theta), R)$ belongs to the edge relation. That is, decide whether there exists a premise Δ of d such that $R = \theta[\Delta]$. By exponential tractability, for all $i \in \mathbb{N}$, we can check in time exponential in the size of $\mathbb{A}$ whether $R = \theta[\Delta_i]$, where Δ_i is the i-th premise of d. So it is sufficient to show that the number of premises of d is bound in a reasonable way.

Recall that the number of formulas in the conclusion Γ of d is at most k. It follows that the number of elements of X occurring in Γ is bound by $n \cdot k$, where n is the maximal arity of the predicate liftings in $\Lambda_{\mathbb{A}}$. Hence the number of premises of d is at most $2^{n \cdot k}$. Since we suppose that the size of a predicate lifting of arity l is at least l, n is less or equal to the size of φ, which is less or equal to the size of $\mathbb{A}$. We observed earlier that $k = |\Lambda_{\mathbb{A}}(Q \times Q)|$ is polynomial in the size of $\mathbb{A}$. Therefore, $2^{n \cdot k}$ is polynomial in the size of $\mathbb{A}$. ◀

Now, since the size of the board is at most exponential in the size of $\mathbb{A}$, the size of the edge relation is also at most exponential in the size of $\mathbb{A}$. Putting this together with the last claim, we can compute the edge relation of the tableau game in time at most exponential in the size of $\mathbb{A}$.

We turn now to the computation of the winning condition. Exactly, as we showed that the satisfiability game is regular (in Proposition 7.3.1), we can prove that the tableau game is regular. We can define the alphabet $\Sigma := \mathcal{P}(Q \times Q) \cup \{*\}$ and a coloring $col' : G_{\exists} \cup G_{\forall} \longrightarrow \Sigma$ such that for all $R \in \mathcal{P}(Q \times Q)$, $col'(R) = R$ and for all positions z of the board which do not belong to $\mathcal{P}(Q \times Q)$, $col'(z) = *$. Finally we let L' be the language $\{R_0 * * R_1 * * \cdots \mid R_0 R_1 \cdots \in NBT(Q, \Omega)\}$. It is easy to see that with these definitions of Σ, col' and L', $Tab(\mathbb{A}, \mathsf{D})$ is a regular game.

It is immediate that we can compute the graph of col' in time exponential in the size of $\mathbb{A}$. Moreover, by slightly modifying the construction of the automaton $\mathbb{B}$ in the proof of Proposition 7.3.1, we can construct a non-deterministic parity ω-automaton $\mathbb{C}$ such that $\mathbb{C}$ recognizes the complement of L'. We may also assume that $\mathbb{C}$ is computable in time exponential in the size of $\mathbb{A}$ and that the size and the index of $\mathbb{C}$ are linear in the size of $\mathbb{A}$. This finishes the proof that we can compute the game $Tab(\mathbb{A}, \mathsf{D})$ in time exponential in the size of $\mathbb{A}$.

Finally, using Theorem 7.1.4, we show that it is decidable in time exponential in the size of $\mathbb{A}$ whether $\exists$ has a winning strategy in $Tab(\mathbb{A}, \mathsf{D})$, from position $\{(q_I, q_I)\}$. As proved earlier, the sizes of the board of the tableau game and of the edge relation are exponential in the size of $\mathbb{A}$. The size and the index of $\mathbb{C}$ are linear in the size of $\mathbb{A}$. Therefore, there exists a deterministic ω-automaton recognizing the same language as $\mathbb{C}$, the size of which is exponential in the size of $\mathbb{A}$ and the index of which is linear in the size of $\mathbb{A}$ (see Theorem 2.4.1). Putting everything together with Theorem 7.1.4, we obtain that it is decidable in time exponential in the size of $\mathbb{A}$ whether $\exists$ has a winning strategy in $Tab(\mathbb{A}, \mathsf{D})$. □

7.5 Conclusions

In this chapter we have introduced Λ-automata which are automata using predicate liftings. We generalize [Ven06b] in that our presentation works for any type of coalgebra i.e. no restriction on the functor.

We introduced an acceptance game for Λ-automata, and established a finite model property (Theorem 7.3.2) using a satisfiability game for Λ-automata. We used games to establish a 2EXPTIME bound on the satisfiability problem for μML_Λ (Theorem 7.3.4). We showed how our approach relates to the work in [CKP09] by means of a game based on tableau rules (Theorem 7.4.2).

There are still some unresolved issues. By proving the finite model property for coalgebraic fixpoint logic, we gave an illustration how Λ-automata can be a tool for showing properties of coalgebraic fixpoint logic. So one could ask whether, using Λ-automata, we could prove other properties. For example, we could be inspired by the fact that the proofs of the existence of a disjunctive normal form and the uniform interpolation theorem for the μ-calculus are based on the automata theoretic approach for the μ-calculus.

Chapter 8

Conclusion

I like the following sentence written by Roland Dorgelès: “Ils sont arrivés parce qu’ils n’allaient pas loin” (“They have arrived because they did not go far”). Throughout this thesis, we saw that there are various perspectives one can have on a μ-sentence and depending on the context, one or another might be more relevant. One could think of the sentence above as being pessimistic regarding the work done. Or it could mean that the horizon is almost infinite.

