

Fragments of Fixpoint Logics

Automata and Expressiveness

Facundo Matías Carreiro

Fragments of Fixpoint Logics

Automata and Expressiveness

ILLC Dissertation Series DS-2015-05



INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

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

Copyright © 2015 by Facundo Matías Carreiro

Cover design by Facundo Matías Carreiro.

Printed and bound by GVO drukkers & vormgevers B.V.

ISBN: 978-90-6464-919-6

Fragments of Fixpoint Logics

Automata and Expressiveness

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 Aula der Universiteit
op vrijdag 11 december 2015, te 11.00 uur

door

Facundo Matías Carreiro

geboren te Buenos Aires, Argentinië.

Promotor:	Prof.dr. Y. Venema	Universiteit van Amsterdam
Overige leden:	Prof.dr. J.F.A.K. van Benthem	Universiteit van Amsterdam
	Prof.dr. D.J.N. van Eijck	Universiteit van Amsterdam
	Prof.dr. M. Otto	TU Darmstadt
	Prof.dr. I. Walukiewicz	Université de Bordeaux
	Dr. A. Facchini	IDSIA
	Dr. M.J. Marx	Universiteit van Amsterdam

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

*Para los abuelos Tinos,
ejemplo de valores, coraje y perseverancia.*

Contents

Acknowledgments	xi
1 Introduction	1
1.1 Featuring logics	5
1.2 Fragments of fixpoint logics	9
1.3 Logic and automata	12
1.4 Expressiveness modulo bisimilarity	19
1.5 Source of the material	21
2 Preliminaries	23
2.1 Terminology, transition systems and trees	23
2.2 Games	25
2.3 Parity automata	27
2.4 The modal μ -calculus	29
2.5 Logics of programs and games	31
2.5.1 Propositional Dynamic Logic	31
2.5.2 Concurrent PDL	33
2.5.3 Game Logic	34
2.6 Bisimulation	36
2.7 First-order logic and extensions	36
2.7.1 First-order logic with generalized quantifiers	37
2.7.2 Fixpoint extension of first-order logic	38
2.7.3 First-order logic with transitive closure	39
2.8 Second-order logics	40
2.9 Notational convention	42
3 Fragments of fixpoint logics	43
3.1 Completely additive fragments	44
3.1.1 Fixpoint theory of completely additive maps	47

3.1.2	Characterization of PDL inside μML	51
3.1.3	Characterization of $\text{FO}(\text{TC}^1)$ inside $\text{FO}(\text{LFP}^1)$	61
3.2	Continuous fragments	64
3.2.1	Fixpoint theory of continuous maps	65
3.2.2	Characterization of CPDL inside μML	67
3.2.3	Finiteness, $\mu_c\text{FOE}^\infty$ and WMSO	72
3.3	The question of Game Logic	76
3.4	Conclusions and open problems	82
4	Subclasses of parity automata	85
4.1	Weak parity automata	85
4.2	Continuous-weak parity automata	87
4.3	Additive-weak parity automata	88
4.4	Partial unraveling of parity automata	89
4.5	Variants of parity automata	92
4.6	Conclusions and open problems	94
5	One-step model theory	97
5.1	Single-sorted first-order languages	99
5.1.1	Normal forms	100
5.1.2	One-step monotonicity	108
5.1.3	One-step continuity	114
5.1.4	One-step additivity	118
5.1.5	Dual fragments	124
5.2	Selected multi-sorted first-order languages	125
5.2.1	Normal forms	126
5.2.2	One-step monotonicity	132
5.2.3	One-step additivity	134
5.2.4	One-step multiplicativity	139
5.3	Selected modal languages	139
5.3.1	Normal forms	140
5.3.2	One-step monotonicity	141
5.3.3	One-step continuity	142
5.3.4	One-step additivity	143
5.3.5	Dual fragments	144
5.4	Effectiveness of the normal forms	146
5.5	Conclusions and open problems	147
6	Concrete modal automata	149
6.1	Automata for test-free PDL	150
6.1.1	From formulas to automata	151
6.1.2	From automata to formulas	153
6.2	Automata for PDL	158

6.2.1	From formulas to automata	159
6.2.2	From automata to formulas	163
6.3	Automata for $\mu_c\text{ML}$	164
6.3.1	From formulas to automata	165
6.3.2	From automata to formulas	169
6.4	Modal automata versus first-order automata	170
6.5	Conclusions and open problems	171
7	Concrete first-order automata	173
7.1	Automata for MSO	173
7.1.1	From MSO to $\text{Aut}(\text{FOE}_1)$	174
7.1.2	From $\text{Aut}(\text{FOE}_1)$ to $\text{FO}(\text{LFP}^1)$	178
7.1.3	From $\text{FO}(\text{LFP}^1)$ to MSO	185
7.1.4	Subtleties of the obtained translations	186
7.2	Automata for WMSO	187
7.2.1	Simulation theorem	188
7.2.2	From WMSO to $\text{Aut}_{wc}(\text{FOE}_1^\infty)$	198
7.2.3	From $\text{Aut}_{wc}(\text{FOE}_1)$ to $\text{FO}(\text{LFP}^1)$	201
7.2.4	From $\mu_c\text{FOE}^\infty$ to WMSO	202
7.3	Automata for WCL	202
7.3.1	Simulation theorem	203
7.3.2	From WCL to $\text{Aut}_{wa}(\text{FOE}_1)$	207
7.3.3	From $\text{Aut}_{wa}(\text{FOE}_1)$ to $\text{FO}(\text{LFP}^1)$	209
7.3.4	From $\mu_a\text{FOE}^\gg$ to WCL	210
7.4	The question of automata for $\text{FO}(\text{TC}^1)$	212
7.5	Conclusions and open problems	213
8	Expressiveness modulo bisimilarity	215
8.1	Continuous-weak automata	216
8.1.1	Bisimulation-invariant fragment of WMSO	219
8.2	Additive-weak automata	220
8.2.1	Bisimulation-invariant fragment of WCL	222
8.2.2	Relative expressive power of PDL, WCL and $\text{FO}(\text{TC}^1)$	223
8.2.3	On the bisimulation-invariant fragment of $\text{FO}(\text{TC}^1)$	226
8.3	Conclusions and open problems	226
	Bibliography	229
	Index	241
	Samenvatting	247
	Abstract	249

Acknowledgments

*“It’s life that matters, nothing but
life – the process of discovering, the
everlasting and perpetual process,
not the discovery itself, at all.”*

— Fyodor Dostoyevsky, *The Idiot*

Chronologically, I should first thank Santiago, Carlos, Lucila and my family. Without their explicit support, even against their own personal interest, I would have never set foot on the ILLC.

I blindly chose Yde as my supervisor, without knowing him personally or talking with his students. Shortly after I first met him in Amsterdam, something happened, which would be excellent for the ILLC and bad for me: he became director of the ILLC. Nothing had prepared me for a four-year relationship with someone who is constantly busy and stressed, and is higher than me in the food chain. Except, of course, living more than twenty years with my parents. In any case, somehow, fueled by cigarettes and who knows what, Yde found (made) the time to meet with me almost *weekly* for four years. Even more, he fully read and commented every paper and draft that I wrote, and even read the whole manuscript of my dissertation, many times. This dedication, however, does not come for free. Yde would not let me get away with an easy PhD. This can be witnessed by the amount of gray hairs in my head, which have grown in biblical proportions since I got to Amsterdam. Yde is strict, yes, and he is demanding; prospective students, be warned. However, I have learned that he has a property that every logic(ian) would want: *consistency*. Yde is only as strict and demanding with you as he is with himself. I have learned much from him. On a scientific level, I think that he is one of the most integrous researchers I have met. He would never do something “just because we can”. He only focuses on questions that matter, conceptually. This is refreshing and inspiring, in an academic world

where many researchers (are forced to) play the game of publishing as much as, and whatever they can.

To Sumit, I will dedicate a paragraph. When I first met him, he thought that every European looked and acted the same, and that what we call hippies were gypsies. By the time we were finishing our dissertations, he was quoting Dante's *Inferno* and, not only could he distinguish Europeans, but he had also mastered the art of Dutch-Italian mediation. He is one of the most laid-back and kindest persons I have met. I am happy that we could share four years of deep conversations and fun activities together.

I learned to dance tango in Amsterdam, where it made for a great distraction from my PhD. As they say, "it takes two to tango". Well, in my case, it took far more than two people for me to dance it. First and foremost I want to thank Arjan and Marianne from Tango Argentino Amsterdam (a.k.a. de Plantage).¹ As tango teachers, they were the only ones that managed to make me feel comfortable with the complexities of the dance. But not only that, they went beyond, and supported me in tough times, always ready to give me an *abrazo*, even if I did not show up for months. I have grown fond of them, and have secretly adopted them as my mother and father figures in Amsterdam. I am also very grateful to Jenny, Nadine, Aafke, Luc, Dieuwke, Annebeth, Andreea, Violet, Will, Peter and Aysil for having shared dances with me. I would still be stepping on people's toes if it wasn't for their patience and support.

Giovanni and Margherita, you are community builders, you integrate people. I still remember Giovanni saying "In our last two parties we had the same group of people, we are doing something wrong." You let me into your lives without asking anything in return, and you also gave me a hand (and a dinner) when I needed it the most. We need more people like you in the world, and I hope to be able to repay your kindness.

Many thanks to the commensals of the *Dutch Lunch*: Hugo, Giovanni, Malvin, Julia, among others, and specially Dieuwke for the organization and supervision. In hindsight, it may not have been the best idea in the world to practice Dutch when we are stuffing food in our mouth, but I think we did learn a lot anyway. Thank you also for being good friends inside and outside the ILLC, and for eagerly listening to my PhD horror stories (this also includes Fatemeh and Paula). As Giovanni said "When you finish your PhD, we have to do a peregrination to some holy place. Everything that could have gone wrong, went wrong." Probably it was not as drastic, but I'm still considering the idea.

The ILLC, as an institute, was an awesome place to do a PhD. The Netherlands is one of the few places in the world where people doing research towards a PhD are not considered students, but workers. It is good to see that the ILLC implements this in full. I have not heard of or seen a place where PhD candidates are taken as seriously into account as at the ILLC. I felt part of the staff. Many

¹<http://www.tangoalma.nl/>

thanks to the ILLC management, as well. Whenever you had a good idea with a concrete plan, they were always ready to make it happen. The ILLC really felt like an oasis amidst the bureaucracy of the FNWI and UvA. Special thanks to Karine, who helped me get a room in Bos en Lommer when I arrived, and later assisted me in the yearly negotiations for Borneostraat 60.

Matt, thanks for being the coolest, most curious and open-minded American I have met, and a great flatmate in Bos en Lommer. You are so cool that I even wrote “American” instead of “of the United States”.

I would like to sincerely thank the members of my dissertation committee. They have kindly agreed to read and evaluate the contents of my dissertation, notwithstanding that their feedback cannot be introduced in the final form of this dissertation, due to the UvA’s doctoral regulations. I realize that this dissertation is not exactly a light read, and I appreciate your efforts to go through it.

Huge thanks to my co-authors, Yde, Alessandro, Fabio, Daniel and Lutz. The whole is clearly greater than the sum of its parts. Without them, this dissertation would not exist. Our discussions, visits, correspondence, theorem-proving efforts and cursing together had a big impact both in my life and work. I would also like to thank the people with whom I did not collaborate directly, but who were always kind enough to answer my questions by email or discuss them at the blackboard. Among others, these include Balder, Dietmar, Jouko and Sebastian.

A mis amigos de Buenos Aires, gracias por mantener el contacto, incluso cuando yo no les hablo por meses. Gracias por seguir invitándome a asados y reuniones (aunque sea de manera simbólica). Javier B., gracias por tu confianza e interés en mi trabajo y vida personal, estoy seguro de que te va a ir muy bien en tu PhD. Javier y Bruno ¡gracias por ser geniales compañeros de viaje! Vivi y Sergio, gracias por visitarme durante mi primer invierno, y sucumbir al frío y la oscuridad de Amsterdam. A mi amiga cordobesa, Inés, muchas gracias por escucharme y aconsejarme, a pesar de ser porteño.

Annebeth, heel erg bedankt voor jouw ondersteuning in alle hoogte- en diepte-punten, letterlijk en figuurlijk. Jij bent stikdapper, lief en zorgzaam. Hartelijk dank ook aan Johan, Martine, J.M., Pieter, Geert, en de naamloze kat, voor een warm welkom.

Amsterdam
August, 2015.

Facundo Carreiro

Chapter 1

Introduction

The general topic of this dissertation is the study of the relative expressive power of formal languages. In order to slowly get to the specific topic of this thesis, we first take a hopefully didactic detour through natural language.

Natural languages (as opposed to formal languages) are mainly used to describe the world and communicate our ideas. However, languages differ in how they cut up the world, in their expressiveness. For example, English has different words for the *blue* and *green* colours. On the other hand, ancient Japanese uses the same word *aoi* to denote these two colours. A more complex example is Russian, which has the word *zelenyy* for green but does not have a word for blue; however, it does have completely different words for light blue (*goluboy*) and dark blue (*siniy*). Discrepancies between languages are also found in translation. For example, the Dutch *gezellig* and Portuguese *saudade* are known to lack a precise translation to English.

While writers use language to convey information, the approach that better relates to the above paragraph is that of *linguists*: among other things, they study the individual properties of (natural) languages and how these languages relate to each other. That is, the languages themselves are *the object of study*.

In this dissertation we deal with formal languages. As opposed to natural languages, formal languages are used to describe properties of mathematical objects (also called *models*). Just as there are different natural languages with different properties, there exists a plethora of formal languages with varying qualities. An example of this variation is witnessed by the natural tension between the expressiveness of a formal language (i.e., *which properties* it can describe) and its computational complexity (i.e., *how difficult* it is to “process” it with a computer).

The use of formal languages is widespread, including but not limited to philosophy, economics, linguistics, mathematics and computer science. Essentially any area can use them to specify properties in a precise way. In this dissertation, we do not use formal languages to formulate properties or model puzzles (i.e., we are not “writers”). Instead, we consider a number of logics and automata, which

are special cases of formal languages, and study their particular properties and interconnections. Among other things, we give translations between automata and logics.

Our particular perspective will be that of theoretical computer science. In this field, one of the most important mathematical structures are the so-called *labeled transition systems*, which are used to represent programs (also called processes). A transition system consists of labeled nodes and edges between them. The nodes represent the states of a process, and the edges represent possible transitions (or actions) taking the process from one state to another. Moreover, the nodes are labeled with local information about the state of the process.

In the following paragraphs we give a brief introduction to logic and automata in computer science. We introduce some basic concepts and motivation for our research; a more precise historical overview with full references is given in the following sections of this chapter.

Logic in Computer Science. A typical application of logic is in the area of software verification. This area is devoted to automatically checking that some piece of software correctly implements its intended behaviour. This behaviour is called the specification of the software, and can be written in a logical language.

The kind of properties that one would want to check are typically expressed in a recursive way: for example “nothing bad ever happens (in the process) if nothing bad happens in the current state and, after a transition, nothing bad ever happens.” Unfortunately, the standard and well-studied logical formalisms of first-order logic and basic modal logic do not have enough expressive power to express recursive definitions. It is then natural to extend these logics with some recursive formalism.

A popular extension consists in adding *fixpoint operators* to these logics, with which we can encode iterative and recursive behaviour. Leaving the precise syntax aside for a moment, these operators provide a way to define equations of the form:

$$\mathbf{safety} \leftrightarrow \text{“everything is locally ok” and “after a transition, safety holds.”}$$

The addition of fixpoint operators to first-order logic results in a very expressive logic, which also has very high computational complexity. It is therefore very good for describing structures, and has become a star in the field of descriptive complexity. However, given its complexity, its *full* power cannot be used for verification.

The basic modal logic extended with fixpoint operators gives a very successful logic called the modal μ -calculus. This logic has a particularly good balance of high expressiveness and relatively low computational complexity. One of its main disadvantages, however, is that fixpoint operators are not very friendly when it comes to actually write complex properties. That is, the resulting formulas are not exactly easy to read and understand.

The literature offers a large range of other options to suit each particular need. For example, sometimes one does not require the full power of recursion, and can already express interesting properties based on *reachability*. The temporal logics LTL (linear temporal logic) and CTL (computation tree logic) can express properties of the form “at some point in the future (of the execution), property x holds” and “property x holds until property y holds.” Moreover, these logics were designed to be easy to read.

Another important logic to reason about programs is called PDL (propositional dynamic logic). Contrary to the rest of the logics that we have discussed so far, it already includes “program constructors” in its syntax, consisting of basic programs, composition and repetition of programs. Typical properties expressible in PDL include “there is a possible execution of program π finishing in state satisfying x ” and “every execution of program π finishes in state satisfying x .”

It is also possible to express full recursive behaviour without using fixpoints. One way to do it, is to use (monadic) second-order logics. These logics extend first-order logic, which can quantify over individual states, with the ability to quantify over *sets* of states. Two noteworthy examples of second-order logic are MSO (monadic second-order logic) which quantifies over arbitrary sets and WMSO (weak MSO) which quantifies over finite sets. Second-order logics usually have high computational complexity, however it is possible to consider *fragments* which are well behaved and provide sufficient expressive power with a reasonable computational complexity.

As a matter of fact, the logics that we named do not exist in isolation. On the contrary, there is a rich interaction among them. To name only a few examples, it is known that PDL can be translated to the μ -calculus; moreover, the μ -calculus can be translated to first-order logic with (unary) fixpoints, and the latter can be translated to MSO. These translations give an indication of the relative expressive power of these logics. On the other hand, for example, the logics MSO and WMSO are known to be incomparable in terms of expressive power. The list of known results goes on, but there are also many unknowns.

The analysis of the interconnections among logics gives insight on the landscape of the logical world. Moreover, it allows for the transfer of results, from the more well-known logics to the less studied ones. This transfer is not restricted to theoretical results, but also includes algorithms. For example, if we can translate formulas of a logic L to some other logic L' then we can use the algorithms for L' to decide problems of L .

Logic and automata. Besides logic, the other central component of this dissertation is the concept of automaton. An automaton is composed of a set of states and a transition map, specifying how to get from one state to another state. That is, an automaton is, itself, very similar to a program or process. Different types of automata are obtained by adding extra structure to this basic guide-

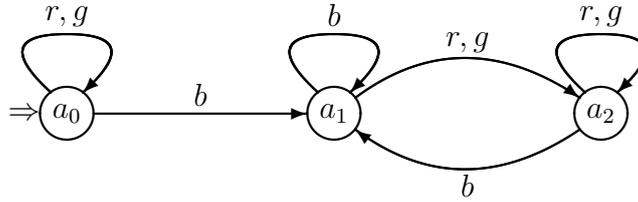


Figure 1.1: A finite state automaton on words.

line. Automata are usually seen as a device which is “run” on some mathematical structure which could be a word, a tree, a transition system or a more general structure.

An automaton starts in a so-called initial state and explores the structure in rounds. At each round, the automaton is standing at a state and “looking at” a part of the given structure. This information, along with the transition map, is used to decide which will be the next state of the automaton. Eventually, the automaton may decide to either *accept* or *reject* the given structure.

The crucial observation is that we can craft automata to express properties of mathematical structures, as we did with logical formulas. For example, we can create an automaton which accepts a transition system if the **safety** condition holds, and rejects it otherwise. In the case of automata, the recursive power of the formalism is explicit in its structure. That is, we can repeat an action if the transition map of an automaton allows us to cycle through some of its states.

One of the advantages of using automata is that, being themselves a model of computation, it is easier to give algorithms to compute acceptance (or rejection) and other tasks. On the other hand, they are not as straightforward to construct as formulas. That is, logical formulas usually have an *inductive* (and compositional) definition, which makes them very attractive when proving theorems. The structure of automata is quite flexible but at the same time less manageable.

All in all, automata and logics complement each other nicely. It is because of this that much effort has been put into translating logical formulas to different types of automata and vice-versa. For example, the modal μ -calculus and MSO are known to precisely correspond, respectively, to different kinds of *parity automata*. This connection has been successfully exploited to give important results about the expressiveness and decidability of both the μ -calculus and MSO.

The logic-automata connection has proven to be a very fruitful area, and there are many open problems which could provide useful insight. For example, there are no known automata models for PDL and WMSO.

Our contribution. This dissertation studies the relative expressive power and properties of several fixpoint and second-order logics. We use the term *fixpoint logic* in a broad sense, referring to any logic which can encode some type of recursion, iteration or repetition.

Our main objective is to systematically identify several important logics as precise fragments of other well-known logics. In order to accomplish this task, we develop automata-theoretic tools to analyze these fragments. The results of this dissertation provide new insight on the relationship of fixpoint and second-order logic and produce further evidence of the successful logic-automata connection.

1.1 Featuring logics

The logics featuring in this dissertation can be roughly divided in three categories: modal logics, (extensions of) first-order logics and (monadic) second-order logics. The analysis performed in this dissertation encompasses both the relationship of these logics inside each category and among categories. We start by introducing the main formalisms of this dissertation. This chapter will be limited to an intuitive and historical introduction, precise definitions are given in Chapter 2.

Modal logics of games and programs. The language now called Propositional Dynamic Logic was first investigated by Fischer and Ladner [FL79] as a logic meant to reason about computer program execution. PDL extends the basic modal logic with an infinite collection of diamonds $\langle \pi \rangle$ where π denotes a non-deterministic program. The intended intuitive interpretation of the formula $\langle \pi \rangle \varphi$ is that “some terminating execution of π from the current state leads to a state satisfying φ ”. The dual assertion $[\pi] \varphi$ states that “every terminating execution of π from the current state leads to a state satisfying φ ”.

The inductive structure of programs is made explicit in PDL’s syntax, as complex programs are built out of atomic programs using four program constructors. Formally, the formulas of PDL are given by the following mutual induction:

$$\begin{aligned} \varphi &::= p \mid \neg \varphi \mid \varphi \vee \varphi \mid \langle \pi \rangle \varphi \\ \pi &::= \ell \mid \pi; \pi \mid \pi \oplus \pi \mid \pi^* \mid \varphi? \end{aligned}$$

where p is a proposition letter and ℓ is an atomic action (or atomic program).

If π_1 and π_2 are programs then $\pi_1; \pi_2$ is a program denoting the execution of π_1 and then π_2 (called sequential composition), $\pi_1 \oplus \pi_2$ is a program denoting a non-deterministic choice between them (usually written $\pi_1 \cup \pi_2$ in the literature) and π_1^* is a program denoting the repetition of π_1 for a finite amount of times (including zero). In addition, for every formula $\varphi \in \text{PDL}$, the program $\varphi?$ tests whether φ holds in the current state.

Given a model \mathbb{M} , each program π induces a relation $R_\pi^{\mathbb{M}} \subseteq M^2$ which is used to give semantics to PDL. The intended meaning of this relation is that $R_\pi^{\mathbb{M}}(s, t)$ iff starting from state s one can successfully execute the program π and arrive at state t . The meaning of PDL-programs can also be seen in terms of a game between \exists and \forall . Under this interpretation, the formula $\langle \pi_1 \oplus \pi_2 \rangle \varphi$ is true if

\exists can choose one of the programs and execute it successfully, arriving at a state where φ holds.

One of the most important and characteristic features of PDL is that the program construction π^* endows PDL with second-order capabilities while still keeping it decidable (EXPTIME for satisfiability [Pra80] and PTIME for model-checking [FL79], even with many additional operators [Lan06]).

Concurrent PDL (CPDL) was introduced by Peleg in [Pel85] and extends the programs of PDL with the concurrent execution operator \otimes (dual of \oplus). The intuition behind this new operator is that the program $\pi_1 \otimes \pi_2$ succeeds when both π_1 and π_2 can be executed simultaneously. If interpreted as a game, this means that the formula $\langle \pi_1 \otimes \pi_2 \rangle \varphi$ is true if, no matter which program \forall chooses, \exists can successfully execute it, arriving to a state where φ holds.

Game Logic (GL) was introduced by Parikh [Par85] and extends PDL with *all* program duals. In this context, the programs are called games. Syntactically, this is obtained by extending the language with a dual operator π^δ for each game π . In this case, the execution of “programs” with the dual operator creates allows for a rich interaction between \exists and \forall . Originally, Game Logic was meant to be interpreted in neighborhood models, used to represent extensive games.¹ In this dissertation we do not consider neighbourhood models and we only focus on relational models as done in [Ber05]. The rationale behind this choice is that every logic in this dissertation but GL is interpreted over relational models, and even there, GL is not well understood. Moreover, relational models are a special case of neighbourhood models.

Modal μ -calculus. The modal μ -calculus (μ ML) extends the basic modal language with a mechanism for forming least (and greatest) fixpoints. It is highly expressive, subsuming a vast amount of dynamic and temporal logics such as PDL, CTL* and Game Logic, while still being computationally well behaved: satisfiability can be solved in EXPTIME [EJ99] and model-checking is in $\text{NP} \cap \text{co-NP}$ [ES95]. Moreover, this logic has a beautiful characterization stating that it can express all bisimulation-invariant properties expressible in monadic second-order logic [JW96]. All in all, it is one of the most significant languages on the modal landscape. It was introduced in its present form by Dexter Kozen [Koz83].

Syntactically, the μ -calculus adds least ($\mu p.\varphi$) and greatest ($\nu p.\varphi$) fixpoint operators to the language of basic modal logic, where p is a propositional variable. In order to give semantics to this operator observe that, given a model \mathbb{S} and a formula φ with a free variable p , the extension $\llbracket \varphi \rrbracket^{\mathbb{S}}$ of φ in \mathbb{S} depends on the set of points where p holds. This dependence can be formalized as a map

¹Neighbourhood models are a generalization of relational models, where the accessibility relation is of the form $R \subseteq S \times \wp(S)$ instead of $R \subseteq S \times S$.

$F_p^\varphi : \wp(S) \rightarrow \wp(S)$ given by:

$$F_p^\varphi(Y) := \{s \in S \mid \varphi(Y) \text{ is true in } \mathbb{S}\}.$$

The extension $\llbracket \mu p.\varphi \rrbracket^{\mathbb{S}}$ can now be defined as the least fixpoint of F_p^φ . The dual operator $\nu p.\varphi$ is defined analogously, or it can also be considered as a macro-expression given by $\nu p.\varphi := \neg \mu p.\neg\varphi[p \mapsto \neg p]$.

Formulas of the modal μ -calculus can be classified according to their *alternation depth*, which roughly is given as the maximal length of a chain of nested alternating least and greatest fixpoint operators [Niw86]. The hierarchy induced by the alternation depth, called the ‘‘alternation hierarchy’’, was shown to be strict [Bra96, Bra98] and is of special interest in the context of model-checking algorithms, since the complexity of the algorithms depends exponentially on the alternation depth of the given formula [EL86, BCM⁺92].

The *alternation-free fragment* of the modal μ -calculus (AFMC) is the collection of μ ML-formulas without nesting of least and greatest fixpoint operators. It is not difficult to see that, over arbitrary models, this fragment is less expressive than the full μ ML. That is, there is a μ ML-formula φ such that there is no equivalent formula of AFMC [Par80]. Despite its simplicity, AFMC already accommodates many temporal and dynamic logics such as PDL, CPDL, CTL and LTL but, for example, it does not cover Game Logic [Ber03].

Extensions of first-order logic. In this dissertation we will consider an extension of first-order logic with equality (FOE) with the generalized quantifier $\exists^\infty x.\varphi$ expressing that there exist infinitely many elements satisfying φ . This quantifier clearly adds new expressive power to FOE. For example, we are now able to characterize the models which have infinite domain, with the formula $\exists^\infty x.\top$. We will denote this extended logic as FOE^∞ .

It is well known that the reflexive-transitive closure R^* of a binary relation R is not expressible in first-order logic [Fag75]. Therefore, a straightforward way to extend first-order logic is to add a reflexive-transitive closure operator:

$$[\text{TC}_{\bar{x},\bar{y}}.\varphi(\bar{x},\bar{y})](\bar{\mathbf{u}},\bar{\mathbf{v}})$$

which states that $(\bar{\mathbf{u}},\bar{\mathbf{v}})$ belongs to the transitive closure of the relation denoted by $\varphi(\bar{x},\bar{y})$. In the above expression, the sequences of variables $\bar{x},\bar{y},\bar{\mathbf{u}},\bar{\mathbf{v}}$ should all be of the same length; this length is called the *arity* of the transitive closure. This extension of first-order logic, called FO(TC) or sometimes transitive-closure logic, was introduced by Immerman in [Imm87] where he showed that it captures the class of NLOGSPACE queries on finite ordered structures.

The fragment $\text{FO}(\text{TC}^k)$ of FO(TC) is given by restricting the arity of the transitive closure to length k . These fragments naturally induce a hierarchy (the arity hierarchy) for FO(TC), which was proven strict for finite models [Gro96].

Moreover, in some restricted classes of trees, full FO(TC) is even more expressive than MSO [TK09]. In this dissertation, however, we restrict our attention to FO(TC¹), which is FO(TC) restricted to sequences of length one; that is, the reflexive-transitive closure can only be applied to formulas $\varphi(x, y)$ defining a *binary relation*. This fragment of FO(TC) is easily seen to be included in MSO.

A more general way to extend first-order logic is to add a fixpoint operator as in [Mos08, Mos74, CH82]. Consider, as an example, a first-order formula $\varphi(p, x)$ where p is a monadic predicate and x is a free variable. The set of elements $s \in M$ of some model \mathbb{M} which satisfy $\varphi(p, s)$ clearly depends on the interpretation of p . This dependency can be formalized as a map

$$F_{p:x}^\varphi(Y) := \{s \in M \mid \varphi(Y, s) \text{ is true in } \mathbb{M}\}.$$

Assuming that φ is monotone in p , the least and greatest fixpoints of this map will exist by the Knaster-Tarski theorem. It is now easy to extend first-order logic with a fixpoint construction $[\text{LFP}_{p:x}.\varphi(p, x)](z)$ which holds iff the interpretation of z belongs to the least fixpoint of the map $F_{p:x}^\varphi$. This extension is called first-order logic with *unary* fixpoints and is usually denoted by FO(LFP¹). This is because the arity of the fixpoint relation (in our case p) is unary. As with the transitive closure operator, we can consider a logic where the arity of the fixpoint is not bounded. This logic is known as FO(LFP) and was shown to capture PTIME queries [Imm87] on finite ordered structures. The reader is referred to [Grä02] for a great survey on the many variants of the model-checking problem for this logic. In this dissertation we focus on FO(LFP¹), which we will denote as μFOE .

Monadic second-order logics. The second-order logics that we will consider are clearly extensions of first-order logic, but they deserve a section of their own. Monadic second-order logic (MSO) extends first-order logic with a new quantifier $\exists X.\varphi$ which quantifies over arbitrary sets (also known as unary predicates).

$$\mathbb{M}, g \models \exists X.\varphi \quad \text{iff} \quad \text{there is } U \subseteq M \text{ such that } \mathbb{M}[X \mapsto U], g \models \varphi.$$

This addition results in huge expressive power, compared to first-order logic. For example, transitive closure and fixpoint of formulas are both expressible in MSO. With set quantification, MSO results in a quite versatile specification language.

Model-checking for monadic second-order logic was shown to be PSPACE-complete [Sto74, Var82]. On the other hand, the satisfiability problem for MSO is clearly undecidable in the general case, since MSO extends first-order logic. However, the satisfiability problem for MSO is decidable *on trees*. This result is of crucial importance and was proved for incrementally bigger classes of models using automata-theoretic techniques. We defer this discussion to a later section.

Another second-order logic that will feature in this dissertation is *weak* MSO. This logic has the same syntax as MSO, but the intended quantification is over

finite sets instead of arbitrary ones.

$$\mathbb{M}, g \models \exists_{\text{fin}} X. \varphi \quad \text{iff} \quad \text{there is a finite } U \subseteq M \text{ such that } \mathbb{M}[X \mapsto U], g \models \varphi.$$

To emphasize the quantification over finite sets we write $\exists_{\text{fin}} X. \varphi$ instead of $\exists X. \varphi$.

1.1.1. REMARK. The adjective “weak” is a bit misleading, since WMSO is in general not a fragment of MSO. Indeed, the class of finitely branching trees is not definable in MSO (see [Wal96]) but is defined using the WMSO-formula $\forall x \exists_{\text{fin}} X \forall y. (R(x, y) \rightarrow y \in X)$. The class of well-founded trees, on the other hand, is definable in MSO but not in WMSO [CF11].

To conclude this section, we introduce the last variant of MSO that we will use. We call it *weak chain logic*, since it quantifies over *finite chains*.

$$\mathbb{T}, g \models \exists_{\text{fch}} X. \varphi \quad \text{iff} \quad \text{there is a finite chain } U \subseteq M \text{ such that } \mathbb{T}[X \mapsto U], g \models \varphi.$$

Here, we define a chain on a tree \mathbb{T} to be a set X such that all elements of X belong to the same branch. The original (non-weak) chain logic was introduced by Thomas in [Tho84], in the context of trees, and further studied in [Tho96, Boj04].

1.2 Fragments of fixpoint logics

The logics mentioned in the previous section do not exist in isolation. On the contrary, the more we know about their interconnection, the more we can reuse results and learn about these logics. In this section we discuss some known results relating the logics of this dissertation, and we outline our main contributions.

Modal realm. It is very well known that PDL can be translated to μML [EL86]. However, the exact fragment of μML that corresponds to PDL is not characterized. A key role in this characterization will be played by the fragment of μML where the fixpoint operator $\mu p. \varphi$ is restricted to formulas φ which are *completely additive in p* (we denote this fragment as $\mu_a\text{ML}$). A formula φ is said to be completely additive in p if for any family of subsets $\{P_i\}_{i \in I}$ with $P_i \subseteq S$ satisfies:

$$F_p^\varphi\left(\bigcup_i P_i\right) = \bigcup_i F_p^\varphi(P_i).$$

Complete additivity implies monotonicity and, in particular, such a map is determined “point-wise.” That is, for every set $P \subseteq S$ we have:

$$F_p^\varphi(P) = \bigcup_{t \in P} F_p^\varphi(\{t\}).$$

Complete additivity has been studied by van Benthem (under the name of ‘continuity’ in [Ben96, Ben98]) when considering operations on relations that preserve bisimulations (called ‘safe for bisimulation’) which can be defined in first-order logic. Hollenberg [Hol98] also studied complete additivity in the context of bisimulation-safe operations definable in monadic second-order logic.

The fragment $\mu_a\text{ML}$ has been syntactically characterized by Fontaine and Venema [FV12, Fon10]. An early observation, made by Venema in an unpublished manuscript [Ven08], is that PDL exactly corresponds to $\mu_a\text{ML}$ and test-free PDL corresponds to a precise syntactical restriction of $\mu_a\text{ML}$, which we call $\mu_{na}^-\text{ML}$.

THEOREM ([Ven08, CV14]). *PDL is effectively equivalent to $\mu_a\text{ML}$ and test-free PDL is effectively equivalent to $\mu_{na}^-\text{ML}$.*

The observation made in [Ven08] was of crucial importance to the development of this dissertation. It provided the necessary insight to build automata for PDL and approach the bisimulation-invariance problem (as we will discuss later). In Section 3.1.2 of this dissertation we give a new presentation of the above theorem, with full proofs. This section also contains a discussion on the peculiarities of complete additivity which allow these equivalences to hold.

The logic CPDL can also be translated to the modal μ -calculus. However, complete additivity will not be enough to capture this logic; we can already conclude this from the above result and the fact that $\text{PDL} \not\subseteq \text{CPDL}$. The key concept related to CPDL is that of (*Scott*) *continuity*. A map $F : \wp(S) \rightarrow \wp(S)$ is called continuous if it is already determined by finite sets (we also say “restricts to finite sets”). That is,

$$F(X) = \bigcup_{Y \subseteq_{\omega} X} F(Y).$$

The topological terminology stems from the observation that this equation expresses the continuity of the map F with respect to the Scott topology on $\wp(S)$.

The notion of continuity is strictly more general than complete additivity. It is easy to see that every completely additive map F is continuous, since F restricts to singletons (and is monotone) then in particular it restricts to finite sets.

Peleg relates CPDL to fragments of μML . First, he considers $\mu_c\text{ML}$, the fragment of μML where the least fixpoint operator is restricted to continuous formulas. This fragment was also studied by van Benthem [Ben06, Definition 5] under the name of ‘ ω - μ -calculus’ (probably due to the constructivity property).

Second, Peleg also considers a fragment of $\mu_c\text{ML}$ which he calls *simple* $\mu_c\text{ML}$ and is obtained by forbidding the interleaving of the fixpoint operators. That is, formulas of the shape $\mu p.\varphi(\mu q.\varphi'(p))$ are not allowed. We follow the terminology of [SV10] and call this fragment *flat* $\mu_c\text{ML}$ and denote it by $\mu_c\text{ML}^b$. Peleg proves that CPDL sits somewhere between these fragments, but leaves the strictness of the inclusions as an open question:

THEOREM ([Pel85, Theorem 2.11]). $\mu_c\text{ML}^b \subseteq \text{CPDL} \subseteq \mu_c\text{ML}$.

Our contribution is to give a syntactic fragment $\mu_{nc}\text{ML}^\vee$ of $\mu_c\text{ML}$ which precisely corresponds to CPDL.

CONTRIBUTION (Section 3.2.2). $\mu_c\text{ML}^b \subseteq \text{CPDL} \equiv \mu_{nc}\text{ML}^\vee \subseteq \mu_c\text{ML}$.

We discuss how $\mu_{nc}\text{ML}^\vee$ is placed inside $\mu_c\text{ML}$, but the question whether these two are equal or distinct is left open. In particular, the strictness of the inclusions is still open.

The modal μ -calculus and Game Logic have very different syntaxes. The former contains explicit fixpoint operators, while the latter only has a seemingly weaker iteration operator. Superficially, these logics look quite different, however, the relationship between them remains an intriguing topic today.

It is known that GL can be translated to the two variable fragment $\mu\text{ML}[2]$ of the μ -calculus (*cf.* [BGL05, Lemma 47]). Even then, the question of whether GL is equivalent to μML was open for a long time, as it was unknown whether $\mu\text{ML}[2]$ was equivalent to the full μML . This question was finally closed by Berwanger by showing that the *variable* hierarchy of the μ -calculus is strict.

THEOREM ([BGL05, Ber05]). $\text{GL} \not\equiv \mu\text{ML}$.

However, the GL question is still not fully solved. The *exact* fragment of μML (or $\mu\text{ML}[2]$) which corresponds to GL is still unknown. In particular, it is not known if GL and $\mu\text{ML}[2]$ coincide.

Following the methodology that we used for PDL and CPDL, we define a fragment μML^\vee of μML and show that it corresponds to GL. As a corollary, we actually get that $\text{GL} \equiv \mu\text{ML}^\vee[2] \equiv \mu\text{ML}^\vee$, but the question of whether $\mu\text{ML}^\vee[2]$ and $\mu\text{ML}[2]$ coincide is left open. We also discuss some intuitions and conjectures that may lead to a separation of these fragments.

CONTRIBUTION (Section 3.3). $\text{GL}, \mu\text{ML}^\vee[2]$ and μML^\vee are effectively equivalent.

First- and second-order realm. It is not difficult to see that $\text{FO}(\text{TC})$ is included in $\text{FO}(\text{LFP})$ and $\text{FO}(\text{TC}^1)$ is included in μFOE . This can be shown by a straightforward translation, but the exact fragment of μFOE that corresponds to $\text{FO}(\text{TC}^1)$ has not been characterized. Similar to what happens with PDL, the key notion leading to such a characterization is again complete additivity.

First of all, we consider a syntactic fragment $\mu_a\text{FOE}$ of μFOE by restricting the application of the unary fixpoint to formulas which are completely additive. We prove that $\text{FO}(\text{TC}^1)$ effectively corresponds to this fragment.

CONTRIBUTION (Section 3.1.3). $\text{FO}(\text{TC}^1)$ is effectively equivalent to $\mu_a\text{FOE}$.

As a minor step towards the above contribution we give, in Section 3.1.1, a general characterization of fixpoints of arbitrary completely additive *maps*, that is, maps which need not be induced by a formula.

It follows from results by Väänänen that, if we restrict to monadic signatures, then WMSO coincides with FOE^∞ , the extension of first-order logic with the generalized quantifier $\exists^\infty x.\varphi$.

THEOREM ([Vää77, Section 6]). $\text{WMSO} \equiv \text{FOE}^\infty$ *on monadic signatures.*

This theorem can also be given a more direct proof using the normal forms that we will develop in Chapter 5. We could ask ourselves what happens with this relationship if we consider (arbitrary) relational signatures, and even consider the presence of a fixpoint operator. The following proposition shows that this relationship cannot be lifted for full μFOE over a relational signature.

CONTRIBUTION (Section 3.2.3). $\mu\text{FOE} \not\subseteq \text{WMSO}$ *and hence* $\mu\text{FOE}^\infty \not\subseteq \text{WMSO}$.

However, if we consider the logic $\mu_c\text{FOE}^\infty$ obtained by restricting the fixpoint operator of μFOE^∞ to *continuous* formulas, then we are in a better shape. The following holds, even for relational signatures:

CONTRIBUTION (Section 3.2.3). *We have* $\text{FOE}^\infty \subseteq \mu_c\text{FOE}^\infty \subseteq \text{WMSO}$ *for relational signatures, but in general* $\text{WMSO} \not\subseteq \mu_c\text{FOE}^\infty$.

If we change our context and focus on the class of tree models, the situation changes dramatically. In the next section we discuss how we use automata-theoretic methods to prove many results connecting the logics of this dissertation. Among others, we prove that *on trees* we do have $\text{WMSO} \equiv \mu_c\text{FOE}^\infty$.

1.3 Logic and automata

It is difficult to over-stress the importance of automata-theoretic techniques in logical questions. Logical languages are declarative and useful to specify structural properties; automata, on the other hand, are a more explicit model of computation. The connection between these two perspectives has led to many advances in both areas. A typical example of a contribution of automata theory to logic is the proof of decidability results of logical languages using the decidability of the emptiness problem for automata. For the other direction, if a logic corresponds to certain automata, and has negation in the language, then one can obtain the closure of the automata under complementation. By automata means only, this closure is usually quite difficult to prove for some automata models.

Automata and classical logics

We start with a historical overview of automata-theoretic methods for classical logics, mostly second-order logic. Our historical account of early automaton-logic interaction is largely influenced by the survey by Wolfgang Thomas [Tho96] and by historical remarks in [Ven11, Boj04, Jac13].

Words. The connection between logic and automata can be traced back to the seminal work of Büchi [Büc60] and independently of Elgot, and Trakthenbrot [Elg61, Tra61]. To be technically correct, they were interested in studying arithmetic, or more specifically *weak (monadic) second-order arithmetic*.

In the terms of this dissertation, this logic can be described as WMSO over the class of models based on the structure $\langle \omega, \text{succ}, p_1, \dots \rangle$ of the natural numbers with a successor relation and additional unary predicates. The theory of (W)MSO over this class of structures is usually denoted (W)S1S meaning “the (weak) theory of (monadic) second-order with one successor”. Another way to see this class of structures, perhaps more natural, is to take the class of *infinite words*.

The main result of Büchi [Büc60], Elgot, and Trakthenbrot is to give finite automata (on infinite words) which corresponds to weak (monadic) second-order arithmetic. As a corollary, by observing that WMSO and MSO coincide on finite words, and giving a proper restriction of Büchi automata to finite words, one obtains the usual statement of their most celebrated theorem:

THEOREM ([Büc60, Elg61, Tra61]). *Finite state automata and MSO are effectively equivalent on finite words.*

These results were later extended by Büchi to infinite words. He introduced a type of finite automata with a new acceptance condition, that we now call the Büchi acceptance condition. Finite automata with such an acceptance condition, now called Büchi automata, accept an infinite word if it goes through a finite state infinitely often.

THEOREM ([Büc62]). *Büchi automata and MSO are effectively equivalent on infinite words.*

The reduction of formulas to automata was the key solution to obtain decidability results for (W)S1S on both finite and infinite words. In particular, this implied the decidability of important theories such as Presburger arithmetic (i.e., FOE with addition and *without multiplication*, on the natural numbers). Even nowadays, WS1S still plays a role in verification of reactive systems [HJJ⁺95].

Infinite trees. After the foundational work on words, results for trees started to appear. The first structure to be considered was the *infinite full binary tree* and its associated theories (W)S2S of “second-order of two successors”. Rabin [Rab69] introduced tree automata with a Muller acceptance condition [Mul63] and proved a correspondence for this case.

THEOREM ([Rab69]). *Tree automata and MSO are effectively equivalent on the infinite binary tree.*

This theorem subsumes previous decidability results, and one obtains many decidability results for several mathematical theories. For more detail the reader is referred to [Rab69, p. 1].

WMSO was also studied on trees. For example, the following result by Rabin gives a characterization of WMSO-definability in terms of tree automata with a Büchi acceptance condition.

THEOREM ([Rab70]). *A class L of infinite binary trees is definable in WMSO iff both L and the complement of L are recognizable by Büchi tree automata.*

A lot later, in the 90's, Muller, Saoudi and Schupp [MSS92] introduced weak alternating automata and gave a characterization for the infinite full k -ary tree for an arbitrary k (that is, for the class of models based on the full k -ary tree).

THEOREM ([MSS92]). *WMSO and weak alternating automata are effectively equivalent on the infinite full k -ary tree.*

The novelty in these automata was the introduction of a *weak acceptance condition*. The ideas behind this condition are quite important, and would continue to appear in future work, including this dissertation. The weakness condition imposes a structural restriction on the transition map of the automata. In its original formulation, it says that the domain of an weak alternating automata should be partitioned as a disjoint union $Q = \bigcup_i Q_i$ such that there is a partial order \sqsubseteq on this partition. Moreover, the transition map should respect this partition. That is, if a state $q \in Q_i$ has a transition to $q' \in Q_j$ then $Q_i \sqsubseteq Q_j$. The ultimate effect of this restriction is that every run of the automaton will ultimately stabilize inside some Q_i . As each Q_i is marked as accepting or rejecting, this gives a straightforward acceptance condition.

Recent history. In a more contemporary paper [Wal96], Walukiewicz introduced *alternating parity automata* for MSO on arbitrary trees. These automata deserve special attention, since throughout this dissertation we will use automata with *parity acceptance condition*. That is, each state a is assigned a “parity” or priority $\Omega(a)$, which is just a natural number; infinite runs are then accepting iff the minimum parity which occurs infinitely often is even.

THEOREM ([Wal96]). *The logics MSO, FO(LFP¹), and alternating parity automata are effectively equivalent on trees.*

Even though this acceptance condition seems far more unnatural than the Büchi or Muller conditions, it has a strong connection with fixpoints. Intuitively,

states with an even parity will relate to greatest fixpoints, while those with an odd parity relate to least fixpoints. Moreover, the relative priority of the states induces a “fixpoint hierarchy”. The invention of the parity condition is usually attributed to Emerson and Jutla [EJ91], Mostowski [Mos85] and Wagner while the connection of automata and fixpoints goes back to Niwiński [Niw86].

Another important milestone for this dissertation is the introduction of what we call *logical automata*. In general, the transition map of an automaton specifies the set of states that can be “reached” from a certain state.

A straightforward way to specify this set is to just give it explicitly, resulting in a transition map of the form $\delta : A \rightarrow \wp(A)$.² This type of transition map is typical of non-deterministic automata where, seen as a game, if the game is standing at a state a then an existential player \exists may choose a state from $\delta(a)$ to continue the game.

With the introduction of alternation in the game, the transition map gets more complex. For example, one could use a map $\Delta : A \rightarrow \wp(\wp(A))$ where first \exists suggests a *set* of states $A' \in \Delta(a)$ but ultimately \forall chooses the state $a \in A'$.

Another way to express alternation and specify the transition map is to use, instead of explicit sets, *logical formulas* denoting those sets. We call these logics *one-step logics*. This approach is already present in [MS85] using formulas from propositional logic (using A as the set of propositions) to specify the transition map. Janin and Walukiewicz [JL04] took this approach one step further, and gave an equivalent formulation of alternating parity automata using formulas of (relation-free) FOE in the transition map. We call these automata MSO-automata and denote them by $Aut(\text{FOE}_1)$. We will see in later sections that having such a transition map allows us to get results for the full class of automata by focusing on properties of the one-step logic.

It is natural to ask if the relation between WMSO and weak automata on k -ary trees lifts to arbitrary trees. Zanasi [Zan12] shows that considering arbitrary branching on trees poses fundamental problems. In this setting, weak parity automata do not correspond to WMSO but to a logic called WFMSO which quantifies over subsets of well-founded trees.³

THEOREM ([Zan12]). *Weak alternating parity automata and WFMSO are effectively equivalent on trees.*

In another effort to characterize WMSO, Jacobi [Jac13] introduced weak parity automata using (relation-free) WMSO as a one-step logic. However, these

²Transition maps are usually more complex even in the most basic cases. For example, finite state automata on words already have a map of the form $\delta : A \times \Sigma \rightarrow \wp(A)$ which specifies the possible next states given the current state and a letter from the alphabet Σ . In this introduction we choose to simplify the left hand side for presentation purposes.

³The results in [Zan12] are stated for arbitrarily branching trees where all their branches are infinite. However, the results easily generalize to arbitrary trees.

automata (which we denote $Aut_w(\text{WMSO}_1)$) are shown to be too strong to correspond to WMSO on trees.

THEOREM ([Jac13]). $\text{WMSO} \subsetneq Aut_w(\text{WMSO}_1)$ on trees.

Our contributions. Our first contribution is to give a class of alternating parity automata which corresponds to WMSO on arbitrary trees. First observe that by König’s lemma, a subset of a tree \mathbb{T} is finite iff it is both a subset of a finitely branching subtree of \mathbb{T} and *well-founded*, that is, a subset of a subtree of \mathbb{T} that has no infinite branches. This suggests that we may change the definition of MSO-automata into one of WMSO-automata via two kinds of modifications, roughly speaking corresponding to a horizontal and a vertical ‘dimension’ of trees.

For the ‘vertical modification’ we can use the weakness condition which, as shown by Zanasi, gives a logic which quantifies over well-founded sets. The hurdle to take, in order to find automata for WMSO on trees of *arbitrary* branching degree, concerns the horizontal dimension; the main problem lies in finding the right one-step language for WMSO-automata.

An obvious candidate for this language would be weak monadic second-order logic itself, or more precisely, its variant WMSO_1 over the signature of monadic predicates (corresponding to the automata states), as done in [Jac13].⁴ A very helpful observation, made by Väänänen [Vää77], states that:

$$\text{WMSO}_1 \equiv \text{FOE}_1^\infty,$$

where FOE_1^∞ is the extension of FOE_1 with the generalized quantifier $\exists^\infty x.\varphi$. Taking the *full* language of WMSO_1 or FOE_1^∞ as our one-step language would give too much expressive power, as shown by Jacobi. It is here that we will crucially involve the notion of *continuity*.

To define our automata for WMSO first observe that for every automaton \mathbb{A} which has formulas on its transition map using the states A as propositions, we can associate a graph where $a, b \in A$ are connected if $b \in \Delta(a)$. We say that a parity automaton \mathbb{A} is *continuous-weak* if for every *maximal* strongly connected component $C \subseteq A$ and states $a, b \in C$ the following conditions hold:⁵

(weakness) $\Omega(a) = \Omega(b)$,

(continuity) If $\Omega(a)$ is odd then the formula $\Delta(a)$ is continuous in the states C .
if $\Omega(a)$ is even then the formula $\Delta(a)$ is co-continuous in the states C .

The effect of the (weakness) condition can be proved to be equivalent to the one based on an ordering \sqsubseteq , for our context. An WMSO-automaton is then defined

⁴Our work on WMSO was done independently of that of Jacobi [Jac13]. Our motivation to use FOE_1^∞ as a one-step language came from the observation by Väänänen [Vää77].

⁵The actual definition that we use has a slightly more complex transition map, but we do not discuss that here to keep the presentation simple.

as a continuous-weak automaton based on FOE_1^∞ ; we denote the class of such automata as $\text{Aut}_{wc}(\text{FOE}_1^\infty)$.

We prove that these automata precisely correspond to WMSO on arbitrary trees and in the process we also obtain a characterization of these formalisms as a fixpoint logic.

CONTRIBUTION (Section 7.2). *$\text{Aut}_{wc}(\text{FOE}_1^\infty)$, $\mu_c\text{FOE}^\infty$ and WMSO are effectively equivalent on trees.*

Our second contribution is to give automata for WCL. In this case, we introduce a new type of additive-weak automata, which replaces the (continuity) restriction of the above automata by an (additivity) restriction.

(additivity) If $\Omega(a)$ is odd then $\Delta(a)$ is completely additive in C .
if $\Omega(a)$ is even then $\Delta(a)$ is completely multiplicative in C .

However, in this case, we use FOE_1 as the one-step language. We denote the class of WCL-automata as $\text{Aut}_{wa}(\text{FOE}_1)$.

CONTRIBUTION (Section 7.3). *$\text{Aut}_{wa}(\text{FOE}_1)$ and WCL are effectively equivalent on trees.*

In the last section of Chapter 7 we discuss how our work makes some progress in the quest of developing (logical) parity automata for $\text{FO}(\text{TC}^1)$, and possible approaches to attack this problem.

Automata and modal logics

Some automata models for the μ -calculus were already introduced in [Niw86] and [JW95]. However, the logical automata that we use in this dissertation was first introduced in [JW96]. These automata are remarkably similar to MSO-automata: they are (logical) alternating parity automata, but the main difference is in the one-step language. While MSO-automata use FOE_1 as their one-step language, automata for the μ -calculus use FO_1 . That is, (relation-free) first-order logic *without equality*. Following our notation, we denote this class of automata as $\text{Aut}(\text{FO}_1)$.

THEOREM ([JW95]). *$\text{Aut}(\text{FO}_1)$ and the modal μ -calculus are effectively equivalent on all models.*

Among other results, automata for the μ -calculus were crucial in proving uniform interpolation for this logic [DH00] and also proving that the μ -calculus is the bisimulation-invariant fragment of MSO (more on this in the next section). Moreover, the above equivalence can be used to give a highly simplified proof of Rabin's complementation lemma [JW95, Remark 5.8].

A very useful observation is that if we add the weakness constraint to automata for the μ -calculus, we obtain automata for AFMC. That is, the weakness condition forces the resulting formula to be alternation-free. We use $Aut_w(\text{FO}_1)$ to denote the class of weak parity automata based on the one-step logic FO_1 .