The work done The idea was to explore the μ-calculus through the “fine structure” approach; that is to say, by specializing the class of models, of frames and of languages that we consider. We believe that the main contributions are the following:

- We gave an easy proof of the completeness of the Kozen’s axiomatization together with the axiom $\mu x.\Box x$ with respect to the class of finite trees. The proof consisted in combining Henkin-style semantics for the μ-calculus together with model theoretic methods.
- We investigated the expressive power of the μ-calculus on frames. More precisely, we showed that an MSO formula is frame definable by a μ-formula on trees iff it is preserved under p-morphic images on trees and under taking subtrees.
- We provided characterizations for the node and path expressions of the fragment of CoreXPath the unique basic axis of which is the descendant relation. These characterizations were obtained by combining well-known results concerning the μ-calculus.
- We gave a syntactic characterization of the continuous fragment. Using the characterization, we showed that it is decidable whether a given μ-sentence is continuous.

- In the framework of universal coalgebra, we introduced the notion of Λ-automaton associated with a set Λ of predicate liftings. We used these automata to obtain, under some very mild condition, a double exponential upper bound for the satisfiability problem for coalgebraic fixpoint logic. It also followed from the proof that coalgebraic fixpoint logic has the finite model property.

A small part of the almost infinite horizon At the end of each chapter, we mentioned a few suggestions for further work. Among those, we would like to recall the following:

- It would be nice to have a better understanding of the completeness of the μ-calculus. In particular, it would be interesting to obtain an easier proof for the completeness of the μ-calculus in general.

  As suggested by Chapter 3, a possible path is to first concentrate on restricted settings. Possibilities other than finite trees include the setting of linear orders and the alternation free fragment of the μ-calculus. Let us recall that the axiomatizability of the μ-calculus on linear orders has been studied by Roope Kaivola [Kai97].

- A natural question following the main result of Chapter 4 is whether we can obtain a characterization of the expressive power of the μ-calculus on arbitrary frames. More precisely, whether we can find semantic conditions that characterize the μ-calculus as a fragment of MSO on frames in general.

- It would be interesting to see whether some of the characterizations presented in Chapter 5 can be adapted for PDL, CTL or CTL*. If we try to modify the proofs of this chapter, a first difficulty that we would encounter is the fact that there is no nabla operator for these logics. But it still seems possible that there are translations similar to the ones presented in Chapter 5 for PDL, CTL or CTL*.

  Another interesting question related to Chapter 5 is to find the scope of the techniques used in this chapter, by for example using these methods to obtain characterizations of other natural fragments of the μ-calculus. A first obvious question is to identify other natural fragments.

- It seems natural to pursue the development of the automata theoretic approach for coalgebras. For example, we could try to show a result similar to the fact that there is a disjunctive normal form for the μ-calculus. This might help to obtain a uniform interpolation result.

- We could also think of lifting some of the results to the setting of coalgebras. The most relevant results in that respect are the completeness of the μ-calculus on finite trees and the characterization results of Chapter 5. The most difficult step would probably be to express the results in coalgebraic terms.

We also mentioned in the introduction that the chapters of this thesis can be seen as an exploration of the possible methods to understand the μ-calculus. In Chapter 3, we used model theoretic methods; it is is not yet clear what the scope of the methods used there is, or if there are other particular model theoretic methods that could help for investigating the μ-calculus.

Chapters 4, 5 and 7 provide several illustrations of the fact that game theory is an adequate formalism to talk about the μ-calculus. Moreover, Chapter 7 is also an example of the power of the automata theoretic approach and the fruitful interaction between automata theory and game theory.

Throughout the remarks of Chapter 5, we sketched how the methods of that chapter could be transferred from the level of formulas to the context of automata. The main point of these remarks is that if we want to address complexity issues (at least related to the problems of Chapter 5), the automata perspective might be more appropriate.

It would be interesting to study this phenomenon on a broader scale. For example, we mentioned in Chapter 5 the question of the complexity of the transformation of a μ-sentence into an equivalent disjunctive sentence. Comparing that complexity with the complexity of the transformation of an alternating μ-automaton into a non-deterministic μ-automaton would contribute to a better understanding of the link between automata and formula.

One of the first steps to investigate the complexity of the transformation of a μ-sentence into an equivalent disjunctive sentence is to establish the complexity of computing the solution of a modal equation system, as defined in [BS07, Section 3.7].

Bibliography

[AJ94] Samson Abramsky and Achim Jung. Domain Theory. In Samson Abramsky, Dov M. Gabbay Dov M., and Tom S. E. Maibaum, editors, *Handbook for Logic in Computer Science*. 1994.

[AN01] André Arnold and Damian Niwiński. *Rudiments of μ-calculus*, volume 146 of *Studies in Logic*. 2001.

[Arn99] André Arnold. The μ-calculus alternation-depth hierarchy is strict on binary trees. *Theoretical Informatics and Applications*, 33(4–5), 1999.

[Bak80] Jaco de Bakker. *Mathematical Theory of Program Correctness*. 1980.

[Bar93] Michael Barr. Terminal coalgebras in well-founded set theory. *Theoretical Computer Sciences*, 114:299–315, 1993.

[BC96] Girish Bhat and Rance Cleaveland. Efficient model checking via the equational μ-calculus. In *Proceedings of LICS'96*, pages 304–312, 1996.

[Ben76] Johan van Benthem. *Modal correspondence theory*. PhD thesis, Institute of Logic, Language and Computation, Amsterdam, 1976.

[Ben84] Johan van Benthem. Correspondence theory. In Dov M. Gabbay and Franz Guenthner, editors, *Handbook of Philosophical Logic*, volume 2, pages 167–248. 1984.

[Ben96] Johan van Benthem. *Exploring logical dynamics*. 1996.

[Ben98] Johan van Benthem. Programming operations that are safe for bisimulation. *Studia Logica*, 60(2):311–330, 1998.

[Ben02] Johan van Benthem. Extensive games as process models. *Journal of logic, language and information*, 11:289–313, 2002.

[Ben05] Johan van Benthem. Guards, bounds, and generalized semantics. *Journal of Logic, Language and Information*, 14:263–279, 2005.

[Ben06] Johan van Benthem. Modal frame correspondences and fixed-points. *Studia Logica*, 83:133–155, 2006.

[BK08] Michael Benedikt and Christoph Koch. XPath leashed. *ACM Computing Surveys*, 41(1), 2008.

[BL69] J.Richard Büchi and Lawrence H. Landweber. Solving sequential conditions by finite state strategies. *Transactions of the American Mathematical Society*, 138:295–311, 1969.

[BM96] Jon Barwise and Lawrence S. Moss. *Vicious Circles: On the Mathematics of Non-Wellfounded Phenomena*, volume 60 of *CSLI Lecture Notes*. 1996.

[Boj07] Mikołaj Bojańczyk. Two-way unary temporal logic over trees. In *Proceedings of LICS*, pages 121–130, 2007.

[BRV01] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal logic*. 2001.

[BS84] Robert Bull and Krister Segerberg. Basic modal logic. In Dov M. Gabbay and Franz Guenthner, editors, *Handbook of Philosophical Logic*, volume 2, pages 1–88. 1984.

[BS07] Julian C. Bradfield and Colin Stirling. Modal μ-calculi. In Patrick Blackburn, Johan van Benthem, and Frank Wolter, editors, *Handbook of Modal Logics*, volume 3 of *Studies in Logic and Practical Reasoning*, pages 721–756. 2007.

[BSd04] Falk Bartels, Ana Sokolova, and Erik de Vink. A hierarchy of probabilistic system types. *Theoretical Computer Science*, 327:3–22, 2004.

[Büc60] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Zeitschrift für Mathematische Logik und Grundlagen der Mathematik*, 6:60–92, 1960.

[BW06] Mikołaj Bojańczyk and Igor Walukiewicz. Characterizing EF and EX tree logics. *Theoretical Computer Science*, 358(255-272), 2006.

[BW07] Mikołaj Bojańczyk and Igor Walukiewicz. Forest algebras. In Jörg Flum, Erich Graedel, and Thomas Wilke, editors, *Automata and logic: history and perspectives*, number 2 in Texts in Logic and Games, pages 107–132. 2007.

[Cat05] Balder ten Cate. *Model theory for extended modal languages*. PhD thesis, University of Amsterdam, 2005. ILLC Dissertation Series DS-2005-01.

[CE81] Edmund M. Clarke and E. Allen Emerson. Design and synthesis of synchronization skeletons using branching time temporal logic. *LNCS*, 131:52–71, 1981.

[CK73] Chen Chung Chang and H. Jerome Keisler. *Model Theory*. 1973.

[CKP09] Corina Cîrstea, Clemens Kupke, and Dirk Pattinson. EXPTIME tableaux for the coalgebraic μ-calculus. In *Proceedings of CSL'09*, pages 179–193, 2009.

[Cza10] Marek Czarnecki. How fast can the fixpoints in modal μ-calculus be reached? Manuscript accepted at FICS10, 2010.

[Dam94] Mads Dam. CTL* and ECTL* as fragments of the modal mu-calculus. *Theoretical Computer Science*, 126(1):77–96, 1994.

[DH00] Giovanna D'Agostino and Marco Hollenberg. Logical questions concerning the μ-calculus: interpolation, Lyndon and Los-Tarski. *Journal of Symbolic Logic*, pages 310–332, 2000.

[DO09] Anuj Dawar and Martin Otto. Modal characterisation theorems over special classes of frames. *Ann. Pure Appl. Logic*, 161(1):1–42, 2009.

[Doe89] Kees Doets. Monadic Π^1_1-theories of Π^1_1-properties. *Notre Dame Journal of Formal Logic*, 30(2):224–240, 1989.

[Don70] John Doner. Tree acceptors and some of their applications. *Journal of Computer and System Sciences*, 4(5):77–96, 1970.

[DV02] Giovanna D'Agostino and Albert Visser. Finality regained: A coalgebraic study of Scott-sets and multisets. *Archive for Mathematical Logic*, 41:267–298, 2002.

[EC80] E. Allen Emerson and Edmund M. Clarke. Characterizing correctness properties of parallel programs using fixpoints. In *Proceedings of ICALP*, volume 85, pages 169–181, 1980.

[ÉI08] Zoltan Ésik and Szabolcs Ivan. Some varieties of finite tree automata related to restricted temporal logics. *Fundamenta Informaticae*, 82:79–103, 2008.