THEOREM ([AN92, KV98]). *$Aut_w(\text{FO}_1)$ and the alternation-free μ -calculus are effectively equivalent on all models.*

These automata are of particular interest for model checking, since many temporal logics can be translated to them [KVW00].

The first automata result for PDL was by Streett [Str81, Str82] who translates PDL (with additional looping and converse operators) to “deterministic two-way automata on infinite trees” and obtains decidability for the satisfiability problem. Later, Vardi and Wolper [VW86] showed that PDL can be translated to Büchi (tree) automata and get sharper complexity results. Muller et al. [MSS88] showed that many dynamic and temporal logics can be uniformly represented using weak alternating automata (see last section). This class of automata recognizes languages that are Büchi and co-Büchi and, on trees, they accept exactly the languages definable in Weak MSO [MSS92, Rab70].

The mentioned papers, among others, use automata-theoretic techniques to prove results about PDL, usually giving a translation into some kind of automata. However, none of them gives a precise characterization. That is, the class of automata in consideration has automata that do not correspond to an equivalent PDL formula.

A further contribution of this dissertation is to define two classes of alternating parity automata (in the spirit of μ -automata) which exactly correspond, respectively, to PDL and its test-free variant PDL^{tf} .

When looking at μ -automata one can see that cycles naturally encode a notion of repetition (or fixpoint, in this case). When considering PDL, it is obvious that not any kind of “action” can be repeated. That is, repetitions can only be done over programs. The slogan that drives the definition of automata for PDL is:

“Maximal strongly connected components correspond to programs.”

Moreover, we will see that connected components correspond to programs crucially involving the iteration operator. Using the insight obtained by the characterization of PDL as a fragment of μML , we define automata for PDL by restricting μ -automata with both the (weakness) and (additivity) conditions. We denote the class of these automata with $Aut_{wa}(\text{FO}_1)$.

CONTRIBUTION (Section 6.1 and 6.2). *The following are equivalent on all models:*

- (i) $Aut_{wa}(\text{FO}_1)$ and PDL,
- (ii) $Aut_{wa}^-(\text{FO}_1)$ and test-free PDL.

Moreover, the equivalences are effective.

Using a similar approach, we can restrict μ -automata with the (weakness) and (continuity) constraints. In this case we obtain automata for $\mu_c\text{ML}$. These automata will play a role in the next section, where we give a characterization of the bisimulation-invariant fragment of WMSO.

CONTRIBUTION (Section 6.3). *Aut_{wc}(FO₁) and $\mu_c\text{ML}$ are effectively equivalent on all models.*

1.4 Expressiveness modulo bisimilarity

The last chapter of this dissertation concerns the relative expressive power of some languages when restricted to properties which are bisimulation-invariant. The interest in such expressiveness questions stems from applications where transition systems model computational processes, and bisimilar structures represent the *same* process. Seen from this perspective, properties of transition structures are relevant only if they are invariant under bisimilarity. This explains the importance of bisimulation-invariance results of the form

$$L' \equiv L / \leftrightarrow,$$

stating that, one language L' is expressively complete with respect to the relevant (i.e., bisimulation-invariant) properties that can be formulated in another language L . In this setting, generally L is some rich yardstick formalism such as first-order or monadic second-order logic, and L' is some modal-style fragment of L , usually displaying much better computational behavior than the full language L .

A seminal result in the theory of modal logic is van Benthem's Characterization Theorem [Ben77], stating that every bisimulation-invariant first-order formula is actually equivalent to (the standard translation of) a modal formula:

$$\text{ML} \equiv \text{FOE} / \leftrightarrow.$$

Over the years, a wealth of variants of the Characterization Theorem have been obtained. For instance, Rosen proved that van Benthem's theorem is one of the few preservation results that transfers to the setting of finite models [Ros97]; for a recent, rich source of van Benthem-style characterization results, see Dawar and Otto [DO09]. In this dissertation we are mainly interested in the work of Janin and Walukiewicz [JW96], who extended van Benthem's result to the setting of fixpoint logics, by proving that the modal μ -calculus is the bisimulation-invariant fragment of monadic second-order logic:

$$\mu\text{ML} \equiv \text{MSO} / \leftrightarrow.$$

Despite the continuous study of the connection between modal and classical logics there are still important logics which are not well understood and represent exciting problems. In particular, the bisimulation-invariant fragments of WMSO and WCL have not been characterized. Also, it is not known whether there is a natural classical logic whose bisimulation-invariant fragment corresponds to PDL (see [Hol98, p. 91]), even though there are results leading towards this direction [Ben98, Hol98, Ben96].

In this dissertation we crucially use the new parity automata developed for WMSO, WCL, PDL, and $\mu_c\text{ML}$ to obtain bisimulation-invariance results.

The one-step approach. When proving results of the form $L' \equiv L/\leftrightarrow$, one of the inclusions is usually given by a translation from L ; to L' . The inclusion $L' \supseteq L/\leftrightarrow$, however, requires much more work. In the context of fixpoint logics, the use of automata is a powerful technique to prove this direction.

In the original work of Janin and Walukiewicz [JW96], an important step of the proof is to define a construction $(-)^{\bullet}$ that transforms automata from $\text{Aut}(\text{FOE}_1)$, which correspond to MSO, to automata of $\text{Aut}(\text{FO}_1)$, which correspond to μML . A key observation made by Venema in [Ven14] is that the construction $(-)^{\bullet}$ is completely determined at the one-step level, by a translation $(-)^{\bullet}_1 : \text{FOE}_1 \rightarrow \text{FO}_1$ satisfying certain properties. Intuitively, if we prove that $\text{FOE}_1/\leftrightarrow \equiv \text{FO}_1$ (recall that these are relation-free logics) then we can get $\text{MSO}/\leftrightarrow \equiv \mu\text{ML}$ without too much work. We call this technique the “one-step approach” to bisimulation-invariance proofs.

In this dissertation we show that this technique provides a nice modular way of giving (automata-based) bisimulation-invariance proofs and, moreover, it works for subclasses of parity automata. This one-step approach, however, requires a detailed development of the model theory of the one-step logics in use.

CONTRIBUTION (Chapter 5). *We give normal forms and syntactic characterizations of the monotone, continuous and completely additive fragments of many (multi-sorted) one-step logics; among others we study the (relation-free) first-order languages FO_1 , FOE_1 , FOE_1^{∞} and the modal language ML_1 .*

Weak MSO. Of particular importance in the setting of weak monadic second-order logic is the difference between structures of finite versus arbitrary branching degree. In the case of finitely branching models, it is not very hard to show that WMSO is a (proper) fragment of MSO, and it seems to be folklore that $\text{WMSO}/\leftrightarrow$ corresponds to AFMC, the alternation-free fragment of the modal μ -calculus. For binary trees, this result was proved by Arnold & Niwiński in [AN92]. In the case of structures of arbitrary branching degree, however, WMSO and MSO have *incomparable* expressive power.

For this reason, the relative expressive power of $\text{WMSO}/\leftrightarrow$ and $\text{MSO}/\leftrightarrow$ is not a priori clear. However, it is reasonable to think that $\text{WMSO}/\leftrightarrow$ is strictly

weaker than AFMC: the class of well-founded trees, which is definable in AFMC by the simple formula $\mu p.\Box p$, is not definable in WMSO. In this dissertation we show that $\text{WMSO}/\leftrightarrow$ is indeed smaller than the alternation-free μ -calculus; moreover, it coincides with $\mu_c\text{ML}$.

CONTRIBUTION (Section 8.1.1). $\text{WMSO}/\leftrightarrow$ is effectively equivalent to $\mu_c\text{ML}$.

Propositional Dynamic Logic. Another contribution of this dissertation is to give a characterization of PDL as the bisimulation-invariant fragment of a second-order logic.

CONTRIBUTION (Section 8.2.1). PDL is effectively equivalent to $\text{WCL}/\leftrightarrow$.

This characterization is admittedly not the most natural one. It is thought in the modal logic community that, as $\text{FOE}/\leftrightarrow \equiv \text{ML}$, then it would be natural to have $\text{FO}(\text{TC}^1)/\leftrightarrow \equiv \text{PDL}$. However, there seems to be no proof of this result. In the last part of Chapter 8 we go into a discussion on the bisimulation-invariant fragment of $\text{FO}(\text{TC}^1)$. We discuss how our work makes progress towards it, and possible approaches to obtain it.

1.5 Source of the material

The distribution of the material in this dissertation is somewhat atypical. This manuscript contains extended and hopefully corrected versions of results published in [CV14, CFVZ14a, CFVZ14b, Car15]. However, there is no direct mapping between the chapters of this dissertation and the mentioned articles.

The results on PDL are joint work with Yde Venema [CV14], and appear in Chapter 3 (syntactic fragments) and Chapter 6 (automata characterizations). The automata and bisimulation-invariance characterization for WCL were published in [Car15] and appear respectively in Chapter 7 and Chapter 8.

The automata characterization of WMSO and $\mu_c\text{ML}$, and the characterization of the bisimulation-invariance fragment of WMSO were given in [CFVZ14a, CFVZ14b] and appear respectively in Chapter 7, Chapter 6 and Chapter 8. This is joint work with Alessandro Facchini, Yde Venema and Fabio Zanasi.

The results and discussions for Concurrent PDL and Game Logic are unpublished, and were obtained in collaboration with Alessandro Facchini during my research visit to the University of Warsaw in 2014.

The analysis of one-step logics done in Chapter 5 is partially taken from the papers [CFVZ14a] and [Car15] but some of the results are still unpublished. The rest of the results and discussions, including the equivalence of $\text{FO}(\text{TC}^1)$ and $\mu_a\text{FOE}$, are mostly unpublished.

2.1 Terminology, transition systems and trees

Throughout this dissertation we fix a set P of elements that will be called *proposition letters* and denoted with small Latin letters p, q , etc. We also fix a set D of *atomic actions*, which will usually correspond to the relations available in our transition systems.

We use overlined boldface letters to represent sequences, for example a list of variables $\bar{x} := x_1, \dots, x_n$ or a sequence of sets $\bar{T} \in \wp(S)^n$. We blur the distinction between sets and sequences: a sequence may be used as a set comprised of the elements of the list; in a similar way, we may assume a fixed order on a set and see it as a list. To simplify notation, given a map $f : A^{n+m} \rightarrow B$ and $\bar{a} \in A^n$, $\bar{a}' \in A^m$ we write $f(\bar{a}, \bar{a}')$ to denote $f(a_1, \dots, a_n, a'_1, \dots, a'_m)$.

Given a binary relation $R \subseteq X \times Y$, for any element $x \in X$, we use $R[x]$ to denote the set $\{y \in Y \mid (x, y) \in R\}$ of R -successors of x . The relations R^+ and R^* are defined respectively as the transitive closure of R and the reflexive and transitive closure of R . The set $\text{Ran}(R)$ is defined as $\bigcup_{x \in X} R[x]$. Sometimes we write p instead of $\{p\}$ and Qp instead of $Q \cup \{p\}$.

Transition systems. A *labeled transition system* (LTS) on the set of propositions P and actions D is a tuple $\mathbb{S} = \langle S, R_{\ell \in D}, \kappa, s_I \rangle$ where S is the universe or domain of \mathbb{S} ; the map $\kappa : S \rightarrow \wp(P)$ is a marking (or coloring) of the elements in S ; $R_\ell \subseteq S^2$ is the accessibility relation for the atomic action $\ell \in D$; and $s_I \in S$ is a distinguished node. We use R without a subscript to denote the binary relation defined as $R := \bigcup_{\ell \in D} R_\ell$. Given a transition system \mathbb{S} and $s \in S$ we use “ \mathbb{S}, s ” to denote the transition system which is exactly like \mathbb{S} but where the distinguished point has been changed from s_I to s .

Observe that a marking $\kappa : S \rightarrow \wp(P)$ can be seen as a valuation $\kappa^\sharp : P \rightarrow \wp(S)$ given by $\kappa^\sharp(p) := \{s \in S \mid p \in \kappa(s)\}$. We say that \mathbb{S} is *p-free* if $p \notin P$ or $p \notin \kappa(s)$ for all $s \in S$. Given a transition system based on P' and a set $X_p \subseteq S$, we use

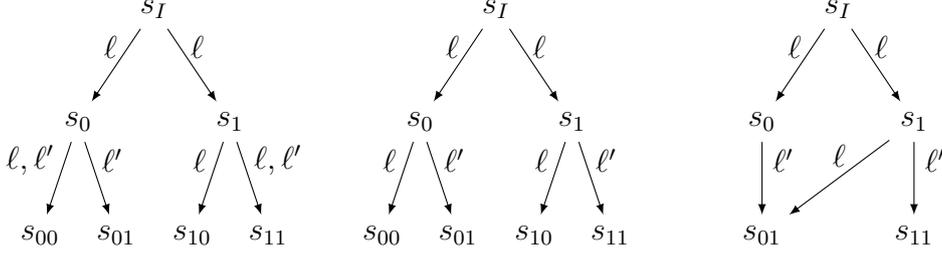


Figure 2.1: A tree, a strict tree, and an LTS which is not a tree.

$\mathbb{S}[p \mapsto X_p]$ to denote the transition system $\langle S, R_{\ell \in \mathbb{D}}, \kappa', s_I \rangle$ over $P' \cup \{p\}$ such that for all $s \in S$ we have $\kappa'(s) \setminus \{p\} = \kappa(s)$ and $p \in \kappa'(s)$ iff $s \in X_p$. Observe that if $p \in P'$, the value of $\kappa^{\sharp}(p)$ basically gets redefined, and if $p \notin P'$ then κ^{\sharp} gets extended with $\kappa^{\sharp}(p) := X_p$. We use $\mathbb{S}[p \upharpoonright X_p]$ to denote $\mathbb{S}[p \mapsto \kappa^{\sharp}(p) \cap X_p]$. This notation extends to tuples of elements, as follows:

$$\begin{aligned} \kappa^{\sharp}(\bar{\mathbf{q}}) &:= \kappa^{\sharp}(q_1), \dots, \kappa^{\sharp}(q_n) \\ \mathbb{S}[\bar{\mathbf{q}} \mapsto \bar{\mathbf{X}}] &:= \mathbb{S}[q_i \mapsto X_i \mid 1 \leq i \leq n] \\ \mathbb{S}[\bar{\mathbf{q}} \upharpoonright \bar{\mathbf{X}}] &:= \mathbb{S}[q_i \mapsto \kappa^{\sharp}(q_i) \cap X_i \mid 1 \leq i \leq n]. \end{aligned}$$

Trees. A P -tree \mathbb{T} is a transition system over P in which every node can be reached from s_I , and every node except s_I has a unique R -predecessor; the distinguished node s_I is called the *root* of \mathbb{T} . A tree is called *strict* when $\text{Ran}(R_{\ell}) \cap \text{Ran}(R_{\ell'}) = \emptyset$ for every $\ell \neq \ell'$. Also observe that if there is only one relation R , the notion of tree and strict tree coincide.

Each node $s \in T$ uniquely defines a subtree of \mathbb{T} with carrier $R^*[s]$ and root s . We denote this subtree by $\mathbb{T}.s$. We use the term *tree language* as a synonym of class of trees.

A *path* through \mathbb{S} is a sequence $\pi = (s_i)_{i < \alpha}$ of elements of S , where α is either ω or a natural number, and $(s_i, s_{i+1}) \in R$ for all i with $i + 1 < \alpha$. The *tree unraveling* of a transition system \mathbb{S} is given by $\hat{\mathbb{S}} := \langle \hat{S}, \hat{R}_{\ell \in \mathbb{D}}, \hat{\kappa}, s_I \rangle$ where \hat{S} is the set of (\mathbb{D} -decorated) finite paths $s_I \rightarrow_{\ell_1} e_1 \rightarrow_{\ell_2} \dots \rightarrow_{\ell_n} e_n$ in \mathbb{S} stemming from s_I ; $\hat{R}_{\ell}(t, t')$ holds iff t' is an extension of t through the relation ℓ ; and the color of a path $t \in \hat{S}$ is given by the color of its last node in S . The ω -unraveling \mathbb{S}^{ω} of \mathbb{S} is an unraveling which has ω -many copies of each node different from the root.

2.1.1. REMARK. The (ω) -unraveling of a transition system is a *strict* tree.

Chains and generalized chains. Let \mathbb{S} be an arbitrary transition system. A *chain* on \mathbb{S} is a set $X \subseteq S$ such that (X, R^*) is a totally ordered set; i.e., the following conditions are satisfied for every $x, y \in X$:

(antisymmetry) if xR^*y and yR^*x then $x = y$,

(transitivity) if xR^*y and yR^*z then xR^*z ,

(totality) xR^*y or yR^*x .

A *finite chain* is a chain based on a finite set. A *generalized chain* is a set $X \subseteq S$ such that $X \subseteq P$, for some path P of \mathbb{S} . A *generalized finite chain* is a finite subset $X \subseteq S$ such that $X \subseteq P$, for some finite path P of \mathbb{S} .

2.1.2. FACT. *Every chain on \mathbb{S} is also a generalized chain on \mathbb{S} .*

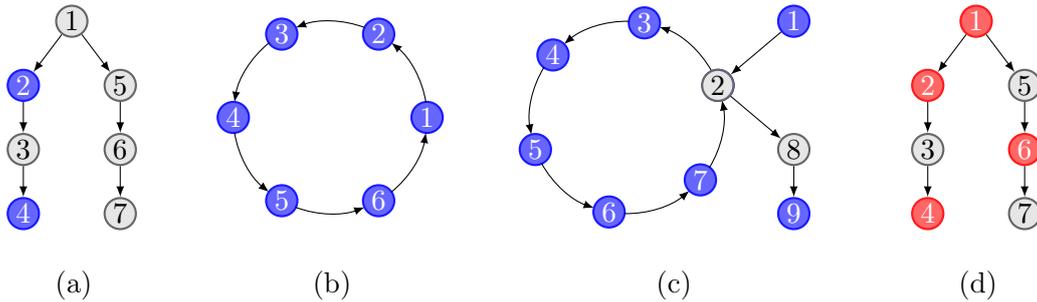


Figure 2.2: Examples and counter-examples of chains.

In Fig. 2.2 we show some examples of (generalized) chains and non-chains: in (a) the set $X_a = \{2, 4\}$ is a finite chain. In (b) the generalized finite chain $X_b = \{1, 2, 3, 4, 5, 6\}$ is witnessed, among others, by the path $3 \rightarrow 4 \rightarrow 5 \rightarrow 6 \rightarrow 1 \rightarrow 2$. Observe, however, that X_b is *not* a chain, since there is no possible total ordering of X_b by R^* (antisymmetry fails). In (c) the generalized finite chain $X_c = \{1, 3, 4, 5, 6, 7, 9\}$ is witnessed by the path $1 \rightarrow 2 \rightarrow \dots \rightarrow 7 \rightarrow 2 \rightarrow 8 \rightarrow 9$; observe that the element 2 is repeated in the path. Again, X_c is not a finite chain. In the last example (d), the set $X_d = \{1, 2, 4, 6\}$ is not a generalized chain (and hence not a chain).

The following proposition states a useful relationship between chains and generalized chains: on trees this distinction vanishes.

2.1.3. PROPOSITION. *On trees, chains and general chains coincide.*

Proof. Observe that every path on a tree \mathbb{T} sits inside some branch of \mathbb{T} . Therefore every generalized chain X can be embedded in some branch of \mathbb{T} and hence (X, R^*) will be a total order. The key concept in the background is that on trees there are no cycles. \square

2.2 Games

We introduce some terminology and background on infinite games. All the games that we consider involve two players called *Éloise* (\exists) and *Abelard* (\forall). In some

contexts we refer to a player Π to specify a generic player in $\{\exists, \forall\}$. Given a set A , by A^* and A^ω we denote respectively the set of words (finite sequences) and streams (or infinite words) over A .

A *board game* \mathcal{G} is a tuple $(G_\exists, G_\forall, E, \text{Win})$, where G_\exists and G_\forall are disjoint sets whose union $G = G_\exists \cup G_\forall$ is called the *board* of \mathcal{G} , $E \subseteq G \times G$ is a binary relation encoding the *admissible moves*, and $\text{Win} \subseteq G^\omega$ is a *winning condition*. An *initialized board game* $\mathcal{G}@u_I$ is a tuple $(G_\exists, G_\forall, u_I, E, \text{Win})$ where $u_I \in G$ is the *initial position* of the game. A special case of winning condition is induced by a parity function $\Omega : G \rightarrow \mathbb{N}$ by defining

$$\text{Win}_\Omega := \{g \in G^\omega \mid \text{the minimum parity occurring infinitely often in } g \text{ is even}\}.$$

In this case, we say that \mathcal{G} is a parity game and write $\mathcal{G} = (G_\exists, G_\forall, E, \Omega)$.

Given a board game \mathcal{G} , a *match* of \mathcal{G} is simply a path through the graph (G, E) ; that is, a sequence $\pi = (u_i)_{i < \alpha}$ of elements of G , where α is either ω or a natural number, and $(u_i, u_{i+1}) \in E$ for all i with $i + 1 < \alpha$. A match of $\mathcal{G}@u_I$ is supposed to start at u_I . Given a finite match $\pi = (u_i)_{i < k}$ for some $k < \omega$, we call $\text{last}(\pi) := u_{k-1}$ the *last position* of the match; the player Π such that $\text{last}(\pi) \in G_\Pi$ is supposed to move at this position, and if $E[\text{last}(\pi)] = \emptyset$, we say that Π *got stuck* in π . A match π is called *total* if it is either finite, with one of the two players getting stuck, or infinite. Matches that are not total are called *partial*. Any total match π is *won* by one of the players: If π is finite, then it is won by the opponent of the player who gets stuck. Otherwise, if π is infinite, the winner is \exists if $\pi \in \text{Win}$, and \forall if $\pi \notin \text{Win}$.

Given a board game \mathcal{G} and a player Π , let $\text{PM}_\Pi^{\mathcal{G}}$ denote the set of partial matches of \mathcal{G} whose last position belongs to player Π . A *strategy* for Π is a function $f : \text{PM}_\Pi^{\mathcal{G}} \rightarrow G$. A match $\pi = (u_i)_{i < \alpha}$ of \mathcal{G} is *f-guided* if for each $i < \alpha$ such that $u_i \in G_\Pi$ we have that $u_{i+1} = f(u_0, \dots, u_i)$. Let $u \in G$ and f be a strategy for Π . We say that f is a *surviving strategy* for Π in $\mathcal{G}@u$ if

- (i) For each f -guided partial match π of $\mathcal{G}@u$, if $\text{last}(\pi)$ is in G_Π then $f(\pi)$ is legitimate, that is, $(\text{last}(\pi), f(\pi)) \in E$.

We say that f is a *winning strategy* for Π in $\mathcal{G}@u$ if, additionally,

- (ii) Π wins each f -guided total match of $\mathcal{G}@u$.

If Π has a winning strategy for $\mathcal{G}@u$ then u is called a *winning position* for Π in \mathcal{G} . The set of positions of \mathcal{G} that are winning for Π is denoted by $\text{Win}_\Pi(\mathcal{G})$. A strategy f is called *positional* if $f(\pi) = f(\pi')$ for each $\pi, \pi' \in \text{Dom}(f)$ with $\text{last}(\pi) = \text{last}(\pi')$. A board game \mathcal{G} with board G is *determined* if $G = \text{Win}_\exists(\mathcal{G}) \cup \text{Win}_\forall(\mathcal{G})$, that is, each $u \in G$ is a winning position for one of the two players. For parity games, strategies can be assumed to be positional and every game is determined.

2.2.1. FACT ([EJ91, MOS91]). *For every parity game \mathcal{G} , there are positional strategies f_\exists and f_\forall respectively for player \exists and \forall , such that for every position $u \in G$ there is a player Π such that f_Π is a winning strategy for Π in $\mathcal{G}@u$.*

From now on, we always assume that each strategy we work with in parity games is positional. Moreover, we will think of a positional strategy f_{II} for player II as a function $f_{\text{II}} : G_{\text{II}} \rightarrow G$.

2.3 Parity automata

We recall the definition of a parity automaton, adapted to our setting. As we will be running parity automata over transition systems with many relations, we will need to use multi-sorted one-step models. Intuitively, each sort corresponds to one of the relations of the transition system. Since we will be comparing parity automata defined in terms of various one-step languages, it makes sense to make the following abstraction.

2.3.1. DEFINITION. Given a finite set A and sorts $\mathcal{S} = \{s_1, \dots, s_n\}$, we define a *one-step model* to be a tuple $\mathbf{D} = (D_{s_1}, \dots, D_{s_n}, V)$ consisting of sets D_{s_1}, \dots, D_{s_n} and a valuation $V : A \rightarrow \wp(\bigcup_{s \in \mathcal{S}} D_s)$. We use D to denote the set $\bigcup_{s \in \mathcal{S}} D_s$ which we call the domain of \mathbf{D} . A one-step model is called *strict* when the sets $D_{s \in \mathcal{S}}$ are pairwise disjoint, that is, when D_{s_1}, \dots, D_{s_n} provide a partition of D . Depending on context, elements of A will be called *monadic predicates*, *names* or *propositional variables*. When the sets $D_{s \in \mathcal{S}}$ are not relevant we will just write the one-step model as (D, V) . The class of all one-step models will be denoted by \mathfrak{M}_1 and the class of all strict one-step models will be denoted by \mathfrak{M}_1^s .

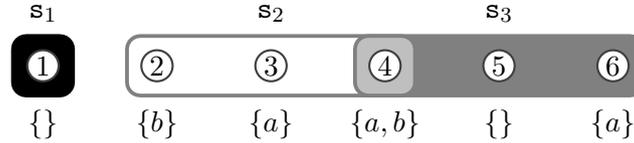


Figure 2.3: One-step model with sorts (above) and valuation (below).

A (*multi-sorted*) *one-step language* is a map \mathcal{L} assigning to each finite set A and sorts \mathcal{S} , a set $\mathcal{L}(A, \mathcal{S})$ of objects called *one-step formulas* over A (on sorts \mathcal{S}). When the sorts are understood from context (or fixed) we simply write $\mathcal{L}(A)$ instead of $\mathcal{L}(A, \mathcal{S})$. We require that $\mathcal{L}(\bigcap_i A_i, \mathcal{S}) = \bigcap_i \mathcal{L}(A_i, \mathcal{S})$, so that for each $\varphi \in \mathcal{L}(A, \mathcal{S})$ there is a smallest $A_\varphi \subseteq A$ such that $\varphi \in \mathcal{L}(A_\varphi, \mathcal{S})$; this A_φ is the set of names that *occur* in φ .

We assume that one-step languages come with a *truth* relation: given a one-step model \mathbf{D} , a formula $\varphi \in \mathcal{L}$ is either *true* or *false* in \mathbf{D} , denoted by, respectively, $\mathbf{D} \models \varphi$ and $\mathbf{D} \not\models \varphi$. We also assume that \mathcal{L} has a *positive fragment* \mathcal{L}^+ characterizing monotonicity. We say that a formula $\varphi \in \mathcal{L}(A, \mathcal{S})$ is monotone in $a \in A$ iff $(D, V) \models \varphi$ implies $(D, V[a \mapsto E]) \models \varphi$ whenever $V(a) \subseteq E$. Hence, we require that $\varphi \in \mathcal{L}(A, \mathcal{S})$ is monotone in all $a \in A$ iff it is equivalent to a formula $\varphi' \in \mathcal{L}^+(A, \mathcal{S})$.

Observe that every valuation $V : A \rightarrow \wp(D)$ can equivalently be seen as a marking (or coloring) $V^\natural : D \rightarrow \wp(A)$ given by $V^\natural(d) := \{a \in A \mid d \in V(a)\}$ and as a relation $Z_V := \{(a, d) \mid d \in V(a)\}$. We will use these perspectives interchangeably.

2.3.2. DEFINITION. A *parity automaton* based on the one-step language \mathcal{L} , actions \mathbb{D} and alphabet $\wp(\mathbb{P})$ is a tuple $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ such that A is a finite set of states of the automaton, $a_I \in A$ is the initial state, $\Delta : A \times \wp(\mathbb{P}) \rightarrow \mathcal{L}^+(A, \mathbb{D})$ is the transition map, and $\Omega : A \rightarrow \mathbb{N}$ is the parity map. The collection of such automata will be denoted by $Aut(\mathcal{L}, \mathbb{P}, \mathbb{D})$. For the rest of the manuscript we fix the set of actions \mathbb{D} and omit it in our notation; we also omit the set \mathbb{P} when clear from context or irrelevant.

Acceptance and rejection of a transition system by an automaton is defined in terms of the following parity game.

2.3.3. DEFINITION. Given an automaton $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ in $Aut(\mathcal{L}, \mathbb{P})$ and a \mathbb{P} -transition system $\mathbb{S} = \langle S, R_{\ell \in \mathbb{D}}, \kappa, s_I \rangle$, the *acceptance game* $\mathcal{A}(\mathbb{A}, \mathbb{S})$ of \mathbb{A} on \mathbb{S} is the parity game defined according to the rules of the following table.

Position	Pl'r	Admissible moves	Parity
$(a, s) \in A \times S$	\exists	$\{V : A \rightarrow \wp(R[s]) \mid (R[s], V) \models \Delta(a, \kappa(s))\}$	$\Omega(a)$
$V : A \rightarrow \wp(S)$	\forall	$\{(b, t) \mid t \in V(b)\}$	$\max(\Omega[A])$

In this case $(R[s], V)$ denotes $(R_{\ell_1}[s], \dots, R_{\ell_n}[s], V)$. A transition system \mathbb{S} is *accepted* by \mathbb{A} if \exists has a winning strategy in $\mathcal{A}(\mathbb{A}, \mathbb{S})@(a_I, s_I)$, and *rejected* if (a_I, s_I) is a winning position for \forall .

Given an automaton \mathbb{A} and a transition system \mathbb{S} we write $\mathbb{S} \Vdash \mathbb{A}$ and $\mathbb{S} \models \mathbb{A}$ when \mathbb{A} accepts \mathbb{S} . The former notation is used when the one-step language of the automaton is modal, and the latter notation is used when the one-step language is first-order (more on this will be discussed in Chapter 4). Given a state $a \in A$ we use “ \mathbb{A}, a ” to denote the automaton which is like \mathbb{A} but where the initial state is now a .

2.3.4. DEFINITION. Observe that given a parity automaton \mathbb{A} we can induce a graph on A by setting a transition from a to b (notation: $a \rightsquigarrow b$) if b occurs in $\Delta(a, c)$ for some $c \in \wp(\mathbb{P})$. We let the *reachability* relation \preceq denote the reflexive-transitive closure of the relation \rightsquigarrow .

A *strongly connected component* (SCC) of an automaton \mathbb{A} is a subset $C \subseteq A$ such that for every $b, c \in C$ we have $b \preceq c$ and $c \preceq b$. The SCC is called *maximal* (MSCC) when no proper extension of C is an SCC.

Closure under complementation. Many properties of parity automata can already be determined at the one-step level. An important example concerns the notion of complementation.

2.3.5. DEFINITION. Two one-step formulas φ and ψ are each other's *Boolean dual* if for every structure (D, V) we have:

$$(D, V) \models \varphi \quad \text{iff} \quad (D, V^c) \not\models \psi,$$

where V^c is the valuation given by $V^c(a) := D \setminus V(a)$, for all a . A one-step language \mathcal{L} is *closed under Boolean duals* if for every set A , each formula $\varphi \in \mathcal{L}(A)$ has a Boolean dual $\varphi^\delta \in \mathcal{L}(A)$.

Following ideas from [MS87, KV09], we can use Boolean duals, together with a *role switch* between \forall and \exists , in order to define a negation or complementation operation on automata.

2.3.6. DEFINITION. Assume that, for some one-step language \mathcal{L} , the map $(-)^{\delta}$ provides, for each set A , a Boolean dual $\varphi^\delta \in \mathcal{L}(A)$ for each $\varphi \in \mathcal{L}(A)$. Given $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ in $\text{Aut}(\mathcal{L})$ we define its *complement* \mathbb{A}^δ as the automaton $\langle A, \Delta^\delta, \Omega^\delta, a_I \rangle$ where $\Delta^\delta(a, c) := (\Delta(a, c))^\delta$, and $\Omega^\delta(a) := 1 + \Omega(a)$, for all $a \in A$ and $c \in \wp(\mathbf{P})$.

2.3.7. PROPOSITION. *Let \mathcal{L} and $(-)^{\delta}$ be as in the previous definition. For each automaton $\mathbb{A} \in \text{Aut}(\mathcal{L})$ and each transition system \mathbb{S} we have that*

$$\mathbb{A}^\delta \text{ accepts } \mathbb{S} \quad \text{iff} \quad \mathbb{A} \text{ rejects } \mathbb{S}.$$

The proof of Proposition 2.3.7 is based on the fact that the power of \exists in $\mathcal{A}(\mathbb{A}^\delta, \mathbb{S})$ is the same as that of \forall in $\mathcal{A}(\mathbb{A}, \mathbb{S})$, as defined in [KV09].

As an immediate consequence of this proposition, one may show that if the one-step language \mathcal{L} is closed under Boolean duals, then the class $\text{Aut}(\mathcal{L})$ is closed under taking complementation. Further on in Chapter 4 we will use Proposition 2.3.7 to show that the same may apply to some subclasses of $\text{Aut}(\mathcal{L})$.

2.4 The modal μ -calculus

The modal μ -calculus extends the basic modal language with a mechanism for forming least (and greatest) fixpoints.

2.4.1. DEFINITION. The language of the modal μ -calculus (μML) on propositions \mathbf{P} and actions \mathbf{D} is given by the following grammar:

$$\varphi ::= q \mid \neg\varphi \mid \varphi \vee \psi \mid \langle \ell \rangle \varphi \mid \mu p. \varphi$$

where $p, q \in \mathbf{P}$, $\ell \in \mathbf{D}$ and p is positive in φ (i.e., p is under an even number of negations).

A formula $\varphi \in \mu\text{ML}$ is clean if no two distinct (occurrences of) fixed point operators in φ bind the same variable, and no variable has both free and bound occurrences in φ . If p is a bound variable of a clean formula, we use δ_p to denote the binding definition of p and σ_p to denote the binding type of p . That is, $\sigma_p p. \delta_p$ is the unique subformula of φ binding p , with $\sigma_p \in \{\mu, \nu\}$. In this dissertation we will assume without loss of generality that our formulas are clean.

The semantics of this language is completely standard. Let \mathbb{S} be a transition system we define it by induction.

$$\begin{aligned} \mathbb{S} \Vdash p & \text{ iff } p \in \kappa(s_I), \\ \mathbb{S} \Vdash \neg\varphi & \text{ iff } \mathbb{S} \not\Vdash \varphi, \\ \mathbb{S} \Vdash \alpha \vee \beta & \text{ iff } \mathbb{S} \Vdash \alpha \text{ or } \mathbb{S} \Vdash \beta, \\ \mathbb{S} \Vdash \langle \ell \rangle \varphi & \text{ iff there exists } t \in S \text{ such that } R_\ell(s_I, t) \text{ and } \mathbb{S}, t \Vdash \varphi, \\ \mathbb{S} \Vdash \mu p. \psi & \text{ iff } s_I \in \bigcap \{X \subseteq S \mid X \supseteq F_p^\psi(X)\}. \end{aligned}$$

The map $F_p^\psi(X)$ is given as follows:

2.4.2. DEFINITION. For every formula $\varphi \in \mu\text{ML}$ and propositional variables $\bar{\mathbf{p}}$ we define, for every transition system \mathbb{S} , the map $F_{\bar{\mathbf{p}}}^\varphi : \wp(S)^n \rightarrow \wp(S)$ induced by φ and $\bar{\mathbf{p}}$ as:

$$F_{\bar{\mathbf{p}}}^\varphi(\bar{\mathbf{X}}) := \{s \in S \mid \mathbb{S}[\bar{\mathbf{p}} \mapsto \bar{\mathbf{X}}], s \Vdash \varphi\}$$

As the formulas under the fixpoints are positive, their corresponding maps will be monotone. Hence, by the Knaster-Tarski theorem we know that the least fixpoint of such a map exists, and that it is precisely given by the expression $\bigcap \{X \subseteq S \mid X \supseteq F_p^\psi(X)\}$. The extension of a formula $\varphi \in \mu\text{ML}$ in a transition system \mathbb{S} is given by $\llbracket \varphi \rrbracket^{\mathbb{S}} := \{s \in S \mid \mathbb{S}, s \Vdash \varphi\}$. The superscript is dropped when the transition system is clear from context.

2.4.3. FACT. *Every formula of μML is equivalent to a formula in negation normal form, given by the following grammar:*

$$\varphi ::= q \mid \neg q \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle \ell \rangle \varphi \mid [\ell] \varphi \mid \mu p. \varphi \mid \nu p. \varphi$$

where $p, q \in \mathbf{P}$, $\ell \in \mathbf{D}$ and p is positive in $\mu p. \varphi$ and $\nu p. \varphi$ (i.e., p is not negated).

We use the symbol \heartsuit to refer to an arbitrary modality $\langle \ell \rangle$ or $[\ell]$ for some ℓ . The free variables $\text{FV}(\varphi)$ of φ are the propositional variables which occur in φ and are not bound by a fixpoint operator. We use $\alpha \trianglelefteq \beta$ to denote that β is a (not necessarily proper) subformula of α . It may be useful to think of this symbol as if β was ‘hanging from’ the syntactic tree of α .

Formulas of the modal μ -calculus are classified according to their *alternation depth*, which roughly is given as the maximal length of a chain of nested alternating least and greatest fixpoint operators [Niw86]. The *alternation-free fragment* of the modal μ -calculus (AFMC) is the collection of μ ML-formulas without nesting of least and greatest fixpoint operators.

2.4.4. DEFINITION. Let φ be a formula of the modal μ -calculus. We say that $\varphi \in \text{AFMC}$ iff for all subformulas $\mu p.\psi_1$ and $\nu q.\psi_2$ we have that p is not free in ψ_2 and q is not free in ψ_1 .

It is not difficult to see that, over arbitrary transition systems, this fragment is less expressive than the whole μ ML. That is, there is a μ ML-formula φ such that there is no equivalent formula of AFMC [Par80].

Another interesting fragment of the μ -calculus is the k -variable fragment, denoted $\mu\text{ML}[k]$. That is, the fragment where only k *fixpoint* variables might be (re)used. These fragments naturally define a hierarchy (the variable hierarchy) which has been shown to be strict by Berwanger [Ber05, BGL05].

Finite approximants of monotone maps. Let $F : \wp(S) \rightarrow \wp(S)$ be a monotone map. The approximants of the least fixpoint of F are the sets $F^\alpha(\emptyset) \subseteq S$, where α is an ordinal. The map F^α is intuitively the α -fold composition of F . Formally,

- $F^0(X) := \emptyset$,
- $F^{\alpha+1}(X) := F(F^\alpha(X))$,
- $F^\lambda(X) := \bigcup_{\alpha < \lambda} F^\alpha(X)$ for limit ordinals λ .

The sets $F^\alpha(\emptyset)$ are called approximants because of the following fact.

2.4.5. FACT. For every $s \in S$ we have that $s \in \text{LFP}(F)$ if and only if $s \in F^\beta(\emptyset)$ for some ordinal β .

Moreover, this approximation starts at $F^0(\emptyset) = \emptyset$ and grows strictly until it stabilizes for some ordinal β . This ordinal is called the closure or unfolding ordinal of F .

2.5 Logics of programs and games

2.5.1 Propositional Dynamic Logic

The language now called Propositional Dynamic Logic was first investigated by Fischer and Ladner [FL79] as a logic to reason about computer program execution. In particular, the focus is on non-deterministic programs. PDL extends

the basic modal logic with an infinite collection of diamonds $\langle \pi \rangle$ where the intended intuitive interpretation of $\langle \pi \rangle \varphi$ is that “some terminating execution of the program π from the current state leads to a state satisfying φ ”.

The inductive structure of programs is made explicit in PDL’s syntax, as complex programs are built out of atomic programs using four program constructors.

2.5.1. DEFINITION. The language of Propositional Dynamic Logic (PDL) on propositions \mathbf{P} and atomic actions \mathbf{D} is given by mutual induction on formulas (φ) and programs (π):

$$\begin{aligned}\varphi &::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle \pi \rangle \varphi \\ \pi &::= \ell \mid \pi; \pi \mid \pi \oplus \pi \mid \pi^* \mid \varphi?\end{aligned}$$

where $p \in \mathbf{P}$ and $\ell \in \mathbf{D}$. We denote this logic by $\text{PDL}(\mathbf{P}, \mathbf{D})$ and drop \mathbf{P}, \mathbf{D} when clear from context. As an abuse of notation we write $\pi \in \text{PDL}(\mathbf{P}, \mathbf{D})$ to mean that π is a program of $\text{PDL}(\mathbf{P}, \mathbf{D})$. The logic called *test-free* PDL (and denoted PDL^{tf}) is defined as PDL without the test program “ $\varphi?$ ”.

Given a formula $\varphi \in \text{PDL}$ we use $\text{FV}(\varphi)$ to denote the propositional variables occurring in φ . This notation extends naturally to programs. Given $X \subseteq \mathbf{P}$ we say that φ is X -free if $X \cap \text{FV}(\varphi) = \emptyset$.

We give the semantics of PDL by mutual induction, together with the relation $R_\pi^{\mathbb{S}}$ induced by a program π on a transition system \mathbb{S} :

$$\begin{aligned}R_\ell^{\mathbb{S}} &:= R_\ell & R_{\pi;\varrho}^{\mathbb{S}} &:= R_\pi^{\mathbb{S}} \circ R_\varrho^{\mathbb{S}} \\ R_{\pi \oplus \varrho}^{\mathbb{S}} &:= R_\pi^{\mathbb{S}} \cup R_\varrho^{\mathbb{S}} & R_{\pi^*}^{\mathbb{S}} &:= (R_\pi^{\mathbb{S}})^* \\ R_{\varphi?}^{\mathbb{S}} &:= \{(s, s) \in S \times S \mid \mathbb{S}, s \Vdash \varphi\}.\end{aligned}$$

The symbol \circ denotes relational composition: if $R \subseteq X \times Y$ and $R' \subseteq Y \times Z$ are two binary relations, then their composition $R \circ R'$ is the relation given by $R \circ R' := \{(x, z) \in X \times Z \mid \exists y \in Y : (x, y) \in R \wedge (y, z) \in R'\}$.

The semantics of PDL is then given as usual on the Boolean operators and as follows on modal operators.

$$\mathbb{S}, s \Vdash \langle \pi \rangle \varphi \quad \text{iff} \quad \text{there exists } t \in S \text{ such that } R_\pi^{\mathbb{S}}(s, t) \text{ and } \mathbb{S}, t \Vdash \varphi.$$

We drop the superscript in $R_\pi^{\mathbb{S}}$ when it is clear from context.

2.5.2. EXAMPLE. One of the most salient properties of PDL is the possibility to express that there is some R_ℓ -path such that p is true at the end, with the formula

$\langle \ell^* \rangle p$. This property is not first-order definable. Moreover, many programming-language constructs can be expressed [GW05, Section 3.2]:

$$\begin{aligned} \text{skip} &:= \top? \\ \text{abort} &:= \perp? \\ \text{if } \varphi \text{ then } \pi_1 \text{ else } \pi_2 &:= (\varphi?; \pi_1) \oplus (\neg\varphi?; \pi_2) \\ \text{while } \varphi \text{ do } \pi &:= (\varphi?; \pi)^*; \neg\varphi? \\ \text{repeat } \pi \text{ until } \varphi &:= \pi; (\neg\varphi?; \pi)^*; \varphi? \end{aligned}$$

For technical reasons, it will be sometimes convenient for us to work with a version of PDL^{tf} that includes the empty program ϵ (or **skip**), which is interpreted as the identity relation in any labeled transition system. Observe that in full PDL, the role of ϵ can be taken by the test program $\top?$.

2.5.3. REMARK. It is not difficult to show that adding the skip program does not add expressive power to PDL^{tf} . To see this, think of the programs of PDL^{tf} and $\text{PDL}^{tf\epsilon}$ as the sets of regular expressions over the set \mathbf{D} that may and may not use the empty string symbol ϵ , respectively. Let \equiv_ℓ denote the relation of language equivalence between regular expressions, that is, write $\pi \equiv_\ell \pi'$ if π and π' denote the same regular language over \mathbf{D} .

One may show, by induction on programs, that for any $\pi \in \text{PDL}^{tf\epsilon}$ either (a) $\pi \equiv_\ell \epsilon$, or there is a program $\varrho \in \text{PDL}^{tf\epsilon}$ such that either (b) $\pi \equiv_\ell \varrho$ or (c) $\pi \equiv_\ell \epsilon \oplus \varrho$. Based on this observation we may inductively define a translation from $\text{PDL}^{tf\epsilon}$ -formulas to PDL^{tf} -formulas; the key clause of this translation uses that $\langle \pi \rangle \varphi$ is equivalent to either (a) φ , (b) $\langle \varrho \rangle \varphi$ or (c) $\varphi \vee \langle \varrho \rangle \varphi$.

2.5.2 Concurrent PDL

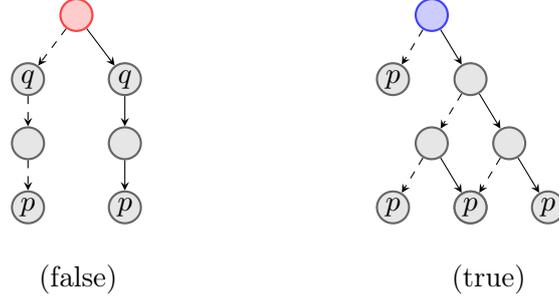
Concurrent PDL is an extension of PDL introduced by Peleg in [Pel85]. This language contains an operator \otimes which is the dual of \oplus and adds expressive power to PDL. The intuition behind this new operator is that the program $\pi_1 \otimes \pi_2$ succeeds when both π_1 and π_2 can be executed simultaneously.

2.5.4. DEFINITION. The language of Concurrent PDL (CPDL) on propositions \mathbf{P} and atomic actions \mathbf{D} is given by mutual induction on formulas and programs:

$$\begin{aligned} \varphi &::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle \pi \rangle \varphi \\ \pi &::= \ell \mid \pi; \pi \mid \pi \oplus \pi \mid \pi \otimes \pi \mid \pi^* \mid \varphi? \end{aligned}$$

where $p \in \mathbf{P}$, $\ell \in \mathbf{D}$.

2.5.5. EXAMPLE. Concurrent PDL features the simultaneous execution of programs. For example the formula $\varphi := \langle (\ell \otimes \ell')^* \rangle p$ expresses that I can reach a state where p holds in a finite amount of steps, while in each step the program must be able to execute both an ℓ and ℓ' transitions.



The above figure shows two transition systems where R_ℓ is represented by solid edges and $R_{\ell'}$ is represented by dashed edges. In the left one φ is false in the root, and in the right one it is true in the root.

From the above example we see that the execution of a CPDL program cannot be seen as a path on a transition system, as it was possible with PDL. In this case, the program $\pi_1 \otimes \pi_2$ forced a branching, given by the (parallel) execution of π_1 and π_2 . This means that programs of CPDL do not (necessarily) induce a relation on a given transition system. We need the following definition (which is taken from Pauly [Pau01]) to give semantics to CPDL. For every transition system \mathbb{S} and program $\pi \in \text{CPDL}$ we define a function $R_\pi^\rightarrow : \wp(S) \rightarrow \wp(S)$.

$$\begin{array}{ll}
 R_\ell^\rightarrow(X) & := R_\ell^{-1}[X] & R_{\pi;\pi'}^\rightarrow(X) & := R_\pi^\rightarrow(R_{\pi'}^\rightarrow(X)) \\
 R_{\pi \oplus \pi'}^\rightarrow(X) & := R_\pi^\rightarrow(X) \cup R_{\pi'}^\rightarrow(X) & R_{\pi \otimes \pi'}^\rightarrow(X) & := R_\pi^\rightarrow(X) \cap R_{\pi'}^\rightarrow(X) \\
 R_{\varphi?}^\rightarrow(X) & := \llbracket \varphi \rrbracket \cap X & R_{\pi^*}^\rightarrow(X) & := \mu Y.(X \cup R_\pi^\rightarrow(Y))
 \end{array}$$

Finally we define the semantics of CPDL as usual on Booleans and

$$\mathbb{S}, s \Vdash \langle \pi \rangle \varphi \quad \text{iff} \quad s \in R_\pi^\rightarrow(\llbracket \varphi \rrbracket).$$

2.5.6. THEOREM ([PEL85, THEOREM 2.8]). $\text{PDL} \subsetneq \text{CPDL}$.

2.5.7. REMARK. Another extension of PDL is the so-called PDL with intersection (see [HTK00, Section 10.4] and [Dan84, Har83]). This language adds the intersection operator $\pi_1 \cap \pi_2$ to the available programs. The semantic of this operator is given as $R_{\pi \cap \varrho}^\mathbb{S} := R_\pi^\mathbb{S} \cap R_\varrho^\mathbb{S}$. We want to remark that the operators \cap and \otimes are *not* the same. For example, the formula $\langle \ell \otimes \ell' \rangle q$ is true on the left transition system of Example 2.5.5 while the formula $\langle \ell \cap \ell' \rangle q$ is false in both of the transition systems.

2.5.3 Game Logic

Game Logic was introduced by Parikh [Par85] and extends PDL with *all* program duals which, in this context, are called games.

2.5.8. DEFINITION. The language of Game Logic (GL) on propositions \mathbf{P} and atomic actions \mathbf{D} is given by mutual induction on formulas and games:

$$\begin{aligned}\varphi &::= p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle \pi \rangle \varphi \\ \pi &::= \ell \mid \pi; \pi \mid \pi \oplus \pi \mid \pi^* \mid \varphi? \mid \pi^\delta\end{aligned}$$

where $p \in \mathbf{P}$ and $\ell \in \mathbf{D}$.

Game Logic was originally meant to be interpreted in (monotone) neighborhood models.¹ However, as transition systems are a special case of neighborhood models, it is easy to define a semantics directly for our case. We extend the semantics of CPDL with the following clause for the dual program:

$$R_{\pi^\delta}^{\rightarrow}(X) := S \setminus R_{\pi}^{\rightarrow}(S \setminus X)$$

Finally we define the semantics of GL as usual on Booleans and

$$\mathbb{S}, s \Vdash \langle \pi \rangle \varphi \quad \text{iff} \quad s \in R_{\pi}^{\rightarrow}(\llbracket \varphi \rrbracket).$$

It is worth remarking that GL (and therefore CPDL and PDL) can be translated to the modal μ -calculus. This connection will be studied in more detail in Section 3.3 of this dissertation.

2.5.9. THEOREM ([PAU01, SECTION 7.2.2]). $\text{GL} \subseteq \mu\text{ML}$.

The syntax of Game Logic can also be presented entirely in dual (and negation) normal form, as follows.

$$\begin{aligned}\varphi &::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle \pi \rangle \varphi \mid [\pi] \varphi \\ \pi &::= \ell \mid \ell^\delta \mid \pi; \pi \mid \pi \oplus \pi \mid \pi \otimes \pi \mid \pi^* \mid \pi^\circ \mid \varphi? \mid \varphi!\end{aligned}$$

where $p \in \mathbf{P}$, $\ell \in \mathbf{D}$ and ℓ^δ represents the dual of ℓ . In this presentation, every program construction has an explicit dual. The semantics of the new constructions is given as $\pi^\circ := ((\pi^\delta)^*)^\delta$ and $\varphi! := ((\varphi^\delta)?)^\delta$.

2.5.10. FACT. *Every formula $\varphi \in \text{GL}$ is equivalent to some formula $\varphi' \in \text{GL}$ in dual normal form.*

¹Neighbourhood models are a generalization of transition systems, where $R \subseteq S \times \wp(S)$.

2.6 Bisimulation

Bisimulation is a notion of behavioral equivalence between processes. For the case of transition systems, it is formally defined as follows.

2.6.1. DEFINITION. Let $\mathbb{S} = \langle S, R_{\ell \in \mathbf{D}}, \kappa, s_I \rangle$ and $\mathbb{S}' = \langle S', R'_{\ell \in \mathbf{D}}, \kappa', s'_I \rangle$ be transition systems. A *bisimulation* is a relation $Z \subseteq S \times S'$ such that for all $(t, t') \in Z$ the following holds:

(atom) $p \in \kappa(t)$ iff $p \in \kappa'(t')$ for all $p \in \mathbf{P}$;

(forth) for all $\ell \in \mathbf{D}$ and $s \in R_\ell[t]$ there is $s' \in R'_\ell[t']$ such that $(s, s') \in Z$;

(back) for all $\ell \in \mathbf{D}$ and $s' \in R'_\ell[t']$ there is $s \in R_\ell[t]$ such that $(s, s') \in Z$.

Two transition systems \mathbb{S} and \mathbb{S}' are *bisimilar* (denoted $\mathbb{S} \Leftrightarrow \mathbb{S}'$) if there is a bisimulation $Z \subseteq S \times S'$ containing (s_I, s'_I) .

The following fact about tree unravelings is central in many theorems of modal logics. It will also play an important role in this dissertation.

2.6.2. FACT. \mathbb{S} and its unraveling $\hat{\mathbb{S}}$ are bisimilar, for every transition system \mathbb{S} .

An important concept in the last part of this dissertation is that of bisimulation invariance. It is defined as follows for an arbitrary language \mathcal{L} :

2.6.3. DEFINITION. A formula $\varphi \in \mathcal{L}$ is *bisimulation-invariant* if $\mathbb{S} \Leftrightarrow \mathbb{S}'$ implies that $\mathbb{S} \Vdash \varphi$ iff $\mathbb{S}' \Vdash \varphi$, for all \mathbb{S} and \mathbb{S}' .

2.6.4. FACT. Every formula of μML (and therefore of GL, CPDL and PDL) is bisimulation-invariant.

2.7 First-order logic and extensions

We start by introducing the syntax and semantics of first-order logic and then discuss some extensions that will be used in this dissertation.