[EJ88] E. Allen Emerson and Charanjit S. Jutla. The complexity of tree automata and logics of programs (extended abstract). In *Proceedings of FOCS*, pages 328–337, 1988.

[EJ91] E. Allen Emerson and Charanjit S. Jutla. Tree automata, μ-calculus and determinacy. In *Proceedings of FOCS*, pages 368–377, 1991.

[EL86] E. Allen Emerson and Chin-Laung Lei. Efficient model checking in fragments of the propositional mu-calculus (extended abstract). In *Proceedings of LICS'86*, pages 267–278, 1986.

[Elg61] Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98:21–52, 1961.

[FBC85] Maurizio Fattorosi-Barnaba and Claudio Cerrato. Graded modalities I. *Studia Logica*, 44:197–221, 1985.

[Fin72] Kit Fine. In so many possible worlds. *Notre Dame Journal of Formal Logic*, 13:516–520, 1972.

[FL79] Michael Fischer and Richard Ladner. Propositional dynamic logic of regular programs. *Journal of Computer and System Sciences*, 18(2):194–211, 1979.

[Fon08] Gaëlle Fontaine. Continuous fragment of the μ-calculus. In *Proceedings of CSL*, pages 139–153, 2008.

[FV59] Solomon Feferman and Robert Lawson Vaught. The first-order properties of algebraic structures. *Fundamenta Informaticae*, 47:57–103, 1959.

[FV10] Gaëlle Fontaine and Yde Venema. Syntactic characterizations of semantic properties of the μ-calculus. submitted soon, 2010.

[GHK+80] Gerhard Gierz, Karl Heinrich Hofmann, Klaus Keimel, John Lawson, Michael Mislove, and Dana Stewart Scott. *A Compendium of Continuous Lattices*. 1980.

[GKP05] Georg Gottlob, Christof Koch, and Reinhard Pichler. Efficient algorithms for processing XPath queries. *ACM Transactions on Database Systems*, 61(2):444–491, 2005.

[Gob70] Lou Goble. Grades of modality. *Logique et Analyse*, 13:323–334, 1970.

[Göd31] Kurt Gödel. Über formal unentscheidbare sätze der principia mathematica und verwandter system i. *Monatschefte für Mathematik und Physik*, 38:173–198, 1931.

[GT75] Rob Goldblatt and S. Thomason. Axiomatic classes in propositional modal logic. In *Algebra and Logic*, volume 450 of *Lecture Notes in Mathematics*, pages 163–173. 1975.

[GTW02] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logic, and Infinite Games*, volume 2500 of *LNCS*. 2002.

[Har84] David Harel. Dynamic logic. In Dov Gabbay and Franz Guenther, editors, *Handbook of Philosophical Logic*, volume 2, pages 497–604. 1984.

[Hen50] Leon Henkin. Completeness in the theory of types. *Journal of Symbolic Logic*, 15(2):81–91, 1950.

[HK04] Helle Hansen and Clemens Kupke. A coalgebraic perspective on monotone modal logic. *Electronic Notes in Theoretical Computer Science*, 106:121 – 143, 2004. Proceedings of CMCS.

[Hol98a] Marco Hollenberg. Characterizations of negative definability in modal logic. *Studia Logica*, 60:357–386, 1998.

[Hol98b] Marco Hollenberg. *Logic and Bisimulation.* PhD thesis, Utrecht University, 1998. Zeno Institute of Philosophy.

[HP78] David Harel and Vaughan R. Pratt. Non-determinism in logics of programs. In *Proceedings of POPL*, pages 203–213, 1978.

[Jan96] David Janin. *Propriétés logiques du non-déterminisme et mu-calcul modal.* PhD thesis, Université de Bordeaux, 1996.

[Jan97] David Janin. Automata, tableaus and a reduction theorem for fixpoint calculi in arbitrary complete lattices. In *Proceedings of LICS*, 1997.

[Jan06] David Janin. Contributions to formal methods: games, logic and automata. Habilitation thesis, 2006.

[JL03] David Janin and Giacomo Lenzi. On the relationship between moandic and weak monadic second order logic on arbitrary trees, with application to the mu-calculus. *Fudamenta Informaticae*, 61:247–265, 2003.

[Jur98] Marcin Jurdziński. Deciding the winner in parity games is in UP $\cap$ co-UP. *Information Processing Letters*, 68(3):119–124, November 1998.

[Jur00] Marcin Jurdziński. Small progress measures for solving parity games. In *Proceedings of STACS*, volume LNCS 1770, pages 290–301, 2000.

[JW95a] David Janin and Igor Walukiewicz. Automata for the modal μ-calculus and related results. In *Proceedings of MFCS*, pages 552–562, 1995. LNCS 969.

[JW95b] David Janin and Igor Walukiewicz. Automata for the modal μ-calculus and related results. In *Proceedings of MFCS'95*, pages 552–562, 1995.

[JW96] David Janin and Igor Walukiewicz. On the expressive completeness of the propositional modal mu-calculus and related results. In *Proceedings of CONCUR*, pages 263–277, 1996.

[Kai97] Roope Kaivola. *Using Automata to Characterize Fixed Point Temporal Logics.* PhD thesis, Department of Computer Science, University of Edinburgh, 1997.

[Koz83] Dexter Kozen. Results on the propositional μ-calculus. *Theoretical Computer Science*, 27(3):333–354, 1983.

[Koz95] Dexter Kozen. Results on the propositional μ-calculus. *Lecture Notes in Computer Science*, 962, 1995.

[KV05] Clemens Kupke and Yde Venema. Closure properties of coalgebra automata. In *Proceedings of LICS 2005*, pages 199–208, 2005.

[KV08] Clemens Kupke and Yde Venema. Coalgebraic automata theory: basic results. *Logical Methods in Computer Science*, 4:1–43, 2008.

[KV09] Christian Kissig and Yde Venema. Complementation of coalgebra automata. In *Proceedings of CALCO'09*, pages 81–96, 2009.

[KVW00] Orna Kupferman, Moshe Y. Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking. *Journal of the ACM*, 47(2):312–360, 2000.

[McN66] Robert McNaughton. Testing and generating infinite sequences by a finite automaton. *Information and Control*, 9(5):521–530, 1966.

[MdR05] Maarten Marx and Maarten de Rijke. Semantic characterizations of navigational XPath. *ACM SIGMOD Report*, 34(3):41–46, 2005.

[Mos74] Yiannis Nicholas Moschovakis. *Elementary induction on abstract structures*, volume 77 of *Studies in Logic and the Foundations of Mathematics.* 1974.

[Mos91] Andrzej Mostowski. Games with forbidden positions. Technical Report 78, University of Gdańsk, 1991.

[Mos99] Lawrence S. Moss. Coalgebraic logic. *Annals of Pure and Applied Logic*, 96:277–317, 1999. (Erratum published *APAL* 99:241–259, 1999).

[MS87] David Muller and Paul Schupp. Alternating automata on infinite trees. *Theoretical Computer Science*, 54:267–276, 1987.

[Niw88] Damian Niwiński. Fixed points vs. infinite generation. In *Proceedings of LICS*, pages 402–0409, 1988.

[Niw97] Damian Niwiński. Fixed point characterization of infinite behavior of finite-state systems. *Theoretical Computer Science*, 189:1–69, 1997.

[NW96] Damian Niwiński and Igor Walukiewicz. Games for the mu-calculus. *Theoretical Computer Science*, 163(1&2):99–116, 1996.

[Ott99] Martin Otto. Eliminating Recursion in the mu-Calculus. In *Proceedings of STACS*, 1999.

[Pap94] Christos H. Papadimitriou. *Computational complexity.* 1994.

[Par69] David Park. Fixed point induction and proof of program properties. *Machine Intelligence*, 5:59–78, 1969.

[Par80] David Park. *On the semantics of fair parallelism*, volume 86, pages 504–526. LNCS, 1980.

[Par81] David Park. Concurrency and automata on infinite sequences. *LNCS*, 154:561–572, 1981.

[Pat03] Dirk Pattinson. Coalgebraic modal logic: Soundness, completeness and decidability of local consequence. *Theoretical Computer Science*, 309:177–193, 2003.

[Pit06] Nir Piterman. From nondeterministic Büchi and Streett automata to deterministic parity automata. In *Proceedings of LICS*, pages 255–264, 2006.

[Pnu77] Amir Pnueli. The temporal logic of programs. In *Proceedings of FOCS*, pages 46–57, 1977.

[Pra76] Vaughan R. Pratt. Semantical considerations on floyd-hoare logic. In *Proceedings of FOCS*, pages 109–121, 1976.

[Pra81] Vaughan R. Pratt. A decidable μ-calculus: Preliminary report. In *Proceedings of FOCS*, pages 421–427, 1981.

[PS09] Thomas Place and Luc Segoufin. A decidable characterization of locally testable tree languages. In *Proceedings of ICALP*, pages 285–296, 2009.

[PS10] Thomas Place and Luc Segoufin. Deciding definability in $FO_2(<)$ (or XPath) on trees. In *Proceedings of LICS'10)*, pages 253–262, 2010.

[Rab69] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, 1969.

[Rab77] Michael O. Rabin. Decidable theories. In Jon Barwise, editor, *Handbook of Mathematical Logic*, pages 595–629. 1977.

[Rij00] Maarten de Rijke. A note on graded modal logic. *Studia Logica*, 64(2):271–283, 2000.

[Saf92] Shmuel Safra. Exponential determinization for ω-automata with strong-fairness acceptance condition. In *Proceedings of STOC*, pages 275–282, 1992.

[Sah75] Henrik Sahlqvist. Correspondence and completeness in the first- and second-order semantics for modal logic. In *Proceedings of the Third Scandinavian Logic Symposium*, pages 110–143, 1975.

[Sal70] Andrzej Salwicki. Formalized algorithmic languages. *Bull. Acad. Polon. Sci., Ser. Sci. Math. Astron. Plys.*, 18:227–232, 1970.

[SdB69] Dana Scott and Jaco de Bakker. A theory of programs. IBM Vienna, 1969.

[SE89] Robert S. Streett and E. Allen Emerson. An automata theoretic decision procedure for the propositional μ-calculus. *Information and Computation*, 81(3):249–264, 1989.

[Smo85] Craig Smorynski. *Self-reference and modal logic.* 1985.

[Sti95] Colin Stirling. Local model checking games. In *Proceedings of CONCUR*, volume 962, pages 1–11, 1995.

[Str81] Robert S. Streett. Propositional dynamic logic of looping and converse. In *Proceedings of STOC*, pages 375–383, 1981.