2.7.1. DEFINITION. The language of *first-order logic with equality* (FOE) on a set of predicates \mathbf{P} , actions \mathbf{D} and individual variables \mathbf{iVar} is given by:

$$\varphi ::= q(x) \mid R_\ell(x, y) \mid x \approx y \mid \exists x. \varphi \mid \neg \varphi \mid \varphi \vee \varphi$$

where $p, q \in \mathbf{P}$, $\ell \in \mathbf{D}$ and $x, y \in \mathbf{iVar}$. We use FO to denote *first-order logic without equality*, which is defined as FOE but without the \approx predicate.

The free variables $\text{FV}(\varphi)$ of a formula $\varphi \in \text{FOE}$ are the *individual* variables which are not bound by a quantifier. The inductive definition of $\text{FV}(\varphi)$ is standard.

2.7.2. REMARK. Every logic in this dissertation will be function-free. That is, we will *not* consider logics with function symbols in their signature.

Formulas of FOE will be interpreted over models $\mathbb{M} = \langle M, R_{\ell \in \mathcal{D}}, \kappa \rangle$ with an assignment $g : \text{iVar} \rightarrow M$. Usually, first-order structures (for the signature that we use) are given as tuples $\langle M, R_{\ell \in \mathcal{D}}, P_1, \dots \rangle$ where $P_i \subseteq M$. However, this same information is encoded in \mathbb{M} , since $P_i = \kappa^{\sharp}(p_i)$. The semantics of FOE (and also FO) is standard, given as follows:

$$\begin{aligned} \mathbb{M}, g \models p_i(x) & \text{ iff } p_i \in \kappa(g(x)) \\ \mathbb{M}, g \models x \approx y & \text{ iff } g(x) = g(y) \\ \mathbb{M}, g \models R_\ell(x, y) & \text{ iff } R_\ell(g(x), g(y)) \\ \mathbb{M}, g \models \neg\varphi & \text{ iff } \mathbb{M}, g \not\models \varphi \\ \mathbb{M}, g \models \varphi \vee \psi & \text{ iff } \mathbb{M}, g \models \varphi \text{ or } \mathbb{M}, g \models \psi \\ \mathbb{M}, g \models \exists x.\varphi & \text{ iff there is } s \in M \text{ such that } \mathbb{M}, g[x \mapsto s] \models \varphi. \end{aligned}$$

2.7.1 First-order logic with generalized quantifiers

In this subsection we introduce an extension of first-order logic with so called generalized quantifiers. Mostowski [Mos57] defined unary generalized quantifiers as follows: a unary generalized quantifier \mathcal{Q} is a collection of pairs (J, X) with $X \subseteq J$, and satisfying the following condition

$$\text{If } ((J, X) \in \mathcal{Q}, |X| = |Y| \wedge |J \setminus X| = |K \setminus Y|) \text{ then } (K, Y) \in \mathcal{Q}.$$

The semantics of \mathcal{Q} is then defined by the following condition

$$\mathbb{M}, g \models \mathcal{Q}x.\phi(x) \text{ iff } (M, \{s \in M \mid \mathbb{M}, g[x \mapsto s] \models \phi(x)\}) \in \mathcal{Q},$$

for every model \mathbb{M} and assignment g .

In this dissertation we will only focus on the generalized quantifier \exists^∞ expressing that there exist infinitely many elements satisfying a certain condition. Formally, it is defined as:

$$\exists^\infty := \{(J, X) \mid |X| \geq \aleph_0\}.$$

The dual of \exists^∞ is $\forall^\infty = \{(J, X) \mid |J \setminus X| < \aleph_0\}$. It is worth observing what is the intended meaning of this quantifier: $\forall^\infty x.\varphi$ expresses that there are *at most finitely many* elements *falsifying* the formula φ .

2.7.3. DEFINITION. The extension of first-order logic with equality (FOE) obtained by adding \exists^∞ to the corresponding first-order language is denoted FOE^∞ .

2.7.2 Fixpoint extension of first-order logic

In this subsection we give an extension of FOE with a unary fixed point operator. This extension is known in the literature as FO(LFP¹) but we will call it μ FOE to keep a consistent notation for fixpoint extensions (e.g., we use $\mu\mathcal{L}$ for a base logic \mathcal{L}) and fragments thereof (e.g. $\mu_X\mathcal{L}$ where X is some restriction on the fixpoint).

Because of the presence of individual variables, the syntax and semantics of the fixpoint operator is considerably more involved than for the modal μ -calculus.

2.7.4. DEFINITION. The language of *first-order logic with equality and unary fixpoints* (μ FOE) on a set of predicates P , actions D and individual variables iVar is given by:

$$\varphi ::= q(x) \mid R_\ell(x, y) \mid x \approx y \mid \exists x.\varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid [\mathsf{LFP}_{p:x}.\varphi(p, x)](z)$$

where $p, q \in \mathsf{P}$, $\ell \in \mathsf{D}$ and $x, y \in \mathsf{iVar}$. Observe that z is free in the fixpoint clause and the fixpoint operator binds the designated variables x and p .

The free variables $\mathsf{FV}(\varphi)$ of a formula $\varphi \in \mu$ FOE are obtained by extending the standard definition of FV for FOE with the clause

$$\mathsf{FV}([\mathsf{LFP}_{p:x}.\varphi(p, x)](z)) := (\mathsf{FV}(\varphi) \setminus \{x\}) \cup \{z\}.$$

The semantics of the fixpoint formula $[\mathsf{LFP}_{p:x}.\varphi(p, x)](z)$ is the expected one (as introduced in [CH82, Mos08, Mos74]). First we give a slightly more general definition than we need right now (which will be useful later). For every model \mathbb{M} , assignment g , and predicates (propositions) Q , the map $F_{\mathsf{Q}:x}^\varphi : \wp(M) \rightarrow \wp(M)$ is given as:

$$F_{\mathsf{Q}:x}^\varphi(\overline{\mathbf{Y}}) := \{t \in M \mid \mathbb{M}[\mathsf{Q} \mapsto \overline{\mathbf{Y}}], g[x \mapsto t] \models \varphi(\mathsf{Q}, x)\}.$$

The formula $\mathbb{M}, g \models [\mathsf{LFP}_{p:x}.\varphi(p, x)](z)$ is then defined to hold iff $g(z) \in \mathsf{LFP}(F_{p:x}^\varphi)$. That is, if $g(z)$ is in the least fixpoint of the map $F_{p:x}^\varphi$.

2.7.5. REMARK. Suppose that $\varphi \in \mu$ FOE has free variables $\mathsf{FV}(\varphi) = \{x, \overline{\mathbf{y}}\}$. If we consider the fixpoint formula $\psi := [\mathsf{LFP}_{p:x}.\varphi(p, x)](z)$ then ψ would have as free variables $\mathsf{FV}(\psi) = \{z, \overline{\mathbf{y}}\}$. The free variables of φ which are not bound by the fixpoint (in this case $\overline{\mathbf{y}}$) are called the *parameters* of the fixpoint.

Parameters can always be avoided at the expense of increasing the arity of the fixpoint [Lib04, p. 184]. That is, for example, taking the fixpoint over a relation $P(x_1, \dots, x_n)$ instead of just a predicate p . However, in this dissertation we will only consider fixpoints over unary predicates, and therefore we will allow the use of parameters unless explicitly stated.

The language of μ FOE can also be further extended with a *greatest* fixpoint operator $[\mathsf{GFP}_{p:x}.\varphi(p, x)](z)$ whose semantics are given by $\mathbb{M}, g \models [\mathsf{GFP}_{p:x}.\varphi(p, x)](z)$ iff $g(z) \in \mathsf{GFP}(F_{p:x}^\varphi)$. However, this extension does not add expressive power, since it is possible to prove that $[\mathsf{GFP}_{p:x}.\varphi(p, x)](z) \equiv \neg[\mathsf{LFP}_{p:x}.\neg\varphi(\neg p, x)](z)$.

2.7.3 First-order logic with transitive closure

It is well known that the reflexive-transitive closure R^* of a binary relation R is not expressible in first-order logic [Fag75]. Therefore, a straightforward way to extend first-order logic is to add a reflexive-transitive closure operator:

$$[\text{TC}_{\bar{x}, \bar{y}} \cdot \varphi(\bar{x}, \bar{y})](\bar{u}, \bar{v})$$

which states that (\bar{u}, \bar{v}) belongs to the transitive closure of the relation denoted by $\varphi(\bar{x}, \bar{y})$. In the above expression, the sequences of variables $\bar{x}, \bar{y}, \bar{u}, \bar{v}$ should all be of the same length; this length is called the *arity* of the transitive closure. This extension of first-order logic, called FO(TC) or sometimes transitive-closure logic, was introduced by Immerman in [Imm87] where he showed that it captures the class of NLOGSPACE queries (on ordered structures).

The fragment FO(TC^k) of FO(TC) is given by restricting the arity of the transitive closure to length at most k . These fragments naturally induce a hierarchy (the arity hierarchy) for FO(TC), which was proven strict for finite models [Gro96]. Moreover, in some restricted classes of trees, full FO(TC) is even more expressive than MSO [TK09]. In this dissertation, however, we restrict our attention to FO(TC¹), which is FO(TC) restricted to sequences of length one; that is, the reflexive-transitive closure can only be applied to formulas $\varphi(x, y)$ defining a *binary relation*. FO(TC¹) is easily seen to be included in MSO.

2.7.6. DEFINITION. The *first-order logic with reflexive-transitive closure* of *binary* formulas is given by the following grammar:

$$\varphi ::= p(x) \mid x \approx y \mid R_\ell(x, y) \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\varphi \mid [\text{TC}_{x,y} \cdot \varphi(x, y)](z, w)$$

where $p, q \in \mathbf{P}$, $\ell \in \mathbf{D}$ and $x, y, z, w \in \mathbf{iVar}$. We denote this logic by FO(TC¹). The semantics for the new operator are:

$$\mathbb{M}, g \models [\text{TC}_{x,y} \cdot \varphi(x, y)](u, v) \quad \text{iff} \quad (g(u), g(v)) \in R_\varphi^*$$

where $R_\varphi := \{(s_x, s_y) \in M \times M \mid \mathbb{M}, g[x \mapsto s_x, y \mapsto s_y] \models \varphi\}$.

The notion of “parameter” also makes sense for this logic. We say that z is a parameter of $[\text{TC}_{x,y} \cdot \varphi(x, y)](u, v)$ if $z \in \text{FV}(\varphi)$ and $z \notin \{x, y\}$.

2.7.7. REMARK. The meaning of $[\text{TC}_{x,y} \cdot \varphi(x, y)](u, v)$ can be rephrased as saying that $v \in \varphi^*[u]$; that is, v is a φ -descendant of u . This can be expressed with the formula $[\text{LFP}_{p,y}.y \approx u \vee (\exists x.p(x) \wedge \varphi(x, y))](v)$ –cf. [GKL⁺05, Example 3.3.8]. However, observe that in order to express the transitive closure with a unary fixpoint, we had to use u as a parameter of the fixpoint. This variable parameter was not a parameter in the original transitive-closure.

2.8 Second-order logics

In this section we define three different monadic second-order logics: standard monadic-second order logic, weak monadic second-order logic and weak chain logic. The syntax of the three logics will be the same, but their semantics will differ in the interpretation of the second-order quantifier $\exists p.\varphi$.

Monadic second-order logic. In this dissertation we will mostly work with one-sorted versions of second-order logics, since these will be better suited to work with automata. That is, instead of having both individual (first-order) variables and set (second-order) variables, we will only have second-order variables. Individual variables can clearly be seen as special *singleton* set variables. We introduce the required definitions and promptly discuss this topic a bit further.

2.8.1. DEFINITION. The single-sorted *monadic second-order logic* (MSO) on a set of predicates \mathbf{P} and actions \mathbf{D} is given by:

$$\varphi ::= \Downarrow p \mid p \sqsubseteq q \mid R_\ell(p, q) \mid \neg\varphi \mid \varphi \vee \psi \mid \exists p.\varphi$$

where $p, q \in \mathbf{P}$ and $\ell \in \mathbf{D}$. We denote this logic by $\text{MSO}(\mathbf{P}, \mathbf{D})$ and omit \mathbf{P} and \mathbf{D} when clear from context. We adopt the standard convention that no letter is both free and bound in φ .

2.8.2. DEFINITION. Let \mathbb{S} be a labeled transition system. The semantics of MSO is defined as follows:

$$\begin{aligned} \mathbb{S} \models \Downarrow p & \text{ iff } \kappa^\sharp(p) = \{s_I\} \\ \mathbb{S} \models p \sqsubseteq q & \text{ iff } \kappa^\sharp(p) \subseteq \kappa^\sharp(q) \\ \mathbb{S} \models R_\ell(p, q) & \text{ iff for all } s \in \kappa^\sharp(p) \text{ there is } t \in \kappa^\sharp(q) \text{ such that } R_\ell(s, t) \\ \mathbb{S} \models \neg\varphi & \text{ iff } \mathbb{S} \not\models \varphi \\ \mathbb{S} \models \varphi \vee \psi & \text{ iff } \mathbb{S} \models \varphi \text{ or } \mathbb{S} \models \psi \\ \mathbb{S} \models \exists p.\varphi & \text{ iff there is a } X \subseteq S \text{ such that } \mathbb{S}[p \mapsto X] \models \varphi. \end{aligned}$$

A digression on second-order languages. The reader may have expected a more standard two-sorted language for second-order logic, for example given by:

$$\varphi ::= p(x) \mid R_\ell(x, y) \mid x \approx y \mid \neg\varphi \mid \varphi \vee \psi \mid \exists x.\varphi \mid \exists p.\varphi$$

where $p \in \mathbf{P}$, $\ell \in \mathbf{D}$, $x, y \in \text{iVar}$ (individual variables), and \approx is the symbol for equality. We call this language 2MSO. This semantics of this language is completely standard, with $\exists x$ denoting first-order quantification (that is, quantification over individual states) and $\exists p$ denoting second-order quantification. Both definitions can be proved to be equivalent.

Formulas of 2MSO are interpreted over models $\mathbb{M} = \langle M, R_{\ell \in \mathcal{D}}, \kappa \rangle$ with an assignment, that is, a map $g : \text{iVar} \rightarrow M$ interpreting the individual variables as elements of M . The key point is that MSO can interpret 2MSO by encoding every individual variable $x \in \text{iVar}$ as a set variable p_x denoting a *singleton*. The following is a more detailed proof of the remark found in [Ven11].

2.8.3. PROPOSITION. *There is a translation $(-)^t : 2\text{MSO}(\mathcal{P}, \mathcal{D}) \rightarrow \text{MSO}(\mathcal{P} \uplus \mathcal{P}_X, \mathcal{D})$ such that*

$$\mathbb{M}, g \models \varphi \quad \text{iff} \quad \mathbb{M}[p_x \in \text{iVar} \mapsto \{g(x)\}] \models \varphi^t,$$

where $\mathcal{P}_X := \{p_x \mid x \in \text{iVar}\}$.

Proof. The translation is inductively defined as follows:

- $(p(x))^t := p_x \sqsubseteq p$,
- $(R_\ell(x, y))^t := R_\ell(p_x, p_y)$,
- $(x \approx y)^t := p_x \sqsubseteq p_y \wedge p_y \sqsubseteq p_x$,
- Negation and disjunction as usual,
- $(\exists p.\varphi)^t := \exists p.\varphi^t$,
- $(\exists x.\varphi)^t := \exists p_x.\text{singleton}(p_x) \wedge \varphi^t$

where the translation crucially uses the predicates

$$\begin{aligned} \text{empty}(p) &:= \forall q.(p \sqsubseteq q) \\ \text{singleton}(p) &:= \forall q.(q \sqsubseteq p \rightarrow (\text{empty}(q) \vee p \sqsubseteq q)) \end{aligned}$$

Observe that the translation does not use the operator $\Downarrow p$ and hence is well-defined on models. We leave the proof of the proposition to the reader. \square

Weak monadic second-order logic. This logic has basically the same syntax as MSO. However, when we refer to this logic we will write its existential quantifier as $\exists_{\text{fin}} p.\varphi$ instead of $\exists p.\varphi$. The subscript of the quantifier emphasizes that the quantification of this logic is over *finite* sets, instead of arbitrary sets:

$$\mathbb{S} \models \exists_{\text{fin}} p.\varphi \quad \text{iff} \quad \text{there is a finite } X \subseteq S \text{ such that } \mathbb{S}[p \mapsto X] \models \varphi.$$

We will denote the one- and two-sorted versions of this logic by $\text{WMSO}(\mathcal{P}, \mathcal{D})$ and $2\text{WMSO}(\mathcal{P}, \mathcal{D})$ respectively.

2.8.4. REMARK. The adjective “weak” is a bit misleading, since WMSO is in general not a fragment of MSO. Indeed, the class of finitely branching trees is not definable in MSO (see [Wal96]) but is defined using the WMSO-formula $\forall x \exists_{\text{fin}} X \forall y.(R(x, y) \rightarrow y \in X)$. The class of well-founded trees, on the other hand, is definable in MSO but not in WMSO [CF11].

Weak chain logic. The non-weak version of chain logic (CL) was defined in [Tho96], and studied in the context of *trees*. CL is a variant of MSO which changes the usual second-order quantifier to the following quantifier over chains:

$$\mathbb{T} \models \exists_{\text{ch}} p. \varphi \quad \text{iff} \quad \text{there is a chain } X \subseteq T \text{ such that } \mathbb{T}[p \mapsto X] \models \varphi.$$

In this dissertation we will only work with a *weak* version of CL, that is, the quantification will be over *finite* chains. On the other hand, we also want to consider this logic on the class of *all* transition systems (as opposed to only trees). To give a definition of weak chain logic we adhere to what we think is the “spirit” of the definition of CL, as opposed to the “letter.” As observed in Section 2.1, the concept of chain on trees coincides with that of “subset of a path.” Therefore, on the class of all models, we choose to define the weak second-order quantifier as:

$$\mathbb{S} \models \exists_{\text{fch}} p. \varphi \quad \text{iff} \quad \text{there is a generalized finite chain } X \subseteq S \text{ s.t. } \mathbb{S}[p \mapsto X] \models \varphi.$$

We will denote the one- and two-sorted versions of this logic by $\text{WCL}(\mathbf{P}, \mathbf{D})$ and $2\text{WCL}(\mathbf{P}, \mathbf{D})$ respectively.

2.8.5. REMARK. On trees, it is not difficult to see that $\text{WCL} \subseteq \text{MSO}$ and $\text{WCL} \subseteq \text{WMSO}$, since finite chains are easily definable in both logics.

2.9 Notational convention

The following table works as a summary of the most used notation in this dissertation. It should be taken as a set of general rules from which we try to divert as little as possible.

Concept	Notation
Transition system (pointed model)	$\mathbb{S} = \langle S, R_{\ell \in \mathbf{D}}, \kappa, s_I \rangle$
Tree (pointed tree)	$\mathbb{T} = \langle T, R_{\ell \in \mathbf{D}}, \kappa, s_I \rangle$
Model	$\mathbb{M} = \langle M, R_{\ell \in \mathbf{D}}, \kappa \rangle$
One-step model	$\mathbf{D} = (D, V : A \rightarrow \wp(D))$
Automaton	$\mathbb{A}, \mathbb{B}, \dots$
Formula	$\varphi, \psi, \alpha, \beta, \xi, \chi, \dots \Phi, \Psi, \dots$
Set	$A, B, C, D, \dots X, Y, Z, W, \dots$
Sequence of objects	$\bar{x}, \bar{y}, \dots \bar{a}, \bar{b}, \dots \bar{\mathbf{X}}, \bar{\mathbf{Y}}, \dots$
Propositional variable	p, q, r, \dots
Individual (first-order) variable	x, y, z, w, \dots
Second-order (set) variable	$X, Y, Z, W, \dots p, q, r, \dots$
Assignment (of individual variables)	$g : \text{iVar} \rightarrow M$
Valuation (of names/propositions)	$V : A \rightarrow \wp(D), \kappa^{\sharp} : \mathbf{P} \rightarrow \wp(S)$
Marking/coloring	$V^{\sharp} : D \rightarrow \wp(A), \kappa : S \rightarrow \wp(\mathbf{P})$

Chapter 3

Fragments of fixpoint logics

In this chapter we define and analyze fragments of both modal and first-order fixpoint logics. The main method that we will use to define these fragments is the restriction of the application of the fixpoint operator $\mu p.\varphi$ (and the first-order equivalent) to formulas φ having a special property. The main properties that we consider are *complete additivity* and *continuity*, but we will also consider other syntactic restrictions and their effects.

Complete additivity. In the first part of the chapter we study the notion of complete additivity: we start with a characterization of the finite approximants and fixpoints of completely additive *maps*. On the modal side, we define several fragments of μML which are based on different variants of complete additivity and perform a detailed analysis of their differences and similarities. The main result related to this analysis are the equivalences

$$\text{PDL} \equiv \mu_a\text{ML} \quad \text{and} \quad \text{PDL}^{tf} \equiv \mu_a^-\text{ML}$$

characterizing PDL and test-free PDL as fragments of the μ -calculus.

Next, we study complete additivity in the context of first-order with fixpoints. In this case, our main result is the equivalence

$$\text{FO}(\text{TC}^1) \equiv \mu_a\text{FOE}.$$

stating that transitive-closure logic is equivalent to the completely additive restriction of μFOE , which in a more standard notation would be called the completely additive restriction of $\text{FO}(\text{LFP}^1)$.

Continuity. In the second part of this chapter we perform a similar analysis of the notion of continuity, also giving a characterization of the fixpoints of continuous maps. After that, we first define the continuous restriction $\mu_c\text{ML}$ of μML and discuss its properties. The main result on the modal side is the equivalence:

$$\text{CPDL} \equiv \mu_{nc}\text{ML}^\vee,$$

where $\mu_{nc}\text{ML}^\vee \subseteq \mu_c\text{ML}$ is basically obtained from $\mu_c\text{ML}$ by adding an extra ‘separation of variables’ constraint under the fixpoint operator.

On the first-order side, we analyze the connection between continuity and (in)finiteness in first-order logics with fixpoints. This connection was studied by Park in [Par76] for μFOE . In our case, we consider the logic μFOE^∞ which has an additional generalized quantifier, and its continuous restriction $\mu_c\text{FOE}^\infty$. Our main result concerning μFOE^∞ is that:

$$\mu_c\text{FOE}^\infty \subsetneq \text{WMSO}.$$

To obtain this result, we show that the (continuous) fixpoint of $\mu_c\text{FOE}^\infty$ can be translated to WMSO, even in the presence of a generalized quantifier. We also discuss a counterexample that entails the strictness of the inclusion. In Chapter 7, however, we use automata-theoretic tools to prove that the converse of this last inclusion holds on the class of *trees*.

The question of Game Logic. In the third and last part of this chapter we discuss the relationship between the modal μ -calculus and Game Logic. In particular we approach the question of whether GL is equivalent to the 2-variable fragment $\mu\text{ML}[2]$ of μML . Our result for this part is the equivalence

$$\text{GL} \equiv \mu\text{ML}^\vee[2]$$

stating that GL is equivalent to the 2-variable fragment of the μ -calculus with the additional restriction of separation of variables. However, we leave as an open question whether $\mu\text{ML}^\vee[2]$ is equivalent to $\mu\text{ML}[2]$ or not. We also discuss particularities of these fragments and possible directions to separate them.

Apart from being of independent interest, as shown in this chapter, many of the introduced fragments will prove of importance in later chapters where we give automata and bisimulation-invariance characterizations for these logics.

3.1 Completely additive fragments

So far, we have only briefly introduced the notion of complete additivity for maps $F : \wp(S) \rightarrow \wp(S)$. We said that such a map is called completely additive if it distributes over *non-empty* families of sets, that is:

$$F\left(\bigcup_i P_i\right) = \bigcup_i F(P_i),$$

for every non-empty family of subsets $\{P_i \subseteq S\}_{i \in I}$. However, it is worth pointing out that the definition of complete additivity which is often found in the literature asks this condition for every *arbitrary* family of subsets, that is, including the

empty family. This variant, which we will call *normal complete additivity*, forces that $F(\emptyset) = \emptyset$. The name comes from the fact that operators preserving the empty set are called normal. From this observation it follows that the least fixpoint of these maps are not very interesting.

3.1.1. FACT. $\text{LFP}(F) = \emptyset$ for every normal and completely additive map F .

In this dissertation we will use both notions. However, depending on the objective of each section, mostly one of the two will be in the spotlight. In any case, the two notions are tightly connected.

3.1.2. FACT ([JT51, THEOREM 1.5]). A map $F : \wp(S) \rightarrow \wp(S)$ is completely additive if and only if $F(X) = K \cup G(X)$ for some $K \subseteq S$ and some normal and completely additive map $G : \wp(S) \rightarrow \wp(S)$.

We now go into an in-depth discussion of the notion of complete additivity (for non-empty families), in the more general context of maps $F : \wp(S)^n \rightarrow \wp(S)$.

3.1.3. DEFINITION. A map $F : \wp(S)^n \rightarrow \wp(S)$ is called *normal* (in the product) if $F(\emptyset, \dots, \emptyset) = \emptyset$ and *normal in the j^{th} -coordinate* if $F(X_1, \dots, \emptyset, \dots, X_n) = \emptyset$ for all $X_1, \dots, X_n \subseteq S$, where the empty set stands (only) in the j^{th} -coordinate.

It is important to observe that these two notions of normality do not coincide. For example, the map $F(A, B) = A \cup B$ is normal in the product but it is not normal in any of the coordinates. We will usually only use the concept of normality in the product and just call it normality, unless explicitly stated.

3.1.4. DEFINITION. A map $F : \wp(S)^n \rightarrow \wp(S)$ is *completely additive in the j^{th} -coordinate* if for every *non-empty* family of subsets $\{Y_i \subseteq S\}_{i \in I}$ and sets $X_1, \dots, X_n \subseteq S$ it satisfies:

$$F(X_1, \dots, \bigcup_i Y_i, \dots, X_n) = \bigcup_i F(X_1, \dots, Y_i, \dots, X_n),$$

where $\bigcup_i Y_i$ and Y_i are standing in the j^{th} coordinate. We say that F is *completely additive* (sometimes called completely additive in the product) if for every *non-empty* family $\{\bar{\mathbf{P}}_i \in \wp(S)^n\}_{i \in I}$ it satisfies:

$$F\left(\bigcup_i \bar{\mathbf{P}}_i\right) = \bigcup_i F(\bar{\mathbf{P}}_i).$$

3.1.5. REMARK. Observe that complete additivity in the j^{th} -coordinate implies monotonicity in the j^{th} -coordinate. Moreover, if a map is completely additive then it is so in every coordinate; however, the converse does not hold. A simple counterexample is given by the map $F(A, B) = A \cap B$.

To see that this map is completely additive in both coordinates consider a non-empty family of subsets $\{Y_i \subseteq S\}_{i \in I}$ and a set $X \subseteq S$. We only show that the property holds for the first coordinate, since the second case is symmetric: $F(\bigcup_i Y_i, X)$ is, by definition $(\bigcup_i Y_i) \cap X$. Using the distributive laws this is equivalent to $\bigcup_i (Y_i \cap X)$ which is, again by definition of F , the same as $\bigcup_i F(Y_i, X)$.

To check that F is not completely additive (in the product) consider the family $P_1 = (A, \emptyset)$ and $P_2 = (\emptyset, B)$ for subsets $A, B \subseteq S$ satisfying $A \cap B \neq \emptyset$. The following holds:

$$F((A, \emptyset) \cup (\emptyset, B)) = F(A, B) = A \cap B \neq \emptyset = F(A, \emptyset) \cup F(\emptyset, B),$$

which contradicts the definition of complete additivity in the product.

An alternative characterization of complete additivity in the j^{th} -coordinate is given by asking that F restricts to singletons (or the empty set) in that coordinate. More formally, F should satisfy, for every $Y \subseteq S$, the following:

$$F(X_1, \dots, Y, \dots, X_n) = F(X_1, \dots, \emptyset, \dots, X_n) \cup \bigcup_{y \in Y} F(X_1, \dots, \{y\}, \dots, X_n),$$

where Y is standing in the j^{th} coordinate. Along the same line, we can give an alternative characterization of complete additivity in the product. First, we need the following definition.

3.1.6. DEFINITION. Given $\bar{\mathbf{X}} \in \wp(S)^n$ we say that $\bar{\mathbf{Y}} \in \wp(S)^n$ is an atom of $\bar{\mathbf{X}}$ if and only if $\bar{\mathbf{Y}} = (\emptyset, \dots, \{x_i\}, \dots, \emptyset)$ for some element $x_i \in X_i$ standing at some coordinate i . We say that $\bar{\mathbf{Q}}$ is a *quasi-atom* if it is an atom or $\bar{\mathbf{Q}} = (\emptyset, \dots, \emptyset)$.

In this terminology, we can formulate the concept of complete additivity in the product by asking that F restricts to quasi-atoms; i.e., for every $\bar{\mathbf{P}} \in \wp(S)^n$, it should satisfy:

$$F(\bar{\mathbf{P}}) = \bigcup \{F(\bar{\mathbf{Q}}) \mid \bar{\mathbf{Q}} \text{ is a quasi-atom of } \bar{\mathbf{P}}\}.$$

Another way to read this last definition is that every $s \in F(\bar{\mathbf{P}})$ only depends on at most one singleton on one of the coordinates. Hence, a remarkable property of these maps is that the coordinates are, in some sense, independent of each other.

3.1.7. THEOREM (SEPARATION OF VARIABLES). A map $F : \wp(S)^n \rightarrow \wp(S)$ is completely additive if and only if it can be decomposed as

$$F(\bar{\mathbf{X}}) = K \cup \bigcup_i G_i(X_i),$$

for a set $K \subseteq S$ and normal completely additive maps $G_1, \dots, G_n : \wp(S) \rightarrow \wp(S)$. Moreover, F is normal and completely additive iff $F(\bar{\mathbf{X}}) = \bigcup_i G_i(X_i)$.

Proof. This theorem was already hinted by Jónsson and Tarski, in a more general algebraic setting [JT51, Theorem 1.5]. For our case, let $K := F(\emptyset, \dots, \emptyset)$ and define $G_i(X) := F(\emptyset, \dots, X, \dots, \emptyset) \setminus K$ where X is standing in the i^{th} coordinate and every other argument is \emptyset . It is easy to see that the required equality holds, using the definition of completely additive map. \square

3.1.1 Fixpoint theory of completely additive maps

As we observed in the previous section, fixpoints of *normal* and completely additive maps are trivial. On the other hand, the theory of fixpoints of (non-normal) completely additive maps is quite rich, and these maps satisfy very nice properties. One example is the following fact.

3.1.8. FACT. *Every completely additive map $F : \wp(S) \rightarrow \wp(S)$ is constructive, that is, $\text{LFP}(F) = \bigcup_{i \in \mathbb{N}} F^i(\emptyset)$.*

In this section we study the fixpoints of completely additive maps, mostly focusing on the approximants of the least fixpoint. We also establish some connections with duality theory.

Suppose now that we are given a map $G(X, Y)$ which is completely additive. A natural question is whether the (least) fixpoint operation preserves complete additivity. That is, whether $G'(Y) := \text{LFP}_X.G(X, Y)$ is completely additive as well. To answer that question, we will have to look at the finite approximants of $F(X) := G(X, Y)$ where Y is now fixed. In this subsection we give a fairly technical and precise characterization of the finite approximants of completely additive maps, and use it to prove the following theorem.

3.1.9. DEFINITION. Let $F : \wp(S) \rightarrow \wp(S)$ and $Y \subseteq S$. We define the restriction of F to Y as the function $F_{\upharpoonright Y} : \wp(Y) \rightarrow \wp(Y)$ given by $F_{\upharpoonright Y}(X) := F(X) \cap Y$.

3.1.10. THEOREM.

- (1) *If $G(X, \bar{Y})$ is completely additive then so is $H(\bar{Y}) := \text{LFP}_X.G(X, \bar{Y})$.*
- (2) *For every completely additive map $F : \wp(S) \rightarrow \wp(S)$ and $s \in S$ we have that*

$$s \in \text{LFP}(F) \quad \text{iff} \quad \text{there exists a finite set } Y \text{ such that } s \in \text{LFP}(F_{\upharpoonright Y})$$

where $Y = \{t_1, \dots, t_k\}$ satisfies $t_{i+1} \in F_{\upharpoonright Y}^{i+1}(\emptyset) \setminus F_{\upharpoonright Y}^i(\emptyset)$ and $t_k = s$.

The following lemma gives a precise characterization of the finite approximants of fixpoints of completely additive functions.

3.1.11. LEMMA. *Let $G : \wp(S)^{n+1} \rightarrow \wp(S)$ be a completely additive map. For every $s \in S$ and $\bar{Y} \in \wp(S)^n$ we have that $s \in \text{LFP}_X.G(X, \bar{Y})$ iff there exist $t_1, \dots, t_k \in S$ such that $t_k = s$ and the following conditions hold:*

- (i) $t_1 \in G(\emptyset, \overline{\mathbf{Q}})$ where $\overline{\mathbf{Q}} \in \wp(S)^n$ is a quasi-atom of $\overline{\mathbf{Y}}$; and
(ii) $t_{i+1} \in G(\{t_i\}, \overline{\emptyset})$, for all $1 \leq i < k$.

Proof. \Rightarrow As an abbreviation, define $F(X) := G(X, \overline{\mathbf{Y}})$. Let $s \in \text{LFP}(F)$ and let $k' \in \mathbb{N}$ be the smallest k' such that $s \in F^{k'}(\emptyset)$. Such k' exists because of Fact 3.1.8 (constructivity). We define elements $u_i \in F^i(\emptyset)$ by downwards induction:

- Case $i = k'$: we set $u_i := s$, which belongs to $F^{k'}(\emptyset)$.
- Case $i < k'$: we want to define u_i in terms of $u_{i+1} \in F^{i+1}(\emptyset)$. By definition we have that $u_{i+1} \in G(F^i(\emptyset), \overline{\mathbf{Y}})$. By complete additivity of G there is a quasi-atom $(T, \overline{\mathbf{Q}}')$ of $(F^i(\emptyset), \overline{\mathbf{Y}})$ such that $u_{i+1} \in G(T, \overline{\mathbf{Q}}')$. We consider the shape of the quasi-atom:
 - (1) If $T = \{t\}$ and $\overline{\mathbf{Q}}' = \overline{\emptyset}$ we set $u_i := t$ which satisfies $u_{i+1} \in G(\{u_i\}, \overline{\emptyset})$.
 - (2) If $T = \emptyset$ and $\overline{\mathbf{Q}}'$ is a quasi-atom of $\overline{\mathbf{Y}}$ we set $\overline{\mathbf{Q}} := \overline{\mathbf{Q}}'$ and finish the process.

Observe that case (2) will eventually occur. In the worst case this it will occur when $i = 1$, because $F^0(\emptyset)$ is defined as \emptyset .

This process defines a series of elements $u_{k'}, u_{k'-1}, \dots, u_j$ where $j \geq 1$. To define the elements t_j we just shift this sequence. That is, we set $k := k' - j + 1$ and $t_i := u_{j+i-1}$ for $1 \leq i \leq k$.

\Leftarrow This direction will easily follow from this claim:

CLAIM 1. $t_i \in F^i(\emptyset)$ for all $1 \leq i \leq k$.

PROOF OF CLAIM. We prove it by induction. For the base case, we have by hypothesis that $t_1 \in G(\emptyset, \overline{\mathbf{Q}})$ where $\overline{\mathbf{Q}}$ is a quasi-atom of $\overline{\mathbf{Y}}$. By monotonicity of G we then have $t_1 \in G(\emptyset, \overline{\mathbf{Y}})$ which means, by definition of F , that $t_1 \in F(\emptyset)$. For the inductive case let $t_{i+1} \in G(\{t_i\}, \overline{\emptyset})$. By inductive hypothesis $t_i \in F^i(\emptyset)$ therefore, by monotonicity of G , we have that $t_{i+1} \in G(F^i(\emptyset), \overline{\emptyset})$. Again by monotonicity, we get that $t_{i+1} \in G(F^i(\emptyset), \overline{\mathbf{Y}})$. By definition of F we can conclude that $t_{i+1} \in F^{i+1}(\emptyset)$. \blacktriangleleft

In particular, $t_k = s \in F^k(\emptyset)$ and therefore we get $s \in \text{LFP}_X.G(X, \overline{\mathbf{Y}})$. \square

Note that the above lemma is not restricted to any particular logic, as it expresses a property about an arbitrary completely additive map G . We can now prove our main theorem about completely additive maps.

Proof of Theorem 3.1.10(1). Let $G(X, \overline{Y})$ be a completely additive map and define $H(\overline{Y}) := \text{LFP}_X.G(X, \overline{Y})$. Suppose that $s \in H(\overline{Y})$. Let \overline{Q} be the quasi-atom of \overline{Y} given by Lemma 3.1.11, we will prove that $s \in H(\overline{Q}) = \text{LFP}_X.G(X, \overline{Q})$. Observe that, by the lemma, $t_1 \in G(\emptyset, \overline{Q})$. The key observation is that as $t_{i+1} \in G(\{t_i\}, \emptyset)$, by monotonicity we get that $t_{i+1} \in G(\{t_i\}, \overline{Q})$. From this it can be easily seen that, as $s \in G(\{t_{k-1}\}, \overline{Q})$ and G is monotone, we get that $s \in \text{LFP}_X.G(X, \overline{Q})$. \square

Proof of Theorem 3.1.10(2). Let $F : \wp(S) \rightarrow \wp(S)$ be a completely additive map and let $s \in S$; we prove that $s \in \text{LFP}(F)$ iff there exists Y such that $s \in \text{LFP}(F_{\upharpoonright Y})$ where $Y = \{t_1, \dots, t_k\}$ satisfies $t_{i+1} \in F_{\upharpoonright Y}^{i+1}(\emptyset) \setminus F_{\upharpoonright Y}^i(\emptyset)$.

\Rightarrow Let $Y = \{t_1, \dots, t_k\}$ be the set obtained using Lemma 3.1.11. In the lemma we already proved that $t_i \in F^i(\emptyset)$ for all i . We now prove the following stronger version of the claim:

CLAIM 2. $t_i \in F_{\upharpoonright Y}^i(\emptyset)$ for all i .

PROOF OF CLAIM. For the base case, we know that $t_1 \in F(\emptyset)$ by Claim 1 (Lemma 3.1.11); moreover, by definition $t_1 \in Y$. Hence $t_1 \in F(\emptyset) \cap Y$ which, by definition of $F_{\upharpoonright Y}$, is equivalent to $t_1 \in F_{\upharpoonright Y}(\emptyset)$. For the inductive case let $t_{i+1} \in F^{i+1}(\emptyset)$. By definition of F^{i+1} we have that $t_{i+1} \in F(F^i(\emptyset))$. Now we use the inductive hypothesis and get that $t_{i+1} \in F(F_{\upharpoonright Y}^i(\emptyset))$. As we did in the base case, because $t_{i+1} \in Y$, we know that $t_{i+1} \in F(F_{\upharpoonright Y}^i(\emptyset)) \cap Y$ which by definition of $F_{\upharpoonright Y}$ and regrouping we can conclude that $t_{i+1} \in F_{\upharpoonright Y}^{i+1}(\emptyset)$. \blacktriangleleft

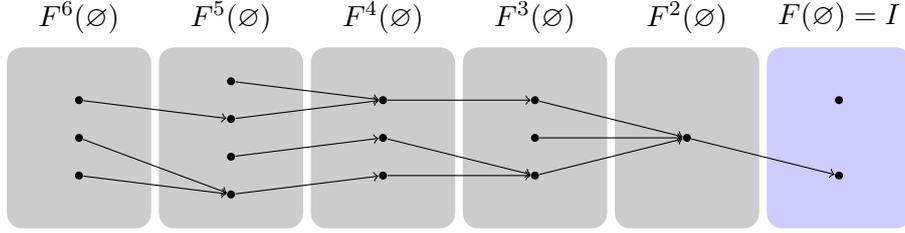
In particular $s \in F_{\upharpoonright Y}^k(\emptyset)$ and therefore $s \in \text{LFP}(F_{\upharpoonright Y})$.

\Leftarrow This direction goes through using a monotonicity argument. Using that for all X we have $F_{\upharpoonright Y}(X) \subseteq F(X)$, it is not difficult to prove that $F_{\upharpoonright Y}^\alpha(X) \subseteq F^\alpha(X)$ for all α , which entails that $\text{LFP}(F_{\upharpoonright Y}) \subseteq \text{LFP}(F)$. \square

Connections with duality theory. Further analysis of Theorem 3.1.10(2) reveals an interesting connection between fixpoints of completely additive maps and *transitive closure* of relations. We first recall a relationship between these maps and relations.

It is known from the early work of Jónsson and Tarski [JT51, Theorem 3.10], that *normal* completely additive maps and relations are *duals*. The (simplified) spirit of this correspondence is as follows: Given a normal and completely additive map $F : \wp(S) \rightarrow \wp(S)$ it is possible to construct a relation $F_\diamond \subseteq S \times S$ by setting $F_\diamond(s, t)$ iff $s \in F(\{t\})$. At the same time, given a relation $R \subseteq S \times S$ one can construct a *normal and completely additive* map $R^\diamond : \wp(S) \rightarrow \wp(S)$ by setting $R^\diamond(X) := R^{-1}[X]$. In a nutshell,

The normal and completely additive map F is dual to the relation F_\diamond .

Figure 3.1: First approximants of F and relation F_\blacklozenge .

3.1.12. REMARK. There is much more to be said about the above correspondence and other important results like the categorical duality developed by Thomason in [Tho75]. Such a discussion falls outside the scope of this dissertation, and we refer the reader to, for example [BRV01, Section 5.4].

On the other hand, if we want to consider *non-normal* completely additive maps then this correspondence does not work anymore, since $R^\diamond(\emptyset) = \emptyset$ for every R . Intuitively, we are losing the information of $F(\emptyset)$.

Broadly speaking, this problem can be fixed by adding a new set among the ingredients, as follows: Given a completely additive map $F : \wp(S) \rightarrow \wp(S)$ it is possible to give a tuple (F_\blacklozenge, I) consisting of a set $I \subseteq S$ of *impossible worlds* and a relation $F_\blacklozenge \subseteq (S \setminus I) \times S$ by setting $I := F(\emptyset)$ and $F_\blacklozenge(s, t)$ iff $s \in F(\{t\})$.¹ At the same time, given a tuple (R, I) with $R \subseteq (S \setminus I) \times S$ and $I \subseteq S$, one can construct a completely additive map $R^\blacklozenge : \wp(S) \rightarrow \wp(S)$ by setting $R^\blacklozenge(X) := R^{-1}[X] \cup I$. In other words,

The completely additive map F corresponds to the tuple $(F_\blacklozenge, F(\emptyset))$.

3.1.13. REMARK. The above paragraph presents a simplified view on a duality between BAOs and *Kripke frames with impossible worlds* (cf. [PSZng]). A Kripke frame with impossible worlds [Kri65, Sections 3,7] is a tuple $\mathbb{W} = \langle W, R, N \rangle$ where W is a non-empty set, $N \subseteq W$ and $R \subseteq N \times W$. The set N is supposed to represent the *normal* worlds, which are the complement of the impossible worlds. Observe that if N is the full set W (which is connected to $F(\emptyset) = \emptyset$), the kind of structures we get are isomorphic to Kripke frames.

Fig. 3.1 shows the approximants of a completely additive map in the light of this duality. If we look at how the relation (F_\blacklozenge) is defined in the Kripke model with impossible worlds we would see that it traverses backwards the approximants. Moreover, the impossible worlds are the worlds that have no outgoing arrows.

We can now finally analyze the statement of Theorem 3.1.10(2). An alternative way to read this result, is that an element s belongs to the least fixed point of a completely additive map F iff there is a finite sequence of elements from

¹Recall that as $F_\blacklozenge \subseteq (S \setminus I) \times S$, in this definition $s \notin I$.

$F(\emptyset), F^2(\emptyset), \dots$ which eventually reaches s . With this in mind and looking at Fig. 3.1, Theorem 3.1.10(2) can be reformulated as follows:

3.1.14. COROLLARY. *Let $F : \wp(S) \rightarrow \wp(S)$ be completely additive. For every $s \in S$ we have that $s \in \text{LFP}(F)$ if and only if $s \in (F_{\blacklozenge}^{-1})^*[F(\emptyset)]$.*

In other words, this corollary builds upon the previous correspondence by stating that to calculate the least fixpoint of F one can look at the transitive closure of (a derived relation based on) the dual relation.

The least fixpoint $\text{LFP}(F)$ corresponds to the set $(F_{\blacklozenge}^{-1})^*[F(\emptyset)]$.

In later sections we will see that if F is induced by a formula and we can express transitive closure in our logic then $(F_{\blacklozenge}^{-1})^*$ becomes definable. Therefore, we will be able to translate fixpoints of completely additive maps to transitive closure of an induced relation.

3.1.2 Characterization of PDL inside μML

It is well known that PDL can be translated to μML . However, the *exact* fragment of μML that corresponds to PDL has not been characterized. As we will see, the key notion leading to such a characterization is that of *complete additivity*.

3.1.15. DEFINITION. We say that $\varphi \in \mu\text{ML}$ is *completely additive in $Q \subseteq \mathbf{P}$* if

$$\mathbb{S} \Vdash \varphi \quad \text{iff} \quad \mathbb{S}[Q \mid \overline{\mathbf{Y}}] \Vdash \varphi \text{ for some quasi-atom } \overline{\mathbf{Y}} \text{ of } \kappa^{\sharp}(Q),$$

for every transition system \mathbb{S} . For a definition of normal complete additivity we just replace quasi-atoms with atoms.

Recall that the map $\kappa : S \rightarrow \wp(\mathbf{P})$ is the coloring of the transition map \mathbb{S} , and the valuation $\kappa^{\sharp} : \mathbf{P} \rightarrow \wp(S)$ was defined as $\kappa^{\sharp}(p) = \{s \in S \mid p \in \kappa(s)\}$.

It is also possible to give the following alternative definition, which enables us to use the fixpoint theory of completely additive maps, when useful.

3.1.16. PROPOSITION. *A formula $\varphi \in \mu\text{ML}$ is (normal and) completely additive in $Q \subseteq \mathbf{P}$ iff the associated functional F_Q^{φ} is (normal and) completely additive, for every transition system \mathbb{S} .*

The property of *normal* complete additivity was studied by van Benthem under the name of ‘continuity’ [Ben96]. He gave a syntactic characterization of the fragment of ML having this property.

3.1.17. THEOREM ([BEN96, THEOREM 5.19]). *A formula $\varphi \in \text{ML}$ is normal and completely additive in Q iff it is equivalent to a formula given by the fragment MLnADD_Q of ML which is defined as follows:*

$$\varphi := q \mid \psi \wedge \varphi \mid \varphi \vee \varphi \mid \langle \ell \rangle \varphi$$

where $\ell \in \text{D}$, $q \in Q$, and $\psi \in \text{ML}$ is Q -free.

Hollenberg [Hol98] also studied this notion, and started to move towards PDL. He proved a characterization for some programs of PDL. As he only considered formulas of ML , he could only capture the star-free programs of PDL.

3.1.18. THEOREM ([HOL98, THEOREM 2.6.6]). *A formula $\varphi \in \text{ML}$ is normal and completely additive in p if and only if $\varphi \equiv \langle \pi \rangle p$ for some p -free and star-free program $\pi \in \text{PDL}$.*

If we consider the modal μ -calculus instead of ML , the following example shows that there is an increase in expressive power, even with respect to completely additive formulas.

3.1.19. EXAMPLE. The formula $\varphi = \mu p. q \vee \langle \ell \rangle p$ induces a normal and completely additive map $F_q^\varphi : \wp(S) \rightarrow \wp(S)$. However, φ is not equivalent to any formula in MLnADD_q . This is easily seen because $\varphi \equiv \langle \ell^* \rangle q$ is not expressible in ML .

Fontaine and Venema [Fon10, FV12] gave a syntactic fragment μMLnADD_Q of μML —which is basically the closure of MLnADD_Q under the least fixpoint operator—and showed that $\varphi \in \mu\text{ML}$ is normal and completely additive in p iff it is equivalent to a formula in that fragment [Fon10, Theorem 5.5.3].

3.1.20. DEFINITION. Given a set $Q \subseteq \text{P}$, the fragment μMLnADD_Q of μML is inductively defined as follows:

$$\varphi := q \mid \alpha \wedge \varphi \mid \varphi \vee \varphi \mid \langle \ell \rangle \varphi \mid \mu p. \varphi'$$

where $\ell \in \text{D}$, $q \in Q$, $p \in \text{P}$, $\alpha \in \mu\text{ML}$ is Q -free and $\varphi' \in \mu\text{MLnADD}_{Qp}$.

3.1.21. THEOREM. *A formula $\varphi \in \mu\text{ML}$ is normal and completely additive in Q iff it is equivalent to a formula of μMLnADD_Q .*

Proof. Fontaine proves this equivalence for $Q = \{p\}$ in [Fon10, Theorem 5.5.3], however, in the inductive proof of this statement she actually uses the loaded statement of this theorem, for the fixpoint case. As we saw in Lemma 3.1.11, complete additivity *in the product* is critical to make the proof go through in the fixpoint case, and this is the reason for the need of the loaded statement. \square

A few syntactic properties of this fragment are stated in the following proposition.

3.1.22. PROPOSITION. *Let $\varphi \in \mu\text{MLnADD}_Q$, the following holds:*

- (i) φ is alternation-free,
- (ii) Every variable bound by a least fixpoint is existential (i.e., is only in the scope of diamonds); dually, every variable bound by a greatest fixpoint is universal (i.e., is only in the scope of boxes).

Unfortunately, no matter which syntactic fragment we choose to characterize normal and complete additivity, we will not be able to lift the exact statement of Theorem 3.1.18 to the μ -calculus. The following proposition shows that there are normal and completely additive formulas of μML which do not correspond to any program of PDL.

3.1.23. PROPOSITION. *There is a formula $\varphi \in \mu\text{ML}$ which is normal and completely additive in p but cannot be written as a PDL formula $\langle \pi \rangle p$.*

Proof. Let $\varphi := (\mu q. \Box q) \wedge p$, this formula is normal and completely additive in p . For example, this can be seen by using that $\varphi \in \mu\text{MLnADD}_p$. However, the formula is true at a point iff p holds in the point and the generated subtree is well-founded. This cannot be expressed in PDL [HTK00, Theorem 10.16]. \square

A close inspection of this counterexample reveals that μMLnADD_Q gives too much freedom in the clause $\alpha \wedge \varphi$, by letting $\alpha \in \mu\text{ML}$. We will see later that this kind of conjunction corresponds to a test on the PDL side, and hence we would be allowing tests of arbitrary μML formulas (in the example, a formula expressing well-foundedness).

It was also suggested in [Fon10, Remark 5.5.1] that PDL might be equivalent to the fragment of μML where the fixpoint operator $\mu p. \varphi$ is restricted to formulas φ which are normal and completely additive in p . However, if φ is normal then $\mu p. \varphi \equiv \perp$. Therefore, this cannot be the case. To be fair, this suggestion was almost right, and we will come back to this issue at the end of the section.

We now define another fragment $\mu_{na}\text{ML}$ of μML which intends to be equivalent to PDL and restricts the use of fixpoints of μML to formulas of a special form. Namely, the least fixpoint should be used as $\mu p. \beta \vee \varphi$ where $\varphi \in \mu\text{MLnADD}_p$ and β is p -free. Observe that if φ is normal and completely additive then $\beta \vee \varphi$ is (non-normal) completely additive.

3.1.24. DEFINITION. Formulas of $\mu_{na}\text{ML}$ are given by the following induction:

$$\alpha := p \mid \neg\alpha \mid \alpha \vee \alpha \mid \langle \ell \rangle \alpha \mid \mu p. \beta \vee \varphi,$$

where $p \in \text{P}$, $\ell \in \text{D}$, $\beta \in \mu_{na}\text{ML}$ is p -free and $\varphi \in \mu\text{MLnADD}_p \cap \mu_{na}\text{ML}$.

Along the same line, we can define a similar pair of fragments $\mu_{na}^- \text{ML}$ and μMLnADD_Q^- to characterize test-free PDL. The main difference between the fragments μMLnADD_Q and μMLnADD_Q^- is that the latter does not contain a clause for conjunction-with-constant (which amounts to testing).

3.1.25. DEFINITION. Given a set of propositions $Q \subseteq \mathbf{P}$, the formulas of fragment μMLnADD_Q^- of μML are defined as follows:

$$\varphi := q \mid \varphi \vee \varphi \mid \langle \ell \rangle \varphi \mid \mu p. \varphi'$$

where $\ell \in \mathbf{D}$, $q \in Q$, $p \in \mathbf{P}$, and $\varphi' \in \mu \text{MLnADD}_{Qp}^-$. Formulas of $\mu_{na}^- \text{ML}$ are given by the following induction:

$$\alpha := p \mid \neg \alpha \mid \alpha \vee \alpha \mid \langle \ell \rangle \alpha \mid \mu p. \beta \vee \varphi,$$

where $p \in \mathbf{P}$, $\ell \in \mathbf{D}$, $\beta \in \mu_{na}^- \text{ML}$ is p -free and $\varphi \in \mu \text{MLnADD}_p^- \cap \mu_{na}^- \text{ML}$.

A remarkable property of both μMLnADD_Q and μMLnADD_Q^- , which can be seen as the syntactic version of Theorem 3.1.7, is the following separation of variables.

3.1.26. PROPOSITION. *Given $\varphi \in \mu \text{MLnADD}_Q$ we can effectively construct a collection of formulas $\{\gamma_p \in \mu \text{MLnADD}_p \mid p \in Q\}$ such that $\varphi \equiv \bigvee_{p \in Q} \gamma_p$ and every γ_p is $Q \setminus \{p\}$ -free. Moreover, if $\varphi \in \mu \text{MLnADD}_Q^-$ then every $\gamma_p \in \mu \text{MLnADD}_p^-$.*

Proof. Using Theorem 3.1.7 we can see that setting $\gamma_p := \varphi[q \mapsto \perp \mid q \in Q \setminus \{p\}]$ works. It is also clear that every γ_p is $Q \setminus \{p\}$ -free and normal and completely additive in p . \square

We are now ready to prove the main theorem of this section, characterizing (test-free) PDL and programs as a fragment of the modal μ -calculus.