[Str82] Robert S. Streett. Propositional dynamic logic of looping and converse is elementarily decidable. *Information and Control*, 54(1/2):121–141, 1982.

[Tar55] Alfred Tarski. A lattice-theoretical fixpoint theorem and its application. *Pacific Journal of Mathematics*, 5:285–309, 1955.

[Tho97] Wolfgang Thomas. Languages, automata and logic. In Grzegorz Rozenberg and Arto Salomaa, editors, *Handbook of formal languages*, volume 3, pages 389–455. 1997.

[TW68] James W. Thatcher and Jesse B. Wright. Generalized finite automata theory with an application to a decision problem for second-order logic. *Mathematical Systems Theory*, 2(1):57–81, 1968.

[Ven93] Yde Venema. Derivation rules as anti-axioms in modal logic. *Journal of Symbolic Logic*, 58:1003–1054, 1993.

[Ven06a] Yde Venema. Algebras and coalgebras. In Patrick Blackburn, Johan van Benthem, and Frank Wolter, editors, *Handbook of Modal Logic*, volume 3 of *Studies in Logic and Practical Reasoning*, pages 331–426. 2006.

[Ven06b] Yde Venema. Automata and fixed point logic: a coalgebraic perspective. *Information and Computation*, 204:637–678, 2006.

[Ven08a] Yde Venema. Lecture notes on μ-calculus. `http://staff.science.uva.nl/~yde/teaching/ml/`, 2008.

[Ven08b] Yde Venema. PDL as a fragment of the modal μ-calculus. unpublished manuscript, 2008.

[VW07] Moshe Y. Vardi and Thomas Wilke. Automata: from logics to algorithms. In Jörg Flum, Erich Graedel, and Thomas Wilke, editors, *Logic and Automata: History and Perspectives*, number 2 in Texts in Logic and Games, pages 629–736. 2007.

[W3C] W3C. XML path language (XPath): Version 1.0. W3C recommendation. `http://www.w3.org/TR/xpath.html`.

[Wal95] Igor Walukiewicz. Completeness of Kozen's axiomatization of the propositional μ-calculus. In *Proceedings of LICS*, pages 14–24, 1995.

[Wal02] Igor Walukiewicz. Monadic second order logic on tree-like structures. *Theoretical Computer Science*, 275:311 – 346, 2002.

[Wu07] Zhilin Wu. A note on the characterization of TL[EF]. *Information Processing Letter*, 102:28–54, 2007.

Index

List of symbols

Samenvatting

Dit proefschrift bestudeert enkele model-theoretische aspecten van de modale μ-calculus, een uitbreiding van de modale logica met kleinste en grootste dekpunt-operatoren. We verkennen deze aspecten via een "fijnstructuur" benadering van de μ-calculus. Met andere woorden, we concentreren ons op speciale klassen van structuren en specifieke fragmenten van de taal. De methoden die wij gebruiken illustreren de vruchtbare interactie tussen de μ-calculus en andere onderzoekgebieden, zoals automatentheorie, speltheorie en modeltheorie.

Hoofdstuk 3 bewerkstelligt een volledigheidsresultaat voor de μ-calculus over eindige bomen. Het volledigheidsbewijs van de μ-calculus over willekeurige structuren [Wal95] staat bekend vanwege de moeilijkheidsgraad, maar eindige bomen staan ons toe een veel eenvoudiger argument te geven. De techniek die we gebruiken bestaat uit het combineren van een Henkin-stijl semantiek voor de μ-calculus met modeltheoretische methoden geïnspireerd op het werk van Kees Doets [Doe89]).

In hoofdstuk 4 bestuderen we de uitdrukkingskracht van de μ-calculus op het niveau van frames. De uitdrukkingskracht van de μ-calculus op het niveau van modellen (gelabelde grafen) is bekend [JW96], terwijl niets bekend is van het niveau van frames (ongelabelde grafen). In de setting van frames komen de propositieletters overeen met universeel gekwantificeerde tweede-orde variabelen. Ons voornaamste resultaat is een karakterisering van die monadische tweede-orde formules die op de klasse van bomen (gezien als frames) equivalent zijn met een formule van de μ-calculus.

In Hoofdstuk 5 laten we karakteriseringen zien van specifieke fragmenten van de μ-calculus, waarvan de belangrijkste het Scott continue fragment en het volledig-additieve fragment zijn. Een interessant aspect van de continue formules is dat ze constructief zijn, dat wil zeggen, hun kleinste dekpunten kunnen worden uitgerekend in hoogstens ω veel stappen. We geven ook een alternatief bewijs voor de karakterisering van het volledig-additieve fragment, een resultaat verkregen door Marco Hollenberg [Hol98b]. Ons bewijs verloopt langs dezelfde lijnen

als dat voor de karakterisering van het continue fragment.

In het daaropvolgende hoofdstuk onderzoeken we de uitdrukkingskracht van een fragment van CoreXPath. XPath is een navigatietaal voor XML documenten en CoreXPath is geíntroduceerd om de logische kern van XPath te vatten. In Hoofdstuk 6 maken we gebruik van de nauwe verwantschap tussen CoreXPath en modale logica: door het combineren van bekende resultaten aangaande de μ-calculus (één daarvan uit Hoofdstuk 5), verkrijgen we een karakterisering van een belangrijk fragment van CoreXPath.