3.1.27. THEOREM. *PDL and test-free PDL are effectively equivalent to the fragments $\mu_{na} \text{ML}$ and $\mu_{na}^- \text{ML}$, respectively.*

The theorem follows directly from Propositions 3.1.29 and 3.1.30 below. We start with the direction from PDL and PDL^{tf} to $\mu_{na} \text{ML}$ and $\mu_{na}^- \text{ML}$.

3.1.28. DEFINITION. By a simultaneous induction on formulas and programs of PDL, we define, for each program $\pi \in \text{PDL}$, a function $f_\pi : \mu \text{ML} \rightarrow \mu \text{ML}$ on the set of modal fixpoint formulas, and a map $(-)^t$ from PDL to μML :

$$\begin{array}{ll} f_\ell(\alpha) & := \langle \ell \rangle \alpha \\ f_{\varphi?}(\alpha) & := \varphi^t \wedge \alpha \\ f_{\pi \oplus \pi'}(\alpha) & := f_\pi(\alpha) \vee f_{\pi'}(\alpha) \\ f_{\pi; \pi'}(\alpha) & := f_\pi(f_{\pi'}(\alpha)) \\ f_{\pi^*}(\alpha) & := \mu p. \alpha \vee f_\pi(p) \end{array} \quad \begin{array}{ll} p^t & := p \\ (\neg \varphi)^t & := \neg \varphi^t \\ (\varphi_0 \vee \varphi_1)^t & := \varphi_0^t \vee \varphi_1^t \\ (\langle \pi \rangle \varphi)^t & := f_\pi(\varphi^t) \end{array}$$

where, in the clause for f_{π^*} , p is a fresh variable.

The following proposition says that the translation $(-)^t$ is the required embedding of PDL into the fragment $\mu_{na}\text{ML}$.

3.1.29. PROPOSITION.

(1) for every program $\pi \in \text{PDL}$, and every formula $\alpha \in \mu_{na}\text{ML}$:

- (1a) $f_\pi(\alpha)$ belongs to $\mu_{na}\text{ML}$,
- (1b) $f_\pi(\alpha) \in \mu\text{MLnADD}_Q$ and $\text{FV}(f_\pi(\alpha)) = \text{FV}(\alpha) \cup \text{FV}(\pi)$,
given $\alpha \in \mu\text{MLnADD}_Q$ and $\text{FV}(\pi) \cap Q = \emptyset$, and
- (1c) $\langle \pi \rangle \alpha \equiv f_\pi(\alpha)$.

(2) for every formula $\alpha \in \text{PDL}$:

- (2a) $\alpha^t \in \mu_{na}\text{ML}$, and
- (2b) $\alpha \equiv \alpha^t$.

Analogous statements can be proved for PDL^{tf} , μMLnADD_Q^- and $\mu_{na}^-\text{ML}$.

Proof. We prove it by simultaneous induction on formulas and programs.

Proof of item (1):

- First consider the case where $\pi = \ell$ for some atomic program ℓ . Recall that $f_\ell(\alpha)$ is then defined as $\langle \ell \rangle \alpha$. Items (1a) and (1b) are immediate by the definition of the fragments $\mu_{na}\text{ML}$ and μMLnADD_Q , and (1c) is immediate by the definition of f_ℓ .
- In case $\pi = \psi?$ for some $\psi \in \text{PDL}$, by the induction hypothesis on formulas we may assume that ψ^t belongs to $\mu_{na}\text{ML}$ and $\text{FV}(\psi^t) = \text{FV}(\psi)$. For (1a), recall that $\alpha \in \mu_{na}\text{ML}$. Then $f_\pi(\alpha) = \psi^t \wedge \alpha$ belongs to $\mu_{na}\text{ML}$ since $\mu_{na}\text{ML}$ is closed under taking conjunctions. For (1b), let Q be a set of variables such that $\alpha \in \mu\text{MLnADD}_Q$ and $\text{FV}(\pi) \cap Q = \emptyset$. By the latter fact, $\text{FV}(\psi) \cap Q = \emptyset$, so ψ (and ψ^t) is a Q -free formula. From this it is immediate that $f_\pi(\alpha) = \psi^t \wedge \alpha$ belongs to μMLnADD_Q . For (1c) observe that the equivalence of $\langle \psi? \rangle \alpha$ and $\psi^t \wedge \alpha$ is immediate by the inductive hypothesis ($\psi \equiv \psi^t$) and the meaning of test programs.
- The case that $\pi = \pi' \oplus \pi''$ is easy and left for the reader.
- For the case that $\pi = \pi'; \pi''$, first consider a formula $\alpha \in \mu_{na}\text{ML}$. Then by successively applying the inductive hypothesis to π' and π'' , we see that first the formula $f_{\pi''}(\alpha)$, and then the formula $f_\pi(\alpha) = f_{\pi'}(f_{\pi''}(\alpha))$ belongs to $\mu_{na}\text{ML}$. This proves (1a). For (1b), let Q be a set of variables such that $\alpha \in \mu\text{MLnADD}_Q$ and no variable from Q occurs in π . Then in particular, no variable from Q occurs in π'' nor in π' . Hence, by the inductive hypothesis for π'' we find that

$f_{\pi''}(\alpha) \in \mu\text{MLnADD}_Q$, and subsequently applying the inductive hypothesis for π' we see that $f_{\pi}(\alpha) = f_{\pi'}(f_{\pi''}(\alpha)) \in \mu\text{MLnADD}_Q$. The statement on the free variables can be verified in the same way. For (1c), it suffices to check the validity of the following chain of equivalences:

$$\langle \pi'; \pi'' \rangle \alpha \equiv \langle \pi' \rangle \langle \pi'' \rangle \alpha \equiv \langle \pi' \rangle f_{\pi''}(\alpha) \equiv f_{\pi'}(f_{\pi''}(\alpha)).$$

- Consider the case where $\pi = \varrho^*$. Take an arbitrary formula $\alpha \in \mu_{na}\text{ML}$. Recall that $f_{\pi}(\alpha)$ is defined as $\mu p.\alpha \vee f_{\varrho}(p)$, where p occurs neither in α nor in ϱ . For (1a), we apply the inductive hypothesis (1a) and (1b) to ϱ and the formula $p \in \mu\text{MLnADD}_p$, and get that $f_{\varrho}(p) \in \mu\text{MLnADD}_p \cap \mu_{na}\text{ML}$. Therefore $\mu p.\alpha \vee f_{\varrho}(p)$ indeed belongs $\mu_{na}\text{ML}$. For (1b), let Q be a set of variables that do not occur in ϱ , assume that $\alpha \in \mu\text{MLnADD}_Q$. Since p does not occur in α this means that $\alpha \in \mu\text{MLnADD}_{Qp}$. We already proved that $f_{\varrho}(p) \in \mu\text{MLnADD}_p$ for (1a). Moreover we also get $\text{FV}(f_{\varrho}(p)) = \text{FV}(\varrho) \cup \{p\}$ and therefore $f_{\varrho}(p)$ is $Q \setminus \{p\}$ -free. From this we can conclude that $f_{\varrho}(p) \in \mu\text{MLnADD}_{Qp}$. Hence the disjunction $\alpha \vee f_{\varrho}(p)$ belongs to μMLnADD_{Qp} , and from this we may conclude that $f_{\pi}(\alpha) \in \mu\text{MLnADD}_Q$. It is also clear that $\text{FV}(f_{\pi}(\alpha)) = \text{FV}(\alpha) \cup \text{FV}(\pi)$. For (1c), we verify that $\langle \varrho^* \rangle \alpha \equiv \mu p.\alpha \vee \langle \varrho \rangle p \equiv \mu p.\alpha \vee f_{\varrho}(p) \equiv f_{\pi}(\alpha)$.

Proof of item (2): we only consider the inductive case where α is of the form $\langle \pi \rangle \beta$. Inductively, we may assume that $\beta^t \in \mu_{na}\text{ML}$ and that $\beta \equiv \beta^t$. Applying the inductive hypothesis to the program π we obtain: by (1a) that $\alpha^t = f_{\pi}(\beta^t)$ belongs to $\mu_{na}\text{ML}$; and by (1c) that $\alpha^t = f_{\pi}(\beta^t)$ is equivalent to the formula $\alpha = \langle \pi \rangle \beta$. This suffices to prove the proposition. \square

The translation in the other direction is provided by the following proposition.

3.1.30. PROPOSITION. *The following procedures can be performed effectively:*

- (i) *Given a formula $\alpha \in \mu_{na}\text{ML}$, return an equivalent formula $\alpha^s \in \text{PDL}$. Moreover, if $\alpha \in \mu_{na}^- \text{ML}$ then $\alpha^s \in \text{PDL}^{\text{tf}}$.*
- (ii) *Given $\varphi \in \mu\text{MLnADD}_p \cap \mu_{na}\text{ML}$, return a p -free program $\pi \in \text{PDL}$ such that $\varphi \equiv \langle \pi \rangle p$. Moreover, if $\varphi \in \mu\text{MLnADD}_p^- \cap \mu_{na}^- \text{ML}$, then $\pi \in \text{PDL}^{\text{tf}}$.*

Proof. We prove the proposition via a mutual induction on the fragments $\mu_{na}\text{ML}$ and μMLnADD_p . The stronger statements concerning formulas in the restricted fragments follow from an easy inspection.

Proof of item (i): Leaving the other cases to the reader, we focus on the inductive step, where we are dealing with a formula $\mu p.\beta \vee \varphi$, where $\beta \in \mu_{na}\text{ML}$ is p -free and $\varphi \in \mu\text{MLnADD}_p$. In order to find the right translation for this formula, we use the induction hypothesis twice. We use item (ii) on φ , that is, we assume

that $\varphi \equiv \langle \pi \rangle p$ where $\pi \in \text{PDL}$ is p -free. We also apply item (i) on β , and hence we know that $\beta^s \equiv \beta$ and $\beta^s \in \text{PDL}$. Hence, if we define the translation as

$$(\mu p.\beta \vee \varphi)^s := \langle \pi^* \rangle \beta^s,$$

it is easy to verify that this definition satisfies the required properties.

Proof of item (ii): If $\varphi = p$, define $\pi := \top?$. (If additionally $\varphi \in \mu\text{MLnADD}_p^-$ and we need to land in test-free PDL, we set $\pi := \epsilon$. This is the reason for adding the skip program ϵ to PDL^{tf}). With this definition, we have that $\varphi = p \equiv \langle \pi \rangle p$. For the inductive step, we do as follows:

- First consider the case that $\varphi = \varphi' \vee \varphi''$. Then inductively $\varphi' \equiv \langle \pi' \rangle p$, and $\varphi'' \equiv \langle \pi'' \rangle p$. It is straightforward to verify that setting $\pi := \pi' \oplus \pi''$ works.
- Consider the case that $\varphi = \alpha \wedge \varphi'$, where $\alpha \in \mu\text{ML}$ is p -free, and the formula $\varphi' \in \mu\text{MLnADD}_Q$. We now crucially use that as $\varphi \in \mu\text{MLnADD}_p \cap \mu_{na}\text{ML}$ then $\alpha \in \mu_{na}\text{ML}$. Using the inductive hypothesis, by item (i), we have $\alpha \equiv \alpha^s$ for $\alpha^s \in \text{PDL}$ and, by item (ii), we have $\varphi' \equiv \langle \pi' \rangle p$. It follows that

$$\varphi \equiv \langle \alpha^s?; \pi' \rangle p.$$

Therefore, setting $\pi := \alpha^s?; \pi'$ works.

- For $\varphi = \langle \ell \rangle \varphi'$, we inductively have that $\varphi' \equiv \langle \pi' \rangle p$. From this it follows that $\varphi \equiv \langle \ell; \pi' \rangle p$. And therefore we set $\pi := \ell; \pi'$.
- Finally, consider the case that $\varphi = \mu q.\varphi'$ where $\varphi' \in \mu\text{MLnADD}_{pq}$. Using Proposition 3.1.26 we assume, without loss of generality, that φ' was already rewritten as a disjunction of formulas separating the variables.² That is, we only consider the case $\varphi = \mu q.\psi_p \vee \psi_q$ where $\psi_p \in \mu\text{MLnADD}_p$ is q -free and $\psi_q \in \mu\text{MLnADD}_q$ is p -free. Using item (ii) of the inductive hypothesis we get that $\varphi \equiv \mu q.\langle \pi_p \rangle p \vee \langle \pi_q \rangle q$ for some $\pi_p, \pi_q \in \text{PDL}$. From this, it is easy to see that the following equivalence holds:

$$\varphi \equiv \langle \pi_q^*; \pi_p \rangle p.$$

The above equation justifies setting $\pi := \pi_q^*; \pi_p$.

This finishes the proof of the proposition. \square

As a corollary of these propositions, we can now extend the connection between MLnADD and star-free PDL-programs implied by Theorem 3.1.18 to μMLnADD and full PDL-programs.

²The induction can be stated in a different way that makes it clear that this increase in formula complexity does not bring problems. We do not do it because it would make the presentation more involved.

3.1.31. COROLLARY. *A formula $\varphi \in \mu\text{ML}$ belongs to $\mu\text{MLnADD}_p \cap \mu_{na}\text{ML}$ (resp. to $\mu\text{MLnADD}_p^- \cap \mu_{na}^-\text{ML}$) iff it is equivalent to $\langle \pi \rangle p$ for some p -free program $\pi \in \text{PDL}$ (resp. $\pi \in \text{PDL}^{tf}$).*

Proof. The left-to-right direction is given by Proposition 3.1.30(ii) and the other direction is given by Proposition 3.1.29(1). \square

Relation to non-normal complete additivity. The results of this section were, in most cases, developed for normal complete additivity since, as witnessed by the above corollary, this notion seems to be closely connected with PDL programs. However, a similar analysis can be done for complete additivity.

We can define two fragments μMLADD_Q and $\mu_a\text{ML}$ of μML . The first one, intends to capture completely additive formulas of μML . The second one restricts the fixpoints *directly* to formulas of μMLADD_Q .

3.1.32. DEFINITION. Given a set of propositions $Q \subseteq \text{P}$, the formulas of the fragment μMLADD_Q of μML are given as follows:

$$\varphi := q \mid \alpha \mid \alpha \wedge \varphi \mid \varphi \vee \varphi \mid \langle \ell \rangle \varphi \mid \mu p. \varphi'$$

where $\ell \in \text{D}$, $q \in Q$, $p \in \text{P}$, $\alpha \in \mu\text{ML}$ is Q -free and $\varphi' \in \mu\text{MLADD}_Q$. Formulas of $\mu_a\text{ML}$ are given by:

$$\alpha := p \mid \neg \alpha \mid \alpha \vee \alpha \mid \langle \ell \rangle \alpha \mid \mu p. \varphi,$$

where $p \in \text{P}$, $\ell \in \text{D}$ and $\varphi \in \mu\text{MLADD}_p \cap \mu_a\text{ML}$.

The difference between μMLnADD_Q and μMLADD_Q is the addition of an extra clause $\alpha \in \mu\text{ML}$; the difference between $\mu_{na}\text{ML}$ and $\mu_a\text{ML}$ is the removal of the disjunct in the fixpoint operator. The following proposition is easily checked.

3.1.33. PROPOSITION. *Every $\varphi \in \mu\text{MLADD}_Q$ is completely additive in Q .*

In Fact 3.1.2 we say that completely additive *maps* can be represented as the union of a constant and a normal and completely additive map. This representation transfers to formulas as well. We prove the following stronger proposition.

3.1.34. PROPOSITION. *If $\varphi \in \mu\text{ML}$ is completely additive in Q (in particular, if $\varphi \in \mu\text{MLADD}_Q$) then $\varphi \equiv \psi \vee \varphi'$ for some $\varphi' \in \mu\text{MLnADD}_Q$ and Q -free $\psi \in \mu\text{ML}$.*

Proof. We set $\psi := \varphi[q \mapsto \perp \mid q \in Q]$ and $\gamma := \varphi \wedge \neg \psi$. It is straightforward to see that $\varphi \equiv \psi \vee \gamma$ and that $\gamma \in \mu\text{ML}$ is *normal* and completely additive in Q . Moreover, as $\psi \in \mu\text{ML}$ is Q -free, then $\psi \in \mu\text{MLADD}_Q$. Using Theorem 3.1.21 on γ we get a formula $\varphi' \in \mu\text{MLnADD}_Q$ such that $\gamma \equiv \varphi'$. Putting everything together, we get that $\varphi \equiv \psi \vee \varphi'$ with $\varphi' \in \mu\text{MLnADD}_Q$. \square

As a corollary of the last two propositions, we get the soundness and completeness of the fragment μMLADD_Q with respect to the property of complete additivity in Q .

3.1.35. COROLLARY. *A formula $\varphi \in \mu\text{ML}$ is completely additive in Q iff it is equivalent to a formula of μMLADD_Q .*

Proof. The right-to-left direction is direct by Proposition 3.1.33. For the other direction, consider a formula $\varphi \in \mu\text{ML}$ which is completely additive in Q . We use Proposition 3.1.34 and get that $\varphi \equiv \psi \vee \varphi'$ where ψ is Q -free and $\varphi' \in \mu\text{MLnADD}_Q$. As $\mu\text{MLnADD}_Q \subseteq \mu\text{MLADD}_Q$, we have $\varphi' \in \mu\text{MLADD}_Q$. Using that the fragment μMLADD_Q is closed under disjunction, we get that $\psi \vee \varphi' \in \mu\text{MLADD}_Q$. \square

In the light of Proposition 3.1.34 and Corollary 3.1.31, the following proposition is straightforward.

3.1.36. PROPOSITION. *A formula $\varphi \in \mu\text{ML}$ belongs to $\mu\text{MLADD}_p \cap \mu_a\text{ML}$ iff it is equivalent to $\psi \vee \langle \pi \rangle p$ for some p -free program $\pi \in \text{PDL}$ and p -free $\psi \in \text{PDL}$.*

It should be observed that $\mu\text{MLADD}_Q \not\equiv \mu\text{MLnADD}_Q$. However, the fragments $\mu_a\text{ML}$ and $\mu_{na}\text{ML}$ are in fact equivalent. This gives us two ways to look at PDL in the light of the μ -calculus.

3.1.37. PROPOSITION. *$\mu_a\text{ML} \equiv \mu_{na}\text{ML}$ and hence $\mu_a\text{ML} \equiv \text{PDL}$.*

Proof. The difference between $\mu_a\text{ML}$ and $\mu_{na}\text{ML}$ lies in the fixpoint operator.

- (i) The fragment $\mu_{na}\text{ML}$ has a clause $\mu p.\varphi$ with $\varphi \in \mu\text{MLADD}_p \cap \mu_a\text{ML}$,
- (ii) The fragment $\mu_a\text{ML}$ has a clause $\mu p.\beta \vee \varphi$ where $\beta \in \mu_{na}\text{ML}$ is p -free and $\varphi \in \mu\text{MLnADD}_p \cap \mu_{na}\text{ML}$.

For the direction from (i) to (ii) we focus on the fixpoint clause $\mu p.\varphi$ with $\varphi \in \mu\text{MLADD}_p \cap \mu_a\text{ML}$. We make a detour through PDL applying Proposition 3.1.36 and getting $\varphi \equiv \psi \vee \langle \pi \rangle p$ for some p -free program $\pi \in \text{PDL}$ and p -free $\psi \in \text{PDL}$. We now apply Proposition 3.1.29(2) on ψ and Proposition 3.1.29(1) on $\langle \pi \rangle p$ and get that $\varphi \equiv \psi \vee \langle \pi \rangle p \equiv \psi' \vee f_\pi(p)$ where $\psi' \in \mu_{na}\text{ML}$ is p -free and $f_\pi(p)$ belongs to $\mu\text{MLnADD}_p \cap \mu_{na}\text{ML}$.

For the other direction, there is no need for a translation (i.e., the identity translation works), we only need a change of perspective. The key argument is that if β is p -free and $\varphi \in \mu\text{MLnADD}_p$ then $\beta \vee \varphi \in \mu\text{MLADD}_p$. \square

From these last propositions we may conclude that PDL programs are better related to *normal* complete additivity, whereas PDL is better related to (non-normal) complete additivity, since we can get it by restricting the fixpoint operator directly to that fragment. If we now look back at the suggestion of [Fon10, Remark 5.5.1] which stated that PDL might be obtained by restricting the least fixpoint of μML to normal complete additivity we can see that even though not correct, the suggestion was on the right path. Moreover, in [Fon08] the author actually suggests that PDL could be obtained by restricting the least fixpoint to *non-normal* complete additivity. Therefore, it could be that this discrepancy is just an artifact from the transcription of [Fon08] to [Fon10].

To close this section, we would like to make a few remarks about some nice properties of complete additivity, and how this properties produce an extremely malleable fragment μMLnADD_Q . The question that we would like to discuss is: what does μMLnADD_Q have, that makes it translatable to PDL? This question is interesting, since in later sections we will try to translate other (not-so-well-behaved) fragments into CPDL and GL.

Separation of variables. It is well-known that star-free PDL (that is, PDL without the iteration operator) has the same expressive power as the multi-modal logic ML based on \mathbf{D} . This can already be seen in the translations of this chapter or, for example, by the equivalences $\langle \pi \oplus \pi' \rangle \varphi \equiv \langle \pi \rangle \varphi \vee \langle \pi' \rangle \varphi$, $\langle \psi? \rangle \varphi \equiv \psi \wedge \varphi$, etc., which provide a way to rewrite any star-free PDL formula to ML.

Therefore, the crucial step in our proofs is, unsurprisingly, the handling of the fixpoint operator. Concretely, in the proof of Theorem 3.1.30(ii) consider the case of a formula $\mu p.\varphi'$ with $\varphi' \in \mu\text{MLnADD}_{Qp}$. In our proof we use Proposition 3.1.26 together with the inductive hypothesis to rewrite this formula as:

$$\mu p.(\langle \pi_q \rangle q \vee \langle \pi_p \rangle p).$$

To simplify and abstract our setting, we can write this as $\gamma = \mu p.(\alpha \vee \langle \pi \rangle p)$, where $\alpha \in \text{PDL}$ is p -free. Observe that if we consider γ in the light of the game semantics for the μ -calculus then its evaluation game would broadly go as follows: first the fixpoint gets discarded and we are left with the main proper subformula; after that, \exists makes the choice of whether she wants to consider a potential repetition (choosing the disjunct which has p) or finish the repetitions (choosing α , which is p -free). Of course, as we are considering a least fixpoint, she can only regenerate p finitely many times if she wants to (have a chance to) win. From this analysis, it should be clear that

$$\mu p.(\langle \pi_q \rangle q \vee \langle \pi_p \rangle p) \equiv \langle \pi^* \rangle \alpha.$$

The main point that we are trying to make is that, from all the nice properties that we get from the completely additive fragment, we mainly used that

if we have a formula $\mu p.\varphi$ then we can separate φ in two disjuncts such that (1) one disjunct has a program-like syntax (obtained inductively) and is q -free; and (2) the other disjunct is p -free. Luckily, these requirements are easily satisfied by completely additive formulas. However, we will see that this need not be the case for continuous formulas.

The separation and freeness constraints are, we think, tightly related to the control that \exists has in a formula of the form $\langle \pi^* \rangle \alpha$, since *she* should be able to always choose the number of repetitions.

Two variables are enough. PDL is not only translatable to μML (or, as we know now, to $\mu_a\text{ML}$) but it can already be translated to the two-variable fragment of μML , that is, $\mu\text{ML}[2]$. This phenomenon also happens with CPDL and Game Logic [BGL05, Lemma 47]. It can be proved that thanks to the separation of variables *under the fixpoint* it is possible to rewrite any formula of $\mu_a\text{ML}$ with just two binding variables. We discuss this further in Section 3.3 in the context of Game Logic.

3.1.3 Characterization of $\text{FO}(\text{TC}^1)$ inside $\text{FO}(\text{LFP}^1)$

It is well-known that the transitive closure of $\text{FO}(\text{TC}^1)$ can be expressed with the fixpoint operators of μFOE . In fact, we explicitly showed how to do it in Remark 2.7.7. In this section we prove that $\text{FO}(\text{TC}^1)$ is equivalent to $\mu_a\text{FOE}$, the fragment of μFOE where the least fixpoint operator is restricted to completely additive formulas. That is,

$$\text{FO}(\text{TC}^1) \equiv \mu_a\text{FOE} \text{ over all models.}$$

3.1.38. DEFINITION. We say that $\varphi \in \mu\text{FOE}$ is *completely additive in* $\mathbf{Q} \subseteq \mathbf{P}$ if for every model \mathbb{M} and assignment g it satisfies

$$\mathbb{M}, g \models \varphi \quad \text{iff} \quad \mathbb{M}[\mathbf{Q} \upharpoonright \bar{\mathbf{Y}}], g \models \varphi \text{ for some quasi-atom } \bar{\mathbf{Y}} \text{ of } \kappa^{\mathfrak{h}}(\mathbf{Q}).$$

3.1.39. PROPOSITION. *If $\varphi \in \mu\text{FOE}$ is completely additive in \mathbf{Q} then the associated functional $F_{\mathbf{Q}:x}^\varphi : \wp(M)^n \rightarrow \wp(M)$ is completely additive, for every model \mathbb{M} and variable $x \in \text{FV}(\varphi)$.*

Proof. Fix a model \mathbb{M} , assignment g and free variable $x \in \text{FV}(\varphi)$. We want to prove that $F_{\mathbf{Q}:x}^\varphi(\bar{\mathbf{Z}})$ is completely additive. An element t belongs to $F_{\mathbf{Q}:x}^\varphi(\bar{\mathbf{Z}})$ iff $\mathbb{M}[\mathbf{Q} \mapsto \bar{\mathbf{Z}}], g[x \mapsto t] \models \varphi$. By complete additivity of φ , this occurs iff $\mathbb{M}[\mathbf{Q} \mapsto \bar{\mathbf{Y}}], g[x \mapsto t] \models \varphi$ for some quasi-atom $\bar{\mathbf{Y}}$ of $\bar{\mathbf{Z}}$. By definition of $F_{\mathbf{Q}:x}^\varphi$, this is equivalent to saying that $t \in F_{\mathbf{Q}:x}^\varphi(\bar{\mathbf{Y}})$. Therefore, $F_{\mathbf{Q}:x}^\varphi$ is completely additive. \square

Next, we provide a definition of a fragment of μFOE , and shortly after that we prove that every formula in this fragment is completely additive.

3.1.40. DEFINITION. Let $\mathbf{Q} \subseteq \mathbf{P}$ be a set of monadic predicates. The fragment $\mu\text{FOEADD}_{\mathbf{Q}}$ is defined by the following rules:

$$\varphi ::= \psi \mid q(x) \mid \exists x.\varphi(x) \mid \varphi \vee \varphi \mid \varphi \wedge \psi \mid [\text{LFP}_{p:x}.\xi(p,x)](z)$$

where $q \in \mathbf{Q}$, $\psi \in \mu\text{FOE}$ is \mathbf{Q} -free, $p \in \mathbf{P} \setminus \mathbf{Q}$ and $\xi(p,x) \in \mu\text{FOEADD}_{\mathbf{Q}_p}$.

Observe that, in this definition, the atomic formulas given by equality and relations are taken into account by the ψ clause.

3.1.41. PROPOSITION. *Every $\varphi \in \mu\text{FOEADD}_{\mathbf{Q}}$ is completely additive in \mathbf{Q} .*

Proof. The proof goes by induction.

- If $\varphi = \psi \in \mu\text{FOE}$ is \mathbf{Q} -free then changes in the \mathbf{Q} part of the valuation will make no difference and hence the condition is trivial.
- Case $\varphi = q(x)$: if $\mathbb{M}, g \models q(x)$ then $g(x) \in \kappa^{\sharp}(q)$. Clearly, we can restrict the valuation of q to $\{g(x)\}$ and get $\mathbb{M}[q \upharpoonright \{g(x)\}; \mathbf{Q} \setminus \{q\} \mapsto \emptyset], g \models q(x)$.
- Case $\varphi = \varphi_1 \vee \varphi_2$: assume $\mathbb{M}, g \models \varphi$. Without loss of generality we can assume that $\mathbb{M}, g \models \varphi_1$ and hence by induction hypothesis there is a quasi-atom $\overline{\mathbf{Y}}$ of $\kappa^{\sharp}(\mathbf{Q})$ such that $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g \models \varphi_1$ which clearly satisfies $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g \models \varphi$.
- Case $\varphi = \varphi_1 \wedge \psi$: assume $\mathbb{M}, g \models \varphi$. By induction hypothesis we have a quasi-atom $\overline{\mathbf{Y}}$ of $\kappa^{\sharp}(\mathbf{Q})$ such that $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g \models \varphi_1$. Observe that $\mathbb{M}, g \models \psi$ and as ψ is \mathbf{Q} -free we also have $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g \models \psi$. Hence, $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g \models \varphi$ holds.
- Case $\varphi = \exists x.\varphi'(x)$ and $\mathbb{M}, g \models \varphi$. By definition there exists $u \in M$ such that $\mathbb{M}, g[x \mapsto u] \models \varphi'(x)$. By induction hypothesis there exists a quasi-atom $\overline{\mathbf{Y}}$ of $\kappa^{\sharp}(\mathbf{Q})$ such that $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g[x \mapsto u] \models \varphi'(x)$ and hence $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g \models \exists x.\varphi'(x)$.
- Let φ be $[\text{LFP}_{p:x}.\psi(p,x)](z)$, we have to prove that

$$\mathbb{M}, g \models \varphi \quad \text{iff} \quad \mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g \models \varphi \text{ for some quasi-atom } \overline{\mathbf{Y}} \text{ of } \kappa^{\sharp}(\mathbf{Q}).$$

By semantics of the fixpoint operator $\mathbb{M}, g \models \varphi$ iff $g(z) \in \text{LFP}(F_{p:x}^{\psi})$. It will be useful to take a slightly more general perspective and consider the map

$$F_{\mathbf{Q}p:x}^{\psi}(P, \overline{\mathbf{Z}}) := \{t \in M \mid \mathbb{M}[p \mapsto P; \mathbf{Q} \mapsto \overline{\mathbf{Z}}], g[x \mapsto t] \models \psi\}$$

and observe that $F_{p:x}^{\psi}(P) = F_{\mathbf{Q}p:x}^{\psi}(P, \kappa^{\sharp}(\mathbf{Q}))$ and therefore their least fixpoints will be the same. By inductive hypothesis and Proposition 3.1.39, we know that $F_{\mathbf{Q}p:x}^{\psi}(P, \overline{\mathbf{Z}})$ is completely additive. Using Theorem 3.1.10(1) we get that $\text{LFP}_P.F_{\mathbf{Q}p:x}^{\psi}(P, \overline{\mathbf{Z}})$ is completely additive as well. In particular,

$$\begin{aligned} t \in \text{LFP}_P.F_{\mathbf{Q}p:x}^{\psi}(P, \kappa^{\sharp}(\mathbf{Q})) & \text{ if and only if} \\ t \in \text{LFP}_P.F_{\mathbf{Q}p:x}^{\psi}(P, \overline{\mathbf{Y}}) & \text{ for some quasi-atom } \overline{\mathbf{Y}} \text{ of } \kappa^{\sharp}(\mathbf{Q}). \end{aligned}$$

From this we can conclude that $\mathbb{M}, g \models \varphi$ iff $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g \models \varphi$, for some quasi-atom $\overline{\mathbf{Y}}$ of $\kappa^{\sharp}(\mathbf{Q})$. Hence, φ is completely additive in \mathbf{Q} .

This finishes all the cases. \square

This proves that the above fragment is “sound” with respect to the property of complete additivity. We conjecture that the fragment is also “complete” with respect to this property, i.e., that every formula of μFOE which is completely additive in \mathbb{Q} is equivalent to a formula in $\mu\text{FOEADD}_{\mathbb{Q}}$.

3.1.42. CONJECTURE. *Every formula of μFOE which is completely additive in \mathbb{Q} is equivalent to some formula of $\mu\text{FOEADD}_{\mathbb{Q}}$.*

Finally, we define $\mu_a\text{FOE}$:

3.1.43. DEFINITION. The fragment $\mu_a\text{FOE}$ of μFOE is given by the following restriction of the fixpoint operator to the completely additive fragment:

$$\varphi ::= q(x) \mid R_\ell(x, y) \mid x \approx y \mid \exists x.\varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid [\text{LFP}_{p:x}.\xi(p, x)](z)$$

where $p, q \in \mathbb{P}$, $\ell \in \mathbb{D}$, $x, y \in \text{iVar}$; and $\xi(p, x) \in \mu\text{FOEADD}_p \cap \mu_a\text{FOE}$.

We are now ready to prove the main theorem of this section.

3.1.44. THEOREM. *The logics $\text{FO}(\text{TC}^1)$ and $\mu_a\text{FOE}$ are effectively equivalent.*

This theorem follows directly from Proposition 3.1.45 and 3.1.46 below, where we give effective translations that witness the equivalence.

3.1.45. PROPOSITION. *$\text{FO}(\text{TC}^1)$ can be effectively translated to $\mu_a\text{FOE}$.*

Proof. In Remark 2.7.7 we observed that the reflexive-transitive closure of a formula can be expressed as a fixed point. That is,

$$[\text{TC}_{x,y}.\varphi(x, y)](u, v) \equiv [\text{LFP}_{p:y}.y \approx u \vee (\exists x.p(x) \wedge \varphi(x, y))](v).$$

It is easy to see (syntactically) that the formula inside the fixpoint is completely additive in p (which is a fresh variable); therefore it belongs to $\mu_a\text{FOE}$. Moreover, the equivalence holds for all models, in particular, for trees. \square

3.1.46. PROPOSITION. *$\mu_a\text{FOE}$ can be effectively translated to $\text{FO}(\text{TC}^1)$.*

Proof. An inductive translation from $\mu_a\text{FOE}$ to $\text{FO}(\text{TC}^1)$ is straightforward for most of the cases. The only difficult step is to show that we can translate a fixpoint into a transitive closure. To do it, we will crucially use the insight developed in Section 3.1.1.

Let $\varphi(p, z)$ belong to μFOEADD_p . We want to show that $[\text{LFP}_{p:z}.\varphi(p, z)](x)$ can be expressed in $\text{FO}(\text{TC}^1)$. Consider first the map $F_{p:z}^\varphi : \wp(M) \rightarrow \wp(M)$ induced by φ . For notational simplicity we will write F instead of $F_{p:z}^\varphi$. By Corollary 3.1.14 we have that $\text{LFP}(F) = (F_\diamond^{-1})^*[F(\emptyset)]$. It only remains to observe that the required relations can be defined in $\text{FO}(\text{TC}^1)$, as follows:

- $v \in F(\emptyset)$ is equivalent to $\varphi(\perp, v)$,
- $F_\blacklozenge(u, v)$ is equivalent to $\varphi(p, u)[p(y) \mapsto v \approx y]$.

To finish, we define the translation of the fixpoint as

$$([\text{LFP}_{p:z}.\varphi(p, z)](x))^t := \exists e.e \in F(\emptyset) \wedge [\text{TC}_{x,y}.F_\blacklozenge(y, x)](e, x).$$

The correctness of this translation is justified by Corollary 3.1.14. \square

3.2 Continuous fragments

We introduced the notion of continuity for maps $F : \wp(S) \rightarrow \wp(S)$ by asking that F should restrict to finite subsets. That is,

$$F(X) = \bigcup_{Y \subseteq_\omega X} F(Y).$$

This notion is strictly more general than complete additivity. It is easy to see that every completely additive map G is continuous, since G restricts to singletons (and is monotone) then in particular it restricts to finite sets.

3.2.1. FACT. *Every completely additive map is continuous.*

As an example of a continuous map which is not completely additive one can define $F(X)$ to be \emptyset for quasi-atoms and X otherwise. However, we shall see that more natural examples can be induced from formulas.

3.2.2. REMARK. The name continuity comes from the fact that this definition coincides with that of continuous maps with respect to *Scott topologies* [GHK⁺80]. This topological perspective will not play a role, and we refrain from explaining it further. The interested reader is referred to the nice and compact presentation given in [Fon10, Section 5.4.1].

As we did with additivity, we now discuss the notion of continuity in the more general context of maps $F : \wp(S)^n \rightarrow \wp(S)$.

3.2.3. DEFINITION. A map $F : \wp(S)^n \rightarrow \wp(S)$ is *continuous in the j^{th} -coordinate* if for every $X_1, \dots, X_n \subseteq S$ it satisfies:

$$F(X_1, \dots, X_j, \dots, X_n) = \bigcup_{Y \subseteq_\omega X_j} F(X_1, \dots, Y, \dots, X_n).$$

We say that F is *continuous* (sometimes called continuous in the product) if for every $\bar{X} \in \wp(S)^n$ it satisfies:

$$F(\bar{X}) = \bigcup_{\bar{Y} \subseteq_\omega \bar{X}} F(\bar{Y}).$$

Even though continuous maps are a generalization of completely additive maps, the following properties continue to hold for them.

3.2.4. FACT. *Every continuous map is monotone and constructive.*

On the other hand, there are still important differences: first observe that continuity does *not* force $F(\emptyset) = \emptyset$. A more important observation is the following.

3.2.5. PROPOSITION. *A map $F : \wp(S)^n \rightarrow \wp(S)$ is continuous (in the product) iff it is continuous in every coordinate.*

The map $F(A, B) = A \cap B$ can be seen to be continuous but, as observed in Remark 3.1.5, we know that $F(A, B) \neq F(A, \emptyset) \cup F(\emptyset, B)$. More generally F cannot be decomposed as two continuous maps G_1, G_2 depending only on A and B , respectively. This shows that, unfortunately, the property of separation of variables does *not* hold for continuous maps.

3.2.1 Fixpoint theory of continuous maps

Suppose now that we are given a map $G(X, Y)$ which is continuous. A natural question is whether the (least) fixpoint operation preserves continuity. That is, whether $G'(Y) := \text{LFP}_X.G(X, Y)$ is continuous as well. To answer that question, we will have to look at the finite approximants of $F(X) = G(X, Y)$ where Y is now fixed. In this subsection we give a fairly technical and precise characterization of the finite approximants of continuous maps, and use it to prove the following theorem.

3.2.6. THEOREM.

(1) *If $G(X, \overline{Y})$ is continuous then so is $H(\overline{Y}) := \text{LFP}_X.G(X, \overline{Y})$.*

(2) *For every continuous map $F : \wp(S) \rightarrow \wp(S)$ and $s \in S$ we have that*

$$s \in \text{LFP}(F) \quad \text{iff} \quad \text{there exists a finite set } Y \text{ such that } s \in \text{LFP}(F \upharpoonright Y).$$

The following lemma gives a precise characterization of the finite approximants of fixpoints of continuous functions.

3.2.7. LEMMA. *Let $G : \wp(S)^{n+1} \rightarrow \wp(S)$ be a continuous map. For every $s \in S$ and $\overline{Y} \in \wp(S)^n$ we have that $s \in \text{LFP}_X.G(X, \overline{Y})$ iff for some k there exist sets $T_1, \dots, T_k \subseteq_\omega S$ and $\overline{Q} \subseteq_\omega \overline{Y}$ such that $s \in T_k$ and the following conditions hold:*

- $T_1 \subseteq_\omega G(\emptyset, \overline{Q})$; and
- $T_{i+1} \subseteq_\omega G(T_i, \overline{Q})$, for all $1 \leq i < k$.

Proof. \Rightarrow As an abbreviation, define $F(X) := G(X, \overline{\mathbf{Y}})$. Let $s \in \text{LFP}(F)$ and $k' \in \mathbb{N}$ be the smallest k' such that $s \in F^{k'}(\emptyset)$. Such k' exists because of Fact 3.2.4 (constructivity). We define sets $U_i \subseteq_\omega F^i(\emptyset)$ by downwards induction:

- Case $i = k'$: we set $U_i := \{s\}$, which satisfies $U_i \subseteq_\omega F^{k'}(\emptyset)$.
- Case $i < k'$: we want to define U_i in terms of $U_{i+1} \subseteq_\omega F^{i+1}(\emptyset)$. By definition we have that $U_{i+1} \subseteq_\omega G(F^i(\emptyset), \overline{\mathbf{Y}})$. By continuity of G there is $(T, \overline{\mathbf{Q}}_i) \subseteq_\omega (F^i(\emptyset), \overline{\mathbf{Y}})$ such that $U_{i+1} \subseteq_\omega G(T, \overline{\mathbf{Q}}_i)$. We consider the shape of T : if $T \neq \emptyset$ we set $U_i = T$ which satisfies $U_{i+1} \subseteq_\omega G(U_i, \overline{\mathbf{Q}}_i)$; otherwise we set $\overline{\mathbf{Q}} := \bigcup_{i < k'} \overline{\mathbf{Q}}_i$ and finish the process. Observe that the second case will eventually occur. In the worst case this it will occur when $i = 1$, because $F^0(\emptyset) = \emptyset$.

This process defines a series of sets $U_{k'}, U_{k'-1}, \dots, U_j$ where $j \geq 1$. To define the sets T_j we just shift this sequence. That is, we set $k := k' - j + 1$ and $T_i := U_{j+i-1}$ for $1 \leq i \leq k$.

\Leftarrow This direction is proved as in Lemma 3.1.11 by first showing that $T_i \subseteq_\omega F^i(\emptyset)$ for all $1 \leq i \leq k$. \square

We can now prove our main theorem about continuous functionals.

Proof of Theorem 3.2.6(1). Let $G(X, \overline{\mathbf{Y}})$ be a continuous functional and define $H(\overline{\mathbf{Y}}) := \text{LFP}_X.G(X, \overline{\mathbf{Y}})$. Suppose that $s \in H(\overline{\mathbf{Y}})$. Let $T_1, \dots, T_k \subseteq_\omega S$ and $\overline{\mathbf{Q}} \subseteq_\omega \overline{\mathbf{Y}}$ be the sets given by Lemma 3.2.7. We will now prove that $s \in H(\overline{\mathbf{Q}}) = \text{LFP}_X.G(X, \overline{\mathbf{Q}})$. Observe that, by the mentioned lemma, we have $T_1 \subseteq_\omega G(\emptyset, \overline{\mathbf{Q}})$ and $T_{i+1} \subseteq_\omega G(T_i, \overline{\mathbf{Q}})$. From this it can be easily seen that $T_i \subseteq \text{LFP}_X.G(X, \overline{\mathbf{Q}})$ for each i . As the lemma states that $s \in T_k$, we can conclude that $s \in \text{LFP}_X.G(X, \overline{\mathbf{Q}})$. \square

Proof of Theorem 3.2.6(2). Let $F : \wp(S) \rightarrow \wp(S)$ be continuous and $s \in S$; we prove that $s \in \text{LFP}(F)$ iff there exists a finite Y such that $s \in \text{LFP}(F_{\upharpoonright Y})$.

\Rightarrow Let $T_1, \dots, T_k \subseteq_\omega S$ be the sets obtained using Lemma 3.2.7 and define $Y = \bigcup_i T_i$. In the lemma we already proved that $T_i \subseteq_\omega F^i(\emptyset)$ for all i . This relationship can be lifted to $T_i \subseteq_\omega F_{\upharpoonright Y}^i(\emptyset)$ for all i (as we did in the proof of Theorem 3.1.10(2)). In particular, we have that $s \in \{s\} = T_k \subseteq_\omega F_{\upharpoonright Y}^k(\emptyset)$ and therefore $s \in \text{LFP}(F_{\upharpoonright Y})$.

\Leftarrow This direction goes through using a monotonicity argument. Using that for all X we have $F_{\upharpoonright Y}(X) \subseteq F(X)$, it is not difficult to prove that $F_{\upharpoonright Y}^\alpha(X) \subseteq F^\alpha(X)$ for all α , which entails that $\text{LFP}(F_{\upharpoonright Y}) \subseteq \text{LFP}(F)$. \square

3.2.2 Characterization of CPDL inside μML

It is known that CPDL can be translated to the μ -calculus, moreover, it can be translated to a smaller fragment $\mu_c\text{ML}$ of μML which relates to the property of continuity. However, even though the concept of continuity is a generalization of complete additivity, it lacks several important features. In this section we identify a fragment of μML which corresponds to CPDL and discuss how continuity and other properties play a role in this correspondence.

3.2.8. DEFINITION. We say that $\varphi \in \mu\text{ML}$ is *continuous in* $Q \subseteq \mathbf{P}$ if

$$\mathbb{S} \Vdash \varphi \quad \text{iff} \quad \mathbb{S}[Q|\overline{\mathbf{Y}}] \Vdash \varphi \text{ for some } \overline{\mathbf{Y}} \subseteq_{\omega} \kappa^{\sharp}(Q),$$

for every transition system \mathbb{S} .

It is useful to remark that the equivalence of continuity in the product and continuity in every variable stated in Proposition 3.2.5 transfers to the setting of formulas, via the following proposition.

3.2.9. PROPOSITION. *A formula $\varphi \in \mu\text{ML}$ is continuous in $Q \subseteq \mathbf{P}$ iff the associated functional F_Q^{φ} is continuous, for every transition system \mathbb{S} .*

Fontaine [Fon08, Fon10] gave a syntactic fragment of μML and showed that $\varphi \in \mu\text{ML}$ is continuous in p iff it is equivalent to a formula in that fragment.

3.2.10. DEFINITION. Given a set $Q \subseteq \mathbf{P}$, the fragment μMLCON_Q of μML is inductively defined as follows:

$$\varphi := q \mid \alpha \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle \ell \rangle \varphi \mid \mu p. \varphi'$$

where $\ell \in \mathbf{D}$, $q \in Q$, $p \in \mathbf{P}$, $\alpha \in \mu\text{ML}$ is Q -free and $\varphi' \in \mu\text{MLCON}_{Qp}$.

3.2.11. THEOREM ([FON10, THEOREM 5.4.4]). *A formula $\varphi \in \mu\text{ML}$ is continuous in p iff it is equivalent to a formula of μMLCON_p .*

This fragment was also studied by van Benthem [Ben06, Definition 5] under the name of ‘ ω - μ -calculus’ (probably due to the constructivity property). A few syntactic properties of this fragment are stated in the following proposition.

3.2.12. PROPOSITION. *Let $\varphi \in \mu\text{MLCON}_Q$, the following holds:*

- (i) φ is alternation-free,
- (ii) Every variable bound by a least fixpoint is existential (i.e., is only in the scope of diamonds); dually, every variable bound by a greatest fixpoint is universal (i.e., is only in the scope of boxes).

Using that a formula is continuous in Q iff it is continuous in every $q \in Q$ (Proposition 3.2.5) we can prove the following corollary.

3.2.13. COROLLARY. *A formula $\varphi \in \mu\text{ML}$ is continuous in Q iff it is equivalent to a formula of μMLCON_Q .*

Proof. Using Proposition 3.2.5 and that $\mu\text{MLCON}_Q \equiv \bigcap_{q \in Q} \mu\text{MLCON}_q$. \square

Given the analysis performed in Section 3.1.2 it is natural to ask if (normal and) continuous formulas of μML correspond to programs of CPDL. Unfortunately, the same counterexample $\varphi = (\mu q. \Box q) \wedge p$ of Proposition 3.1.23 applies, since CPDL also fails to express well-foundedness.

3.2.14. PROPOSITION. *There is a formula $\varphi \in \mu\text{ML}$ which is normal and continuous in p but cannot be written as a CPDL-formula $\langle \pi \rangle p$.*

Proof. Direct from the observation that CPDL can be translated to the continuous fragment $\mu_c\text{FOE}$ of the first-order μ -calculus μFOE , but $\mu_c\text{FOE}$ cannot express well-foundedness [Par76, Section 4]. \square

We now finally introduce the fragment $\mu_c\text{ML}$ of μML considered in [Pel85, Fon08, Fon10], which restricts the least fixpoint operator $\mu p. \varphi$ to μMLCON_p . This fragment will play an important role in later chapters, independently of CPDL.

3.2.15. DEFINITION. Formulas of $\mu_c\text{ML}$ are given by the following induction:

$$\alpha := p \mid \neg \alpha \mid \alpha \vee \alpha \mid \langle \ell \rangle \alpha \mid \mu p. \varphi,$$

where $p \in \mathbf{P}$, $\ell \in \mathbf{D}$ and $\varphi \in \mu\text{MLCON}_p \cap \mu_c\text{ML}$.

Peleg also considers a fragment of $\mu_c\text{ML}$ which he calls *simple* $\mu_c\text{ML}$, and which is obtained by forbidding the interleaving of the fixpoint operators. That is, formulas of the shape $\mu p. \varphi(\mu q. \varphi'(p))$ are not allowed. We follow the terminology of [SV10] and call this fragment *flat* $\mu_c\text{ML}$ and denote it by $\mu_c\text{ML}^b$.

3.2.16. THEOREM ([PEL85, THEOREM 2.11]). $\mu_c\text{ML}^b \subseteq \text{CPDL} \subseteq \mu_c\text{ML}$.

The strictness of these two inclusions is still an open problem, which is discussed in the last part of this chapter. We conjecture that they are both strict.

3.2.17. CONJECTURE. $\mu_c\text{ML}^b \subsetneq \text{CPDL} \subsetneq \mu_c\text{ML}$.

In the remainder of this section we introduce a (syntactic) restriction $\mu_{nc}\text{ML}^\vee$ of $\mu_c\text{ML}$ and prove that $\text{CPDL} \equiv \mu_{nc}\text{ML}^\vee$. Therefore the relationship between the mentioned languages is as follows:

$$\mu_c\text{ML}^b \subseteq \text{CPDL} \equiv \mu_{nc}\text{ML}^\vee \subseteq \mu_c\text{ML}.$$

3.2.18. DEFINITION. Given a set of propositions $Q \subseteq P$, the formulas of the fragment μMLnCON_Q^\vee of μML are given as follows:

$$\varphi := q \mid \alpha \wedge \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle \ell \rangle \varphi \mid \mu p. \varphi' \vee \varphi''$$

where $\ell \in D$, $q \in Q$, $p \in P$, $\alpha \in \mu\text{ML}$ is Q -free, $\varphi' \in \mu\text{MLnCON}_Q^\vee$ is p -free and $\varphi'' \in \mu\text{MLnCON}_p^\vee$ is Q -free. Formulas of $\mu_{nc}\text{ML}^\vee$ are given as follows:

$$\alpha := p \mid \neg \alpha \mid \alpha \vee \alpha \mid \langle \ell \rangle \alpha \mid \mu p. \beta \vee \varphi,$$

where $p \in P$, $\ell \in D$, $\beta \in \mu_{nc}\text{ML}^\vee$ is p -free and $\varphi \in \mu\text{MLnCON}_p^\vee \cap \mu_{nc}\text{ML}^\vee$.

The fragment μMLnCON_Q^\vee has two main differences with μMLCON_p . The first difference is the removal of the $\alpha \in \mu\text{ML}$ clause and the addition of $\alpha \wedge \varphi$. This change amounts to getting ‘normality’ without losing the power to do tests (that is, having conjunction-with-constant).

However, this change is not crucial, as we will see later in Proposition 3.2.27. The second (and key) difference is the separation of variables under the fixpoint operator, forced by the $\mu p. \varphi' \vee \varphi''$ clause where $\varphi' \in \mu\text{MLnCON}_Q^\vee$ is p -free and $\varphi'' \in \mu\text{MLnCON}_p^\vee$ is Q -free.

3.2.19. PROPOSITION. *Every $\varphi \in \mu\text{MLnCON}_Q^\vee$ is normal and continuous in Q .*

Proof. Continuity is direct because $\mu\text{MLnCON}_p^\vee \subseteq \mu\text{MLCON}_p$. Normality is easily proved by induction. \square

A natural question is whether the logics $\mu_c\text{ML}$ and $\mu_{nc}\text{ML}^\vee$, which use the fragments μMLCON_Q and μMLnCON_p^\vee respectively, are equivalent or not. In the case of complete additivity, it is not too difficult to see that an analogous statement holds (mainly because of the separation of variables, *cf.* Proposition 3.1.26 and Proposition 3.1.37). In the current case, however, we were not able to establish the precise relationship between these fragments. We conjecture that these logics are different, some intuitions will be discussed in Section 3.3 in the context of Game Logic.

3.2.20. CONJECTURE. $\mu_c\text{ML} \not\subseteq \mu_{nc}\text{ML}^\vee$.

Finally, we prove the characterization of CPDL inside μML .

3.2.21. THEOREM. *CPDL is effectively equivalent to $\mu_{nc}\text{ML}^\vee$.*

The theorem follows directly from Propositions 3.2.23 and 3.2.24 below. We start with the direction from CPDL to $\mu_{nc}\text{ML}^\vee$.

3.2.22. DEFINITION. We extend the translations $f_\pi : \mu\text{ML} \rightarrow \mu\text{ML}$ and $(-)^t : \text{PDL} \rightarrow \mu\text{ML}$ of Definition 3.1.28 with the clause

$$f_{\pi \otimes \pi'}(\alpha) := f_\pi(\alpha) \wedge f_{\pi'}(\alpha),$$

and get a translation $(-)^t : \text{CPDL} \rightarrow \mu\text{ML}$.

The following proposition says that the translation $(-)^t$ is the required embedding of CPDL into the fragment $\mu_{nc}\text{ML}^\vee$.

3.2.23. PROPOSITION.

(1) for every program $\pi \in \text{CPDL}$, and every formula $\alpha \in \mu_{nc}\text{ML}^\vee$:

- (1a) $f_\pi(\alpha)$ belongs to $\mu_{nc}\text{ML}^\vee$,
- (1b) $f_\pi(\alpha) \in \mu\text{MLnCON}_Q^\vee$ and $\text{FV}(f_\pi(\alpha)) = \text{FV}(\alpha) \cup \text{FV}(\pi)$,
given $\alpha \in \mu\text{MLnCON}_Q^\vee$ and $\text{FV}(\pi) \cap Q = \emptyset$, and
- (1c) $\langle \pi \rangle \alpha \equiv f_\pi(\alpha)$.

(2) for every formula $\alpha \in \text{CPDL}$:

- (2a) $\alpha^t \in \mu_{nc}\text{ML}^\vee$, and
- (2b) $\alpha \equiv \alpha^t$.

Proof. Most of the proof is exactly as Proposition 3.1.29. We only prove item (1) for the case of the new program and the star.

- Suppose $\pi = \pi' \otimes \pi''$ and consider a formula $\alpha \in \mu_{nc}\text{ML}^\vee$. By inductive hypothesis we have $f_{\pi'}(\alpha) \in \mu_{nc}\text{ML}^\vee$, $f_{\pi'}(\alpha) \equiv \langle \pi' \rangle \alpha$ and analogous statements for π'' . To prove (1a) just observe that $\mu_{nc}\text{ML}^\vee$ is closed under conjunction, therefore $f_\pi(\alpha) = f_{\pi'}(\alpha) \wedge f_{\pi''}(\alpha)$ belongs to $\mu_{nc}\text{ML}^\vee$. Item (1b) is proved similarly, using (1b) of the inductive hypothesis. For (1c) we simply verify that $\langle \pi' \rangle \alpha \wedge \langle \pi'' \rangle \alpha \equiv \langle \pi' \otimes \pi'' \rangle \alpha$ by the semantics of CPDL.
- Suppose $\pi = \varrho^*$ and consider a formula $\alpha \in \mu_{nc}\text{ML}^\vee$. Recall that $f_\pi(\alpha)$ is defined as $\mu p. \alpha \vee f_\varrho(p)$, where p does not occur in neither α nor ϱ . To prove (1a), we apply the inductive hypothesis (1a) and (1b) to ϱ and the formula $p \in \mu\text{MLnCON}_p^\vee$, and get that $f_\varrho(p) \in \mu\text{MLnCON}_p^\vee \cap \mu_{nc}\text{ML}^\vee$. Therefore $\mu p. \alpha \vee f_\varrho(p)$ indeed belongs to $\mu_{nc}\text{ML}^\vee$. For (1b), let Q be a set of variables that do not occur in ϱ , assume that $\alpha \in \mu\text{MLnCON}_Q^\vee$. We already proved for (1a) that $f_\varrho(p) \in \mu\text{MLnCON}_p^\vee$. Observing that $\text{FV}(f_\varrho(p)) = \text{FV}(\varrho) \cup \{p\}$, we can conclude that $\mu p. \alpha \vee f_\varrho(p)$ belongs to the set μMLnCON_Q^\vee and $\text{FV}(f_\pi(\alpha)) = \text{FV}(\alpha) \cup \text{FV}(\pi)$. For (1c), it is obvious that $\langle \varrho^* \rangle \alpha \equiv \mu p. \alpha \vee \langle \varrho \rangle p \equiv \mu p. \alpha \vee f_\varrho(p) \equiv f_\pi(\alpha)$.

This finishes the proof. □

The translation in the other direction is provided by the following proposition.

3.2.24. PROPOSITION. *The following procedures can be performed effectively:*

- (i) *Given a formula $\alpha \in \mu_{nc}\text{ML}^\vee$, return an equivalent formula $\alpha^s \in \text{CPDL}$.*
- (ii) *Given a formula $\varphi \in \mu\text{MLnCON}_p^\vee \cap \mu_{nc}\text{ML}^\vee$, return a p -free program $\pi \in \text{CPDL}$ such that $\varphi \equiv \langle \pi \rangle p$.*

Proof. Most of the proof is done by mutual induction like Proposition 3.1.30, we focus on proving point (ii) for the case of the conjunction and fixpoint.

Suppose $\varphi = \varphi' \wedge \varphi''$ belongs to $\mu_{nc}\text{ML}^\vee$, then by definition of the fragment both $\varphi', \varphi'' \in \mu_{nc}\text{ML}^\vee$ as well. Applying the inductive hypothesis to both formulas we get that $\varphi \equiv \langle \pi' \rangle p \wedge \langle \pi'' \rangle p$ for p -free programs $\pi', \pi'' \in \text{CPDL}$. It is only left to observe that $\langle \pi' \rangle p \wedge \langle \pi'' \rangle p \equiv \langle \pi' \otimes \pi'' \rangle p$.

Suppose $\varphi = \mu q. \varphi' \vee \varphi''$ where $\varphi' \in \mu\text{MLnCON}_p^\vee$ is q -free and $\varphi'' \in \mu\text{MLnCON}_q^\vee$ is p -free. Using item (ii) of the inductive hypothesis on both formulas we get that $\varphi \equiv \mu p. (\langle \pi' \rangle p \vee \langle \pi'' \rangle q)$ where $\pi', \pi'' \in \text{CPDL}$ are pq -free. From this, it is easy to see that $\varphi \equiv \langle (\pi'')^* ; \pi' \rangle p$. \square

As a corollary of these propositions, we get a characterization for CPDL-programs.

3.2.25. COROLLARY. *A formula $\varphi \in \mu\text{ML}$ belongs to $\mu\text{MLnCON}_p^\vee \cap \mu_{nc}\text{ML}^\vee$ iff it is equivalent to $\langle \pi \rangle p$ for some p -free program $\pi \in \text{CPDL}$.*

Proof. The left-to-right direction is given by Proposition 3.2.24(ii) and the other direction is given by Proposition 3.2.23(1). \square

Relation to non-normal continuity. As for additivity, the results of this section were mostly developed for *normal* continuity to get a clean characterization for CPDL programs (*cf.* Corollary 3.2.25). The same analysis could have been done by defining fragments based on non-normal continuity.

3.2.26. DEFINITION. Given a set $Q \subseteq P$, the fragment μMLCON_Q^\vee of μML is inductively defined as follows:

$$\varphi := q \mid \alpha \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle \ell \rangle \varphi \mid \mu p. \varphi' \vee \varphi''$$

where $\ell \in D$, $q \in Q$, $p \in P$, $\alpha \in \mu\text{ML}$ is Q -free, $\varphi' \in \mu\text{MLCON}_Q^\vee$ is p -free and $\varphi'' \in \mu\text{MLCON}_p^\vee$ is Q -free. Formulas of $\mu_c\text{ML}^\vee$ are given as follows:

$$\alpha := p \mid \neg \alpha \mid \alpha \vee \alpha \mid \langle \ell \rangle \alpha \mid \mu p. \varphi,$$

where $p \in P$, $\ell \in D$ and $\varphi \in \mu\text{MLCON}_p^\vee \cap \mu_c\text{ML}^\vee$.

It should be observed that $\mu\text{MLCON}_Q^\vee \not\equiv \mu\text{MLnCON}_Q^\vee$. However, the fragments $\mu_c\text{ML}^\vee$ and $\mu_{nc}\text{ML}^\vee$ are in fact equivalent. The following proposition is proved similar to Proposition 3.1.37.

3.2.27. PROPOSITION. $\mu_c\text{ML}^\vee \equiv \mu_{nc}\text{ML}^\vee$ and hence $\mu_c\text{ML}^\vee \equiv \text{CPDL}$.

3.2.3 Finiteness, $\mu_c\text{FOE}^\infty$ and WMSO

In this section we study the relationship of continuity in first-order languages and (in-)finiteness in second order languages. Park [Par76, Section 4] defined a continuous fragment of μFOE and proved that it is included in the infinitary language $\mathcal{L}_{\omega_1\omega}$; that is, first-order logic with countably infinite conjunctions and disjunctions in addition to the usual finite operations.

For this section, however, we are more interested in considering WMSO as a target language. To begin with, we consider the case of monadic signatures, that is, signatures with relations of arity at most one (i.e, predicates). It follows from results by Väänänen that, for monadic signatures, WMSO coincides with an extension FOE^∞ of first-order logic, with an additional generalized quantifier $\exists^\infty x.\varphi$ stating that there are *infinitely many elements* satisfying φ .

3.2.28. THEOREM ([VÄÄ77, SECTION 6]). *WMSO on a monadic signature is equivalent to monadic FOE^∞ .*

This theorem can also be given a more direct proof using the normal forms that we will develop in Chapter 5. We could ask ourselves what happens with this relationship if we consider (arbitrary) relational signatures, and even consider the presence of a fixpoint operator. The following proposition shows that this relationship cannot be lifted for full μFOE over a relational signature.

3.2.29. PROPOSITION. $\mu\text{FOE} \not\subseteq \text{WMSO}$ and hence $\mu\text{FOE}^\infty \not\subseteq \text{WMSO}$.

Proof. In μFOE we can define the class of well-founded trees but this cannot be done in WMSO. This follows from the fact that WMSO can only define properties of trees that, from a topological point of view, are Borel, which is not the case of the class of well-founded trees. See e.g. [CF11]. \square

However, if we consider the logic $\mu_c\text{FOE}^\infty$ obtained by restricting the fixpoint operator of μFOE^∞ to *continuous* formulas, then we are in a better shape. The main result of this section is that, even for relational signatures, the following inclusion holds:

$$\mu_c\text{FOE}^\infty \subseteq \text{WMSO}.$$

Unfortunately, other inclusion does *not* hold in general.

3.2.30. PROPOSITION. $\text{WMSO} \not\subseteq \mu_c\text{FOE}^\infty$.

Proof. The first observation is that, *on finite models* we have that $\text{MSO} \not\subseteq \mu\text{FOE}$. This result is discussed in [Sch06, p. 2] as a corollary of [Daw98, Theorem 4.11]. The next observation is that, again on finite models, we have $\text{WMSO} \equiv \text{MSO}$ and also $\mu_c\text{FOE}^\infty \subseteq \mu\text{FOE}$, since the generalized quantifier \exists^∞ trivializes. Hence on finite models we get $\text{WMSO} \not\subseteq \mu_c\text{FOE}^\infty$. The final step is to realize that if $\text{WMSO} \not\subseteq \mu_c\text{FOE}^\infty$ for finite models then then inclusion cannot hold for arbitrary models. \square

This proposition shows that WMSO and $\mu_c\text{FOE}^\infty$ cannot be equivalent for arbitrary models. Nevertheless, in Chapter 7, we use automata-theoretic tools to prove that the equivalence does hold for tree models. For the moment, we begin with the necessary definitions to develop the main theorem of this section.

3.2.31. DEFINITION. We say that $\varphi \in \mu\text{FOE}^\infty$ is *continuous in* $\mathbf{Q} \subseteq \mathbf{P}$ if for every model \mathbb{M} and assignment g it satisfies

$$\mathbb{M}, g \models \varphi \quad \text{iff} \quad \mathbb{M}[\mathbf{Q} \mapsto \overline{\mathbf{Y}}], g \models \varphi \text{ for some finite set } \overline{\mathbf{Y}} \subseteq_\omega \kappa^{\mathbf{h}}(\mathbf{Q}).$$

3.2.32. PROPOSITION. *If $\varphi \in \mu\text{FOE}^\infty$ is continuous in \mathbf{Q} then the functional $F_{\mathbf{Q}:x}^\varphi : \varphi(M)^n \rightarrow \varphi(M)$ is continuous, for every model \mathbb{M} and variable $x \in \text{FV}(\varphi)$.*

Proof. Fix a model \mathbb{M} , assignment g and free variable $x \in \text{FV}(\varphi)$. We want to prove that $F_{\mathbf{Q}:x}^\varphi(\overline{\mathbf{Z}})$ is continuous. An element t belongs to $F_{\mathbf{Q}:x}^\varphi(\overline{\mathbf{Z}})$ iff $\mathbb{M}[\mathbf{Q} \mapsto \overline{\mathbf{Z}}], g[x \mapsto t] \models \varphi$. By continuity of φ , this occurs iff $\mathbb{M}[\mathbf{Q} \mapsto \overline{\mathbf{Y}}], g[x \mapsto t] \models \varphi$ for some $\overline{\mathbf{Y}} \subseteq_\omega \overline{\mathbf{Z}}$. By definition of $F_{\mathbf{Q}:x}^\varphi$, this is equivalent to saying that $t \in F_{\mathbf{Q}:x}^\varphi(\overline{\mathbf{Y}})$. Therefore, $F_{\mathbf{Q}:x}^\varphi$ is continuous. \square

Next, we provide a definition of a fragment of μFOE^∞ , and shortly after that we prove that every formula in this fragment is continuous.

3.2.33. DEFINITION. Let $\mathbf{Q} \subseteq \mathbf{P}$ be a set of monadic predicates. The fragment $\mu\text{FOE}^\infty\text{CON}_{\mathbf{Q}}(\mathbf{P}, \mathbf{D})$ is defined by the following rules:

$$\varphi ::= \psi \mid q(x) \mid \exists x.\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \mathbf{W}x.(\varphi, \psi) \mid [\text{LFP}_{p:x}.\xi(p, x)](z)$$

where $q \in \mathbf{Q}$, $\psi \in \mu\text{FOE}^\infty(\mathbf{P} \setminus \mathbf{Q}, \mathbf{D})$, $p \in \mathbf{P} \setminus \mathbf{Q}$ and $\xi(p, x) \in \mu\text{FOE}^\infty\text{CON}_{\mathbf{Q}_p}(\mathbf{P}, \mathbf{D})$. The shorthand quantifier $\mathbf{W}x.(\varphi, \psi)$ is defined as $\forall x.(\varphi \vee \psi) \wedge \forall^\infty x.\psi$.

Observe that the atomic formulas given by equality and relations are taken into account by this definition in the ψ clause.

Universal quantification is usually problematic for preserving continuity because of its potentially infinite nature. However, in the case of $\mathbf{W}x.(\varphi, \psi)$, the combination of both quantifiers ensures that all the elements are covered by $\varphi \vee \psi$ but only *finitely many* are required to make φ true (which contains $q \in \mathbf{Q}$). This gives no trouble for continuity in \mathbf{Q} .

3.2.34. PROPOSITION. *Every $\varphi \in \mu\text{FOE}^\infty\text{CON}_{\mathbf{Q}}$ is continuous in \mathbf{Q} .*

Proof. The proof goes by induction. Most cases are proved exactly as in Proposition 3.1.41, we focus on the inductive step of the fixpoint operator, conjunction and the new quantifier.

- Case $\varphi = \varphi_1 \wedge \varphi_2$: assume $\mathbb{M}, g \models \varphi$. By induction hypothesis we have sets $\overline{\mathbf{Y}}_1, \overline{\mathbf{Y}}_2 \subseteq_{\omega} \kappa^{\sharp}(\mathbf{Q})$ such that $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}_i], g \models \varphi_i$. By monotonicity we also have that $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}_1 \cup \overline{\mathbf{Y}}_2], g \models \varphi_i$ and hence $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}_1 \cup \overline{\mathbf{Y}}_2], g \models \varphi$. This finishes the case because $\overline{\mathbf{Y}}_1 \cup \overline{\mathbf{Y}}_2$ is finite.
- Let φ be $[\text{LFP}_{p:x}.\psi(p, x)](z)$, we have to prove that

$$\mathbb{M}, g \models \varphi \quad \text{iff} \quad \mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g \models \varphi \text{ for some } \overline{\mathbf{Y}} \subseteq_{\omega} \kappa^{\sharp}(\mathbf{Q}).$$

By semantics of the fixpoint operator $\mathbb{M}, g \models \varphi$ iff $g(z) \in \text{LFP}(F_{p:x}^{\psi})$. It will be useful to take a slightly more general perspective and consider the map

$$F_{\mathbf{Q}p:x}^{\psi}(P, \overline{\mathbf{Z}}) := \{t \in M \mid \mathbb{M}[p \mapsto P; \mathbf{Q} \mapsto \overline{\mathbf{Z}}], g[x \mapsto t] \models \psi\}$$

and observe that $F_{p:x}^{\psi}(P) = F_{\mathbf{Q}p:x}^{\psi}(P, \kappa^{\sharp}(\mathbf{Q}))$ and therefore their least fixpoints will be the same. By inductive hypothesis and Proposition 3.2.32, we know that $F_{\mathbf{Q}p:x}^{\psi}(P, \overline{\mathbf{Z}})$ is continuous. Using Theorem 3.2.6(1) we get that $\text{LFP}_P.F_{\mathbf{Q}p:x}^{\psi}(P, \overline{\mathbf{Z}})$ is continuous as well. In particular,

$$\begin{aligned} t \in \text{LFP}_P.F_{\mathbf{Q}p:x}^{\psi}(P, \kappa^{\sharp}(\mathbf{Q})) & \text{ if and only if} \\ t \in \text{LFP}_P.F_{\mathbf{Q}p:x}^{\psi}(P, \overline{\mathbf{Y}}) & \text{ for some } \overline{\mathbf{Y}} \subseteq_{\omega} \kappa^{\sharp}(\mathbf{Q}). \end{aligned}$$

From this we conclude that $\mathbb{M}, g \models \varphi$ iff $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g \models \varphi$, for some $\overline{\mathbf{Y}} \subseteq_{\omega} \kappa^{\sharp}(\mathbf{Q})$. Hence, φ is continuous in \mathbf{Q} .

- Case $\mathbf{W}x.\varphi'$: by definition of the quantifier this formula is equivalent to

$$\forall x. \underbrace{(\varphi(x) \vee \psi(x))}_{\alpha(x)} \wedge \underbrace{\forall^{\infty} x.\psi(x)}_{\beta}.$$

Let $\mathbb{M}, g \models \mathbf{W}x.\varphi'$. By induction hypothesis, for every $g_u := g[x \mapsto u]$ which satisfies $\mathbb{M}, g_u \models \alpha(x)$ there is $\overline{\mathbf{Y}}_u \subseteq_{\omega} \kappa^{\sharp}(\mathbf{Q})$ such that $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}_u], g_u \models \alpha(x)$. The crucial observation is that because of β , only finitely many elements make $\psi(x)$ false. Let $\overline{\mathbf{Y}} := \bigcup \{\overline{\mathbf{Y}}_u \mid \mathbb{M}, g_u \not\models \psi(x)\}$. Note that $\overline{\mathbf{Y}}$ is a finite union of finite sets, hence finite.

CLAIM 1. $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g \models \varphi'$.

PROOF OF CLAIM. It is clear that $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g \models \beta$ because ψ is \mathbf{Q} -free. To show that $\forall x.\alpha(x)$ is true we have to show that $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g_u \models \varphi(x) \vee \psi(x)$ for every $u \in M$. We consider two cases: (i) if $\mathbb{M}, g_u \models \psi(x)$ we are done, again because ψ is \mathbf{Q} -free; (ii) if the former is not the case then $\overline{\mathbf{Y}}_u \subseteq \overline{\mathbf{Y}}$; moreover, we knew that $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}_u], g_u \models \alpha(x)$ and by monotonicity of $\alpha(x)$ we can conclude that $\mathbb{M}[\mathbf{Q} \upharpoonright \overline{\mathbf{Y}}], g_d \models \alpha(x)$. \blacktriangleleft

This concludes all the new cases. \square

This proves that the above fragment is “sound” with respect to the property of continuity. We conjecture that the fragment is also “complete” with respect to this property, i.e., that every formula of μFOE^∞ which is continuous in \mathbf{Q} is equivalent to a formula in $\mu\text{FOE}^\infty\text{CON}_\mathbf{Q}$.

3.2.35. CONJECTURE. *Every formula $\varphi \in \mu\text{FOE}^\infty$ which is continuous in \mathbf{Q} is equivalent to some formula $\varphi' \in \mu\text{FOE}^\infty\text{CON}_\mathbf{Q}$.*

Finally, we define $\mu_c\text{FOE}^\infty$:

3.2.36. DEFINITION. The fragment $\mu_c\text{FOE}^\infty$ of μFOE^∞ is given by the following restriction of the fixpoint operator to the continuous fragment:

$$\varphi ::= q(x) \mid R_\ell(x, y) \mid x \approx y \mid \exists x.\varphi \mid \exists^\infty x.\varphi \mid \neg\varphi \mid \varphi \vee \varphi \mid [\text{LFP}_{p;x}.\xi(p, x)](z)$$

where $p, q \in \mathbf{P}$, $\ell \in \mathbf{D}$, $x, y \in \text{iVar}$; and $\xi(p, x) \in \mu\text{FOE}^\infty\text{CON}_p \cap \mu_c\text{FOE}^\infty$.

We are now ready to prove the main theorem of this section. In this case we will make use of the correspondence between single-sorted WMSO and the two-sorted version 2WMSO given in Section 2.8.

3.2.37. PROPOSITION. *There is an effective translation $(-)^t$ from $\mu_c\text{FOE}^\infty$ to 2WMSO such that for every model \mathbb{M} , assignment g and $\varphi \in \mu_c\text{FOE}^\infty$ we have: $\mathbb{M}, g \models \varphi$ if and only if $\mathbb{M}, g \models \varphi^t$.*

Proof. Clearly the interesting cases are the generalized quantifier and fixpoint operator. The former is easy to translate, as follows:

$$(\exists^\infty x.\varphi)^t := \forall_{\text{fin}} Y.\exists x.(\neg Y(x) \wedge \varphi^t(x))$$

Turning to the fixpoint case, in this proof we will use F^ψ to denote $F_{p;y}^\psi$. We define the translation of the fixpoint as follows:

$$\begin{aligned} ([\text{LFP}_{p;y}.\psi(p, y)](x))^t &:= \exists_{\text{fin}} Y.(\forall_{\text{fin}} W \subseteq Y.W \in \text{PRE}(F_Y^\psi) \rightarrow x \in W) \\ W \in \text{PRE}(F_Y^\psi) &:= \forall v.\psi^t(W, v) \wedge v \in Y \rightarrow v \in W. \end{aligned}$$

CLAIM 1. The translation of the fixpoint is correct.

First recall that the translation of $[\text{LFP}_{p;y}.\psi(p, y)](x)$ into MSO is given by:

$$\forall W.(W \in \text{PRE}(F^\psi) \rightarrow x \in W). \quad (\text{t-MSO})$$

where $W \in \text{PRE}(F^\psi)$ expresses that W is a prefixpoint of $F^\psi : \wp(S) \rightarrow \wp(S)$. This translation is based on the following fact about fixpoints of monotone maps:

$$s \in \text{LFP}(F^\psi) \quad \text{iff} \quad s \in \bigcap \{W \subseteq S \mid W \in \text{PRE}(F^\psi)\}. \quad (\text{PRE})$$

It is easy to see that (t-MSO) exactly expresses that $g(x)$ has to belong to every prefixpoint of F^ψ . In our translation $(-)^t$, however, we cannot make use of the set quantifier $\exists W$, since we are dealing with WMSO. The crucial observation is that, as $F^\psi : \wp(S) \rightarrow \wp(S)$ is continuous, then we can use Theorem 3.2.6 to prove that, without loss of generality, we can restrict ourselves to finite sets, in the following sense:

$$\begin{aligned} s \in \text{LFP}(F^\psi) \quad \text{iff} \quad s \in \text{LFP}(F^\psi_{|Y}) \text{ for } Y \subseteq_\omega S & \quad (\text{Theorem 3.2.6}) \\ \text{iff} \quad s \in \bigcap \{W \subseteq Y \mid W \in \text{PRE}(F^\psi_{|Y})\} \text{ for } Y \subseteq_\omega S & \quad (\text{PRE}) \end{aligned}$$

A crucial observation about these equations is that we have $W \subseteq Y$ instead of $W \subseteq S$ because $F^\psi_{|Y} : \wp(Y) \rightarrow \wp(Y)$. Therefore, the translation $(-)^t$ basically expresses the same as (t-MSO) but relativized to a finite set Y . The correctness of the translation is then justified by the above equations. \square

3.3 The question of Game Logic

The syntax of Game Logic is very different from that of the modal μ -calculus. The latter contains explicit fixpoint operators, while the former only has a seemingly weaker iteration operator. Superficially, these logics look quite different, however, the relationship between them remains an intriguing topic today.

Initially, there was considerable evidence to think that these logics could be equivalent. For example, GL cannot be embedded in any fixed level of the alternation hierarchy of μML . To see this, consider the following fact of μML : given a number n of parities, the μ -calculus can express the existence of a winning strategy for any given parity game, with a formula W^n [EJ91]. The number of parities in such a game is strongly related to the *alternation* hierarchy of μML .

3.3.1. THEOREM ([BRA96, BRA98]). *Every $W^n \in \mu\text{ML}$ can be expressed with alternation n and not with alternation $n - 1$.*

Berwanger showed that GL can also express these formulas, and therefore it traverses the whole alternation hierarchy of μML . This result contrasts with the cases of PDL and CPDL, which belong to the alternation-free fragment of μML .

3.3.2. THEOREM ([BER03, THEOREM 7]). *No finite level of the alternation hierarchy of μML captures the expressive power of GL.*

It was known that GL can be translated to the two variable fragment $\mu\text{ML}[2]$ of the μ -calculus (cf. [BGL05, Lemma 47]). Even then, the question of whether GL is equivalent to μML was open, as it was unknown whether $\mu\text{ML}[2]$ is equivalent to the full μML . This question was finally closed by Berwanger by showing that the *variable* hierarchy of the μ -calculus is strict.

3.3.3. THEOREM ([BGL05, BER05]). $GL \not\equiv \mu ML$.

In spite of the remarkable results that we just named, the GL question is still not fully solved. The *exact* fragment of μML (or $\mu ML[2]$) which corresponds to GL is still unknown. In particular, it is not known whether GL and $\mu ML[2]$ coincide. In this section we address this question, but unfortunately we do not provide a complete answer.

Following the methodology of the previous sections we define a fragment μML^\vee of μML and show that it corresponds to GL. As a corollary, we actually get that $GL \equiv \mu ML^\vee[2] \equiv \mu ML^\vee$, but the question of whether $\mu ML^\vee[2]$ and $\mu ML[2]$ coincide is left open. We also discuss some intuitions and conjectures that may lead to a separation of these fragments.

In the previous sections we played with several restrictions on the fixpoint operator, most notably, complete additivity and continuity, together with a separation of variables. In this section we only keep the separation of variables and monotonicity. As we saw, the former seems to be connected to PDL-like syntaxes. The latter, on the other hand, is no restriction at all, since already in μML formulas under fixpoints are required to be monotone.

3.3.4. DEFINITION. Given a set $Q \subseteq P$, the fragment $\mu MLMON_Q^\vee$ of μML is inductively defined as follows:

$$\varphi := q \mid p' \mid \neg p' \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \langle \ell \rangle \varphi \mid [\ell] \varphi \mid \mu p. \varphi' \vee \varphi'' \mid \nu p. \varphi' \wedge \varphi''$$

where $\ell \in D$, $q \in Q$, $p \in P$, $p' \in P \setminus Q$, $\varphi' \in \mu MLMON_Q^\vee$ is p -free and $\varphi'' \in \mu MLMON_p^\vee$ is Q -free. Formulas of μML^\vee are given as follows:

$$\alpha := p \mid \neg \alpha \mid \alpha \vee \alpha \mid \langle \ell \rangle \alpha \mid \mu p. \beta \vee \varphi,$$

where $p \in P$, $\ell \in D$, $\beta \in \mu ML^\vee$ is p -free and $\varphi \in \mu MLMON_p^\vee \cap \mu ML^\vee$.

3.3.5. REMARK. The fragment $\mu MLMON_Q^\vee$ is just the monotone (positive) fragment of μML (see [DH00] and [Fon10, Section 5.1.4]) with the additional constraint of separation of variables. We introduced this fragment in negation normal form, hoping that it would more clearly show its structure. However, it can be equivalently defined as follows:

$$\varphi := p \in P \mid \neg \varphi \mid \varphi \vee \varphi \mid \langle \ell \rangle \varphi \mid \mu q. \varphi' \vee \varphi''$$

asking that every $q \in Q$ is positive (i.e., under an even number of negations).

3.3.6. PROPOSITION. *Every $\mu MLMON_Q^\vee$ is monotone in Q .*

Proof. Corollary of the characterization of monotonicity given in [DH00]. \square

A crucial property of the fragments that we have just defined is that they are closed under Boolean duals. This feature acts as a counterpart to the closure of GL games under duals. We now define the concept of Boolean dual for μML .

3.3.7. DEFINITION. For every formula $\varphi \in \mu\text{ML}$ we define the Boolean dual $\varphi^\delta \in \mu\text{ML}$ of φ by induction:

$$\begin{aligned} p^\delta &:= p & (\neg\varphi)^\delta &:= \neg\varphi^\delta \\ (\varphi \wedge \psi)^\delta &:= \varphi^\delta \vee \psi^\delta & (\varphi \vee \psi)^\delta &:= \varphi^\delta \wedge \psi^\delta \\ (\langle \ell \rangle \varphi)^\delta &:= [\ell] \varphi^\delta & ([\ell] \varphi)^\delta &:= \langle \ell \rangle \varphi^\delta \\ (\mu p. \varphi)^\delta &:= \nu p. \varphi^\delta & (\nu p. \varphi)^\delta &:= \mu p. \varphi^\delta \end{aligned}$$

3.3.8. PROPOSITION. For every transition system \mathbb{S} we have that

$$\mathbb{S} \Vdash \varphi \quad \text{iff} \quad \mathbb{S}[p \mapsto M \setminus \kappa^{\text{h}}(p) \mid p \in \mathbf{P}] \not\Vdash \varphi.$$

Moreover, the transformation $(-)^\delta$ preserves the positivity of the variables.

The following proposition is easily verified using the definition of the fragments.

3.3.9. PROPOSITION.

- (i) If $\varphi \in \mu\text{ML}^\vee[k]$ then $\varphi^\delta \in \mu\text{ML}^\vee[k]$,
- (ii) If $\varphi \in \mu\text{MLMON}_Q^\vee[k]$ then $\varphi^\delta \in \mu\text{MLMON}_Q^\vee[k]$.

Finally, we prove that GL and μML^\vee are equivalent.

3.3.10. THEOREM. The logics GL, μML^\vee and $\mu\text{ML}^\vee[2]$ are effectively equivalent.

The theorem follows directly from Propositions 3.3.13 and 3.3.15 below. We first prove that GL can be translated to $\mu\text{ML}^\vee[2]$.

3.3.11. DEFINITION. By a simultaneous induction on formulas and games of GL, we define, for each game $\pi \in \text{GL}$, functions $f_\pi^x, f_\pi^y : \mu\text{ML} \rightarrow \mu\text{ML}$ on the set of modal fixpoint formulas, and a map $(-)^t : \text{GL} \rightarrow \mu\text{ML}$:

$$\begin{aligned} f_\ell^x(\alpha) &:= \langle \ell \rangle \alpha & f_\ell^y(\alpha) &:= \langle \ell \rangle \alpha \\ f_{\varphi?}^x(\alpha) &:= \varphi^t \wedge \alpha & f_{\varphi?}^y(\alpha) &:= \varphi \wedge \alpha \\ f_{\pi \oplus \pi'}^x(\alpha) &:= f_\pi^x(\alpha) \vee f_{\pi'}^x(\alpha) & f_{\pi \oplus \pi'}^y(\alpha) &:= f_\pi^y(\alpha) \vee f_{\pi'}^y(\alpha) \\ f_{\pi; \pi'}^x(\alpha) &:= f_\pi^x(f_{\pi'}^x(\alpha)) & f_{\pi; \pi'}^y(\alpha) &:= f_\pi^y(f_{\pi'}^y(\alpha)) \\ f_{\pi^*}^x(\alpha) &:= \mu y. \alpha \vee f_\pi^y(y) & f_{\pi^*}^y(\alpha) &:= \mu x. \alpha \vee f_\pi^x(x) \\ f_{\pi^\delta}^x(\alpha) &:= \neg f_\pi^x(\neg \alpha) & f_{\pi^\delta}^y(\alpha) &:= \neg f_\pi^y(\neg \alpha) \\ p^t &:= p & (\varphi_0 \vee \varphi_1)^t &:= \varphi_0^t \vee \varphi_1^t \\ (\neg\varphi)^t &:= \neg\varphi^t & (\langle \pi \rangle \varphi)^t &:= f_\pi^x(\varphi^t) \end{aligned}$$

3.3.12. REMARK. Since we do not want the variables x, y to conflict with other variables in the translation, we take them to be propositional variables which are available in $\mu\text{ML}[2]$ but not in GL. An alternative way to prevent the conflict is to think that both GL and $\mu\text{ML}[2]$ share the same variables but the translation $(\varphi)^t$ of a formula $\varphi \in \text{GL}$ uses the functions f^x, f^y for some distinct $x, y \notin \text{FV}(\varphi)$.

The following proposition says that the translation $(-)^t$ is the required embedding of GL into the fragment $\mu\text{ML}^\vee[2]$.

3.3.13. PROPOSITION.

(1) for every game $\pi \in \text{GL}$, and y -free $\alpha_x \in \mu\text{ML}^\vee[2]$, x -free $\alpha_y \in \mu\text{ML}^\vee[2]$:

- (1a) $f_\pi^x(\alpha_x)$ is y -free, $f_\pi^y(\alpha_y)$ is x -free and both belong to $\mu\text{ML}^\vee[2]$,
- (1b) $f_\pi^z(\alpha_z) \in \mu\text{MLMON}_Q^\vee$ and $\text{FV}(f_\pi^z(\alpha_z)) = \text{FV}(\alpha_z) \cup \text{FV}(\pi)$,
given $\alpha_z \in \mu\text{MLMON}_Q^\vee$ and $\text{FV}(\pi) \cap Q = \emptyset$ for $z \in \{x, y\}$; and
- (1c) $\langle \pi \rangle \alpha \equiv f_\pi(\alpha)$.

(2) for every formula $\alpha \in \text{GL}$:

- (2a) $\alpha^t \in \mu\text{ML}^\vee[2]$,
- (2b) $\alpha \equiv \alpha^t$.

Proof. Most of the proof is exactly as that of Proposition 3.1.29. We only prove item (1) for the case of the star and dual and we prove the statements only for f_π^x , as the case of f_π^y is completely symmetric.

Suppose $\pi = \varrho^*$ and consider a y -free formula $\alpha_x \in \mu\text{ML}^\vee[2]$. Recall that $f_\pi^x(\alpha_x)$ is defined as $\mu y. \alpha_x \vee f_\varrho^y(y)$. For (1a) first observe that as y gets bound by the least fixpoint then clearly $f_\pi^x(\alpha_x)$ is y -free. By Remark 3.3.12 $y \notin \text{FV}(\pi)$, moreover $y \in \mu\text{MLMON}_y^\vee$. Hence, by inductive hypothesis $f_\varrho^y(y) \in \mu\text{MLMON}_y^\vee$ and therefore $f_\pi^x(\alpha_x) \in \mu\text{ML}^\vee[2]$. For (1b) assume that $\alpha_x \in \mu\text{MLMON}_Q^\vee$ and $\text{FV}(\pi) \cap Q = \emptyset$. The first observation is that $\alpha_x \in \mu\text{MLMON}_Q^\vee$ is y -free by hypothesis. We already proved for (1a) that $f_\varrho^y(y) \in \mu\text{MLMON}_y^\vee$. It is only left to observe that $\text{FV}(f_\varrho^y(y)) = \text{FV}(\varrho) \cup \{y\}$ to conclude that $\mu y. \alpha_x \vee f_\varrho^y(y)$ belongs to the set μMLMON_Q^\vee and $\text{FV}(f_\pi^x(\alpha_x)) = \text{FV}(\alpha_x) \cup \text{FV}(\pi)$. For (1c), it is obvious that $\langle \varrho^* \rangle \alpha \equiv \mu y. \alpha \vee \langle \varrho \rangle y \equiv \mu y. \alpha \vee f_\varrho^y(y) \equiv f_\pi^x(\alpha)$.

For the dual recall that $f_{\pi^\delta}^x(\alpha)$ is defined as $\neg f_\pi^x(\neg \alpha)$. Items (1a) and (1b) are direct by Proposition 3.3.9, that is, the closure of the relevant fragments under Boolean duals. For (1c) first we use the inductive hypothesis and get that $f_\pi^x(\neg \alpha) \equiv \langle \pi \rangle \neg \alpha$. Therefore, $\neg f_\pi^x(\neg \alpha) \equiv \neg \langle \pi \rangle \neg \alpha$ from which it is straightforward to show that $\neg f_\pi^x(\neg \alpha) \equiv \langle \pi^\delta \rangle \alpha$. \square

A close look at this translation and proof reveals that the exact same technique works for PDL and CPDL. That is, not only can PDL and CPDL be translated to $\mu\text{ML}^\vee[2]$ but they can also be translated to $\mu_{na}\text{ML}[2]$ and $\mu_{nc}\text{ML}^\vee[2]$, respectively.

3.3.14. COROLLARY. $\text{PDL} \equiv \mu_{na}\text{ML}[2]$ and $\text{CPDL} \equiv \mu_{nc}\text{ML}^\vee[2]$.

The translation in the other direction is provided by the following proposition.

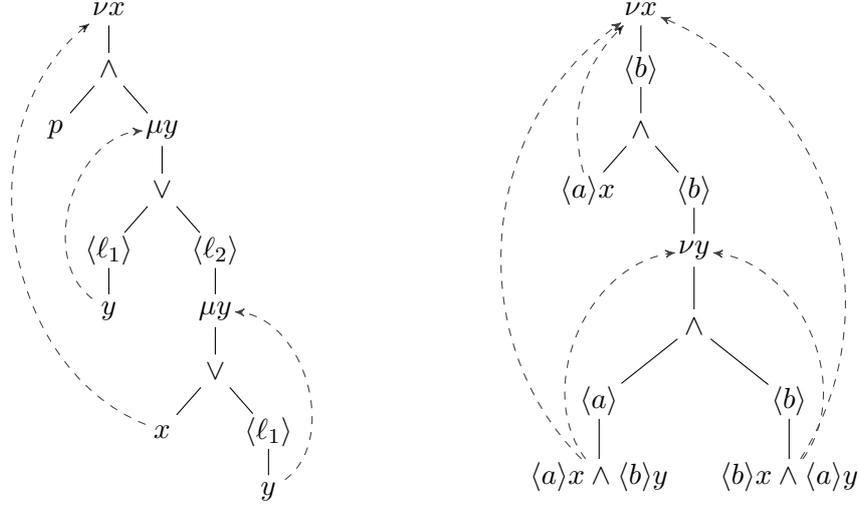
3.3.15. PROPOSITION. *The following procedures can be performed effectively:*

- (i) *Given a formula $\alpha \in \mu\text{ML}^\vee$, return an equivalent formula $\alpha^s \in \text{GL}$.*
- (ii) *Given a formula $\varphi \in \mu\text{MLMON}_p^\vee \cap \mu\text{ML}^\vee$, return a p -free game $\pi \in \text{GL}$ such that $\varphi \equiv \langle \pi \rangle p$.*

Proof. We prove the proposition via a mutual induction on the fragments μML^\vee and μMLMON_p^\vee . Leaving item (i) to the reader (which is proved just like in Proposition 3.1.30) we focus on item (ii). Let $\varphi \in \mu\text{MLMON}_p \cap \mu\text{ML}^\vee$, we prove some of the most interesting cases and leave the rest to the reader.

- In case $\varphi = p$, simply take $\pi := \top?$; clearly $\varphi \equiv \langle \top? \rangle p$.
- In case $\varphi = q$ (the case $\neg q$ is similar) we use $\alpha := \langle q?; \neg q! \rangle p$ as a formula. To prove that $\varphi \equiv \alpha$ assume first that q is false, then \exists fails at $q?$ and α is false. If q is true then \exists goes through $q?$ but then \forall fails at $\neg q!$ ($\varphi!$ is the dual of $\varphi?$), therefore α is true. Observe that the p here is irrelevant.
- As an example of a Boolean connective consider $\varphi = \varphi_1 \wedge \varphi_2$. Using the inductive hypothesis we get $\varphi_i \equiv \langle \pi_i \rangle p$ and hence $\varphi \equiv \langle \pi_1 \rangle p \wedge \langle \pi_2 \rangle p$. It is straightforward to verify that $\varphi \equiv \langle \pi_1 \otimes \pi_2 \rangle p$.
- As an example of a modality consider $\varphi = [d]\varphi'$. By inductive hypothesis we get $\varphi' \equiv \langle \pi' \rangle p$ and hence $\varphi \equiv [d]\langle \pi' \rangle p$. It is straightforward to verify that $\varphi \equiv \langle d^\delta; \pi' \rangle p$.
- Let φ be $\mu q. \varphi' \vee \varphi''$ where $\varphi' \in \mu\text{MLMON}_p^\vee$ is q -free and $\varphi'' \in \mu\text{MLMON}_q^\vee$ is p -free. By inductive hypothesis on both formulas we get that $\varphi' \equiv \langle \pi' \rangle q$ and $\varphi'' \equiv \langle \pi'' \rangle p$. We have already seen that then $\varphi \equiv \langle (\pi'')^*; \pi' \rangle q$.
- The case of the greatest fixpoint is solved dually. Let φ be of the form $\nu q. \varphi' \wedge \varphi''$ where $\varphi' \in \mu\text{MLMON}_p^\vee$ is q -free and $\varphi'' \in \mu\text{MLMON}_q^\vee$ is p -free. By inductive hypothesis on both formulas we get that $\varphi' \equiv \langle \pi' \rangle q$ and $\varphi'' \equiv \langle \pi'' \rangle p$. In this case we use the dual $(-)^{\circ}$ of $(-)^*$ and verify that $\varphi \equiv \langle (\pi'')^{\circ}; \pi' \rangle q$.

This finishes the proof of both statements. □

Figure 3.2: Fixpoint interleaving of $\mu\text{ML}^\vee[2]$ and $\mu\text{ML}[2]$.

What is special about GL? Given what we proved in this section, the question of whether GL is equivalent to $\mu\text{ML}[2]$ can be reduced to the question of the equivalence of $\mu\text{ML}[2]$ and $\mu\text{ML}^\vee[2]$. The only (but not minor) difference between these fragments is the separation constraint under the fixpoint operator. We conjecture that these two fragments have different expressive power, and proceed to give some intuitions that may help for further research.

3.3.16. CONJECTURE. $\mu\text{ML}[2] \not\subseteq \mu\text{ML}^\vee[2]$.

Our main observation is that formulas of $\mu\text{ML}^\vee[2]$ have a very special structure in the interleaving of the fixpoints. To give an example, consider the GL formula $\varphi = \langle (\ell_1^*; \ell_2; \ell_1^*)^\circ \rangle p$. This formula can be translated to the fragment $\mu\text{ML}^\vee[2]$ as:

$$\varphi^t = \nu x. p \wedge (\mu y. \langle \ell_2 \rangle (\mu y. x \vee \langle \ell_1 \rangle y) \vee \langle \ell_1 \rangle y).$$

In Fig. 3.2 we draw the formula structure of $\varphi^t \in \mu\text{ML}^\vee[2]$ and of a formula which we call $\chi^2 \in \mu\text{ML}[2]$, which is hard for the second level of the variable hierarchy of μML . That is, there is no formula in $\mu\text{ML}[1]$ equivalent to χ^2 (see [Ber05, Corollary 5.3.4]).

The formula structure of φ^t clearly mimics the sequentiality of the game (or program) structure of φ . For example, after the definition of each fixpoint, the interested player has to already choose whether (s)he is interested to possibly regenerate the fixpoint variable, or to exit the component altogether. This corresponds to the iteration construct.

Another feature that we consider even more crucial is the following: the *regeneration structure* of μML^\vee is very simple. Consider the evaluation of the formula $\langle (\ell_1^*; \ell_2; \ell_1^*)^\circ \rangle p$. We start with the game $(\ell_1^*; \ell_2; \ell_1^*)^\circ$ and \forall has to choose if he wants

to regenerate the star-dual. Suppose he does. Then \exists has to play ℓ_1 a number of times. This is clearly done in μML^\vee with a fixpoint operator. The observation is that after \exists is done, the game *has to continue* with $\ell_2; \ell_1^*$ before \forall gets the option to regenerate the star-dual again. That is, it is not possible to *break* the play of the game in the middle, and regenerate a star or star-dual. In other words, this means that if x and y bind two different fixpoints (i.e., stars or star-duals) then we shouldn't be able to have a formula like $\langle \ell_1 \rangle x \wedge \langle \ell_2 \rangle y$ in the scope of these fixpoints, as this formula is regenerating different variables depending on the action.

The formula structure of χ^2 is clearly more complex, and it is not known to us whether this formula can be taken (maybe with a complexity blowup) to the simpler form of $\mu\text{ML}^\vee[2]$.

3.4 Conclusions and open problems

In this chapter we studied some fragments of modal and first-order fixpoint logics. In particular, we made a thorough analysis of complete additivity and continuity both abstractly (at the level of maps) and of the logics resulting from a restriction of the fixpoint operator to these notions.

On the modal side we gave characterizations of PDL-like logics as fragments of the modal μ -calculus. Namely, we identified fragments of μML corresponding to PDL, CPDL and GL. We saw that in the case of PDL everything ran smoothly thanks to the multiple nice properties of complete additivity. However, the cases of CPDL and GL required some extra constraints which need to be further contemplated.

On the first-order side, we showed that first-order logic with transitive closure is equivalent to first-order logic with completely additive fixpoints. We also considered the effect of continuous fixpoints in $\mu_c\text{FOE}^\infty$ and showed that this logic is properly included in WMSO.

Open problems. We already discussed several open problems in each of the sections of this chapter. As a summary, we provide a list with some additional problems.

1. *Separation of variables:* In the modal sections of this chapter we saw that, in order to translate a fragment of μML into a PDL-like syntax, it seemed crucial to 'separate variables' under the fixpoint operator. This constraint may really be crucial or it could be possible that every formula of μML can be transformed into an equivalent formula in such a form. Conjecture 3.2.20 and 3.3.16 depend on an answer to this question, which would separate or prove equivalent GL and $\mu\text{ML}[2]$.

2. *Succinctness of the fragments*: In this chapter we proved that some (fragments of) logics are effectively equivalent. Even then, one of the formalisms may be more succinct than its correspondent in expressing certain properties. For example, if we take PDL and $\mu_a\text{ML}$, there seems to be a blowup in formula complexity when going from PDL to $\mu_a\text{ML}$ (see Propositions 3.1.30 and 3.1.26). It would be interesting to understand the relationship between these formalisms in terms of succinctness.
3. *Characterization of full FO(TC) inside FO(LFP)*: In this chapter we gave a precise characterization of the relationship between $\text{FO}(\text{TC}^1)$ and $\text{FO}(\text{LFP}^1)$. It would be worth checking if this relationship lifts to $\text{FO}(\text{TC})$ and $\text{FO}(\text{LFP})$.
4. *Equivalence of fragments modulo fixpoint*: In [Fon10, Fon08] Fontaine and Venema study the continuous fragment μMLCON_Q of μML . They remark that every continuous formula is constructive [FLV10, Proposition 5.4.2], but that the converse is not true. For example the formula $\varphi = \Box p \wedge \Box \Box \perp$ is constructive but not continuous. However, they observe that $\mu p.\varphi \equiv \mu p.\Box \Box \perp$ and therefore there is a continuous formula $\varphi' = \Box \Box \perp$ such that φ and φ' are equivalent *modulo fixpoint*. The question is raised of whether it is always possible to find such a continuous formula.

Going back to this chapter, our study of the property of separation of variables suggests further research in this direction. In Proposition 3.2.5 we discussed that the map $F(A, B) = A \cap B$ cannot be expressed as the union of two maps $G(A), G'(B)$. However, we *can* do that modulo fixpoint. That is, $\text{LFP}_A.\text{LFP}_B.F(A, B) = \emptyset$ and therefore if we define $G(A) = \emptyset$ and $G'(B) = \emptyset$ we certainly have $\text{LFP}_A.\text{LFP}_B.F(A, B) = \text{LFP}_A.\text{LFP}_B.G(A) \cup G'(B)$.

The main point is, to prove that $\mu\text{ML}[2]$ is equivalent to $\mu\text{ML}^\vee[2]$ (or not) we can focus on checking whether the property of separation of variables is crucial or not *modulo fixpoint*.

5. *Models definable by PDL, CPDL and GL*: One of the main problems with working with these logics is that we lack a good semantic characterization of the classes of transition systems, or properties, which are definable in them. Suppose that we have a class of transition systems \mathfrak{C} : which properties should \mathfrak{C} satisfy to be definable in, for example, PDL? We know that it should be closed under bisimulation, however, that should also be the case for \mathfrak{C} to be definable in μML and even in ML . It would be very helpful to have clear model theoretic properties to tear all these logics apart.
6. *Semantic characterizations inside μML* : We have given a characterization of PDL as a syntactic fragment of μML . Namely, the fragment $\mu_a\text{ML}$ which is related to the property of complete additivity. This is a good start, but it would also be interesting to have a characterization of the following style: “a formula

$\varphi \in \mu\text{ML}$ is equivalent to a formula $\varphi' \in \text{PDL}$ iff the semantic property P holds on φ ". A candidates for P could be of the style ‘invariance under a different notion of (bi-)simulation or EF game.’ Similar characterizations would also prove useful for CPDL and GL.

7. *Decidability of membership in PDL, CPDL and GL:* In [Ott99] Otto showed that given a formula of $\varphi \in \mu\text{ML}$ it is decidable to know if it is equivalent to some formula in the basic modal logic. However, it is unknown if we can decide whether φ is equivalent to a formula in $\{\text{PDL}, \text{CPDL}, \text{GL}\}$. The fragments presented in this chapter give some insight, since now the question can equivalently be cast with respect to the fragments $\{\mu_a\text{ML}, \mu_c\text{ML}^\vee, \mu\text{ML}^\vee\}$. However, the question remains open.
8. *Separate $\mu_c\text{ML}^b$ and $\mu_c\text{ML}$:* Peleg leaves the open question of whether $\mu_c\text{ML}^b$ and $\mu_c\text{ML}$ are equivalent [Pel85, after Theorem 2.11]. We conjecture that $\text{CPDL} \not\subseteq \mu_c\text{ML}^b$. One way to prove it would be to prove that $\text{PDL} \not\subseteq \mu\text{ML}[1]$. Although quite intuitive, we could not find a proof of this statement. Suppose that it holds, then we can separate $\mu_c\text{ML}^b$ and $\mu_c\text{ML}$ as follows:

As observed in [SV10], flat fragments of the μ -calculus can be reduced to their one-variable fragment. That is, in particular $\mu_c\text{ML}^b \equiv \mu_c\text{ML}^b[1]$. Intuitively, the non-interleaving of the fixpoint operators lets us reuse the same binding variable over and over again. From this observation we get $\text{CPDL} \not\subseteq \mu_c\text{ML}^b$, since already $\text{PDL} \not\subseteq \mu\text{ML}[1]$.

Intuitively, $\mu_c\text{ML}^b$ would only be able to express (in the best case) the fragment of CPDL without nesting of star operators, but already the star-height of PDL formulas is strict [Egg63, Corollary 1].

Chapter 4

Subclasses of parity automata

One of the main objectives of this dissertation is to provide a set of automata-theoretic tools to analyze fragments of fixpoint logics. In particular, we are interested in studying the fragments given in Chapter 3. In the current chapter we make a first step in this direction by introducing several subclasses of parity automata. These subclasses are inspired by the fragments of Chapter 3 and will try to parallel, on the automata side, the additivity and continuity constraints of the syntactic fragments. Our definitions will be given for parity automata over arbitrary one-step languages, and we will focus our discussion to the intuitions and motivations behind these definitions. Concrete automata (i.e., over particular one-step languages) of these classes will be examined in later chapters.

In the last part of this chapter we introduce a general technique (due to Janin [Jan06]) to bring parity automata into a tree-like shape. That is, we show how every parity automata can be unraveled to obtain an equivalent automata which looks like a tree with back edges. This structure has the advantage of being ‘almost a (fixpoint) formula’ and therefore it is easy to translate it to an appropriate fixpoint language.

To finish, we introduce other possible equivalent definitions of parity automata. These different perspectives will become useful in later chapters.

Even though we use parity automata running over labeled transition systems, in this chapter we choose to restrict most of our discussion (but not the definitions) to *trees*. The only reason is that the intuitions are easier to visualize over tree structures than on arbitrary models.

4.1 Weak parity automata

In this section we introduce and briefly discuss the notion of *weak (alternating) parity automata*. This class of automata is defined by posing an additional restriction on the parity map, which results in weaker expressive power.

Weak automata were introduced in [MSS92] to study weak definability [Rab70] in *trees with fixed finite branching degree*. That is, to study the classes (also called languages) of k -ary trees which can be defined in WMSO.

4.1.1. DEFINITION. The class $Aut_w(\mathcal{L})$ of *weak automata* is given by the automata $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ from $Aut(\mathcal{L})$ such that the following condition holds:

(weakness) if $a \preceq b$ and $b \preceq a$ then $\Omega(a) = \Omega(b)$.

The intuition is that every run of a weak automaton \mathbb{A} stabilizes on (‘gets trapped into’) some strongly connected component $C \subseteq A$ after finitely many steps, and therefore the only parity seen infinitely often after that point will be the parity of C . Moreover, as only *one* parity can be repeated infinitely often, the precise number does not matter; only the parity does:

4.1.2. FACT ([NSW02]). *Every weak automaton $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ is equivalent to a weak automaton $\mathbb{A}' = \langle A, \Delta, \Omega', a_I \rangle$ with parity map $\Omega' : A \rightarrow \{0, 1\}$.*

Proof. Just define $\Omega'(a) := \Omega(a) \bmod 2$. □

From now on we assume such a map for weak parity automata. The special structure of weak alternating automata is reflected in their attractive computational properties [KV01, KVW00]. If we think about trees, the leading intuition is that the weakness condition restricts the processing of the ‘vertical dimension’ of input trees. In the context of trees of bounded branching, this restriction is all that is needed to characterize WMSO.

4.1.3. THEOREM ([MSS92, THEOREM 1]). *A k -ary tree language is accepted by a weak automaton iff it is definable in WMSO.*

However, if the branching of the tree is not bounded, the story is quite different. This scenario was studied in [Zan12, FVZ13] where it was shown that weak automata capture a different logic.

4.1.4. THEOREM ([FVZ13, THEOREM 2]). *An (arbitrarily branching) tree language is accepted by a weak automaton iff it is definable in WFMSO.*

In this case, WFMSO stands for *well-founded* MSO, a variant of MSO which quantifies over (subsets of) well-founded trees. Moreover, it is shown that in the class of arbitrary trees, the logics WFMSO and WMSO are incomparable (see [Zan12, Corollary 5.16]).

The different behaviour of weak automata depending on the branching degree of the trees can be explained if we look at the runs of such automata. Intuitively, the problem is that weak automata can process well-founded but infinitely branching trees. On the other hand, a finite subset of a tree (as quantified by WMSO) is always embedded in a well-founded *and finitely branching* subtree. In the following sections we will consider additional constraints to solve this problem.

No alternation. If we think of parity automata as the automata counterpart of fixpoint logics, it is known that the (Mostowski) index of the automata (i.e., the range of the parity map) is tightly connected to the alternation of fixpoints [Wil01]. As weak automata can be thought of as having a parity map with range $\{0, 1\}$ (cf. Fact 4.1.2) this means that, on the fixpoint side, the corresponding logic will be *alteration-free*. This correspondence was proved between concrete weak automata (based on FO) and the alternation-free μ -calculus, for increasingly more general structures in [AN92, KV05, KV98, KV03].