Ten slotte, in Hoofdstuk 7, ontwikkelen we automaten-theoretische hulpmiddelen voor co-algebraische dekpuntlogica's, dat wil zeggen, generaliseringen van de μ-calculus naar het abstractieniveau van co-algebras. Co-algebras geven een abstract kader voor het wiskundig representeren van evoluerende systemen. We gebruiken deze hulpmiddelen om zowel de beslisbaarheid van het vervulbaarheidsprobleem als de kleine-model eigenschap voor co-algebraische dekpuntlogica's in een algemene setting te laten zien. We verkrijgen een dubbel-exponentiële bovengrens voor de complexiteit van het vervulbaarheidsprobleem.

Abstract

This thesis is a study into some model-theoretic aspects of the modal μ-calculus, the extension of modal logic with least and greatest fixpoint operators. We explore these aspects through a "fine-structure" approach to the μ-calculus. That is, we concentrate on special classes of structures and particular fragments of the language. The methods we use also illustrate the fruitful interaction between the μ-calculus and other methods from automata theory, game theory and model theory.

Chapter 3 establishes a completeness result for the μ-calculus on finite trees. The proof of the completeness of the μ-calculus on arbitrary structures [Wal95] is well-known for its difficulty, but it turns out that on finite trees, we can provide a much simpler argument. The technique we use consists in combining an Henkin-type semantics for the μ-calculus together with model theoretic methods (inspired by the work of Kees Doets [Doe89]).

In Chapter 4, we study the expressive power of the μ-calculus at the level of frames. The expressive power of the μ-calculus on the level of models (labeled graphs) is well understood [JW96], while nothing is known on the level of frames (graphs without labeling). In the setting of frames, the proposition letters correspond to second-order variables over which we quantify universally. Our main result is a characterization of those monadic second-order formulas that are equivalent on trees (seen as frames) to a formula of the μ-calculus.

In Chapter 5, we provide characterizations of particular fragments of the μ-calculus, the main ones being the Scott continuous fragment and the completely additive fragment. An interesting feature of the continuous formulas is that they are constructive, that is, their least fixpoints can be calculated in at most ω steps. We also give an alternative proof of the characterization of the completely additive fragment obtained by Marco Hollenberg [Hol98b], following the lines of the proof for the characterization of the continuous fragment.

In the next chapter, we investigate the expressive power of a fragment of CoreXPath. XPath is a navigation language for XML documents and CoreXpath

has been introduced to capture the logical core of XPath. In Chapter 6, we exploit the tight connection between CoreXPath and modal logic: by combining well-known results concerning the μ-calculus (one of them appearing in Chapter 5), we establish a characterization of an important fragment of CoreXPath.

Finally, in Chapter 7, we develop automata-theoretic tools for coalgebraic fixpoint logics, viz. generalizations of the μ-calculus to the abstraction level of coalgebras. Coalgebras provide an abstract way of representing evolving systems. We use those tools to show the decidability of the satisfiability problem as well as a small model property for coalgebraic fixpoint logics in a general setting. We also obtain a double exponential upper bound on the complexity of the satisfiability problem.

Titles in the ILLC Dissertation Series:

ILLC DS-2001-01: **Maria Aloni**
Quantification under Conceptual Covers

ILLC DS-2001-02: **Alexander van den Bosch**
Rationality in Discovery - a study of Logic, Cognition, Computation and Neuropharmacology

ILLC DS-2001-03: **Erik de Haas**
Logics For OO Information Systems: a Semantic Study of Object Orientation from a Categorial Substructural Perspective

ILLC DS-2001-04: **Rosalie Iemhoff**
Provability Logic and Admissible Rules

ILLC DS-2001-05: **Eva Hoogland**
Definability and Interpolation: Model-theoretic investigations

ILLC DS-2001-06: **Ronald de Wolf**
Quantum Computing and Communication Complexity

ILLC DS-2001-07: **Katsumi Sasaki**
Logics and Provability

ILLC DS-2001-08: **Allard Tamminga**
Belief Dynamics. (Epistemo)logical Investigations

ILLC DS-2001-09: **Gwen Kerdiles**
Saying It with Pictures: a Logical Landscape of Conceptual Graphs

ILLC DS-2001-10: **Marc Pauly**
Logic for Social Software

ILLC DS-2002-01: **Nikos Massios**
Decision-Theoretic Robotic Surveillance

ILLC DS-2002-02: **Marco Aiello**
Spatial Reasoning: Theory and Practice

ILLC DS-2002-03: **Yuri Engelhardt**
The Language of Graphics

ILLC DS-2002-04: **Willem Klaas van Dam**
On Quantum Computation Theory

ILLC DS-2002-05: **Rosella Gennari**
Mapping Inferences: Constraint Propagation and Diamond Satisfaction

ILLC DS-2002-06: **Ivar Vermeulen**
A Logical Approach to Competition in Industries

ILLC DS-2003-01: **Barteld Kooi**
Knowledge, chance, and change

ILLC DS-2003-02: **Elisabeth Catherine Brouwer**
Imagining Metaphors: Cognitive Representation in Interpretation and Understanding