4.2 Continuous-weak parity automata

We will introduce the notion of *continuous-weak (alternating) parity automata* which combines the ‘vertical’ constraint given by the weakness condition (cf. Section 4.1) with an additional restriction on the transition map, which amounts to a ‘horizontal’ constraint.

As we discussed, the weakness condition on parity automata does not seem to be enough to capture WMSO, since we can still define infinitely branching well-founded trees. The intuition on what is missing comes the fragments of Chapter 3: If we look at the fragments $\mu_c\text{ML}$ and $\mu_c\text{FOE}^\infty$, and particularly to Proposition 3.2.37 we see that restricting the (least) fixpoints to continuous formulas is tightly connected to finiteness. Namely, we observed that the least fixpoint of a continuous formula can be assumed to be finite (in the sense of Theorem 3.2.6). As the fixpoints are matched with cycles in the automata, we would like to impose some kind of continuity constraint on cycles. The resulting notion is as follows:

4.2.1. DEFINITION. The class $\text{Aut}_{wc}(\mathcal{L})$ of *continuous-weak automata* is given by the automata $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ from $\text{Aut}(\mathcal{L})$ such that for every maximal strongly connected component $C \subseteq A$ and states $a, b \in C$ the following conditions hold:

(weakness) $\Omega(a) = \Omega(b)$,

(continuity) for every color $c \in \wp(\mathbf{P})$:

If $\Omega(a)$ is odd then $\Delta(a, c)$ is continuous in C .

if $\Omega(a)$ is even then $\Delta(a, c)$ is co-continuous in C .

For this definition to make sense, we need to give a notion of (co-)continuity for one-step languages. We introduce it promptly, but postpone the discussion of this particular one-step version to Chapter 5.

4.2.2. DEFINITION. We say that $\varphi \in \mathcal{L}(A)$ is *continuous in* $a \in A$ if φ is monotone in a and additionally, for every (D, V) and assignment $g : \text{iVar} \rightarrow D$,

if $(D, V), g \models \varphi$ then $\exists U \subseteq_\omega V(a)$ such that $(D, V[a \mapsto U]), g \models \varphi$.

We say that φ is *co-continuous in* $a \in A$ if the Boolean dual φ^δ of φ (cf. Definition 2.3.5) is continuous in $a \in A$.

Recall from Section 3.2 that continuity in the product coincides with continuity in every variable. Therefore we say that φ is continuous in $C \subseteq A$ iff φ is continuous in every $a \in C$.

Intuitively, the continuity restriction has the following effect when combined with the weakness restriction: suppose that the run of a continuous-weak automaton stays inside a connected component C for some rounds of the acceptance game. Moreover, suppose that the parity of C is odd. For this case, the continuity condition lets us assume without loss of generality that the nodes of the tree coloured with some state of C form a *finitely branching and well-founded* subtree. The reason for this is that at each round of the acceptance game –because of continuity– player \exists can play a valuation where at most finitely many nodes are colored with C . After that, \forall subsequently chooses an element coloured by C , a new round starts. Repeating this strategy for a finite number of rounds will define a finitely branching well-founded subtree. Observe also that, on trees, every finite set is included in a finitely branching and well-founded subtree; and every such subtree is finite. This is the rationale behind trying to characterize WMSO with continuous-weak automata.

4.3 Additive-weak parity automata

In this section we introduce and briefly discuss the notion of *additive-weak (alternating) parity automata*. This class of automata combines the ‘vertical’ constraint given by the weakness condition (cf. Section 4.1) with an additional restriction on the transition map, which amounts to a ‘horizontal’ constraint.

The driving intuition behind this kind of automata is that we would like to have an automata counterpart for WCL. In Section 4.2 we used an extra continuity constraint on the cycles of the automata, to make continuous-weak automata ‘work with’ finite trees. In this case, we will use an additivity constraint to make additive-weak automata work with finite *paths*.

4.3.1. DEFINITION. We say that $\varphi \in \mathcal{L}(A)$ is *completely additive in* $\{\bar{\mathbf{a}}\} \subseteq A$ if φ is monotone in every a_i and, for every (D, V) and assignment $g : \text{iVar} \rightarrow D$,

if $(D, V), g \models \varphi$ then $(D, V[\bar{\mathbf{a}} \mapsto \bar{\mathbf{Q}}]), g \models \varphi$ for some quasi-atom $\bar{\mathbf{Q}}$ of $V(\bar{\mathbf{a}})$.

We say that φ is *completely multiplicative in* $\{\bar{\mathbf{a}}\} \subseteq A$ if the Boolean dual φ^δ of φ (cf. Definition 2.3.5) is completely additive in $\{\bar{\mathbf{a}}\} \subseteq A$.

A more systematic study of the notions of one-step complete additivity and multiplicativity, together with concrete cases will be given in Chapter 5. We now formally define additive-weak automata for an arbitrary one-step language.

4.3.2. DEFINITION. The class $Aut_{wa}(\mathcal{L})$ of *additive-weak automata* is given by the automata $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ from $Aut(\mathcal{L})$ such that for every *maximal* strongly connected component $C \subseteq A$ and states $a, b \in C$ the following conditions hold:

(weakness) $\Omega(a) = \Omega(b)$,

(additivity) for every color $c \in \wp(\mathbf{P})$:

If $\Omega(a)$ is odd then $\Delta(a, c)$ is completely additive in C .

if $\Omega(a)$ is even then $\Delta(a, c)$ is completely multiplicative in C .

Intuitively, the additivity restriction has the following effect: while a run of an additive automaton stays inside a connected component with odd parity, we can assume without loss of generality that the nodes of the tree coloured with some state of C form a *path* in the tree. The reason for this is that at each step –because of complete additivity– player \exists can play a valuation where at most one node is coloured with C . Therefore, if \forall chooses the element coloured by C , a repetition of this step will define a path.

4.4 Partial unraveling of parity automata

Regarding their structure, parity automata are much more flexible than formulas. In general, the syntax tree of a formula (e.g., of first-order logic or modal logic) induces, as the name hints, a tree. In the case of formulas of fixpoint logics (like the first-order and modal μ -calculus) we can see the formulas as a tree with extra edges going from fixpoint variables to their binding definitions (see e.g., Fig. 3.2). On the other hand, parity automata can induce arbitrary graphs.

In this section we present a general procedure to transform an arbitrary parity automaton into a parity automaton whose induced graph is a tree with back edges. As observed, this kind of structure has a natural counterpart as a formula. Moreover, we show that this transformation preserves the weakness, continuity and additivity conditions.

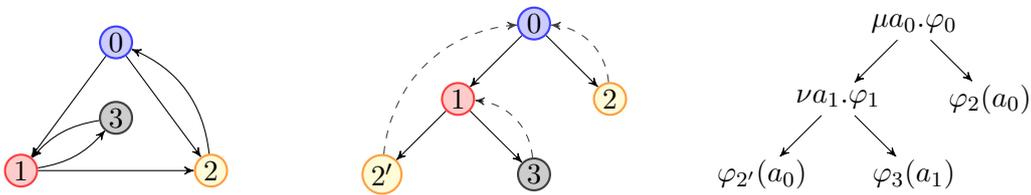


Figure 4.1: Automata, finite unraveling and formula structure.

Once we have automata of this shape, it is easy to translate them to fixpoint formulas, as we will do in later chapters. Intuitively, the tree part of the automaton is used to define the scaffolding of the corresponding formulas. On top of

that, the nodes which are the target of back-edges will correspond to binding definitions of fixpoint variables. Fig. 4.1 gives an example where the target formula is taken to be in the μ -calculus. This is done for illustrative reasons.

4.4.1. DEFINITION. A directed graph $(G, R \subseteq G^2)$ is a *tree with back-edges* if there is a partition $R = E \uplus B$ of the edges into tree edges and back edges such that (G, E) is indeed a directed tree, and whenever $(u, v) \in B$, then $(v, u) \in E^*$.

Berwanger [Ber05] shows that every finite transition system can be transformed, via partial unraveling, into a bisimilar finite model which is a tree with back edges. An unraveling technique is also present in Janin's habilitation thesis [Jan06, Section 3.2.3], where he puts modal parity automata into the shape of trees with back edges. We adapt these ideas to our setting by defining a similar transformation on parity automata of an arbitrary one-step language \mathcal{L} .

For every automaton \mathbb{A} we will define an unraveling \mathbb{A}^u . We want the latter automaton to satisfy the following two properties:

- (i) \mathbb{A}^u is a tree with back-edges,
- (ii) For every cycle, the state which is located *highest* in the tree (i.e., closest to the root) has the *minimum* parity among the states of the cycle (i.e., the maximum priority).

Item (ii) is not necessary for \mathbb{A}^u to be equivalent to \mathbb{A} . However, as illustrated in Fig. 4.1, we will later use the tree structure of the unraveled automaton to define a fixpoint formula. In these formulas, the fixpoint operator which is higher in the tree has the highest priority. It is because of this that we want higher states to have higher priority. We refer the reader to Fig. 4.2 for an example of unraveling, and in particular, of the requirements of item (ii).

4.4.2. DEFINITION. The *finite (or partial) unraveling* of a parity automaton $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ is the parity automaton $\mathbb{A}^u = \langle A^u, \Delta^u, \Omega^u, a_I^u \rangle$ such that

1. A^u is made of finite sequences $\bar{a} \in A^+$ where $a_0 = a_I$ and $a_i \rightsquigarrow_{\mathbb{A}} a_{i+1}$,
2. a_I^u is the one-element sequence containing only a_I ,
3. Every element of A^u is reachable from a_I^u ,
4. $\Omega^u(\bar{a} \cdot a_k) = \Omega(a_k)$, and
5. $\Delta^u(\bar{a} \cdot a_k, c) = \Delta(a_k, c)[b \mapsto \text{update}(\bar{a} \cdot a_k, b) \mid b \in A]$ where $\text{update}(a_0, \dots, a_k, b)$ is defined as the shortest prefix a_0, \dots, a_i of a_0, \dots, a_k, b such that (a) $a_i = b$ and, (b) for every $i < j \leq k$ we have that $\Omega(a_i) \leq \Omega(a_j)$; that is, the minimum parity encountered in the cycle a_i, a_{i+1}, \dots, a_i is $\Omega(a_i)$.

It is worth observing that \mathbb{A}^u can be constructed from \mathbb{A} with a cardinality at most doubly exponential in the size of \mathbb{A} ; in particular, condition (5) can be satisfied [Jan06, Lemma 3.2.3.2].

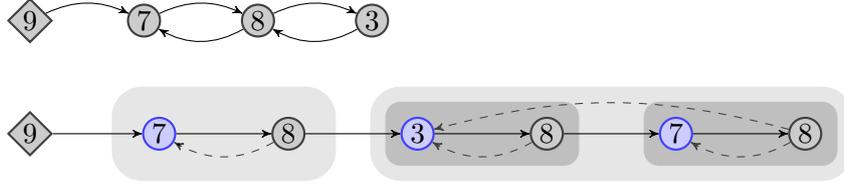


Figure 4.2: Partial unraveling with parities (grows to right).

4.4.3. REMARK. Condition (5b) is there to ensure that the target of a back-edge is ‘of maximum priority’ (i.e., minimum parity) among the elements of the given cycle. In our case, as the automata that we use are *weak*, all the parities of the elements of a given cycle are the same. Since the resulting formula will not have any alternation, we could have simply left condition (5b) out. We chose to keep it for compatibility with the results of [Jan06], and because we will need it in Section 7.1.2, where we recall how to turn MSO-automata into formulas.

4.4.4. LEMMA ([JAN06, LEMMA 3.2.3.2]). $\mathbb{A} \equiv \mathbb{A}^u$.

We devote the rest of this section to proving that the unraveling construction preserves the weakness, continuity and additivity constraints. Define the projection $\text{last} : A^+ \rightarrow A$ as $\text{last}(a_0, \dots, a_k) := a_k$. For sets $B \subseteq A^+$ the projection is extended to $\text{last} : \wp(A^+) \rightarrow \wp(A)$ by defining $\text{last}(B) := \{\text{last}(\bar{a}) \mid \bar{a} \in B\}$. The following observations will be useful:

4.4.5. PROPOSITION.

1. If $C \subseteq A^u$ is a strongly connected component in \mathbb{A}^u then $\text{last}(C)$ is a strongly connected component in \mathbb{A} .
2. $\Omega(\text{last}(C)) = \Omega^u(C)$, for every strongly connected component $C \subseteq A^u$.

Proof. Item (2) is direct by definition of Ω^u . To prove item (1) it is enough to prove that if $a_0, \dots, a_k \prec_{\mathbb{A}^u} b_0, \dots, b_{k'}$ then $a_k \prec_{\mathbb{A}} b_{k'}$, as strongly connected components are defined in terms of \prec . Now, because \prec is the reflexive-transitive closure of \rightsquigarrow , it will actually be enough to prove that if $a_0, \dots, a_k \rightsquigarrow_{\mathbb{A}^u} b_0, \dots, b_{k'}$ then $a_k \rightsquigarrow_{\mathbb{A}} b_{k'}$. For this, just observe that if $a_0, \dots, a_k \rightsquigarrow_{\mathbb{A}^u} b_0, \dots, b_{k'}$ then, by construction of Δ^u in Definition 4.4.2, we have that $b_0, \dots, b_{k'}$ is the result of replacing the name $b_{k'}$ in $\Delta(a_k, c)$ with the outcome of the function `update`. Therefore, to have $b_0, \dots, b_{k'}$ in $\Delta^u(a_0, \dots, a_k, c)$, we must have $b_{k'}$ in $\Delta(a_k, c)$ in the first place. By definition of $\rightsquigarrow_{\mathbb{A}}$ this means that $a_k \rightsquigarrow_{\mathbb{A}} b_{k'}$. \square

We start with the preservation of the weakness condition.

4.4.6. PROPOSITION. If $\mathbb{A} \in \text{Aut}_w(\mathcal{L})$ then $\mathbb{A}^u \in \text{Aut}_w(\mathcal{L})$.

Proof. By Proposition 4.4.5(1) we know that if C is a maximal strongly connected component in \mathbb{A}^u then $\text{last}(C)$ will also be a strongly connected component in \mathbb{A} . As \mathbb{A} is weak, then every element of $\text{last}(C)$ will have the same parity, which we call $\Omega^u(\text{last}(C))$. Using Proposition 4.4.5(2), we know that $\Omega(\text{last}(C)) = \Omega^u(C)$, and therefore get that every element of C has the same parity. \square

4.4.7. PROPOSITION. *If $\mathbb{A} \in \text{Aut}_{wa}(\mathcal{L})$ then $\mathbb{A}^u \in \text{Aut}_{wa}(\mathcal{L})$.*

Proof. The weakness condition is preserved by Proposition 4.4.6. For the additivity condition let $C \subseteq A^u$ be a maximally connected component with odd $\Omega^u(C)$ and let \bar{a} be an element of C . We want to prove that $\Delta^u(\bar{a}, c)$ is completely additive in C , for every color $c \in \wp(\mathbf{P})$. Define $\varphi := \Delta(\text{last}(\bar{a}), c)$. It is not difficult to observe that, as $\text{last}(\bar{a})$ is in the connected component $\text{last}(C)$, then φ is completely additive in $\text{last}(C)$. The key observation now is that if we substitute all the names in φ from $\text{last}(C)$ with some new set of names A' then the new formula will be completely additive in A' . To conclude, we just recall that $\Delta^u(\bar{a}, c)$ is obtained by substituting the names from $\text{last}(C)$ in φ with new names that belong to C . Using the previous observation, we get that $\Delta^u(\bar{a}, c)$ is completely additive in C . We leave the case of even $\Omega^u(C)$ to the reader. \square

4.4.8. PROPOSITION. *If $\mathbb{A} \in \text{Aut}_{wc}(\mathcal{L})$ then $\mathbb{A}^u \in \text{Aut}_{wc}(\mathcal{L})$.*

Proof. Same as additivity. \square

4.5 Variants of parity automata

In Section 2.3 we defined parity automata for an arbitrary language \mathcal{L} as a tuple $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ with $\Delta : A \times \wp(\mathbf{P}) \rightarrow \mathcal{L}^+(A, \mathbf{D})$ and whose semantics is given by the following parity game.¹

Position	Pl'r	Admissible moves	Parity
$(a, s) \in A \times S$	\exists	$\{V : A \rightarrow \wp(R[s]) \mid (R[s], V) \models \Delta(a, \kappa(s))\}$	$\Omega(a)$
$V : A \rightarrow \wp(S)$	\forall	$\{(b, t) \mid t \in V(b)\}$	$\max(\Omega[A])$

This will be the main definition of parity automata used throughout this dissertation. Nevertheless, in some special situations, it is useful to consider other definitions of such automata which will turn out to be equivalent but (in that context) easier to manipulate. In this section we discuss variants of parity automata and how they are connected.

¹Recall that \mathcal{L}^+ is the positive fragment of \mathcal{L} .

Modal and first-order automata. One of the most important families of one-step languages that we will use is that of first-order languages. For example, we have already seen that $Aut(\text{FOE}_1)$ captures MSO on trees [Wal96], and that $Aut(\text{FO}_1)$ captures μML on all models [JW95]. However, there is a clear difference between these logics: MSO is an extension of first-order logic, and in most cases it is useful to work with automata based on FOE_1 when trying to study MSO; on the other hand, μML is a *modal* logic, and hence it is sometimes tedious to work with FO_1 when a modal language would be a closer match.

If we compare FO_1 with, say, modal logic we see that the first language is fit to describe one-step models of the form (D, V) while formulas of modal logic are evaluated at *points*. However, observe that if the acceptance game of some automaton is standing at some basic position (a, s) then the formulas

$$\exists x.(b(x) \wedge c(x)) \vee \forall y.d(y) \quad \text{and} \quad \diamond(b \wedge c) \vee \square d$$

basically describe the same requirements over the set $R[s]$. The only difference is that while the former is directly evaluated at $R[s]$, the latter should be evaluated directly at s . It is possible, then, to slightly modify our definition of parity automata to work with modal one-step languages.

4.5.1. DEFINITION. A *modal parity automaton* is a parity automaton based on a modal language, whose semantics is given as follows. Given a model \mathbb{S} and a parity automaton \mathbb{A} based on a *modal language* we define the rules for the *acceptance game* $\mathcal{A}(\mathbb{A}, \mathbb{S})$ as follows:

Position	Pl'r	Admissible moves	Parity
$(a, s) \in A \times S$	\exists	$\{V : A \rightarrow \wp S \mid \mathbb{S}, V, s \Vdash \Delta(a, \kappa(s))\}$	$\Omega(a)$
$V : A \rightarrow \wp S$	\forall	$\{(a, s) \mid a \in A, s \in V(a)\}$	$\max(\Omega[A])$

We also use $Aut(\mathcal{L})$ to denote the class of these automata, since the acceptance game that should be played will be clear from the choice of \mathcal{L} .

In Chapter 6 we define concrete cases of modal automata. We will then show that those automata can be equivalently seen as automata based on FO_1 . Moreover, this equivalence will carry over to the subclasses of parity automata that we use.

Chromatic and achromatic automata. Our definition of parity automata on a set of propositions \mathbf{P} includes a transition map $\Delta : A \times \wp(\mathbf{P}) \rightarrow \mathcal{L}^+(A)$. In the acceptance game of such automata, the second coordinate of the transition map is used to get different formulas depending on the coloring $\kappa(s)$ of the current node. We call this kind of transition map and automata *chromatic*, since we think of $\wp(\mathbf{P})$ as an alphabet of colors [Ven11]. This perspective is very useful, as we will see, to prove simulation theorems and closure under projection.

In other situations, which arise mostly when considering modal automata, it is more useful to have a transition map of the form $\Delta' : A \rightarrow \mathcal{L}^+(A, \mathbf{P})$ and transfer the requirements on the coloring of the current node to the formula itself. We call this type of automata and transition map *achromatic*. For example, suppose that $\Delta : A \times \wp(\mathbf{P}) \rightarrow \mathcal{L}^+(A)$ is based on a modal language $\mathcal{L}^+(A)$ without propositions, we can define $\Delta' : A \rightarrow \mathcal{L}^+(A, \mathbf{P})$ as follows:

$$\Delta'(a) := \bigvee \left\{ \left(\bigwedge_{p \in c} p \wedge \bigwedge_{p \notin c} \neg p \right) \wedge \Delta(a, c) \mid c \in \wp(\mathbf{P}) \right\}.$$

It is clear that Δ and Δ' carry the same information, however, the acceptance game has to be modified in order to work with transition maps like Δ' .

4.5.2. DEFINITION. An *achromatic* modal parity automaton is a tuple $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ with $\Delta : A \rightarrow \mathcal{L}^+(A, \mathbf{D}, \mathbf{P})$ and whose semantics is given by the following parity game.

Position	Pl'r	Admissible moves	Parity
$(a, s) \in A \times \mathcal{S}$	\exists	$\{V : A \rightarrow \wp \mathcal{S} \mid \mathbb{S}, V, s \Vdash \Delta(a)\}$	$\Omega(a)$
$V : A \rightarrow \wp \mathcal{S}$	\forall	$\{(a, s) \mid a \in A, s \in V(a)\}$	$\max(\Omega[A])$

We also use $\text{Aut}(\mathcal{L})$ to denote the class of these automata, since the acceptance game that should be played will be clear from the shape of Δ .

4.5.3. REMARK. Chromatic and achromatic automata can also be analyzed under the light of coalgebra, as observed in Chapter 1. Chromatic automata present an asymmetry between the models they run on, and the kind of one-step formulas they contain. That is, these automata (and all the automata considered in this dissertation) run on labeled transition systems on propositions \mathbf{P} , which can be seen as coalgebras for the functor $F(X) := \wp(\mathbf{P}) \times \wp(X)$. However, the transition map contains formulas without propositions, that is, they are more suited to describe coalgebras for the functor $F'(X) := \wp(X)$. In the case of achromatic automata, the symmetry is restored since both the automata and the formulas are intended for the functor F .

We do not define all combinations of (a)chromatic and (non-)modal automata since not all of them are natural. Moreover, in this dissertation we will only use chromatic first-order automata and achromatic modal automata.

4.6 Conclusions and open problems

Observe that the additivity and continuity conditions on parity automata were given semantically. In Chapter 5 we formally define every one-step language used in this dissertation and give *syntactic characterizations* of their completely additive and continuous fragments. In later chapters we take advantage of the mentioned characterizations and give concrete definitions of the automata introduced in this chapter.

Open problems. The aim of this chapter was to introduce some new definitions and discuss some known notions and techniques on parity automata. Most of the results that use these definitions and techniques will be in later chapters, where, in turn, open problems will be stated.

Chapter 5

One-step model theory

One of the advantages of taking an automata approach to fixpoint logics is that their complexity can be divided in two simpler and clearly defined parts: a graph structure representing the repetitions (i.e., the states of the automata) and a transition map with a simple one-step logic.

In this chapter we focus on the latter part. We introduce the one-step logics that we use in this dissertation and carry on an in-depth study of them. Our objective is to provide normal forms and characterize several fragments of this logics (continuous, completely additive, etc.) The results of this analysis will be crucial in later chapters, when we prove properties of automata based on these languages.

The one-step languages that we will consider are of two types: first-order based, and modal. The first-order languages that we study can be further divided into *single-sorted* and *multi-sorted*. While the relationship between first-order and modal languages has already been discussed in Section 4.5, the introduction of sorts in first-order languages deserves a short discussion.

As mentioned in Section 2.3, the one-step formulas in the transition map of an automaton are used in the acceptance game of parity automata. In each round, the game is standing at a position (a, s) where a is a state of the automaton and s is an element of the labeled transition system. At this point, \exists has to provide a coloring of the successors of s , precisely specified by (the formula in) the transition map of a . If the labeled transition system contains just one binary relation R , we can unambiguously specify the valid colorings on $R[s]$ using a *single-sorted* formula of $\text{FO}_1(A)$. For example, the formula $\exists x.a(x) \wedge \forall y.b(y)$ specifies all colorings of $R[s]$ where there is at least one element colored with state a and all elements are colored with state b . However, if the labeled transition system contains many binary relations R_1, \dots, R_n we need some way to specify, without ambiguity, to which relation do the quantified elements belong. This leads us to adding sorted quantification $\exists x:\mathbf{s}.\varphi$ to the first-order language, where \mathbf{s} belongs to some set of sorts $\mathcal{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_n\}$ representing the relations.

Single-sorted	Normal form	Monotone	Continuous	Additive
FO ₁	Fact 5.1.6	Thm. 5.1.22	Thm. 5.1.37	Thm. 5.1.49
FOE ₁	Thm. 5.1.12	Thm. 5.1.27	–	Thm. 5.1.45
FOE ₁ [∞]	Thm. 5.1.18	Thm. 5.1.31	Thm. 5.1.41	–
Multi-sorted	Normal form	Monotone	Continuous	Additive
FO ₁	Prop. 5.2.5	Thm. 5.2.18	–	Thm. 5.2.30
FOE ₁	Thm. 5.2.17	Thm. 5.2.22	–	Thm. 5.2.26
ML ₁	Thm. 5.3.6	Thm. 5.3.7	Thm. 5.3.12	Thm. 5.3.16

Table 5.1: Results of Chapter 5.

5.0.1. REMARK. This issue could have been solved in other ways. For example, instead of taking a transition map $\Delta : A \times \wp(\mathbf{P}) \rightarrow \text{FO}_1(A, \mathcal{S})$ as suggested in the above paragraph, one could have taken a transition map of the form $\Delta : A \times \wp(\mathbf{P}) \times \{1, \dots, n\} \rightarrow \text{FO}_1(A)$ which basically provides a *single-sorted* formula for *each* relation. This approach avoids adding sorts to the one-step language, but entails other technical problems for the kind of (fragment) analysis that we want to perform.

In this chapter we start with an analysis of single-sorted first-order languages. Next, we perform a multi-sorted analysis for a selected subset of these languages, which will later be used in automata running on labeled transition systems with many relations. The generalization to many sorts does not contain any fundamentally new concept or technique, but it requires a careful crafting of more complex normal forms. To finish the chapter, we perform a similar study on some modal one-step languages. Table 5.1 shows a summary of the results which are proved in this chapter.

Basic definitions. The notions of (multi-sorted) one-step model and language were given in Section 2.3 and will be promptly recalled in the following sections, we now give a few general definitions. Given a finite set of names A and $S \subseteq A$, we introduce the notation

$$\tau_S(x) := \bigwedge_{a \in S} a(x) \wedge \bigwedge_{a \in A \setminus S} \neg a(x).$$

The formula $\tau_S(x)$ is called an *A-type*, we usually blur the distinction between $\tau_S(x)$ and S and call S an *A-type* as well. A *positive A-type* only bears positive information, and is defined as $\tau_S^+(x) := \bigwedge_{a \in S} a(x)$. We use the convention that, if S is the empty set, then $\tau_S^+(x)$ is \top and we call it an *empty positive A-type*.

Given a one-step model \mathbf{D} we use $|S|_{\mathbf{D}}$ to denote the number of elements that realize the *A-type* τ_S in \mathbf{D} . Formally, $|S|_{\mathbf{D}} := |\{d \in |\mathbf{D}| : \mathbf{D} \models \tau_S(d)\}|$.

5.1 Single-sorted first-order languages

5.1.1. DEFINITION. The set $\text{FOE}_1(A)$ of one-step first-order sentences (with equality) is given by the sentences formed by

$$\varphi ::= \top \mid \perp \mid a(x) \mid x \approx y \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x.\varphi$$

where $x, y \in \text{iVar}$, $a \in A$. The one-step logic $\text{FO}_1(A)$ is as $\text{FOE}_1(A)$ but without equality. The set $\text{FOE}_1^\infty(A)$ of one-step first-order sentences with generalized quantifier \exists^∞ (with equality) is defined analogously by just adding the clauses for the generalized quantifiers $\exists^\infty x.\varphi$ and $\forall^\infty x.\varphi$.

5.1.2. REMARK. The elements \top and \perp are added for technical reasons. Even though they are already definable in $\text{FOE}_1(A)$, this will not necessarily be the case in other fragments that will be defined later.

Recall from Section 2.3 that formulas of an arbitrary (single-sorted) one-step language \mathcal{L} are interpreted over (single-sorted) one-step models, that is, a tuple $\mathbf{D} = (D, V : A \rightarrow \wp D)$. Whenever we say ‘one-step model’ in this section we will be referring to *single-sorted* one-step models. Recall that the class of all one-step models is denoted by \mathfrak{M}_1 and that we write $\mathcal{L}^+(A)$ to denote the fragment where every predicate $a \in A$ occurs only positively. Without loss of generality, from now on we always assume that every bound variable occurring in a sentence is bound by an unique quantifier (generalized or not).

5.1.3. DEFINITION. Let $\varphi \in \text{FOE}_1^\infty(A)$ be a formula, $\mathbf{D} = (D, V)$ be a one-step model and $g : \text{iVar} \rightarrow \wp(D)$ be an assignment. The semantics of $\text{FOE}_1^\infty(A)$ is given as follows:

$$\begin{aligned} \mathbf{D}, g \models a(x) & \text{ iff } g(x) \in V(a), \\ \mathbf{D}, g \models x \approx y & \text{ iff } g(x) = g(y), \\ \mathbf{D}, g \models \exists x.\varphi & \text{ iff } \mathbf{D}, g[x \mapsto d] \models \varphi \text{ for some } d \in D, \\ \mathbf{D}, g \models \exists^\infty x.\varphi & \text{ iff } \mathbf{D}, g[x \mapsto d] \models \varphi \text{ for infinitely many distinct } d \in D, \end{aligned}$$

while the Boolean connectives are defined as expected.

Recall that $\forall^\infty x.\varphi$ expresses that there are *at most finitely many* elements *falsifying* the formula φ .

5.1.4. DEFINITION. The quantifier rank $qr(\varphi)$ of a formula $\varphi \in \text{FOE}_1^\infty$ (hence also for FO_1 and FOE_1) is defined as follows

- If φ is atomic then $qr(\varphi) = 0$,
- If $\varphi = \neg\psi$ then $qr(\varphi) = qr(\psi)$,

- If $\varphi = \psi_1 \wedge \psi_2$ or $\varphi = \psi_1 \vee \psi_2$ then $qr(\varphi) = \max\{qr(\psi_1), qr(\psi_2)\}$,
- If $\varphi = Qx.\psi$ for $Q \in \{\exists, \forall, \exists^\infty, \forall^\infty\}$ then $qr(\varphi) = 1 + qr(\psi)$.

Given a one-step logic \mathcal{L} we write $\mathbf{D} \equiv_k^{\mathcal{L}} \mathbf{D}'$ to indicate that the one-step models \mathbf{D} and \mathbf{D}' satisfy exactly the same formulas $\varphi \in \mathcal{L}$ with $qr(\varphi) \leq k$. The logic \mathcal{L} will be omitted when it is clear from context.

A *partial isomorphism* between two (single-sorted) one-step models (D, V) and (D', V') is a *partial* function $f : D \rightarrow D'$ which is injective and satisfies that $d \in V(a) \Leftrightarrow f(d) \in V'(a)$ for all $a \in A$ and $d \in \text{Dom}(f)$.

Given two sequences $\bar{\mathbf{d}} \in D^k$ and $\bar{\mathbf{d}}' \in D'^k$ we use $f : \bar{\mathbf{d}} \mapsto \bar{\mathbf{d}}'$ to denote the partial function $f : D \rightarrow D'$ defined as $f(d_i) := d'_i$. We explicitly avoid cases where there exist d_i, d_j such that $d_i = d_j$ but $d'_i \neq d'_j$.

5.1.1 Normal forms

In this section we provide normal forms for the single-sorted one-step logics FO_1 , FOE_1 and FOE_1^∞ . These normal forms will be pivotal to characterize the different fragments of these logics, in later sections.

Normal form for FO_1

We start by stating a normal form for one-step first-order logic without equality. A formula in *basic form* gives a complete description of the types that are satisfied in a one-step model.

5.1.5. DEFINITION. A formula $\varphi \in \text{FO}_1(A)$ is in *basic form* if $\varphi = \bigvee \nabla_{\text{FO}}(\Sigma)$ where each disjunct is of the form

$$\nabla_{\text{FO}}(\Sigma) = \bigwedge_{S \in \Sigma} \exists x. \tau_S(x) \wedge \forall x. \bigvee_{S \in \Sigma} \tau_S(x)$$

for some set of types $\Sigma \subseteq \wp A$.

It is easy to prove, using Ehrenfeucht-Fraïssé games, that every formula of first-order logic without equality over a unary signature (i.e., FO_1) is equivalent to a formula in basic form. Proof sketches can be found in [GTW02, Lemma 16.23] and [Ven14, Proposition 4.14]. We omit a full proof because it is very similar to the following more complex cases.

5.1.6. FACT. *Every formula of $\text{FO}_1(A)$ is equivalent to a formula in basic form.*

Normal form for FOE₁

When considering a normal form for FOE₁, the fact that we can ‘count types’ using equality yields a more involved basic form.

5.1.7. DEFINITION. We say that a formula $\varphi \in \text{FOE}_1(A)$ is in *basic form* if $\varphi = \bigvee \nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)$ where each disjunct is of the form

$$\nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi) = \exists \overline{\mathbf{x}}. (\text{diff}(\overline{\mathbf{x}}) \wedge \bigwedge_i \tau_{T_i}(x_i) \wedge \forall z. (\text{diff}(\overline{\mathbf{x}}, z) \rightarrow \bigvee_{S \in \Pi} \tau_S(z)))$$

such that $\overline{\mathbf{T}} \in \wp(A)^k$ for some k and $\Pi \subseteq \overline{\mathbf{T}}$. The predicate $\text{diff}(\overline{\mathbf{y}})$, stating that the elements $\overline{\mathbf{y}}$ are distinct, is defined as $\text{diff}(y_1, \dots, y_n) := \bigwedge_{1 \leq m < m' \leq n} (y_m \not\approx y_{m'})$.

We prove that every formula of monadic first-order logic with equality (i.e., FOE₁) is equivalent to a formula in basic form. This result seems to be folklore, however, we provide a detailed proof because some of its ingredients will be used later, when we give a normal form for FOE₁[∞]. We start by defining the following relation between one-step models.

5.1.8. DEFINITION. Let \mathbf{D} and \mathbf{D}' be one-step models. For every $k \in \mathbb{N}$, the relation $\mathbf{D} \sim_k^- \mathbf{D}'$ is defined as

$$\mathbf{D} \sim_k^- \mathbf{D}' \iff \forall S \subseteq A \left(|S|_{\mathbf{D}} = |S|_{\mathbf{D}'} < k \text{ or } |S|_{\mathbf{D}}, |S|_{\mathbf{D}'} \geq k \right)$$

Intuitively, two models are related by \sim_k^- when their type information coincides ‘modulo k ’. Later we will prove that this is the same as saying that they cannot be distinguished by a formula of FOE₁ with quantifier rank lower or equal to k . For the moment, we prove the following properties of \sim_k^- .

5.1.9. PROPOSITION. *The following hold:*

- (i) \sim_k^- is an equivalence relation,
- (ii) \sim_k^- has finite index,
- (iii) Every $E \in \mathfrak{M}_1 / \sim_k^-$ is characterized by a formula $\varphi_E^- \in \text{FOE}_1(A)$ with $qr(\varphi_E^-) = k$.

Proof. We only prove the last point. Let $E \in \mathfrak{M}_1 / \sim_k^-$ and let $\mathbf{D} \in E$ be a representative. Call $S_1, \dots, S_n \subseteq A$ to the types such that $|S_i|_{\mathbf{D}} = n_i < k$ and $S'_1, \dots, S'_m \subseteq A$ to those satisfying $|S'_i|_{\mathbf{D}} \geq k$. Now define

$$\begin{aligned} \varphi_E^- := & \bigwedge_{i \leq n} (\exists x_1, \dots, x_{n_i}. \text{diff}(x_1, \dots, x_{n_i}) \wedge \\ & \bigwedge_{j \leq n_i} \tau_{S_i}(x_j) \wedge \forall z. \text{diff}(x_1, \dots, x_{n_i}, z) \rightarrow \neg \tau_{S_i}(z)) \wedge \\ & \bigwedge_{i \leq m} (\exists x_1, \dots, x_k. \text{diff}(x_1, \dots, x_k) \wedge \bigwedge_{j \leq k} \tau_{S'_i}(x_j)) \end{aligned}$$

First note that the union of all the S_i and S'_i yields all the possible A -types, and that if a type is not realized at all, then it will correspond to some S_j with $n_j = 0$. It is easy to see that $qr(\varphi_E^-) = k$ and that $\mathbf{D}' \models \varphi_E^-$ iff $\mathbf{D}' \in E$. Observe that φ_E^- gives a specification of E “type by type”. \square

Next we recall a (standard) notion of Ehrenfeucht-Fraïssé game for FOE_1 which will be used to establish the connection between \sim_k^- and \equiv_k^{FOE} .

5.1.10. DEFINITION. Let $\mathbf{D}_0 = (D_0, V_0)$ and $\mathbf{D}_1 = (D_1, V_1)$ be one-step models. We define the game $\text{EF}_k^-(\mathbf{D}_0, \mathbf{D}_1)$ between \forall and \exists . If \mathbf{D}_i is one of the models we use \mathbf{D}_{-i} to denote the other model. A position in this game is a pair of sequences $\overline{\mathbf{s}}_0 \in D_0^n$ and $\overline{\mathbf{s}}_1 \in D_1^n$ with $n \leq k$. The game consists of k rounds where in round $n + 1$ the following steps are made

1. \forall chooses an element d_i in one of the \mathbf{D}_i ,
2. \exists responds with an element d_{-i} in the model \mathbf{D}_{-i} .
3. Let $\overline{\mathbf{s}}_i \in D_i^n$ be the sequences of elements chosen up to round n , they are extended to $\overline{\mathbf{s}}_i' := \overline{\mathbf{s}}_i \cdot d_i$. Player \exists survives the round iff she does not get stuck and the function $f_{n+1} : \overline{\mathbf{s}}_0' \mapsto \overline{\mathbf{s}}_1'$ is a partial isomorphism of one-step models.

Player \exists wins iff she can survive all k rounds. Given $n \leq k$ and $\overline{\mathbf{s}}_i \in D_i^n$ such that $f_n : \overline{\mathbf{s}}_0 \mapsto \overline{\mathbf{s}}_1$ is a partial isomorphism, we use $\text{EF}_k^-(\mathbf{D}_0, \mathbf{D}_1) @ (\overline{\mathbf{s}}_0, \overline{\mathbf{s}}_1)$ to denote the (initialized) game where n moves have been played and $k - n$ moves are left to be played.

5.1.11. LEMMA. *The following are equivalent*

1. $\mathbf{D}_0 \equiv_k^{\text{FOE}} \mathbf{D}_1$,
2. $\mathbf{D}_0 \sim_k^- \mathbf{D}_1$,
3. \exists has a winning strategy in $\text{EF}_k^-(\mathbf{D}_0, \mathbf{D}_1)$.

Proof. Step (1) to (2) is direct by Proposition 5.1.9. For (2) to (3) we give a winning strategy for \exists in $\text{EF}_k^-(\mathbf{D}_0, \mathbf{D}_1)$. We do it by showing the following claim

CLAIM 1. Let $\mathbf{D}_0 \sim_k^- \mathbf{D}_1$ and $\overline{\mathbf{s}}_i \in D_i^n$ be such that $n < k$ and $f_n : \overline{\mathbf{s}}_0 \mapsto \overline{\mathbf{s}}_1$ is a partial isomorphism; then \exists can survive one more round in $\text{EF}_k^-(\mathbf{D}_0, \mathbf{D}_1) @ (\overline{\mathbf{s}}_0, \overline{\mathbf{s}}_1)$.

PROOF OF CLAIM. Let \forall pick $d_i \in D_i$ such that the type of d_i is $T \subseteq A$. If d_i had already been played then \exists picks the same element as before and $f_{n+1} = f_n$. If d_i is new and $|T|_{\mathbf{D}_i} \geq k$ then, as at most $n < k$ elements have been played, there is always some new $d_{-i} \in D_{-i}$ that \exists can choose that matches d_i . If $|T|_{\mathbf{D}_i} = m < k$ then we know that $|T|_{\mathbf{D}_{-i}} = m$. Therefore, as d_i is new and f_n is injective, there must be a $d_{-i} \in D_{-i}$ that \exists can choose. \blacktriangleleft

Step (3) to (1) is a standard result [EF95, Corollary 2.2.9] which we prove anyway because we will need to extend it later. We prove the following loaded statement.

CLAIM 2. Let $\bar{\mathbf{s}}_i \in D_i^n$ and $\varphi(z_1, \dots, z_n) \in \text{FOE}_1(A)$ be such that $qr(\varphi) \leq k - n$. If \exists has a winning strategy in the game $\text{EF}_k^{\bar{=}}(\mathbf{D}_0, \mathbf{D}_1) @ (\bar{\mathbf{s}}_0, \bar{\mathbf{s}}_1)$ then $\mathbf{D}_0 \models \varphi(\bar{\mathbf{s}}_0)$ iff $\mathbf{D}_1 \models \varphi(\bar{\mathbf{s}}_1)$.

PROOF OF CLAIM. If φ is atomic the claim holds because of $f_n : \bar{\mathbf{s}}_0 \mapsto \bar{\mathbf{s}}_1$ being a partial isomorphism. Boolean cases are straightforward. Let $\varphi(z_1, \dots, z_n) = \exists x. \psi(z_1, \dots, z_n, x)$ and suppose $\mathbf{D}_0 \models \varphi(\bar{\mathbf{s}}_0)$. Hence, there exists $d_0 \in D_0$ such that $\mathbf{D}_0 \models \psi(\bar{\mathbf{s}}_0, d_0)$. By hypothesis we know that \exists has a winning strategy for $\text{EF}_k^{\bar{=}}(\mathbf{D}_0, \mathbf{D}_1) @ (\bar{\mathbf{s}}_0, \bar{\mathbf{s}}_1)$. Therefore, if \forall picks $d_0 \in D_0$ she can respond with some $d_1 \in D_1$ and has a winning strategy for $\text{EF}_k^{\bar{=}}(\mathbf{D}_0, \mathbf{D}_1) @ (\bar{\mathbf{s}}_0 \cdot d_0, \bar{\mathbf{s}}_1 \cdot d_1)$. By induction hypothesis, because $qr(\psi) \leq k - (n + 1)$, we have that $\mathbf{D}_0 \models \psi(\bar{\mathbf{s}}_0, d_0)$ iff $\mathbf{D}_1 \models \psi(\bar{\mathbf{s}}_1, d_1)$ and hence $\mathbf{D}_1 \models \exists x. \psi(\bar{\mathbf{s}}_1, x)$. The opposite direction is proved by a symmetric argument. \blacktriangleleft

Combining these claims finishes the proof of the lemma. \square

5.1.12. THEOREM. *Every $\psi \in \text{FOE}_1(A)$ is equivalent to a formula in basic form.*

Proof. Let $qr(\psi) = k$ and let $\llbracket \psi \rrbracket$ be the class of models satisfying ψ . As $\mathfrak{M}_1 / \equiv_k^{\text{FOE}}$ is the same as $\mathfrak{M}_1 / \sim_k^{\bar{=}}$ by Lemma 5.1.11, it is easy to see that ψ is equivalent to $\bigvee \{ \varphi_E^{\bar{=}} \mid E \in \llbracket \psi \rrbracket / \sim_k^{\bar{=}} \}$. Now it only remains to see that each $\varphi_E^{\bar{=}}$ is equivalent to $\nabla_{\text{FOE}}(\bar{\mathbf{T}}, \Pi)$ for some $\Pi \subseteq \wp A$ and $T_i \subseteq A$.

The crucial observation is that we will use $\bar{\mathbf{T}}$ and Π to give a specification of the types “element by element”. Let $\mathbf{D} \in E$ be a representative. Call $S_1, \dots, S_n \subseteq A$ to the types such that $|S_i|_{\mathbf{D}} = n_i < k$ and $S'_1, \dots, S'_m \subseteq A$ to those satisfying $|S'_j|_{\mathbf{D}} \geq k$. The size of the sequence $\bar{\mathbf{T}}$ is defined to be $(\sum_{i=1}^n n_i) + k \times m$ where $\bar{\mathbf{T}}$ contains exactly n_i occurrences of type S_i and k occurrences of each S'_j . On the other hand $\Pi = \{S'_1, \dots, S'_m\}$. It is straightforward to check that $\varphi_E^{\bar{=}}$ is equivalent to $\nabla_{\text{FOE}}(\bar{\mathbf{T}}, \Pi)$, however, the quantifier rank of the latter is only bounded by $k \times 2^{|A|} + 1$. \square

Normal form for FOE_1^∞

The logic FOE_1^∞ extends FOE_1 with the capacity to tear apart finite and infinite sets of elements. This is reflected in the normal form for FOE_1^∞ by adding extra constraints to the normal form of FOE_1 .

5.1.13. DEFINITION. We say that a formula $\varphi \in \text{FOE}_1^\infty(A)$ is in *basic form* if $\varphi = \bigvee \nabla_{\text{FOE}^\infty}(\bar{\mathbf{T}}, \Pi, \Sigma)$ where each disjunct is of the form

$$\nabla_{\text{FOE}^\infty}(\bar{\mathbf{T}}, \Pi, \Sigma) = \nabla_{\text{FOE}}(\bar{\mathbf{T}}, \Pi \cup \Sigma) \wedge \nabla_\infty(\Sigma)$$

where

$$\nabla_\infty(\Sigma) := \bigwedge_{S \in \Sigma} \exists^\infty y. \tau_S(y) \wedge \forall^\infty y. \bigvee_{S \in \Sigma} \tau_S(y)$$

for some set of types $\Pi, \Sigma \subseteq \wp A$ and each $T_i \subseteq A$.

Intuitively, the formula $\nabla_\infty(\Sigma)$ says that (1) for every type $S \in \Sigma$, there are infinitely many elements satisfying S and (2) only finitely many elements do not satisfy any type in Σ .

A short argument reveals that, intuitively, every disjunct expresses that each one-step model satisfying it admits a partition of its domain in three parts:

- (i) distinct elements t_1, \dots, t_n with type T_1, \dots, T_n ,
- (ii) finitely many elements whose types belong to Π , and
- (iii) for each $S \in \Sigma$, infinitely many elements with type S .

In the same way as before, we define a relation \sim_k^∞ which refines \sim_k^ω by adding information about the (in-)finiteness of the types.

5.1.14. DEFINITION. Let \mathbf{D} and \mathbf{D}' be one-step models. For every $k \in \mathbb{N}$, the relation $\mathbf{D} \sim_k^\infty \mathbf{D}'$ is defined as follows:

$$\begin{aligned} \mathbf{D} \sim_0^\infty \mathbf{D}' &\iff \text{always} \\ \mathbf{D} \sim_{k+1}^\infty \mathbf{D}' &\iff \mathbf{D} \sim_{k+1}^\omega \mathbf{D}' \text{ and} \\ &\quad \forall S \subseteq A \ (|S|_{\mathbf{D}}, |S|_{\mathbf{D}'} < \omega \text{ or } |S|_{\mathbf{D}}, |S|_{\mathbf{D}'} \geq \omega) \end{aligned}$$

5.1.15. PROPOSITION. *The following hold:*

- (i) \sim_k^∞ is an equivalence relation,
- (ii) \sim_k^∞ has finite index,
- (iii) \sim_k^∞ is a refinement of \sim_k^ω ,
- (iv) Every $E \in \mathfrak{M}_1 / \sim_k^\infty$ is characterized by a formula $\varphi_E^\infty \in \text{FOE}_1^\infty(A)$ with $qr(\varphi) = k$.

Proof. We only prove the last point, for $k > 0$. Let $E \in \mathfrak{M}_1 / \sim_k^\infty$ and let $\mathbf{D} \in E$ be a representative of the class. Let $E' \in \mathfrak{M}_1 / \sim_k^\omega$ be the equivalence class of \mathbf{D} with respect to \sim_k^ω . Let $S_1, \dots, S_n \subseteq A$ be all the types such that $|S_i|_{\mathbf{D}} \geq \omega$.

$$\varphi_E^\infty := \varphi_{E'}^\omega \wedge \nabla_\infty(\{S_1, \dots, S_n\}).$$

It is not difficult to see that $qr(\varphi_E^\infty) = k$ and that $\mathbf{D}' \models \varphi_E^\infty$ iff $\mathbf{D}' \in E$. \square

Now we give a notion of Ehrenfeucht-Fraïssé game for FOE_1^∞ . In this case the game extends EF_k^ω with a move for \exists^∞ .

5.1.16. DEFINITION. Let $\mathbf{D}_0 = (D_0, V_0)$ and $\mathbf{D}_1 = (D_1, V_1)$ be one-step models. We define the game $\text{EF}_k^\infty(\mathbf{D}_0, \mathbf{D}_1)$ between \forall and \exists . A position in this game is a pair of sequences $\overline{\mathbf{s}}_0 \in D_0^n$ and $\overline{\mathbf{s}}_1 \in D_1^n$ with $n \leq k$. The game consists of k rounds, where in round $n + 1$ the following steps are made. First \forall chooses to perform one of the following types of moves:

(a) Second-order move

1. \forall chooses an infinite set $X_i \subseteq D_i$,
2. \exists responds with an infinite set $X_{-i} \subseteq D_{-i}$,
3. \forall chooses an element $x_{-i} \in X_{-i}$,
4. \exists responds with an element $x_i \in X_i$.

(b) First-order move

1. \forall chooses an element $d_i \in D_i$,
2. \exists responds with an element $d_{-i} \in D_{-i}$.

Let $\overline{\mathbf{s}}_i \in D_i^n$ be the sequences of elements chosen up to round n , they are extended to $\overline{\mathbf{s}}_i' := \overline{\mathbf{s}}_i \cdot d_i$. \exists survives the round iff she does not get stuck and the function $f_{n+1} : \overline{\mathbf{s}}_0' \mapsto \overline{\mathbf{s}}_1'$ is a partial isomorphism of one-step models.

This game can be seen as an adaptation of the Ehrenfeucht-Fraïssé game for monotone generalized quantifiers found in [KV95] to the case of full monadic first-order logic.

5.1.17. LEMMA. *The following are equivalent:*

1. $\mathbf{D}_0 \equiv_k^{\text{FOE}^\infty} \mathbf{D}_1$,
2. $\mathbf{D}_0 \sim_k^\infty \mathbf{D}_1$,
3. \exists has a winning strategy in $\text{EF}_k^\infty(\mathbf{D}_0, \mathbf{D}_1)$.

Proof. Step (1) to (2) is direct by Proposition 5.1.15. For (2) to (3) we show

CLAIM 1. Let $\mathbf{D}_0 \sim_k^\infty \mathbf{D}_1$ and $\overline{\mathbf{s}}_i \in D_i^n$ be such that $n < k$ and $f_n : \overline{\mathbf{s}}_0 \mapsto \overline{\mathbf{s}}_1$ is a partial isomorphism; then \exists can survive one more round in $\text{EF}_k^\infty(\mathbf{D}_0, \mathbf{D}_1) @ (\overline{\mathbf{s}}_0, \overline{\mathbf{s}}_1)$.

PROOF OF CLAIM. We focus on the second-order moves because the first-order moves are the same as in the corresponding Claim of Lemma 5.1.11. Let \forall choose an infinite set $X_i \subseteq D_i$, we would like \exists to choose a set $X_{-i} \subseteq D_{-i}$ such that the following conditions hold:

- (a) The map f_n is a well-defined partial isomorphism between the restricted one-step models $\mathbf{D}_0|X_0$ and $\mathbf{D}_1|X_1$,

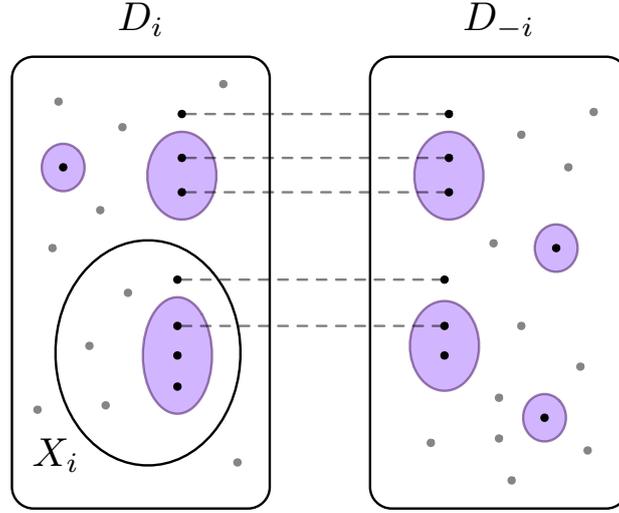


Figure 5.1: Elements of type S have coloured background.

- (b) For every type S we have that there is an element $d \in X_i$ of type S which is *not* connected by f_n iff there is such an element in X_{-i} ,
- (c) X_{-i} is infinite.

First we prove that such a set exists. To satisfy item (a) she just needs to add to X_{-i} the elements connected to X_i by f_n ; this is not a problem.

For item (b) we proceed as follows: for every type S such that there is an element $d \in X_i$ of type S , we add a new element $d' \in D_{-i}$ of type S to X_{-i} . To see that this is always possible, observe first that $\mathbf{D}_0 \sim_k^\infty \mathbf{D}_1$ implies $\mathbf{D}_0 \sim_k^- \mathbf{D}_1$. Using the properties of this relation, we divide in two cases:

- If $|S|_{D_i} \geq k$ we know that $|S|_{D_{-i}} \geq k$ as well. From the elements of D_{-i} of type S , at most $n < k$ are used by f_n . Hence, there is at least one $d' \in D_{-i}$ of type S to choose from.
- If $|S|_{D_i} < k$ we know that $|S|_{D_i} = |S|_{D_{-i}}$. From the elements of D_i of type S , at most $|S|_{D_i} - 1$ are used by f_n . The reason for the -1 is that we are assuming that we have just chosen a $d \in X_i$ which is not in f_n . Using that $|S|_{D_i} = |S|_{D_{-i}}$ and that f_n is a partial isomorphism we can again conclude that there is at least one $d' \in D_{-i}$ of type S to choose from.

For item (c) observe that as X_i is infinite but there are only finitely many types, there must be some S such that $|S|_{X_i} \geq \omega$. It is then safe to add infinitely many elements for S in X_{-i} while considering point (b). Moreover, the existence of infinitely many elements satisfying S in D_{-i} is guaranteed by $\mathbf{D}_0 \sim_k^\infty \mathbf{D}_1$.

Having shown that \exists can choose a set X_{-i} satisfying the above conditions, it is now clear that using point (b) \exists can survive the “first-order part” of the second-order move we were considering. This finishes the proof of the claim. \blacktriangleleft

Going back to the proof of Lemma 5.1.17, for step (3) to (1) we prove the following.

CLAIM 2. Let $\bar{\mathbf{s}}_i \in D_i^n$ and $\varphi(z_1, \dots, z_n) \in \text{FOE}_1^\infty(A)$ be such that $qr(\varphi) \leq k - n$. If \exists has a winning strategy in $\text{EF}_k^\infty(\mathbf{D}_0, \mathbf{D}_1) @ (\bar{\mathbf{s}}_0, \bar{\mathbf{s}}_1)$ then $\mathbf{D}_0 \models \varphi(\bar{\mathbf{s}}_0)$ iff $\mathbf{D}_1 \models \varphi(\bar{\mathbf{s}}_1)$.

PROOF OF CLAIM. All the cases involving operators of FOE_1 are the same as in Lemma 5.1.11. We prove the inductive case for the generalized quantifier. Let $\varphi(z_1, \dots, z_n)$ be of the form $\exists^\infty x. \psi(z_1, \dots, z_n, x)$ and let $\mathbf{D}_0 \models \varphi(\bar{\mathbf{s}}_0)$. Hence, there is an *infinite* set $X_0 \subseteq D_0$ such that

$$\mathbf{D}_0 \models \psi(\bar{\mathbf{s}}_0, x_0) \text{ if and only if } x_0 \in X_0. \quad (5.1)$$

By hypothesis we know that \exists has a winning strategy for $\text{EF}_k^\infty(\mathbf{D}_0, \mathbf{D}_1) @ (\bar{\mathbf{s}}_0, \bar{\mathbf{s}}_1)$. Therefore, if \forall plays a second-order move by picking $X_0 \subseteq D_0$ she can respond with some infinite set $X_1 \subseteq D_1$. We claim that $\mathbf{D}_1 \models \psi(\bar{\mathbf{s}}_1, x_1)$ for every $x_1 \in X_1$. First observe that if this holds then the set $X'_1 := \{d_1 \in D_1 \mid \mathbf{D}_1 \models \psi(\bar{\mathbf{s}}_1, d_1)\}$ must be infinite, and hence $\mathbf{D}_1 \models \exists^\infty x. \psi(\bar{\mathbf{s}}_1, x)$.

Assume, for a contradiction, that $\mathbf{D}_1 \not\models \psi(\bar{\mathbf{s}}_1, x'_1)$ for some $x'_1 \in X_1$. Let \forall play that x'_1 as the second part of his move. Then, as \exists has a winning strategy, she will respond with some $x'_0 \in X_0$ such that she has a winning strategy for $\text{EF}_k^\infty(\mathbf{D}_0, \mathbf{D}_1) @ (\bar{\mathbf{s}}_0 \cdot x'_0, \bar{\mathbf{s}}_1 \cdot x'_1)$. By induction hypothesis, as $qr(\psi) \leq k - (n + 1)$, this means that $\mathbf{D}_0 \models \psi(\bar{\mathbf{s}}_0, x'_0)$ iff $\mathbf{D}_1 \models \varphi(\bar{\mathbf{s}}_1, x'_1)$ which contradicts (5.1). The other direction is symmetric. \blacktriangleleft

Combining the claims finishes the proof of the lemma. \square

5.1.18. THEOREM. *Every formula $\varphi \in \text{FOE}_1^\infty(A)$ is equivalent to a formula in basic form.*

Proof. This can be proved using the same technique as in Theorem 5.1.12. Hence we only focus on showing that $\varphi_E^\infty \equiv \nabla_{\text{FOE}^\infty}(\bar{\mathbf{T}}, \Pi, \Sigma)$ for some $\Pi, \Sigma \subseteq \wp A$ and $T_i \subseteq A$. Recall that

$$\varphi_E^\infty = \varphi_{E'}^\infty \wedge \nabla_\infty(\{S_1, \dots, S_n\})$$

where $\{S_1, \dots, S_n\}$ are all the types that should be satisfied by infinitely many elements. Using Theorem 5.1.12 on $\varphi_{E'}^\infty$, we know that this is equivalent to

$$\varphi_E^\infty = \nabla_{\text{FOE}}(\bar{\mathbf{T}}', \Pi') \wedge \nabla_\infty(\{S_1, \dots, S_n\})$$

for some $\Pi' \subseteq \wp A$ and $T'_i \subseteq A$. Now separate Π' as $\Pi' = \Pi \uplus \Sigma$ where $\Sigma := \{S_1, \dots, S_n\}$ is composed of the infinite types and $\Pi := \Pi' \setminus \Sigma$ is composed of the finite types. After a minor rewriting, we get that

$$\varphi_E^\infty \equiv \nabla_{\text{FOE}}(\bar{\mathbf{T}}', \Pi \cup \Sigma) \wedge \nabla_\infty(\Sigma).$$

Therefore, we can conclude that $\varphi_E^\infty \equiv \nabla_{\text{FOE}^\infty}(\bar{\mathbf{T}}', \Pi, \Sigma)$. \square

The following stronger normal form will be useful in later chapters.

5.1.19. PROPOSITION. *For every formula in the basic form $\bigvee \nabla_{\text{FOE}^\infty}(\overline{\mathbf{T}}, \Pi, \Sigma)$ it is possible to assume, without loss of generality, that $\Sigma \subseteq \Pi$.*

Proof. This is direct from observing that $\nabla_{\text{FOE}^\infty}(\overline{\mathbf{T}}, \Pi, \Sigma)$ is equivalent to $\nabla_{\text{FOE}^\infty}(\overline{\mathbf{T}}, \Pi \cup \Sigma, \Sigma)$. To check it we just unravel the definitions and observe that $\nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi \cup \Sigma) \wedge \nabla_\infty(\Sigma)$ is equivalent to $\nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi \cup \Sigma \cup \Sigma) \wedge \nabla_\infty(\Sigma)$. \square

5.1.2 One-step monotonicity

Given a one-step logic $\mathcal{L}(A)$ and formula $\varphi \in \mathcal{L}(A)$. We say that φ is *monotone in $\{\overline{\mathbf{a}}\} \subseteq A$* if for every one step model (D, V) and assignment $g : \text{iVar} \rightarrow D$,

$$\text{if } (D, V), g \models \varphi \text{ and } V(\overline{\mathbf{a}}) \subseteq \overline{\mathbf{E}} \text{ then } (D, V[\overline{\mathbf{a}} \mapsto \overline{\mathbf{E}}]), g \models \varphi.$$

5.1.20. REMARK. It is easy to prove that a formula is monotone in $\{\overline{\mathbf{a}}\} \subseteq A$ iff it is monotone in every a_i . Therefore, in the following proofs we will, in general, consider monotonicity in every single a_i instead of in the full $\overline{\mathbf{a}}$. This is equivalent, and only done to avoid an even more complex notation.

Monotonicity is usually tightly related to positivity. If the quantifiers are well-behaved (i.e., monotone) then a formula φ will usually be monotone in $a \in A$ iff a has positive polarity in φ , that is, if it only occurs under an even number of negations. This is the case for all one-step logics considered in this dissertation. In this section we give a syntactic characterization of monotonicity for several one-step logics.

5.1.21. DEFINITION. Given $S \subseteq A$ and $A' \subseteq A$ we use the following notation

$$\tau_S^{A'}(x) := \bigwedge_{b \in S} b(x) \wedge \bigwedge_{b \in A \setminus (S \cup A')} \neg b(x),$$

for what we call the *A' -positive A -type $\tau_S^{A'}$* . Intuitively, $\tau_S^{A'}$ works almost like the A -type τ_S , but discarding the negative information for the names in A' . If $A' = \{a\}$ we write τ_S^a instead of $\tau_S^{\{a\}}$. Observe that with this notation, τ_S^+ is equivalent to τ_S^A .

Monotone fragment of FO_1

5.1.22. THEOREM. *A formula of $\text{FO}_1(A)$ is monotone in $A' \subseteq A$ iff it is equivalent to a sentence given by:*

$$\varphi ::= \psi \mid a(x) \mid \exists x.\varphi \mid \forall x.\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

where $a \in A'$, $\psi \in \text{FO}_1(A \setminus A')$. We denote this fragment as $\text{FO}_1\text{MON}_{A'}(A)$.

The result will follow from the following two lemmas and Remark 5.1.20.

5.1.23. LEMMA. *Every $\varphi \in \text{FO}_1\text{MON}_a(A)$ is monotone in a .*

Proof. The proof is a routine argument by induction on the complexity of φ . \square

Before going on we need to introduce a bit of new notation. In Section 5.1.1 we introduced the formula $\nabla_{\text{FO}}(\Sigma)$. We now give a few variants of this notation, which will be crucial to build the normal forms of the fragments discussed in this dissertation.

5.1.24. DEFINITION. Let $A' \subseteq A$ be a finite set of names. The A' -positive variant of $\nabla_{\text{FO}}(\Sigma)$ is given as follows:

$$\nabla_{\text{FO}}^{A'}(\Sigma) := \bigwedge_{S \in \Sigma} \exists x. \tau_S^{A'}(x) \wedge \forall x. \bigvee_{S \in \Sigma} \tau_S^{A'}(x).$$

We also introduce the following generalized forms of the above notation:

$$\nabla_{\text{FO}}^{A'}(\Sigma, \Pi) := \bigwedge_{S \in \Sigma} \exists x. \tau_S^{A'}(x) \wedge \forall x. \bigvee_{S \in \Pi} \tau_S^{A'}(x).$$

The *positive* variants of the above notations are defined as $\nabla_{\text{FO}}^+(\Sigma) := \nabla_{\text{FO}}^A(\Sigma)$ and $\nabla_{\text{FO}}^+(\Sigma, \Pi) := \nabla_{\text{FO}}^A(\Sigma, \Pi)$.

To prove that the fragment FO_1MON_a is complete for monotonicity in a , we need to show that every formula which is monotone in a is equivalent to some formula in FO_1MON_a . We prove a stronger result: we give a translation that constructively maps arbitrary formulas into FO_1MON_a . The interesting observation is that the translation will preserve truth iff the given formula is monotone in a .

5.1.25. LEMMA. *There exists a translation $(-)^{\circ} : \text{FO}_1(A) \rightarrow \text{FO}_1\text{MON}_a(A)$ such that a formula $\varphi \in \text{FO}_1(A)$ is monotone in $a \in A$ if and only if $\varphi \equiv \varphi^{\circ}$.*

Proof. To define the translation we assume, without loss of generality, that φ is in the normal form $\bigvee \nabla_{\text{FO}}(\Sigma)$ given in Definition 5.1.5, that is:

$$\nabla_{\text{FO}}(\Sigma) := \bigwedge_{S \in \Sigma} \exists x. \tau_S(x) \wedge \forall x. \bigvee_{S \in \Sigma} \tau_S(x).$$

We define the translation as $(\bigvee \nabla_{\text{FO}}(\Sigma))^{\circ} := \bigvee \nabla_{\text{FO}}^a(\Sigma)$.

From the construction it is clear that $\varphi^{\circ} \in \text{FO}_1\text{MON}_a(A)$ and therefore the right-to-left direction of the lemma is immediate by Lemma 5.1.23. For the left-to-right direction assume that φ is monotone in a , we have to prove that $(D, V) \models \varphi$ if and only if $(D, V) \models \varphi^{\circ}$.

\Rightarrow This direction is trivial.

\Leftarrow Assume $(D, V) \models \varphi^\circ$ and let Σ be such that $(D, V) \models \nabla_{\text{FO}}^a(\Sigma)$. Because of the universal part of $\nabla_{\text{FO}}^a(\Sigma)$, it is safe to assume that the *only* (a -positive) types realized in (D, V) are exactly those in Σ ; moreover, it is also safe to assume that every type has a (single) distinct witness (this is because (D, V) can be proved to be FO_1 -equivalent to such a model). For every $S \in \Sigma$, let d_S be the witness of the a -positive type $\tau_S^a(x)$ in (D, V) . Let $U := \{d_S \mid S \in \Sigma, a \notin S\}$ and $V' := V[a \mapsto V(a) \setminus U]$.

CLAIM 1. $(D, V') \models \nabla_{\text{FO}}(\Sigma)$.

PROOF OF CLAIM. First we show that the existential part of the normal form is satisfied. That is, that for every $S \in \Sigma$ we have a witness for the *full* type $\tau_S(x)$. If $a \in S$ the witness is given by φ° , that is, d_S . If $a \notin S$ then we specially crafted d_S to be a witness. The universal part is clearly satisfied. \blacktriangleleft

To finish observe that, by monotonicity of φ , we get $(D, V) \models \varphi$. \square

Putting together the above lemmas we obtain Theorem 5.1.22. Moreover, a careful analysis of the translation gives us the following corollary, providing normal forms for the monotone fragment of FO_1 .

5.1.26. COROLLARY. *Let $\varphi \in \text{FO}_1(A)$, the following hold:*

- (i) *The formula φ is monotone in $A' \subseteq A$ iff it is equivalent to a formula in the basic form $\bigvee \nabla_{\text{FO}}^{A'}(\Sigma)$ for some types $\Sigma \subseteq \wp A$.*
- (ii) *The formula φ is monotone in every $a \in A$ (i.e., $\varphi \in \text{FO}_1^+(A)$) iff φ is equivalent to a formula $\bigvee \nabla_{\text{FO}}^+(\Sigma)$ for some types $\Sigma \subseteq \wp A$.*

Monotone fragment of FOE_1

5.1.27. THEOREM. *A formula of $\text{FOE}_1(A)$ is monotone in $A' \subseteq A$ iff it is equivalent to a sentence given by:*

$$\varphi ::= \psi \mid a(x) \mid \exists x.\varphi \mid \forall x.\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

where $a \in A'$ and $\psi \in \text{FOE}_1(A \setminus A')$. We denote this fragment as $\text{FOE}_1\text{MON}_{A'}(A)$.

Observe that, in this definition, the equality predicate is taken into account by the ψ clause. Before going on we need to introduce a bit of new notation.

5.1.28. DEFINITION. Let $A' \subseteq A$ be a finite set of names. The monotone variant of $\nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)$ is given as follows:

$$\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi) := \exists \overline{\mathbf{x}}. (\text{diff}(\overline{\mathbf{x}}) \wedge \bigwedge_i \tau_{T_i}^{A'}(x_i) \wedge \forall z. (\text{diff}(\overline{\mathbf{x}}, z) \rightarrow \bigvee_{S \in \Pi} \tau_S^{A'}(z))).$$

When the set A' is a singleton $\{a\}$ we will write a instead of A' . The positive variant of $\nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)$ is defined as above but with $+$ in place of A' .

The result follows from the following lemma.

5.1.29. LEMMA. *The following hold:*

1. Every $\varphi \in \text{FOE}_1 \text{MON}_{A'}(A)$ is monotone in A' .
2. There exists a translation $(-)^{\circ} : \text{FOE}_1(A) \rightarrow \text{FOE}_1 \text{MON}_{A'}(A)$ such that a formula $\varphi \in \text{FOE}_1(A)$ is monotone in A' if and only if $\varphi \equiv \varphi^{\circ}$.

Proof. In Theorem 5.1.31 this result is proved for FOE_1^{∞} (i.e., FOE_1 extended with generalized quantifiers). It is not difficult to adapt the proof for FOE_1 . Intuitively, the translation is defined as $\varphi^{\circ} := \varphi[\neg a(x) \mapsto \top \mid a \in A']$ for φ in negation normal form. \square

Combining the normal form for FOE_1 and the above lemma, we obtain the following corollary providing a normal form for the monotone fragment of FOE_1 .

5.1.30. COROLLARY. *Given $\varphi \in \text{FOE}_1(A)$, the following hold:*

- (i) The formula φ is monotone in $A' \subseteq A$ iff it is equivalent to a formula in the basic form $\bigvee \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)$ where for each disjunct we have $\overline{\mathbf{T}} \in \wp(A)^k$ for some k and $\Pi \subseteq \overline{\mathbf{T}}$,
- (ii) The formula φ is monotone in all $a \in A$ (i.e., $\varphi \in \text{FOE}_1^+(A)$) iff it is equivalent to a formula in the basic form $\bigvee \nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)$ where for each disjunct we have $\overline{\mathbf{T}} \in \wp(A)^k$ for some k and $\Pi \subseteq \overline{\mathbf{T}}$.

Monotone fragment of FOE_1^{∞}

5.1.31. THEOREM. *A formula of $\text{FOE}_1^{\infty}(A)$ is monotone in $A' \subseteq A$ iff it is equivalent to a sentence given by:*

$$\varphi ::= \psi \mid a(x) \mid \exists x. \varphi \mid \forall x. \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \exists^{\infty} x. \varphi \mid \forall^{\infty} x. \varphi$$

where $a \in A'$ and $\psi \in \text{FOE}_1^{\infty}(A \setminus A')$. We call this fragment $\text{FOE}_1^{\infty} \text{MON}_{A'}(A)$.

Observe that $x \approx y$ and $x \not\approx y$ are included in the case $\psi \in \text{FOE}_1^{\infty}(A \setminus A')$. The result will follow from the following two lemmas and Remark 5.1.20.

5.1.32. LEMMA. *Every $\varphi \in \text{FOE}_1^\infty \text{MON}_a(A)$ is monotone in a .*

Proof. The proof is basically the same as Lemma 5.1.23. That is, we show by induction, that any one-step formula φ in the fragment (which may not be a sentence) satisfies, for every one-step model (D, V) and assignment $g : \text{iVar} \rightarrow D$,

$$\text{if } (D, V), g \models \varphi \text{ and } V(a) \subseteq E \text{ then } (D, V[a \mapsto E]), g \models \varphi.$$

We focus on the generalized quantifiers. Let $(D, V), g \models \varphi$ and $V(a) \subseteq E$.

- Case $\varphi = \exists^\infty x. \varphi'(x)$. By definition there exists an infinite set $I \subseteq D$ such that for all $d \in I$ we have $(D, V), g[x \mapsto d] \models \varphi'(x)$. By induction hypothesis $(D, V[a \mapsto E]), g[x \mapsto d] \models \varphi'(x)$ for all $d \in I$. Therefore $(D, V[a \mapsto E]), g \models \exists^\infty x. \varphi'(x)$.
- Case $\varphi = \forall^\infty x. \varphi'(x)$. Hence there exists $I \subseteq D$ such that for all $d \in I$ we have $(D, V), g[x \mapsto d] \models \varphi'(x)$ and $D \setminus I$ is *finite*. By induction hypothesis $(D, V[a \mapsto E]), g[x \mapsto d] \models \varphi'(x)$ for all $d \in I$. Therefore $(D, V[a \mapsto E]), g \models \forall^\infty x. \varphi'(x)$.

This finishes the proof. □

Before going on, we introduce some notation.

5.1.33. DEFINITION. Let $A' \subseteq A$ be a finite set of names. The A' -positive variant of $\nabla_{\text{FOE}^\infty}(\overline{\mathbf{T}}, \Pi, \Sigma)$ is given as follows:

$$\begin{aligned} \nabla_{\text{FOE}^\infty}^{A'}(\overline{\mathbf{T}}, \Pi, \Sigma) &:= \nabla_{\text{FOE}^\infty}^{A'}(\overline{\mathbf{T}}, \Pi \cup \Sigma) \wedge \nabla_\infty^{A'}(\Sigma) \\ \nabla_{\text{FOE}^\infty}^{A'}(\overline{\mathbf{T}}, \Lambda) &:= \exists \overline{\mathbf{x}}. (\text{diff}(\overline{\mathbf{x}}) \wedge \bigwedge_i \tau_{T_i}^{A'}(x_i) \wedge \forall z. (\text{diff}(\overline{\mathbf{x}}, z) \rightarrow \bigvee_{S \in \Lambda} \tau_S^{A'}(z))) \\ \nabla_\infty^{A'}(\Sigma) &:= \bigwedge_{S \in \Sigma} \exists^\infty y. \tau_S^{A'}(y) \wedge \forall^\infty y. \bigvee_{S \in \Sigma} \tau_S^{A'}(y). \end{aligned}$$

When the set A' is a singleton $\{a\}$ we will write a instead of A' . The positive variant of $\nabla_{\text{FOE}^\infty}(\overline{\mathbf{T}}, \Pi, \Sigma)$ is defined as $\nabla_{\text{FOE}^\infty}^+(\overline{\mathbf{T}}, \Pi, \Sigma) := \nabla_{\text{FOE}^\infty}^A(\overline{\mathbf{T}}, \Pi, \Sigma)$.

We are now ready to give the translation.

5.1.34. LEMMA. *There is a translation $(-)^{\circ} : \text{FOE}_1^\infty(A) \rightarrow \text{FOE}_1^\infty \text{MON}_a(A)$ such that a formula $\varphi \in \text{FOE}_1^\infty(A)$ is monotone in a if and only if $\varphi \equiv \varphi^{\circ}$.*

Proof. We assume that φ is in the normal form $\bigvee \nabla_{\text{FOE}^\infty}(\overline{\mathbf{T}}, \Pi, \Sigma)$ where

$$\nabla_{\text{FOE}^\infty}(\overline{\mathbf{T}}, \Pi, \Sigma) = \nabla_{\text{FOE}^\infty}(\overline{\mathbf{T}}, \Pi \cup \Sigma) \wedge \nabla_\infty(\Sigma).$$

for some sets of types $\Pi, \Sigma \subseteq \wp A$ and each $T_i \subseteq A$. For the translation we define

$$(\bigvee \nabla_{\text{FOE}^\infty}(\bar{\mathbf{T}}, \Pi, \Sigma))^\circ := \bigvee \nabla_{\text{FOE}^\infty}^a(\bar{\mathbf{T}}, \Pi, \Sigma).$$

From the construction it is clear that $\varphi^\circ \in \text{FOE}_1^\infty \text{MON}_a(A)$ and therefore the right-to-left direction of the lemma is immediate by Lemma 5.1.32. For the left-to-right direction assume that φ is monotone in a , we have to prove that $(D, V) \models \varphi$ if and only if $(D, V) \models \varphi^\circ$.

\Rightarrow This direction is trivial.

\Leftarrow Assume $(D, V) \models \varphi^\circ$, and in particular that $(D, V) \models \nabla_{\text{FOE}^\infty}^a(\bar{\mathbf{T}}, \Pi, \Sigma)$. Observe that the elements of D can be partitioned in the following way:

- (a) Distinct elements $t_i \in D$ such that each t_i satisfies $\tau_{T_i}^a(x)$,
- (b) Disjoint sets $\{D_S \subseteq D \mid S \in \Sigma\}$ such that each D_S is *infinite* and every $d \in D_S$ is a witness for the a -positive type $S \in \Sigma$,
- (c) A *finite* set $D_\Pi \subseteq D$ of witnesses of the a -positive types in Π .

Following this partition, every element $d \in D$ is be the witness of an a -type in either (a) $\bar{\mathbf{T}}$, (b) Σ , or (c) Π . We use $S_d \in \bar{\mathbf{T}} \cup \Pi \cup \Sigma$ to denote the a -type which d witnesses. Now, we are talking about a -types, there might be a slight difference between S_d and the actual type that each d has (namely $V^\natural(d)$). That is, it could be that $d \in V(a)$ but that $a \notin S_d$. What we want to do now is to shrink V in such a way that the witnessed (S_d) type and the actual type coincide. We give a new valuation U defined as $U^\natural(d) := S_d$.¹ Observe that $U(a) \subseteq V(a)$ and $U(b) = V(b)$ for $b \in A \setminus \{a\}$.

CLAIM 1. $(D, U) \models \varphi$.

PROOF OF CLAIM. First we check that $(D, U) \models \nabla_{\text{FOE}}(\bar{\mathbf{T}}, \Pi \cup \Sigma)$. It is easy to see that the elements t_i work as witnesses for the *full* types T_i . That is $(D, U) \models \tau_{T_i}(t_i)$ for every i . To prove the universal part of the formula it is enough to show that:

1. Every element $d \in D_\Pi$ realizes the full type $S_d \in \Pi$,
2. For all $S \in \Sigma$, every element of D_S realizes the full type S .

Let d be an element of either D_Π or any of the D_S . By (b) and (c) we know $(D, V) \models \tau_{S_d}^a(d)$. If $a \in S_d$ we can trivially conclude $(D, U) \models \tau_{S_d}(d)$. If $a \notin S_d$, by definition of U we know that $d \notin U(a)$ and hence we can also conclude that $(D, U) \models \tau_{S_d}(d)$.

To prove that $(D, U) \models \bigwedge_{S \in \Sigma} \exists^\infty y. \tau_S(y) \wedge \forall^\infty y. \bigvee_{S \in \Sigma} \tau_S(y)$ we only need to observe that the existential part is satisfied because each D_S is infinite by (c) and the universal part is satisfied because the set $D_\Pi \cup \bar{\mathbf{T}}$ is finite by (b). \blacktriangleleft

¹Recall that a valuation $U : A \rightarrow \wp D$ can also be represented as a marking $U^\natural : D \rightarrow \wp A$ given by $U^\natural(d) := \{a \in A \mid d \in V(a)\}$.

To finish the proof, note that by monotonicity of φ we get $(D, V) \models \varphi$. \square

Putting together the above lemmas we obtain Theorem 5.1.31. Moreover, a careful analysis of the translation gives us the following corollary, providing normal forms for the monotone fragment of FOE_1^∞ .

5.1.35. COROLLARY. *Let $\varphi \in \text{FOE}_1^\infty(A)$, the following hold:*

- (i) *The formula φ is monotone in $A' \subseteq A$ iff it is equivalent to a formula $\bigvee \nabla_{\text{FOE}^\infty}^{A'}(\overline{\mathbf{T}}, \Pi, \Sigma)$ for $\Sigma \subseteq \Pi \subseteq \wp A$ and $\overline{\mathbf{T}} \in \wp(A)^k$ for some k .*
- (ii) *The formula φ is monotone in every $a \in A$ (i.e., $\varphi \in \text{FOE}_1^{\infty+}(A)$) iff it is equivalent to a formula in the basic form $\bigvee \nabla_{\text{FOE}^\infty}^+(\overline{\mathbf{T}}, \Pi, \Sigma)$ for types $\Sigma \subseteq \Pi \subseteq \wp A$ and $\overline{\mathbf{T}} \in \wp(A)^k$ for some k .*

Proof. We only remark that to obtain $\Sigma \subseteq \Pi$ in the above normal forms it is enough to use Proposition 5.1.19 before applying the translation. \square

5.1.3 One-step continuity

Recall from Chapter 4 that a formula $\varphi \in \mathcal{L}(A)$ is continuous in $\{\bar{\mathbf{a}}\} \subseteq A$ if φ is monotone in $\bar{\mathbf{a}}$ and additionally, for every (D, V) and assignment $g : \text{iVar} \rightarrow D$,

$$\text{if } (D, V), g \models \varphi \text{ then } \exists \bar{\mathbf{U}} \subseteq_\omega V(\bar{\mathbf{a}}) \text{ such that } (D, V[\bar{\mathbf{a}} \mapsto \bar{\mathbf{U}}]), g \models \varphi.$$

5.1.36. REMARK. It was proved in Proposition 3.2.5 that continuity in the product coincides with continuity in every variable. Therefore, in the following proofs we will, in general, consider continuity in every single a_i instead of in the full $\bar{\mathbf{a}}$. This is equivalent, and only done to avoid an even more complex notation.

In this section we will characterize the continuous fragment of FO_1 and FOE_1^∞ but we will not characterize that of FOE_1 , since it is not used in this dissertation.

Continuous fragment of FO_1

5.1.37. THEOREM. *A formula of $\text{FO}_1(A)$ is continuous in $A' \subseteq A$ iff it is equivalent to a sentence given by:*

$$\varphi ::= \psi \mid a(x) \mid \exists x. \varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

where $a \in A'$ and $\psi \in \text{FO}_1(A \setminus A')$. We denote this fragment as $\text{FO}_1\text{CON}_{A'}(A)$.

The theorem will follow from the next two lemmas and Remark 5.1.36.

5.1.38. LEMMA. *Every $\varphi \in \text{FO}_1\text{CON}_a(A)$ is continuous in a .*

Proof. First observe that φ is monotone in a by Theorem 5.1.22. We show, by induction, that any one-step formula φ in the fragment (which may not be a sentence) satisfies, for every one-step model (D, V) , assignment $g : \text{iVar} \rightarrow D$,

if $(D, V), g \models \varphi$ then $\exists U \subseteq_{\omega} V(a)$ such that $(D, V[a \mapsto U]), g \models \varphi$.

- If $\varphi = \psi \in \text{FO}_1(A \setminus \{a\})$ changes in the a part of the valuation will make no difference and hence the condition is trivial.
- Case $\varphi = a(x)$: if $(D, V), g \models a(x)$ then $g(x) \in V(a)$. Clearly, $g(x) \in V[a \mapsto \{g(x)\}](a)$ and hence $(D, V[a \mapsto \{g(x)\}]), g \models a(x)$.
- Case $\varphi = \varphi_1 \vee \varphi_2$: assume $(D, V), g \models \varphi$. Without loss of generality we can assume that $(D, V), g \models \varphi_1$ and hence by induction hypothesis we have that there is $U \subseteq_{\omega} V(a)$ such that $(D, V[a \mapsto U]), g \models \varphi_1$ which clearly implies $(D, V[a \mapsto U]), g \models \varphi$.
- Case $\varphi = \varphi_1 \wedge \varphi_2$: assume $(D, V), g \models \varphi$. By induction hypothesis we have $U_1, U_2 \subseteq_{\omega} V(a)$ such that $(D, V[a \mapsto U_1]), g \models \varphi_1$ and $(D, V[a \mapsto U_2]), g \models \varphi_2$. By monotonicity this also holds with $V[a \mapsto U_1 \cup U_2]$ and therefore $(D, V[a \mapsto U_1 \cup U_2]), g \models \varphi$.
- Case $\varphi = \exists x. \varphi'(x)$ and $(D, V), g \models \varphi$. By definition there exists $d \in D$ such that $(D, V), g[x \mapsto d] \models \varphi'(x)$. By induction hypothesis there exists $U \subseteq_{\omega} V(a)$ such that $(D, V[a \mapsto U]), g[x \mapsto d] \models \varphi'(x)$ and hence $(D, V[a \mapsto U]), g \models \exists x. \varphi'(x)$.

This finishes the proof. □

5.1.39. LEMMA. *There is a translation $(-)^{\ominus} : \text{FO}_1\text{MON}_a(A) \rightarrow \text{FO}_1\text{CON}_a(A)$ such that a formula $\varphi \in \text{FO}_1\text{MON}_a(A)$ is continuous in a if and only if $\varphi \equiv \varphi^{\ominus}$.*

Proof. To define the translation we assume, without loss of generality, that φ is in the basic form $\bigvee \nabla_{\text{FO}}^a(\Sigma)$. For the translation, let $(\bigvee \nabla_{\text{FO}}^a(\Sigma))^{\ominus} := \bigvee \nabla_{\text{FO}}^a(\Sigma, \Sigma_a^-)$ where $\Sigma_a^- := \{S \in \Sigma \mid a \notin S\}$.

From the construction it is clear that $\varphi^{\ominus} \in \text{FO}_1\text{CON}_a(A)$ and therefore the right-to-left direction of the lemma is immediate by Lemma 5.1.38. For the left-to-right direction assume that φ is continuous in a , we have to prove that $(D, V) \models \varphi$ iff $(D, V) \models \varphi^{\ominus}$, for every one-step model (D, V) . We will take a slightly different but equivalent approach.

It is easy to prove that $(D, V) \equiv_{\text{FO}} (D \times \omega, V_{\pi})$ where $D \times \omega$ has countably many copies of each element in D and $V_{\pi}(a) := \{(d, k) \mid d \in V(a), k \in \omega\}$. Moreover, as φ is continuous in a there is $U \subseteq_{\omega} V_{\pi}(a)$ such that $V'_{\pi} := V[a \mapsto U]$ satisfies $(D \times \omega, V_{\pi}) \models \varphi$ iff $(D \times \omega, V'_{\pi}) \models \varphi$. Therefore, it is enough to prove that $(D \times \omega, V_{\pi}) \models \varphi$ iff $(D \times \omega, V'_{\pi}) \models \varphi^{\ominus}$.

$\boxed{\Rightarrow}$ Let $(D \times \omega, V'_\pi) \models \nabla_{\text{FO}}^a(\Sigma)$, we show that $(D \times \omega, V'_\pi) \models \nabla_{\text{FO}}^a(\Sigma, \Sigma_a^-)$. The existential part of $\nabla_{\text{FO}}^a(\Sigma, \Sigma_a^-)$ is trivially true. We have to show that every element of $(D \times \omega, V'_\pi)$ realizes an a -positive type in Σ_a^- . Take $(d, k) \in D \times \omega$ and let T be the (full) type of (d, k) . If $a \notin T$ then trivially $T \in \Sigma_a^-$ and we are done. Suppose $a \in T$. Observe that in $D \times \omega$ we have infinitely many copies of $d \in D$. However, as $V'_\pi(a)$ is finite, there must be some (d, k') with type $T' := T \setminus \{a\}$. For $\nabla_{\text{FO}}^a(\Sigma)$ to be true we must have $T' \in \Sigma$ and hence $T' \in \Sigma_a^-$. It is easy to see that (d, k) realizes the a -positive type T' .

$\boxed{\Leftarrow}$ Let $(D \times \omega, V'_\pi) \models \nabla_{\text{FO}}^a(\Sigma, \Sigma_a^-)$, we show that $(D \times \omega, V'_\pi) \models \nabla_{\text{FO}}^a(\Sigma)$. The existential part is trivial. For the universal part just observe that $\Sigma_a^- \subseteq \Sigma$. \square

Putting together the above lemmas we obtain Theorem 5.1.37. Moreover, a careful analysis of the translation gives us the following corollary, providing normal forms for the continuous fragment of FO_1 .

5.1.40. COROLLARY. *Let $\varphi \in \text{FO}_1(A)$, the following hold:*

- (i) *The formula φ is continuous in $a \in A$ iff it is equivalent to a formula $\bigvee \nabla_{\text{FO}}^a(\Sigma, \Sigma_a^-)$ for some types $\Sigma \subseteq \wp A$, where $\Sigma_a^- := \{S \in \Sigma \mid a \notin S\}$.*
- (ii) *If φ is monotone in A (i.e., $\varphi \in \text{FO}_1^+(A)$) then φ is continuous in $a \in A$ iff it is equivalent to a formula in the basic form $\bigvee \nabla_{\text{FO}}^+(\Sigma, \Sigma_a^-)$ for some types $\Sigma \subseteq \wp A$, where $\Sigma_a^- := \{S \in \Sigma \mid a \notin S\}$.*

Continuous fragment of FOE_1^∞

5.1.41. THEOREM. *A formula of $\text{FOE}_1^\infty(A)$ is continuous in $A' \subseteq A$ iff it is equivalent to a sentence given by:*

$$\varphi ::= \psi \mid a(x) \mid \exists x.\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{W}x.(\varphi, \psi)$$

where $a \in A'$ and $\psi \in \text{FOE}_1^\infty(A \setminus A')$. Recall from Definition 3.2.33 that $\mathbf{W}x.(\varphi, \psi)$ is defined as $\forall x.(\varphi(x) \vee \psi(x)) \wedge \forall^\infty x.\psi(x)$. We denote this fragment as $\text{FOE}_1^\infty \text{CON}_{A'}(A)$.

The theorem will follow from the next two lemmas and Remark 5.1.36.

5.1.42. LEMMA. *Every $\varphi \in \text{FOE}_1^\infty \text{CON}_a(A)$ is continuous in a .*

Proof. Observe that monotonicity of φ is guaranteed by Theorem 5.1.31. We show, by induction, that any formula of the fragment (which may not be a sentence) satisfies, for every one-step model (D, V) and assignment $g : i\text{Var} \rightarrow D$,

$$\text{if } (D, V), g \models \varphi \text{ then } \exists U \subseteq_\omega V(a) \text{ such that } (D, V[a \mapsto U]), g \models \varphi.$$

We focus on the inductive case of the new quantifier. Let $\varphi' = \mathbf{W}x.(\varphi, \psi)$, i.e.,

$$\varphi' = \forall x. \underbrace{(\varphi(x) \vee \psi(x))}_{\alpha(x)} \wedge \underbrace{\forall^\infty x. \psi(x)}_{\beta}.$$

Let $(D, V), g \models \varphi'$. By induction hypothesis, for every $g_d := g[x \mapsto d]$ which satisfies $(D, V), g_d \models \alpha(x)$ there is $U_d \subseteq_\omega V(a)$ such that $(D, V[a \mapsto U_d]), g_d \models \alpha(x)$. The crucial observation is that because of β , only finitely many elements of d make $\psi(d)$ false. Let $U := \bigcup \{U_d \mid (D, V), g_d \not\models \psi(x)\}$. Note that U is a finite union of finite sets, hence finite.

CLAIM 1. Let $V_U := V[a \mapsto U]$; then we have $(D, V_U), g \models \varphi'$.

PROOF OF CLAIM. It is clear that $(D, V_U), g \models \beta$ because ψ is a -free. To show that the first conjunct is true we have to show that $(D, V_U), g_d \models \varphi(x) \vee \psi(x)$ for every $d \in D$. We consider two cases: (i) if $(D, V), g_d \models \psi(x)$ we are done, again because ψ is a -free; (ii) if the former is not the case then $U_d \subseteq U$; moreover, we knew that $(D, V[a \mapsto U_d]), g_d \models \alpha(x)$ and by monotonicity of $\alpha(x)$ we can conclude that $(D, V_U), g_d \models \alpha(x)$. \blacktriangleleft

This finishes the proof of the lemma. \square

5.1.43. LEMMA. *There is a translation $(-)^{\ominus} : \text{FOE}_1^\infty \text{MON}_a(A) \rightarrow \text{FOE}_1^\infty \text{CON}_a(A)$ such that a formula $\varphi \in \text{FOE}_1^\infty \text{MON}_a(A)$ is continuous in a if and only if $\varphi \equiv \varphi^{\ominus}$.*

Proof. We assume that φ is in basic normal form, i.e., $\varphi = \bigvee \nabla_{\text{FOE}^\infty}^a(\overline{\mathbf{T}}, \Pi, \Sigma)$. For the translation let $(\bigvee \nabla_{\text{FOE}^\infty}^a(\overline{\mathbf{T}}, \Pi, \Sigma))^{\ominus} := \bigvee \nabla_{\text{FOE}^\infty}^a(\overline{\mathbf{T}}, \Pi, \Sigma)^{\ominus}$ where

$$\nabla_{\text{FOE}^\infty}^a(\overline{\mathbf{T}}, \Pi, \Sigma)^{\ominus} := \begin{cases} \perp & \text{if } a \in \bigcup \Sigma \\ \nabla_{\text{FOE}^\infty}^a(\overline{\mathbf{T}}, \Pi, \Sigma) & \text{otherwise.} \end{cases}$$

First we prove the right-to-left direction of the lemma. By Lemma 5.1.42 it is enough to show that $\varphi^{\ominus} \in \text{FOE}_1^\infty \text{CON}_a(A)$. We focus on the disjuncts of φ^{\ominus} . The interesting case is when $a \notin \bigcup \Sigma$. If we rearrange $\nabla_{\text{FOE}^\infty}^a(\overline{\mathbf{T}}, \Pi, \Sigma)$ and define the formulas φ', ψ as follows:

$$\begin{aligned} \nabla_{\text{FOE}^\infty}^a(\overline{\mathbf{T}}, \Pi, \Sigma) \equiv \exists \overline{\mathbf{x}}. & \left(\text{diff}(\overline{\mathbf{x}}) \wedge \bigwedge_i \tau_{T_i}^a(x_i) \wedge \right. \\ & \forall z. \underbrace{(\neg \text{diff}(\overline{\mathbf{x}}, z) \vee \bigvee_{S \in \Pi} \tau_S^a(z))}_{\varphi'(\overline{\mathbf{x}}, z)} \vee \underbrace{\bigvee_{S \in \Sigma} \tau_S^a(z)}_{\psi(z)} \Big) \wedge \\ & \forall^\infty y. \underbrace{\bigvee_{S \in \Sigma} \tau_S^a(y)}_{\psi(y)} \Big) \wedge \bigwedge_{S \in \Sigma} \exists^\infty y. \tau_S^a(y), \end{aligned}$$

then we get that

$$\nabla_{\text{FOE}^\infty}^a(\overline{\mathbf{T}}, \Pi, \Sigma) \equiv \exists \overline{\mathbf{x}}. \left(\text{diff}(\overline{\mathbf{x}}) \wedge \bigwedge_i \tau_{T_i}^a(x_i) \wedge \mathbf{W}z. (\varphi'(\overline{\mathbf{x}}, z), \psi(z)) \right) \wedge \bigwedge_{S \in \Sigma} \exists^\infty y. \tau_S^a(y)$$

which, because $a \notin \bigcup \Sigma$, is in the required fragment.

For the left-to-right direction of the lemma we have to prove that $\varphi \equiv \varphi^\ominus$.

$\boxed{\Leftarrow}$ Let $(D, V) \models \varphi^\ominus$. The only difference between φ and φ^\ominus is that some disjuncts may have been replaced by \perp . Therefore this direction is trivial.

$\boxed{\Rightarrow}$ Let $(D, V) \models \varphi$. Because φ is continuous in a we may assume that $V(a)$ is finite. Let $\nabla_{\text{FOE}^\infty}^a(\overline{\mathbf{T}}, \Pi, \Sigma)$ be a disjunct of φ such that $(D, V) \models \nabla_{\text{FOE}^\infty}^a(\overline{\mathbf{T}}, \Pi, \Sigma)$. If $a \notin \bigcup \Sigma$ we trivially conclude that $(D, V) \models \varphi^\ominus$ because the disjunct remains unchanged. Suppose now that $a \in \bigcup \Sigma$, then there must be some $S \in \Sigma$ with $a \in S$. Because $(D, V) \models \nabla_{\text{FOE}^\infty}^a(\overline{\mathbf{T}}, \Pi, \Sigma)$ we have, in particular, that $(D, V) \models \exists^\infty y. \tau_S^a(x)$ and hence $V(a)$ must be infinite which is absurd. \square

Putting together the above lemmas we obtain Theorem 5.1.41. Moreover, a careful analysis of the translation gives us the following corollary, providing normal forms for the continuous fragment of FOE_1^∞ .

5.1.44. COROLLARY. *Let $\varphi \in \text{FOE}_1^\infty(A)$, the following hold:*

- (i) *The formula φ is continuous in $a \in A$ iff it is equivalent to a formula in the basic form $\bigvee \nabla_{\text{FOE}^\infty}^a(\overline{\mathbf{T}}, \Pi, \Sigma)$ for some types $\Sigma \subseteq \Pi \subseteq \wp A$ and $T_i \subseteq A$ such that $a \notin \bigcup \Sigma$.*
- (ii) *If φ is monotone in every element of A (i.e., $\varphi \in \text{FOE}_1^{\infty+}(A)$) then φ is continuous in $a \in A$ iff it is equivalent to a formula in the basic form $\bigvee \nabla_{\text{FOE}^\infty}^+(\overline{\mathbf{T}}, \Pi, \Sigma)$ for some types $\Sigma \subseteq \Pi \subseteq \wp A$ and $T_i \subseteq A$ such that $a \notin \bigcup \Sigma$.*

Proof. We only remark that to obtain $\Sigma \subseteq \Pi$ in the above normal forms it is enough to use Proposition 5.1.19 before applying the translation. \square

5.1.4 One-step additivity

Recall from Chapter 4 that a formula $\varphi \in \mathcal{L}(A)$ is completely additive in $\{\overline{\mathbf{a}}\} \subseteq A$ if φ is monotone in every a_i and, for every (D, V) and assignment $g : \text{iVar} \rightarrow D$,

if $(D, V), g \models \varphi$ then $(D, V[\overline{\mathbf{a}} \mapsto \overline{\mathbf{Q}}]), g \models \varphi$ for some quasi-atom $\overline{\mathbf{Q}}$ of $V(\overline{\mathbf{a}})$.

We start by giving a characterization of the completely additive fragment of FOE_1 and then use it to give a similar characterization for FO_1 .

Completely additive fragment of FOE_1

5.1.45. THEOREM. *A formula of $\text{FOE}_1(A)$ is completely additive in $A' \subseteq A$ iff it is equivalent to a sentence given by:*

$$\varphi ::= \psi \mid a(x) \mid \exists x.\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \psi$$

where $a \in A'$ and $\psi \in \text{FOE}_1(A \setminus A')$. Observe that the equality is included in ψ . We denote this fragment as $\text{FOE}_1\text{ADD}_{A'}(A)$.

The theorem will follow from the next two lemmas.

5.1.46. LEMMA. *Every $\varphi \in \text{FOE}_1\text{ADD}_{A'}(A)$ is completely additive in A' .*

Proof. First observe that φ is monotone in every $a \in A'$ by Theorem 5.1.27. We show, by induction, that any formula φ in the fragment (which may not be a sentence) satisfies, for every one-step model (D, V) and assignment $g : i\text{Var} \rightarrow D$,

if $(D, V), g \models \varphi$ then $(D, V[A' \mapsto \overline{\mathbf{Q}}]), g \models \varphi$ for some quasi-atom $\overline{\mathbf{Q}}$ of $V(A')$.

The cases are as follows:

- If $\varphi = \psi \in \text{FO}_1(A \setminus A')$ changes in the A' -part of the valuation will make no difference and hence the condition is trivial.
- Case $\varphi = a_i(x)$ with $a_i \in A'$: if $(D, V), g \models a_i(x)$ then $g(x) \in V(a_i)$. If we take $\overline{\mathbf{Q}}$ to be an atom of $V(A')$ such that $Q_i := \{g(x)\}$ it is clear that $g(x) \in V[A' \mapsto \overline{\mathbf{Q}}](a_i)$ and hence $(D, V[A' \mapsto \overline{\mathbf{Q}}]), g \models a_i(x)$.
- Case $\varphi = \varphi_1 \vee \varphi_2$: this case is solved applying the inductive hypothesis to one of the disjuncts. Details are left to the reader.
- Case $\varphi = \varphi_1 \wedge \psi$: assume $(D, V), g \models \varphi$. By induction hypothesis we have that $(D, V[A' \mapsto \overline{\mathbf{Q}}]), g \models \varphi_1$ for some $\overline{\mathbf{Q}}$. Observe that $(D, V), g \models \psi$ and as ψ is A' -free we also have $(D, V[A' \mapsto \overline{\mathbf{Q}}]), g \models \psi$. Therefore we can conclude that $(D, V[A' \mapsto \overline{\mathbf{Q}}]), g \models \varphi$.
- Case $\varphi = \exists x.\varphi'$: assume $(D, V), g \models \varphi$. By definition there exists $d \in D$ such that $(D, V), g[x \mapsto d] \models \varphi'$. By induction hypothesis we have that $(D, V[A' \mapsto \overline{\mathbf{Q}}]), g[x \mapsto d] \models \varphi'$ for some $\overline{\mathbf{Q}}$. Therefore we can conclude that $(D, V[A' \mapsto \overline{\mathbf{Q}}]), g \models \exists x.\varphi'$.

This finishes the proof. □

5.1.47. LEMMA. *There is a translation $(-)^{\oplus} : \text{FOE}_1\text{MON}_{A'}(A) \rightarrow \text{FOE}_1\text{ADD}_{A'}(A)$ such that $\varphi \in \text{FOE}_1\text{MON}_{A'}(A)$ is completely additive in A' if and only if $\varphi \equiv \varphi^{\oplus}$.*

Proof. We assume that φ is in basic form, i.e., $\varphi = \bigvee \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)$ with $\Pi \subseteq \overline{\mathbf{T}}$. First, we intuitively consider some conditions on an arbitrary disjunct $\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)$ of φ that would force the existence of at least two elements, each satisfying $a(x)$ for some $a \in A'$. Clearly, any formula that forces this, goes against the spirit of complete additivity.

- (i) There are $a, b \in T_i \cap T_j$ for distinct $a, b \in A'$ or distinct i, j .
- (ii) There is some $S \in \Pi$ such that $S \cap A' \neq \emptyset$.

Now, we give a translation which eliminates (replaces with \perp) the subformulas forcing the above cases. We define $(\bigvee \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi))^\oplus := \bigvee \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)^\oplus$ and

$$\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)^\oplus := \begin{cases} \perp & \text{if (i) holds,} \\ \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi_{A'}^\times) & \text{otherwise,} \end{cases}$$

where $\Pi_{A'}^\times := \{S \in \Pi \mid A' \cap S = \emptyset\}$.

First we prove the right-to-left direction of the lemma. Inspecting the syntactic form of φ^\oplus it is not difficult to see that $\varphi^\oplus \in \text{FOE}_1\text{ADD}_{A'}(A)$, as given in Theorem 5.1.45. Using Lemma 5.1.46 we can conclude that φ^\oplus (and therefore φ as well) is completely additive in A' . For the left-to-right direction of the lemma we assume φ to be completely additive in A' and have to prove $\varphi \equiv \varphi^\oplus$.

$\boxed{\Leftarrow}$ Let $(D, V) \models \varphi^\oplus$. It is enough to show that $(D, V) \models \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi_{A'}^\times)$ implies $(D, V) \models \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)$ for every disjunct. The key observation is that $\Pi_{A'}^\times \subseteq \Pi$.

$\boxed{\Rightarrow}$ Let $(D, V) \models \varphi$. By complete additivity in A' we have that $(D, V[A' \mapsto \overline{\mathbf{Q}}]) \models \varphi$ for some quasi-atom $\overline{\mathbf{Q}}$ of $V(A')$. To improve readability we define $V' := V[A' \mapsto \overline{\mathbf{Q}}]$. We now work with (D, V') because (by monotonicity, which is implied by complete additivity) it will be enough to prove that $(D, V') \models \varphi^\oplus$.

As $(D, V') \models \varphi$, we know there is some disjunct $\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)$ of φ witnessing the satisfaction. We fix it and prove the following claim.

CLAIM 2. For every $b, b' \in A'$, if $b \in T_i$ and $b' \in T_j$ then $b = b'$ and $i = j$.

PROOF OF CLAIM. Suppose that there are distinct $T_i, T_j \in \overline{\mathbf{T}}$ such that $b \in T_i \cap T_j$. This would require at least two *distinct* elements to satisfy $b(x)$. However, this cannot occur because $V'(A')$ is a quasi-atom. The case where $b \neq b'$ is handled in a similar way: suppose that $b \in T_i$, $b' \in T_j$ and $b \neq b'$. Using what we just proved, let us assume that $i = j$. Therefore, this requires the existence of an element which is colored with both b and b' . However, as $V'(A')$ is a quasi-atom, this cannot occur if $b \neq b'$. \blacktriangleleft

To finish, we show that condition (ii) is taken care of.

CLAIM 3. If $(D, V') \models \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)$ then $(D, V') \models \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi_{A'}^\times)$.

PROOF OF CLAIM. Assume $(D, V') \models \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)$ and that $d \in D$ is one of the elements which is not a witness for $\overline{\mathbf{T}}$; therefore, d has to satisfy some type $S_d \in \Pi$. If $S_d \cap A' = \emptyset$ we are done, because in that case $S_d \in \Pi_{A'}^\times$. Suppose that $S_d \cap A' \neq \emptyset$, this means that d is colored with some $a \in A'$. As $V'(A')$ is a quasi-atom, this means that no other element can be colored with A' . The final observation is that, as $\Pi \subseteq \overline{\mathbf{T}}$, then $S_d \in \overline{\mathbf{T}}$. This means that there should exist an element $d' \neq d$ which is colored with the same $a \in A'$ but we have just observed that this cannot occur. We conclude that every $d \in D$ has to satisfy some type $S \in \Pi$ with $S \cap A' = \emptyset$. ◀

This finishes the proof. ◻

Putting together the above lemmas we obtain Theorem 5.1.45. Moreover, a careful analysis of the translation gives us the following corollary, providing normal forms for the completely additive fragment of FOE_1 .

5.1.48. COROLLARY. *Let $\varphi \in \text{FOE}_1(A)$, the following hold:*

- (i) *The formula φ is completely additive in $A' \subseteq A$ iff it is equivalent to a formula in the basic form $\bigvee \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)$ where $\overline{\mathbf{T}} \in \wp(A)^k$ for some k , $\Pi \subseteq \overline{\mathbf{T}}$ and for every disjunct: Π is A' -free and there is at most one element of A' in the concatenation of the lists $T_1 \cdot T_2 \cdot \dots \cdot T_k$.*
- (ii) *If φ is monotone in A (i.e., $\varphi \in \text{FOE}_1^+(A)$) then φ is completely additive in $A' \subseteq A$ iff it is equivalent to a formula $\bigvee \nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)$ where $\overline{\mathbf{T}} \in \wp(A)^k$ for some k , $\Pi \subseteq \overline{\mathbf{T}}$ and for every disjunct: Π is A' -free and there is at most one element of A' in the concatenation of the lists $T_1 \cdot T_2 \cdot \dots \cdot T_k$.*

Completely additive fragment of FO_1

5.1.49. THEOREM. *A formula of $\text{FO}_1(A)$ is completely additive in $A' \subseteq A$ iff it is equivalent to a sentence generated by the following grammar:*

$$\varphi ::= \psi \mid a(x) \mid \exists x.\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \psi$$

where $a \in A'$ and $\psi \in \text{FO}_1(A \setminus A')$. We denote this fragment as $\text{FO}_1\text{ADD}_{A'}(A)$.

The theorem will follow from the next two lemmas.

5.1.50. LEMMA. *Every $\varphi \in \text{FO}_1\text{ADD}_{A'}(A)$ is completely additive in A' .*

Proof. As $\text{FO}_1\text{ADD}_{A'}(A)$ is included in $\text{FOE}_1\text{ADD}_{A'}(A)$ we can simply use Lemma 5.1.46 for $\text{FOE}_1\text{ADD}_{A'}(A)$ and conclude what we want. ◻

In Lemma 5.1.47 we gave a translation which transforms every formula φ of $\text{FOE}_1(A)$ to an equivalent formula in $\text{FOE}_1\text{ADD}_{A'}(A)$, given that φ is completely additive in A' . As $\text{FO}_1(A) \subseteq \text{FOE}_1(A)$, it is tempting to use this same translation to obtain a characterization for $\text{FO}_1(A)$. The main problem is that the target of this translation always uses *equality* (in the predicate ‘diff’) and therefore would not work for FOE_1 . The next definition and proposition show that we can still make use of this translation but on top of that we need to forget (erase) the equality constraints. If we do that, we will obtain a normal form for the completely additive fragment of $\text{FO}_1(A)$.