ILLC DS-2003-03: **Juan Heguiabehere**
Building Logic Toolboxes

ILLC DS-2003-04: **Christof Monz**
From Document Retrieval to Question Answering

ILLC DS-2004-01: **Hein Philipp Röhrig**
Quantum Query Complexity and Distributed Computing

ILLC DS-2004-02: **Sebastian Brand**
Rule-based Constraint Propagation: Theory and Applications

ILLC DS-2004-03: **Boudewijn de Bruin**
Explaining Games. On the Logic of Game Theoretic Explanations

ILLC DS-2005-01: **Balder David ten Cate**
Model theory for extended modal languages

ILLC DS-2005-02: **Willem-Jan van Hoeve**
Operations Research Techniques in Constraint Programming

ILLC DS-2005-03: **Rosja Mastop**
What can you do? Imperative mood in Semantic Theory

ILLC DS-2005-04: **Anna Pilatova**
A User's Guide to Proper names: Their Pragmatics and Semanics

ILLC DS-2005-05: **Sieuwert van Otterloo**
A Strategic Analysis of Multi-agent Protocols

ILLC DS-2006-01: **Troy Lee**
Kolmogorov complexity and formula size lower bounds

ILLC DS-2006-02: **Nick Bezhanishvili**
Lattices of intermediate and cylindric modal logics

ILLC DS-2006-03: **Clemens Kupke**
Finitary coalgebraic logics

ILLC DS-2006-04: **Robert Špalek**
Quantum Algorithms, Lower Bounds, and Time-Space Tradeoffs

ILLC DS-2006-05: **Aline Honingh**
The Origin and Well-Formedness of Tonal Pitch Structures

ILLC DS-2006-06: **Merlijn Sevenster**
Branches of imperfect information: logic, games, and computation

ILLC DS-2006-07: **Marie Nilsenova**
Rises and Falls. Studies in the Semantics and Pragmatics of Intonation

ILLC DS-2006-08: **Darko Sarenac**
Products of Topological Modal Logics

ILLC DS-2007-01: **Rudi Cilibrasi**
Statistical Inference Through Data Compression

ILLC DS-2007-02: **Neta Spiro**
What contributes to the perception of musical phrases in western classical music?

ILLC DS-2007-03: **Darrin Hindsill**
It's a Process and an Event: Perspectives in Event Semantics

ILLC DS-2007-04: **Katrin Schulz**
Minimal Models in Semantics and Pragmatics: Free Choice, Exhaustivity, and Conditionals

ILLC DS-2007-05: **Yoav Seginer**
Learning Syntactic Structure

ILLC DS-2008-01: **Stephanie Wehner**
Cryptography in a Quantum World

ILLC DS-2008-02: **Fenrong Liu**
Changing for the Better: Preference Dynamics and Agent Diversity

ILLC DS-2008-03: **Olivier Roy**
Thinking before Acting: Intentions, Logic, Rational Choice

ILLC DS-2008-04: **Patrick Girard**
Modal Logic for Belief and Preference Change

ILLC DS-2008-05: **Erik Rietveld**
Unreflective Action: A Philosophical Contribution to Integrative Neuroscience

ILLC DS-2008-06: **Falk Unger**
Noise in Quantum and Classical Computation and Non-locality

ILLC DS-2008-07: **Steven de Rooij**
Minimum Description Length Model Selection: Problems and Extensions

ILLC DS-2008-08: **Fabrice Nauze**
Modality in Typological Perspective

ILLC DS-2008-09: **Floris Roelofsen**
Anaphora Resolved

ILLC DS-2008-10: **Marian Counihan**
Looking for logic in all the wrong places: an investigation of language, literacy and logic in reasoning

ILLC DS-2009-01: **Jakub Szymanik**
Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language

ILLC DS-2009-02: **Hartmut Fitz**
Neural Syntax

ILLC DS-2009-03: **Brian Thomas Semmes**
A Game for the Borel Functions

ILLC DS-2009-04: **Sara L. Uckelman**
Modalities in Medieval Logic

ILLC DS-2009-05: **Andreas Witzel**
Knowledge and Games: Theory and Implementation

ILLC DS-2009-06: **Chantal Bax**
Subjectivity after Wittgenstein. Wittgenstein's embodied and embedded subject and the debate about the death of man.

ILLC DS-2009-07: **Kata Balogh**
Theme with Variations. A Context-based Analysis of Focus

ILLC DS-2009-08: **Tomohiro Hoshi**
Epistemic Dynamics and Protocol Information

ILLC DS-2009-09: **Olivia Ladinig**
Temporal expectations and their violations

ILLC DS-2009-10: **Tikitu de Jager**
"Now that you mention it, I wonder…": Awareness, Attention, Assumption

ILLC DS-2009-11: **Michael Franke**
Signal to Act: Game Theory in Pragmatics

ILLC DS-2009-12: **Joel Uckelman**
More Than the Sum of Its Parts: Compact Preference Representation Over Combinatorial Domains

ILLC DS-2009-13: **Stefan Bold**
Cardinals as Ultrapowers. A Canonical Measure Analysis under the Axiom of Determinacy.

ILLC DS-2010-01: **Reut Tsarfaty**
Relational-Realizational Parsing

ILLC DS-2010-02: **Jonathan Zvesper**
Playing with Information

ILLC DS-2010-03: **Cédric Dégremont**
The Temporal Mind. Observations on the logic of belief change in interactive systems