5.1.51. DEFINITION. The translation $(-)^{\bullet} : \text{FOE}_1(A) \rightarrow \text{FO}_1(A)$ on formulas of $\text{FOE}_1(A)$ which are in normal form, is defined as follows:

$$(\nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi))^{\bullet} := \nabla_{\text{FO}}(\overline{\mathbf{T}}, \Pi)$$

and for $\alpha = \bigvee_i \psi_i$ we define $\alpha^{\bullet} := \bigvee_i \psi_i^{\bullet}$. We extend this translation to the monotone and positive fragments as expected, i.e.: $(\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi))^{\bullet} := \nabla_{\text{FO}}^{A'}(\overline{\mathbf{T}}, \Pi)$ and $(\nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi))^{\bullet} := \nabla_{\text{FO}}^+(\overline{\mathbf{T}}, \Pi)$.

Observe that on the right hand side of the above definitions, $\overline{\mathbf{T}}$ is seen as a set. The key property of this translation is the following.

5.1.52. PROPOSITION. *For every one-step model (D, V) and every $\alpha \in \text{FOE}_1(A)$ in normal form we have:*

$$(D, V) \models \alpha^{\bullet} \quad \text{iff} \quad (D \times \omega, V_{\pi}) \models \alpha,$$

where the valuation V_{π} is given by $V_{\pi}^{\natural}((d, k)) := V^{\natural}(d)$.

We call these one-step models \mathbf{D} and \mathbf{D}_{ω} respectively. Observe that the model \mathbf{D}_{ω} has ω -many copies of each element of \mathbf{D} .

Proof. We will prove that $\mathbf{D} \models \nabla_{\text{FO}}(\overline{\mathbf{T}}, \Pi)$ iff $\mathbf{D}_{\omega} \models \nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)$. The cases of the monotone and positive fragments are proved in the same way.

\Rightarrow Let $\mathbf{D} \models \nabla_{\text{FO}}(\overline{\mathbf{T}}, \Pi)$, we prove that $\mathbf{D}_{\omega} \models \nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)$. The existential part (i.e., $\overline{\mathbf{T}}$) is straightforward, by observing that in \mathbf{D}_{ω} we can choose as many distinct witnesses for each T_i as we want, because of the ω -expansion. For the universal part, observe that $\nabla_{\text{FO}}(\overline{\mathbf{T}}, \Pi)$ states that *every* $d \in D$ satisfies some type in Π . Therefore, the same happens with the elements of \mathbf{D}_{ω} . In particular, for the elements that are not witnesses for $\overline{\mathbf{T}}$. Therefore, $\mathbf{D}_{\omega} \models \nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)$.

\Leftarrow Let $\mathbf{D}_{\omega} \models \nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)$, we prove that $\mathbf{D} \models \nabla_{\text{FO}}(\overline{\mathbf{T}}, \Pi)$. For the existential part, consider an arbitrary T_i , we show that it has a witness in \mathbf{D} . We know by hypothesis that there is some $(d, k) \in \mathbf{D}_{\omega}$ which is a witness for T_i . It is easy to see that d works as a witness for T_i in \mathbf{D} . For the universal part, consider an element

$d \in D$, we show that it satisfies some type in Π . If there is some $(d, k) \in \mathbf{D}_\omega$ such that (d, k) is not a witness of $\overline{\mathbf{T}}$ then we are done, as it should satisfy some type in Π by the semantics of $\nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)$. The key observation is that there is always such an element, because at most $|\overline{\mathbf{T}}|$ elements of $\{(d, n) \mid n \in \mathbb{N}\}$ function as witnesses for $\overline{\mathbf{T}}$. \square

We are now ready to state the lemma which provides a translation for FO_1 .

5.1.53. LEMMA. *There is a translation $(-)^{\oplus} : \text{FO}_1\text{MON}_{A'}(A) \rightarrow \text{FO}_1\text{ADD}_{A'}(A)$ such that $\varphi \in \text{FO}_1\text{MON}_{A'}(A)$ is completely additive in A' if and only if $\varphi \equiv \varphi^{\oplus}$.*

Proof. To define the translation $(-)^{\oplus} : \text{FO}_1\text{MON}_{A'}(A) \rightarrow \text{FO}_1\text{ADD}_{A'}(A)$ we will use the translation for FOE_1 given in Lemma 5.1.47. To avoid confusion, we call it $(-)_{\text{FOE}}^{\oplus} : \text{FOE}_1\text{MON}_{A'}(A) \rightarrow \text{FOE}_1\text{ADD}_{A'}(A)$. We define, for every formula $\varphi \in \text{FO}_1\text{MON}_{A'}(A)$:

$$\varphi^{\oplus} := (\varphi_{\text{FOE}}^{\oplus})^{\bullet}.$$

$\boxed{\Leftarrow}$ A short argument reveals that indeed $\varphi^{\oplus} \in \text{FO}_1\text{ADD}_{A'}(A)$, and therefore by Lemma 5.1.50 the formula φ^{\oplus} is completely additive in A' . As φ is equivalent to φ^{\oplus} , it is also completely additive in A' .

$\boxed{\Rightarrow}$ For this direction we assume that $\varphi \in \text{FO}_1\text{MON}_{A'}(A)$ is completely additive in A' and we prove that for every (D, V) we have $(D, V) \models \varphi$ iff $(D, V) \models \varphi^{\oplus}$.

$$\begin{aligned} (D, V) \models \varphi & \text{ iff } (D \times \omega, V_\pi) \models \varphi && \text{(Properties of FO}_1\text{)} \\ & \text{ iff } (D \times \omega, V_\pi) \models \varphi_{\text{FOE}}^{\oplus} && \text{(Lemma 5.1.47)} \\ & \text{ iff } (D, V) \models (\varphi_{\text{FOE}}^{\oplus})^{\bullet} && \text{(Proposition 5.1.52)} \\ & \text{ iff } (D, V) \models \varphi^{\oplus}. && \text{(Definition of } \varphi^{\oplus}\text{)} \end{aligned}$$

This finishes the proof. \square

Putting together the above lemmas we obtain Theorem 5.1.49. Moreover, a careful analysis of the translation gives us normal forms for the completely additive fragment of FO_1 .

5.1.54. COROLLARY. *Let $\varphi \in \text{FO}_1(A)$ and, given $\Sigma \subseteq \wp(A)$, let $L_\Sigma \in A^*$ be the list with repetitions of elements of A in Σ . The following hold:*

- (i) *The formula φ is completely additive in $A' \subseteq A$ iff it is equivalent to a formula in the basic form $\bigvee \nabla_{\text{FO}}^{A'}(\Sigma, \Pi)$ where $\Sigma \subseteq \wp(A)$ and for every disjoint: Π is A' -free and there is at most one element of A' in L_Σ .*
- (ii) *If φ is monotone in A (i.e., $\varphi \in \text{FO}_1^+(A)$) then φ is completely additive in $A' \subseteq A$ iff it is equivalent to a formula of the form $\bigvee \nabla_{\text{FO}}^+(\Sigma, \Pi)$ where $\Sigma \subseteq \wp(A)$ and for every disjoint: Π is A' -free and there is at most one element of A' in L_Σ .*

5.1.5 Dual fragments

In this section we give syntactic characterizations of the co-continuous and completely multiplicative fragments of several one-step logics. These notions are dual to continuity and complete additivity, respectively. We first give a concrete definition of the dualization operator of Definition 2.3.5.

5.1.55. DEFINITION. The *dual* $\varphi^\delta \in \text{FOE}_1^\infty(A)$ of $\varphi \in \text{FOE}_1^\infty(A)$ is given by:

$$\begin{array}{ll}
(a(x))^\delta := a(x) & (\neg a(x))^\delta := \neg a(x) \\
(\top)^\delta := \perp & (\perp)^\delta := \top \\
(x \approx y)^\delta := x \not\approx y & (x \not\approx y)^\delta := x \approx y \\
(\varphi \wedge \psi)^\delta := \varphi^\delta \vee \psi^\delta & (\varphi \vee \psi)^\delta := \varphi^\delta \wedge \psi^\delta \\
(\exists x.\psi)^\delta := \forall x.\psi^\delta & (\forall x.\psi)^\delta := \exists x.\psi^\delta \\
(\exists^\infty x.\psi)^\delta := \forall^\infty x.\psi^\delta & (\forall^\infty x.\psi)^\delta := \exists^\infty x.\psi^\delta
\end{array}$$

5.1.56. REMARK. Observe that if $\varphi \in \mathcal{L}(A)$ for $\mathcal{L} \in \{\text{FO}_1, \text{FOE}_1, \text{FOE}_1^\infty\}$ then $\varphi^\delta \in \mathcal{L}(A)$. Moreover, the operator preserves positivity of the predicates, that is, if $\varphi \in \mathcal{L}^+(A)$ then $\varphi^\delta \in \mathcal{L}^+(A)$.

The proof of the following proposition is a routine check.

5.1.57. PROPOSITION. *For every $\varphi \in \text{FOE}_1^\infty(A)$, φ and φ^δ are Boolean duals.*

We are now ready to give the syntactic definition of the dual fragments for the one-step logics into consideration.

5.1.58. DEFINITION. The fragment $\text{FOE}_1^\infty \overline{\text{CON}}_{A'}(A)$ is given by the sentences generated by:

$$\varphi ::= \psi \mid a(x) \mid \forall x.\varphi \mid \forall^\infty x.\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi$$

where $a \in A'$ and $\psi \in \text{FOE}_1^\infty(A \setminus A')$. Observe that the equality is included in ψ . The fragment $\text{FO}_1 \overline{\text{CON}}_{A'}(A)$ is defined as $\text{FOE}_1^\infty \overline{\text{CON}}_{A'}(A)$ but without the clause for \forall^∞ and with $\psi \in \text{FO}_1(A \setminus A')$.

The fragment $\text{FOE}_1 \text{MUL}_{A'}(A)$ is given by the sentences generated by:

$$\varphi ::= \psi \mid a(x) \mid \forall x.\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi$$

where $a \in A'$ and $\psi \in \text{FOE}_1(A \setminus A')$. Again, the equality is included in ψ . The fragment $\text{FO}_1 \text{MUL}_{A'}(A)$ is defined as $\text{FOE}_1 \text{MUL}_{A'}(A)$ but with $\psi \in \text{FO}_1(A \setminus A')$.

The following proposition states that the above fragments are actually the duals of the fragments defined earlier in this chapter.

5.1.59. PROPOSITION. *The following hold:*

$$\begin{aligned}\text{FOE}_1^\infty \overline{\text{CON}}_{A'}(A) &= \{\varphi \mid \varphi^\delta \in \text{FOE}_1^\infty \text{CON}_{A'}(A)\} \\ \text{FO}_1 \overline{\text{CON}}_{A'}(A) &= \{\varphi \mid \varphi^\delta \in \text{FO}_1 \text{CON}_{A'}(A)\} \\ \text{FO}_1 \text{MUL}_{A'}(A) &= \{\varphi \mid \varphi^\delta \in \text{FO}_1 \text{ADD}_{A'}(A)\} \\ \text{FOE}_1 \text{MUL}_{A'}(A) &= \{\varphi \mid \varphi^\delta \in \text{FOE}_1 \text{ADD}_{A'}(A)\}.\end{aligned}$$

Proof. Easily proved by induction. \square

As a corollary, we get a characterization for co-continuity and multiplicativity.

5.1.60. COROLLARY.

- (i) *Let $\mathcal{L} \in \{\text{FO}_1, \text{FOE}_1^\infty\}$. A formula $\varphi \in \mathcal{L}(A)$ is co-continuous in $a \in A$ if and only if it is equivalent to some $\varphi' \in \mathcal{L} \overline{\text{CON}}_a(A)$.*
- (ii) *Let $\mathcal{L} \in \{\text{FO}_1, \text{FOE}_1\}$. A formula $\varphi \in \mathcal{L}(A)$ is completely multiplicative in $A' \subseteq A$ if and only if it is equivalent to some $\varphi' \in \mathcal{L} \text{MUL}_{A'}(A)$.*

Proof. Consequence of Proposition 5.1.59 and 5.1.57. \square

5.2 Selected multi-sorted first-order languages

In this section we give normal forms and characterizations for a few selected multi-sorted one-step languages. These results will be generalizations of the analogous results for the single-sorted scenario.

5.2.1. DEFINITION. The set $\text{FOE}_1(A, \mathcal{S})$ of (multi-sorted) one-step first-order sentences (with equality) is given by the sentences formed by

$$\varphi ::= \top \mid \perp \mid a(x) \mid x \approx y \mid \neg\varphi \mid \varphi \vee \varphi \mid \exists x:\mathbf{s}.\varphi$$

where $x, y \in \text{iVar}$, $a \in A$ and $\mathbf{s} \in \mathcal{S}$. The one-step logic $\text{FO}_1(A, \mathcal{S})$ of multi-sorted first-order sentences without equality is defined similarly.

Recall from Section 2.3 that formulas of an arbitrary multi-sorted one-step language \mathcal{L} are interpreted over multi-sorted one-step models, that is, a tuple $\mathbf{D} = (D_{\mathbf{s}_1}, \dots, D_{\mathbf{s}_n}, V : A \rightarrow \wp(\bigcup_{\mathbf{s}} D_{\mathbf{s}}))$ where the \mathbf{s}_i belong to a set of sorts \mathcal{S} and we use D to denote $\bigcup_{\mathbf{s}} D_{\mathbf{s}}$. Whenever we say ‘one-step model’ in this section we will be referring to a *multi-sorted* one-step model. A one-step model is called strict when the sets $D_{\mathbf{s} \in \mathcal{S}}$ are pairwise disjoint, that is, when $D_{\mathbf{s}_1}, \dots, D_{\mathbf{s}_n}$ is a partition of D . When the sets $D_{\mathbf{s} \in \mathcal{S}}$ are not relevant we will just write the one-step model as (D, V) . The class of all one-step models will be denoted by \mathfrak{M}_1 and the class of all strict one-step models will be denoted by \mathfrak{M}_1^s .

The multi-sorted semantics that we will use in this dissertation is slightly non-standard: for example, the individual variables and names (predicates) do not have a fixed sort. We define the semantics formally to avoid confusion.

5.2.2. DEFINITION. Let $\varphi \in \text{FOE}_1(A, \mathcal{S})$ be a formula, $\mathbf{D} = (D_{s_1}, \dots, D_{s_n}, V)$ be a one-step model on sorts \mathcal{S} and $g : i\text{Var} \rightarrow \wp(D)$ be an assignment. The semantics of $\text{FOE}_1(A, \mathcal{S})$ is given as follows:

$$\begin{aligned} \mathbf{D}, g \models a(x) & \text{ iff } g(x) \in V(a), \\ \mathbf{D}, g \models x \approx y & \text{ iff } g(x) = g(y), \\ \mathbf{D}, g \models \exists x:\mathbf{s}.\varphi & \text{ iff } \mathbf{D}, g[x \mapsto d] \models \varphi \text{ for some } d \in D_{\mathbf{s}}, \end{aligned}$$

while the Boolean connectives are defined as expected.

Given a one-step model \mathbf{D} we use $|S|_{\mathbf{D}}^{\mathbf{s}}$ to denote the number of elements of sort $s \in \mathcal{S}$ that realize the A -type τ_S in \mathbf{D} . Formally, $|S|_{\mathbf{D}}^{\mathbf{s}} := |\{d \in D_s : \mathbf{D} \models \tau_S(d)\}|$.

A *partial isomorphism* between two one-step models $\mathbf{D} = (D_{s_1}, \dots, D_{s_n}, V)$ and $\mathbf{D}' = (D'_{s_1}, \dots, D'_{s_n}, V')$ is a *partial* function $f : D \rightarrow D'$ which is injective and for all $d \in \text{Dom}(f)$ it satisfies the following conditions:

(sorts) d and $f(d)$ have the same sorts,

(atom) $d \in V(a) \Leftrightarrow f(d) \in V'(a)$, for all $a \in A$.

Given two sequences $\bar{\mathbf{d}} \in D^k$ and $\bar{\mathbf{d}}' \in D'^k$ we use $f : \bar{\mathbf{d}} \mapsto \bar{\mathbf{d}}'$ to denote the partial function $f : D \rightarrow D'$ defined as $f(d_i) := d'_i$. We explicitly avoid cases where there exist d_i, d_j such that $d_i = d_j$ but $d'_i \neq d'_j$.

5.2.1 Normal forms

In this section we provide normal forms for the multi-sorted one-step logics $\text{FO}_1(A, \mathcal{S})$ and $\text{FOE}_1(A, \mathcal{S})$.

Normal form for FO_1

We start by stating a normal form for one-step first-order logic without equality. A formula in *basic form* gives a complete description of the types that are satisfied in a one-step model.

5.2.3. DEFINITION. We say that a formula $\varphi \in \text{FO}_1(A, \mathcal{S})$ is in *basic form* if $\varphi = \bigvee \bigwedge_{\mathbf{s}} \nabla_{\text{FO}}(\Sigma)_{\mathbf{s}}$ where in each conjunct

$$\nabla_{\text{FO}}(\Sigma)_{\mathbf{s}} := \bigwedge_{S \in \Sigma} \exists x:\mathbf{s}.\tau_S(x) \wedge \forall x:\mathbf{s}.\bigvee_{S \in \Sigma} \tau_S(x)$$

for some set of types $\Sigma \subseteq \wp(A)$.

The subindex $\mathbf{s} \in \mathcal{S}$ in $\nabla_{\text{FO}}(\Sigma)_{\mathbf{s}}$ denotes that this formula describes the elements of sort \mathbf{s} of the one-step model. Therefore, a multi-sorted formula $\varphi \in \text{FO}_1(A, \mathcal{S})$ in basic form is comprised of many disjuncts, each one having a description of the elements of each sort.

5.2.4. REMARK. FO_1 cannot distinguish between arbitrary and strict one-step models. More formally, every arbitrary one-step model $\mathbf{D} = (D_1, \dots, D_n, V)$ is equivalent (for FO_1) to the model $\mathbf{D}^n := (D_1 \times \{1\}, \dots, D_n \times \{n\}, V_\pi)$ where $V_\pi(a) := \{(d, k) \mid d \in V(a), k \in \{0, \dots, n\}\}$. Therefore, when proving results for FO_1 , it is not difficult to see that we can restrict to the class of strict models.

It is not difficult to prove, using Ehrenfeucht-Fraïssé games, that every formula of multi-sorted FO_1 is equivalent to a formula in basic form over strict models. By Remark 5.2.4 the normal form also holds over arbitrary models. We omit a full proof for this case because it is very similar to that of FOE_1 .

5.2.5. PROPOSITION. *Every formula of $\text{FO}_1(A, \mathcal{S})$ is equivalent to a formula in basic form.*

Normal form for FOE_1

In this subsection we have to pay particular attention to the kind of one-step models that we are working with. The case of FOE_1 is much more complicated, as this logic *can* distinguish between strict and arbitrary one-step models. We first give a normal form for *strict* models and then lift it to arbitrary models.

The strict case. We prove that every formula of *multi-sorted* monadic first-order logic with equality (i.e., FOE_1) is equivalent to a formula in strict basic form over strict models.

5.2.6. DEFINITION. We say that a formula $\varphi \in \text{FOE}_1(A, \mathcal{S})$ is in *strict basic form* if $\varphi = \bigvee \bigwedge_{\mathbf{s}} \nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ where in each conjunct we have:

$$\nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}} := \exists \overline{\mathbf{x}}:\mathbf{s}. (\text{diff}(\overline{\mathbf{x}}) \wedge \bigwedge_i \tau_{T_i}(x_i) \wedge \forall z:\mathbf{s}. (\text{diff}(\overline{\mathbf{x}}, z) \rightarrow \bigvee_{S \in \Pi} \tau_S(z)))$$

such that $\overline{\mathbf{T}} \in \wp(A)^k$ for some k , $\mathbf{s} \in \mathcal{S}$ and $\Pi \subseteq \overline{\mathbf{T}}$.

We start by defining the following relation between strict one-step models.

5.2.7. DEFINITION. Let \mathbf{D} and \mathbf{D}' be strict one-step models. For every $k \in \mathbb{N}$ we define the following relation:

$$\mathbf{D} \sim_k^{\overline{}} \mathbf{D}' \iff \forall S \subseteq A, \mathbf{s} \in \mathcal{S}. (|S|_{\mathbf{D}}^{\mathbf{s}} = |S|_{\mathbf{D}'}^{\mathbf{s}} < k \text{ or } |S|_{\mathbf{D}}^{\mathbf{s}}, |S|_{\mathbf{D}'}^{\mathbf{s}} \geq k).$$

Intuitively, two models are related by $\sim_k^{\overline{}}$ when their type information coincides ‘modulo k ’. Later we will prove that this is the same as saying that they cannot be distinguished by a formula of FOE_1 with quantifier rank lower or equal to k . For the moment, we prove the following properties of $\sim_k^{\overline{}}$.

5.2.8. PROPOSITION. *The following hold*

- (i) \sim_k^- is an equivalence relation,
- (ii) \sim_k^- has finite index,
- (iii) Every $E \in \mathfrak{M}_1^s / \sim_k^-$ is characterized by a formula $\varphi_E^- \in \text{FOE}_1(A, \mathcal{S})$ with $qr(\varphi_E^-) = k$.

Proof. We only prove the last point. Let $E \in \mathfrak{M}_1^s / \sim_k^-$ and let $\mathbf{D} \in E$ be a representative. For every $\mathbf{s} \in \mathcal{S}$ call $S_1, \dots, S_n \subseteq A$ to the types such that $|S_i|_{\mathbf{D}}^{\mathbf{s}} = n_i < k$ and $S'_1, \dots, S'_m \subseteq A$ to those satisfying $|S'_i|_{\mathbf{D}}^{\mathbf{s}} \geq k$. Now define

$$\begin{aligned} \varphi_{E, \mathbf{s}}^- := & \bigwedge_{i \leq n} (\exists x_1, \dots, x_{n_i} : \mathbf{s}. \text{diff}(x_1, \dots, x_{n_i}) \wedge \\ & \bigwedge_{j \leq n_i} \tau_{S_i}(x_j) \wedge \forall z : \mathbf{s}. \text{diff}(x_1, \dots, x_{n_i}, z) \rightarrow \neg \tau_{S_i}(z)) \wedge \\ & \bigwedge_{i \leq m} (\exists x_1, \dots, x_k : \mathbf{s}. \text{diff}(x_1, \dots, x_k) \wedge \bigwedge_{j \leq k} \tau_{S'_i}(x_j)) \end{aligned}$$

Finally set $\varphi_E^- := \bigwedge_{\mathbf{s}} \varphi_{E, \mathbf{s}}^-$.

First note that the union of all the S_i and S'_i yields all the possible A -types, and that if a type is not realized at all, then it will correspond to some S_j with $n_j = 0$. It is easy to see that $qr(\varphi_E^-) = k$ and that $\mathbf{D}' \models \varphi_E^-$ iff $\mathbf{D}' \in E$. Observe that the formula φ_E^- gives a specification of E “sort by sort and type by type”. \square

In the following definition we recall the notion of Ehrenfeucht-Fraïssé game for FOE_1 , slightly adapted for the multi-sorted setting, which will be used to establish the connection between \sim_k^- and \equiv_k^{FOE} .

5.2.9. DEFINITION. Let $\mathbf{D}_0 = (D_0, V_0)$ and $\mathbf{D}_1 = (D_1, V_1)$ be strict multi-sorted one-step models. We define the game $\text{EF}_k^-(\mathbf{D}_0, \mathbf{D}_1)$ between \forall and \exists . If \mathbf{D}_i is one of the models we use \mathbf{D}_{-i} to denote the other model, and we do the same with elements. Note that in this definition the index i will never refer to a sort, but to one of the models. A position in this game is a pair of sequences $\bar{\mathbf{s}}_0 \in D_0^n$ and $\bar{\mathbf{s}}_1 \in D_1^n$ with $n \leq k$. The game consists of k rounds where in round $n + 1$ the following steps are made

1. \forall chooses an element d_i in one of the \mathbf{D}_i ,
2. \exists responds with an element d_{-i} in the model \mathbf{D}_{-i} .
3. Let $\bar{\mathbf{s}}_i \in D_i^n$ be the sequences of elements chosen up to round n , they are extended to $\bar{\mathbf{s}}_i' := \bar{\mathbf{s}}_i \cdot d_i$. Player \exists survives the round iff she does not get stuck and the function $f_{n+1} : \bar{\mathbf{s}}_0' \mapsto \bar{\mathbf{s}}_1'$ is a partial isomorphism of one-step models.

Player \exists wins iff she can survive all k rounds. Given $n \leq k$ and $\bar{\mathbf{s}}_i \in D_i^n$ such that $f_n : \bar{\mathbf{s}}_0 \mapsto \bar{\mathbf{s}}_1$ is a partial isomorphism, we use $\text{EF}_k^{\bar{}}(\mathbf{D}_0, \mathbf{D}_1) @ (\bar{\mathbf{s}}_0, \bar{\mathbf{s}}_1)$ to denote the (initialized) game where n moves have been played and $k - n$ moves are left to be played.

5.2.10. LEMMA. *The following are equivalent*

1. $\mathbf{D}_0 \equiv_k^{\text{FOE}} \mathbf{D}_1$,
2. $\mathbf{D}_0 \sim_k^{\bar{}} \mathbf{D}_1$,
3. \exists has a winning strategy in $\text{EF}_k^{\bar{}}(\mathbf{D}_0, \mathbf{D}_1)$.

Proof. Step (1) to (2) is direct by Proposition 5.2.8. For (2) to (3) we give a winning strategy for \exists in $\text{EF}_k^{\bar{}}(\mathbf{D}_0, \mathbf{D}_1)$. We do it by showing the following claim

CLAIM 1. Let $\mathbf{D}_0 \sim_k^{\bar{}} \mathbf{D}_1$ and $\bar{\mathbf{s}}_i \in D_i^n$ be such that $n < k$ and $f_n : \bar{\mathbf{s}}_0 \mapsto \bar{\mathbf{s}}_1$ is a partial isomorphism; then \exists can survive one more round in $\text{EF}_k^{\bar{}}(\mathbf{D}_0, \mathbf{D}_1) @ (\bar{\mathbf{s}}_0, \bar{\mathbf{s}}_1)$.

PROOF OF CLAIM. Let \forall pick $d_i \in D_i$ such that d_i has type $T \subseteq A$ and sort $\mathbf{s} \in \mathcal{S}$. If d_i had already been played then \exists picks the same element as before and $f_{n+1} = f_n$. If d_i is new and $|T|_{\mathbf{D}_i}^{\mathbf{s}} \geq k$ then, as at most $n < k$ elements have been played, there is always some new $d_{-i} \in D_{-i}$ that \exists can choose that matches d_i . If $|T|_{\mathbf{D}_i}^{\mathbf{s}} = m < k$ then we know that $|T|_{\mathbf{D}_{-i}}^{\mathbf{s}} = m$. Therefore, as d_i is new and f_n is injective, there must be a $d_{-i} \in D_{-i}$ of sort \mathbf{s} that \exists can choose. \blacktriangleleft

Step (3) to (1) is a standard result [EF95, Corollary 2.2.9] in the unsorted setting, we prove it for the multi-sorted setting and for completeness sake.

CLAIM 2. Let $\bar{\mathbf{s}}_i \in D_i^n$ and $\varphi(z_1, \dots, z_n) \in \text{FOE}_1(A)$ be such that $qr(\varphi) \leq k - n$. If \exists has a winning strategy in $\text{EF}_k^{\bar{}}(\mathbf{D}_0, \mathbf{D}_1) @ (\bar{\mathbf{s}}_0, \bar{\mathbf{s}}_1)$ then $\mathbf{D}_0 \models \varphi(\bar{\mathbf{s}}_0)$ if and only if $\mathbf{D}_1 \models \varphi(\bar{\mathbf{s}}_1)$.

PROOF OF CLAIM. If φ is atomic the claim holds because of $f_n : \bar{\mathbf{s}}_0 \mapsto \bar{\mathbf{s}}_1$ being a partial isomorphism (more specifically, the *atom* condition). Boolean cases are straightforward. Let $\varphi(z_1, \dots, z_n) = \exists x:\mathbf{s}.\psi(z_1, \dots, z_n, x)$ and suppose $\mathbf{D}_0 \models \varphi(\bar{\mathbf{s}}_0)$. Hence, there exists $d_0 \in D_0$ of sort \mathbf{s} such that $\mathbf{D}_0 \models \psi(\bar{\mathbf{s}}_0, d_0)$. By hypothesis we know that \exists has a winning strategy for $\text{EF}_k^{\bar{}}(\mathbf{D}_0, \mathbf{D}_1) @ (\bar{\mathbf{s}}_0, \bar{\mathbf{s}}_1)$. Therefore, if \forall picks $d_0 \in D_0$ she can respond with some $d_1 \in D_1$ and has a winning strategy for $\text{EF}_k^{\bar{}}(\mathbf{D}_0, \mathbf{D}_1) @ (\bar{\mathbf{s}}_0 \cdot d_0, \bar{\mathbf{s}}_1 \cdot d_1)$. First observe that, as \exists survives the round, then $\bar{\mathbf{s}}_0 \cdot d_0 \mapsto \bar{\mathbf{s}}_1 \cdot d_1$ is a partial isomorphism and hence (by the *sorts* condition) the elements d_0 and d_1 will have the same sort. By induction hypothesis, because $qr(\psi) \leq k - (n + 1)$, we have that $\mathbf{D}_0 \models \psi(\bar{\mathbf{s}}_0, d_0)$ iff $\mathbf{D}_1 \models \psi(\bar{\mathbf{s}}_1, d_1)$ and hence $\mathbf{D}_1 \models \exists x:\mathbf{s}.\psi(\bar{\mathbf{s}}_1, x)$. The other direction is symmetric. \blacktriangleleft

Combining these claims finishes the proof of the lemma. \square

5.2.11. THEOREM. *Over strict models, every formula $\varphi \in \text{FOE}_1(A, \mathcal{S})$ is equivalent to a formula $\psi \in \text{FOE}_1(A, \mathcal{S})$ in strict basic form.*

Proof. Let $qr(\varphi) = k$ and let $\llbracket \varphi \rrbracket$ be the class of models satisfying φ . As $\mathfrak{M}_1^s / \equiv_k^{\text{FOE}}$ is the same as $\mathfrak{M}_1^s / \sim_k^-$ by Lemma 5.2.10, it is easy to see that $\varphi \equiv \bigvee \{ \varphi_E^- \mid E \in \llbracket \varphi \rrbracket / \sim_k^- \}$. Remember that φ_E^- is defined as $\bigwedge_{\mathbf{s}} \varphi_{E,s}^-$. Therefore, it is enough to see that each $\varphi_{E,s}^-$ is equivalent to some $\nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ where $T_i \subseteq A$ and $\Pi \subseteq \overline{\mathbf{T}}$. From this, we can conclude that φ is equivalent to $\psi := \bigvee \{ \varphi_E^- \mid E \in \llbracket \varphi \rrbracket / \sim_k^- \}$. This can be done exactly as in Theorem 5.1.12. \square

The arbitrary case. We now prove that we can also give a normal form for arbitrary models. As an intuition on why the strict normal form “lifts” to arbitrary models observe that any one-step model on sorts \mathcal{S} can be seen as a *strict* one-step model on sorts $\wp(\mathcal{S})$.

5.2.12. DEFINITION. For an arbitrary one-step model \mathbf{D} on sorts \mathcal{S} we define \mathbf{D}^\uparrow to be the strict one-step model on sorts $\wp(\mathcal{S})$ obtained by redefining the sorts of \mathbf{D} as follows: an element d of \mathbf{D}^\uparrow belongs to the sort $\mathbf{S} \subseteq \mathcal{S}$ iff it belongs to all the sorts $\mathbf{s} \in \mathbf{S}$ in \mathbf{D} and it does *not* belong to any sort $\mathbf{s}' \in \mathcal{S} \setminus \mathbf{S}$ in \mathbf{D} .

For every $\varphi \in \text{FOE}_1(A, \mathcal{S})$ we define the translation $\varphi^\uparrow \in \text{FOE}_1(A, \wp(\mathcal{S}))$ inductively: it behaves homomorphically in every operator which is not a quantifier, and it is defined as follows for quantifiers.

$$\begin{aligned} (\exists x:\mathbf{s}.\varphi(x))^\uparrow &:= \bigvee \{ \exists x:\mathbf{S}.\varphi^\uparrow(x) \mid \{\mathbf{s}\} \subseteq \mathbf{S} \subseteq \mathcal{S} \} \\ (\forall x:\mathbf{s}.\varphi(x))^\uparrow &:= \bigwedge \{ \forall x:\mathbf{S}.\varphi^\uparrow(x) \mid \{\mathbf{s}\} \subseteq \mathbf{S} \subseteq \mathcal{S} \}. \end{aligned}$$

The following proposition states the expected relationship.

5.2.13. PROPOSITION. *For every $\varphi \in \text{FOE}_1(A, \mathcal{S})$ and arbitrary one-step model \mathbf{D} on sorts \mathcal{S} we have that $\mathbf{D} \models \varphi$ iff $\mathbf{D}^\uparrow \models \varphi^\uparrow$.*

Proof. This proposition is proved by induction, we focus on the existential case. That is, we prove that $\mathbf{D} \models \exists x:\mathbf{s}.\varphi(x)$ iff $\mathbf{D}^\uparrow \models (\exists x:\mathbf{s}.\varphi(x))^\uparrow$.

\Rightarrow Let $d \in D_{\mathbf{s}}$ be the witness of sort \mathbf{s} for φ . Let $\mathbf{S} \subseteq \mathcal{S}$ be the set of sorts to which d belongs in \mathbf{D} . Observe that by definition of \mathbf{D}^\uparrow , the element d (only) belongs to the sort \mathbf{S} in \mathbf{D}^\uparrow . Hence, using the inductive hypothesis we get that $\mathbf{D}^\uparrow \models \exists x:\mathbf{S}.\varphi^\uparrow(x)$. Finally, as $\{\mathbf{s}\} \subseteq \mathbf{S}$, this implies that $\mathbf{D}^\uparrow \models (\exists x:\mathbf{s}.\varphi(x))^\uparrow$.

\Leftarrow This direction is easier and left to the reader. \square

The next step is to use Theorem 5.2.11 (over strict models) to get a strict normal form ψ of φ^\uparrow . After that we want to transfer the normal form to arbitrary

models, therefore we need something like a converse of Proposition 5.2.13. With this in mind, we introduce the following abbreviation $\exists x:\mathbf{S}!$:

$$\exists x:\mathbf{S}!. \varphi(x) := \exists x, x_1:\mathbf{s}_1, \dots, x_n:\mathbf{s}_n. \text{equal}(x, x_1, \dots, x_n) \wedge \left(\bigwedge_{\mathbf{s} \in \mathcal{S} \setminus \mathbf{S}} \forall z:\mathbf{s}. x \neq z \right) \wedge \varphi(x)$$

where $\mathbf{S} = \{\mathbf{s}_1, \dots, \mathbf{s}_n\}$ and $\text{equal}(y_1, \dots, y_n) := \bigwedge_{1 \leq m < n} (y_m \approx y_{m+1})$. Intuitively, the quantifier $\exists x:\mathbf{S}!$ says that there is an element x which belongs *exactly* to the sorts $\mathbf{S} \subseteq \wp(\mathcal{S})$.

5.2.14. DEFINITION. For every $\psi \in \text{FOE}_1(A, \wp(\mathcal{S}))$ we define the translation $\psi^\downarrow \in \text{FOE}_1(A, \mathcal{S})$ inductively: it behaves homomorphically in every operator which is not a quantifier, and it is defined as follows for quantifiers.

$$(\exists x:\mathbf{S}.\psi(x))^\downarrow := \exists x:\mathbf{S}!. \psi^\downarrow(x) \quad \text{and} \quad (\forall x:\mathbf{S}.\psi(x))^\downarrow := \neg \exists x:\mathbf{S}!. \neg \psi^\downarrow(x)$$

for $\mathbf{S} \in \wp(\mathcal{S})$.

The following proposition states the expected relationship.

5.2.15. PROPOSITION. For every $\psi \in \text{FOE}_1(A, \wp(\mathcal{S}))$ and arbitrary one-step model \mathbf{D} on sorts \mathcal{S} we have that $\mathbf{D} \models \psi^\downarrow$ iff $\mathbf{D}^\uparrow \models \psi$.

We are now ready to state the normal form of FOE_1 for arbitrary models and generalize Theorem 5.2.11.

5.2.16. DEFINITION. We say that a formula $\varphi \in \text{FOE}_1(A, \mathcal{S})$ is in *basic form* if $\varphi = \bigvee \bigwedge_{\mathbf{s}} \nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ where in each conjunct we have:

$$\nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}} := \exists \overline{\mathbf{x}}:\mathbf{S}!. (\text{diff}(\overline{\mathbf{x}}) \wedge \bigwedge_i \tau_{T_i}(x_i) \wedge \forall z:\mathbf{S}!. (\text{diff}(\overline{\mathbf{x}}, z) \rightarrow \bigvee_{S \in \Pi} \tau_S(z)))$$

such that $\overline{\mathbf{T}} \in \wp(A)^k$ for some k , $\mathbf{S} \subseteq \mathcal{S}$ is non-empty and $\Pi \subseteq \overline{\mathbf{T}}$.

5.2.17. THEOREM. Every $\varphi \in \text{FOE}_1(A, \mathcal{S})$ is equivalent to a formula in basic form.

Proof. We use the notation that we have developed in this subsection and proceed as follows:

$$\begin{aligned} \mathbf{D} \models \varphi & \text{ iff } \mathbf{D}^\uparrow \models \varphi^\uparrow && \text{(Proposition 5.2.13)} \\ & \text{ iff } \mathbf{D}^\uparrow \models \psi && \text{(Theorem 5.2.11: strict normal form)} \\ & \text{ iff } \mathbf{D} \models \psi^\downarrow. && \text{(Proposition 5.2.15)} \end{aligned}$$

Observe that by construction ψ^\downarrow is in basic normal form. \square

5.2.2 One-step monotonicity

Monotone fragment of FO_1

5.2.18. THEOREM. *A formula of $\text{FO}_1(A, \mathcal{S})$ is monotone in $A' \subseteq A$ iff it is equivalent to a sentence given by the following grammar:*

$$\varphi ::= \psi \mid a(x) \mid \exists x:\mathbf{s}.\varphi \mid \forall x:\mathbf{s}.\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

where $a \in A'$, $s \in \mathcal{S}$ and $\psi \in \text{FO}_1(A \setminus A', \mathcal{S})$. We denote this fragment as $\text{FO}_1\text{MON}_{A'}(A, \mathcal{S})$.

Before going on we need to introduce a bit of new notation.

5.2.19. DEFINITION. Let $A' \subseteq A$ be a finite set of names. The monotone and positive variants of $\nabla_{\text{FO}}(\Sigma)_{\mathbf{s}}$ are given as follows:

$$\begin{aligned} \nabla_{\text{FO}}^{A'}(\Sigma)_{\mathbf{s}} &:= \bigwedge_{s \in \Sigma} \exists x:\mathbf{s}.\tau_S^{A'}(x) \wedge \forall x:\mathbf{s}.\bigvee_{s \in \Sigma} \tau_S^{A'}(x) \\ \nabla_{\text{FO}}^+(\Sigma)_{\mathbf{s}} &:= \bigwedge_{s \in \Sigma} \exists x:\mathbf{s}.\tau_S^+(x) \wedge \forall x:\mathbf{s}.\bigvee_{s \in \Sigma} \tau_S^+(x). \end{aligned}$$

We also introduce the following generalized forms of the above notation:

$$\begin{aligned} \nabla_{\text{FO}}^{A'}(\Sigma, \Pi)_{\mathbf{s}} &:= \bigwedge_{s \in \Sigma} \exists x:\mathbf{s}.\tau_S^{A'}(x) \wedge \forall x:\mathbf{s}.\bigvee_{s \in \Pi} \tau_S^{A'}(x) \\ \nabla_{\text{FO}}^+(\Sigma, \Pi)_{\mathbf{s}} &:= \bigwedge_{s \in \Sigma} \exists x:\mathbf{s}.\tau_S^+(x) \wedge \forall x:\mathbf{s}.\bigvee_{s \in \Pi} \tau_S^+(x). \end{aligned}$$

When the set A' is a singleton $\{a\}$ we will write a instead of A' .

The result follows from the following lemma.

5.2.20. LEMMA. *The following hold:*

1. *Every $\varphi \in \text{FO}_1\text{MON}_{A'}(A, \mathcal{S})$ is monotone in A' .*
2. *There exists a translation $(-)^{\circ} : \text{FO}_1(A, \mathcal{S}) \rightarrow \text{FO}_1\text{MON}_{A'}(A, \mathcal{S})$ such that a formula $\varphi \in \text{FO}_1(A, \mathcal{S})$ is monotone in A' if and only if $\varphi \equiv \varphi^{\circ}$.*

Proof. In Section 5.1.2 this result is proved for single-sorted FO_1 . It is not difficult to adapt the proof for multi-sorted FO_1 . For item (2), if we assume that φ is in negation normal form, the translation is basically defined as $\varphi^{\circ} := \varphi[\neg a(x) \mapsto \top \mid a \in A']$ for φ in negation normal form. \square

Combining the normal form theorem for FO_1 and the above lemma, we obtain the following corollary providing a normal form for the monotone fragment of FO_1 .

5.2.21. COROLLARY. *Let $\varphi \in \text{FO}_1(A, \mathcal{S})$, the following hold:*

- (i) *The formula φ is monotone in $A' \subseteq A$ iff it is equivalent to a formula in the basic form $\bigvee \bigwedge_{\mathbf{s}} \nabla_{\text{FO}}^{A'}(\Sigma)_{\mathbf{s}}$ for some types $\Sigma \subseteq \wp A$.*
- (ii) *The formula φ is monotone in all $a \in A$ (i.e., $\varphi \in \text{FO}_1^+(A, \mathcal{S})$) iff φ is equivalent to a formula in the basic form $\bigvee \bigwedge_{\mathbf{s}} \nabla_{\text{FO}}^+(\Sigma)_{\mathbf{s}}$ for some types $\Sigma \subseteq \wp A$.*

Monotone fragment of FOE_1

5.2.22. THEOREM. *A formula of $\text{FOE}_1(A, \mathcal{S})$ is monotone in $A' \subseteq A$ iff it is equivalent to a sentence given by:*

$$\varphi ::= \psi \mid a(x) \mid \exists x:\mathbf{s}.\varphi \mid \forall x.\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi$$

where $a \in A'$, $\mathbf{s} \in \mathcal{S}$ and $\psi \in \text{FOE}_1(A \setminus A', \mathcal{S})$. We denote this fragment as $\text{FOE}_1\text{MON}_{A'}(A, \mathcal{S})$.

Before going on we need to introduce a bit of new notation.

5.2.23. DEFINITION. Let $A' \subseteq A$ be a finite set of names. The monotone variant of $\nabla_{\text{FOE}}(\Sigma, \Pi)_{\mathbf{s}}$ is given as follows:

$$\nabla_{\text{FOE}}^{A'}(\bar{\mathbf{T}}, \Pi)_{\mathbf{s}} := \exists \bar{\mathbf{x}}:\mathbf{S}!. (\text{diff}(\bar{\mathbf{x}}) \wedge \bigwedge_i \tau_{T_i}^{A'}(x_i) \wedge \forall z:\mathbf{S}!. (\text{diff}(\bar{\mathbf{x}}, z) \rightarrow \bigvee_{S \in \Pi} \tau_S^{A'}(z))).$$

When the set A' is a singleton $\{a\}$ we will write a instead of A' . The positive variant of $\nabla_{\text{FOE}}(\Sigma, \Pi)$ is defined as above but with $+$ in place of A' . All these variants can also be defined with $\mathbf{s} \in \mathcal{S}$ instead of $\mathbf{S} \subseteq \mathcal{S}$.

The result follows from the following lemma.

5.2.24. LEMMA. *The following hold:*

1. *Every $\varphi \in \text{FOE}_1\text{MON}_{A'}(A, \mathcal{S})$ is monotone in A' .*
2. *There exists a translation $(-)^{\circ} : \text{FOE}_1(A, \mathcal{S}) \rightarrow \text{FOE}_1\text{MON}_{A'}(A, \mathcal{S})$ such that a formula $\varphi \in \text{FOE}_1(A, \mathcal{S})$ is monotone in A' if and only if $\varphi \equiv \varphi^{\circ}$.*

Proof. In Section 5.1.2 this result is proved for single-sorted FOE_1 . It is not difficult to adapt the proof for multi-sorted FOE_1 . Intuitively, the translation is defined as $\varphi^{\circ} := \varphi[\neg a(x) \mapsto \top \mid a \in A']$ for φ in negation normal form. \square

Combining the normal form theorem for FOE_1 and the above lemma, we obtain the following normal forms for the monotone fragment of FOE_1 .

5.2.25. COROLLARY. *Given $\varphi \in \text{FOE}_1(A, \mathcal{S})$, the following hold:*

- (i) *The formula φ is monotone in $A' \subseteq A$ iff it is equivalent to a formula in the basic form $\bigvee \bigwedge_{\mathcal{S}} \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathcal{S}}$ where for each conjunct there are $\overline{\mathbf{T}}$ and Π satisfying $T_i \in \wp(A)$ and $\Pi \subseteq \overline{\mathbf{T}}$,*
- (ii) *The formula φ is monotone in all $a \in A$ (i.e., $\varphi \in \text{FOE}_1^+(A)$) iff it is equivalent to a formula in the basic form $\bigvee \bigwedge_{\mathcal{S}} \nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathcal{S}}$ where for each conjunct there are $\overline{\mathbf{T}}$ and Π satisfying $T_i \in \wp(A)$ and $\Pi \subseteq \overline{\mathbf{T}}$.*
- (iii) *Over strict one-step models the normal forms hold with \mathcal{S} replaced by \mathfrak{s} .*

5.2.3 One-step additivity

As we did in the single-sorted case, we start by giving a characterization of the completely additive fragment of $\text{FOE}_1(A, \mathcal{S})$ and then use it to give a similar characterization for $\text{FO}_1(A, \mathcal{S})$.

Completely additive fragment of FOE_1

5.2.26. THEOREM. *A formula of $\text{FOE}_1(A, \mathcal{S})$ is completely additive in $A' \subseteq A$ iff it is equivalent to a sentence generated by the following grammar:*

$$\varphi ::= \psi \mid a(x) \mid \exists x:\mathfrak{s}.\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \psi$$

where $a \in A'$, $\mathfrak{s} \in \mathcal{S}$ and $\psi \in \text{FOE}_1(A \setminus A', \mathcal{S})$. Observe that the equality is included in ψ . We denote this fragment as $\text{FOE}_1\text{ADD}_{A'}(A, \mathcal{S})$.

The theorem will follow from the next two lemmas.

5.2.27. LEMMA. *Every $\varphi \in \text{FOE}_1\text{ADD}_{A'}(A, \mathcal{S})$ is completely additive in A' .*

Proof. First observe that φ is monotone in every $a \in A'$ by Theorem 5.2.22. We show, by induction, that any one-step formula φ in the fragment (which may not be a sentence) satisfies, for every one-step model (D, V) , assignment $g : i\text{Var} \rightarrow D$,

if $(D, V), g \models \varphi$ then $(D, V[A' \mapsto \overline{\mathbf{Q}}]), g \models \varphi$ for some quasi-atom $\overline{\mathbf{Q}}$ of $V(A')$.

Most cases are proved exactly as in the single-sorted version of this lemma (cf. Lemma 5.1.46). We focus on the sorted existential quantifier. Consider $\varphi = \exists x:\mathfrak{s}.\varphi'$ and assume $(D, V), g \models \varphi$. By definition there is $d \in D_{\mathfrak{s}}$ such that $(D, V), g[x \mapsto d] \models \varphi'$. By induction hypothesis $(D, V[A' \mapsto \overline{\mathbf{Q}}]), g[x \mapsto d] \models \varphi'$ for some $\overline{\mathbf{Q}}$. Therefore we can conclude that $(D, V[A' \mapsto \overline{\mathbf{Q}}]), g \models \exists x:\mathfrak{s}.\varphi'$. \square

5.2.28. LEMMA. *There is an effective translation $(-)^{\oplus} : \text{FOE}_1\text{MON}_{A'}(A, \mathcal{S}) \rightarrow \text{FOE}_1\text{ADD}_{A'}(A, \mathcal{S})$ such that $\varphi \in \text{FOE}_1\text{MON}_{A'}(A, \mathcal{S})$ is completely additive in A' if and only if $\varphi \equiv \varphi^{\oplus}$.*

Proof. We assume that φ is in basic form, i.e., $\varphi = \bigvee \bigwedge_{\mathbf{s}} \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ with $\Pi \subseteq \overline{\mathbf{T}}$. First, we intuitively consider some conditions on subformulas of φ that would force the existence of at least two elements colored with A' . Clearly, any formula that forces this, goes against the spirit of complete additivity:

- (i) Some $\bigwedge_{\mathbf{s}} \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ has $\overline{\mathbf{T}}_{\mathbf{S}_1}$ and $\overline{\mathbf{T}}_{\mathbf{S}_2}$ with $a \in \overline{\mathbf{T}}_{\mathbf{S}_1}$ and $b \in \overline{\mathbf{T}}_{\mathbf{S}_2}$ for $a, b \in A'$, $\mathbf{S}_1 \neq \mathbf{S}_2$.
- (ii) For any $\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ there are $a, b \in T_i \cap T_j$ for distinct $a, b \in A'$ or distinct i, j .
- (iii) For any $\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ we have $S \cap A' \neq \emptyset$ for some $S \in \Pi$.

Now, we give a translation which eliminates (replaces with \perp) the subformulas forcing the above cases. We first take care of case (i) with the following definition

$$\left(\bigvee \bigwedge_{\mathbf{s}} \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}\right)^{\oplus} := \bigvee \left\{ \bigwedge_{\mathbf{s}} \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}^{\oplus} \mid \text{(i) is not the case} \right\}$$

and we take care of the remaining cases as follows

$$\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}^{\oplus} := \begin{cases} \perp & \text{if (ii) holds,} \\ \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi_{A'}^{\times})_{\mathbf{s}} & \text{otherwise,} \end{cases}$$

where $\Pi_{A'}^{\times} := \{S \in \Pi \mid A' \cap S = \emptyset\}$.

First we prove the right-to-left direction of the lemma. Inspecting the syntactic form of φ^{\oplus} it is not difficult to see that $\varphi^{\oplus} \in \text{FOE}_1\text{ADD}_{A'}(A, \mathcal{S})$. Using Lemma 5.2.27 we can conclude that φ^{\oplus} (and therefore φ as well) is completely additive in A' . For the left-to-right direction of the lemma we assume φ to be completely additive in A' and have to prove $\varphi \equiv \varphi^{\oplus}$.

$\boxed{\Leftarrow}$ Let $(D, V) \models \varphi^{\oplus}$. It is enough to show that $(D, V) \models \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi_{A'}^{\times})_{\mathbf{s}}$ implies $(D, V) \models \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ for every conjunct. The key observation is that $\Pi_{A'}^{\times} \subseteq \Pi$.

$\boxed{\Rightarrow}$ Let $(D, V) \models \varphi$. By complete additivity in A' we have that $(D, V[A' \mapsto \overline{\mathbf{Q}}]) \models \varphi$ for some quasi-atom $\overline{\mathbf{Q}}$ of $V(A')$. To improve readability we define $V' := V[A' \mapsto \overline{\mathbf{Q}}]$. We now work with (D, V') because (by monotonicity, which is implied by complete additivity) it will be enough to prove that $(D, V') \models \varphi^{\oplus}$.

As $(D, V') \models \varphi$, we know there is some disjunct $\bigwedge_{\mathbf{s}} \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ of φ witnessing the satisfaction. First, we prove that this disjunct is preserved by the translation.

CLAIM 1. The disjunct $\bigwedge_{\mathbf{s}} \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ does not satisfy case (i).

PROOF OF CLAIM. Suppose that, for this disjunct, there are two conjuncts corresponding to sorts \mathbf{S}_1 and \mathbf{S}_2 such that $a \in \overline{\mathbf{T}}_{\mathbf{S}_1}$ and $b \in \overline{\mathbf{T}}_{\mathbf{S}_2}$ for $a, b \in A'$. As the sorts are disjoint, this implies that there should be two *distinct* elements colored by elements of A' . As $V'_{\pi}(A')$ is a quasi-atom, this cannot be the case. \blacktriangleleft

From the above claim it follows that, for the previously fixed disjunct, there is at most one sort (i.e, one conjunct) which can possibly use A' in the existential part (that is, in $\overline{\mathbf{T}}$). Hence, the disjunct is (so far) preserved by the translation. We still have to check that every conjunct is preserved, that is, we now focus on cases (ii) and (iii). We fix an arbitrary conjunct $\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ and state the following claims, which are proved exactly as in Lemma 5.1.47.

CLAIM 2. For every $b, b' \in A'$, if $b \in T_i$ and $b' \in T_j$ then $b = b'$ and $i = j$.

CLAIM 3. If $(D, V') \models \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ then $(D, V') \models \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi_{A'}^{\times})_{\mathbf{s}}$.

The combination of these claims yields the desired result. \square

Putting together the above lemmas we obtain Theorem 5.2.26. Moreover, a careful analysis of the translation gives us the following corollary, providing normal forms for the completely additive fragment of FOE_1 .

5.2.29. COROLLARY. *Let $\varphi \in \text{FOE}_1(A)$, the following hold:*

(i) *The formula φ is completely additive in $A' \subseteq A$ iff it is equivalent to a formula in the basic form $\bigvee \bigwedge_{\mathbf{S}} \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ where $\overline{\mathbf{T}} \in \wp(A)^k$, $\Pi \subseteq \overline{\mathbf{T}}$ and for every disjunct $\bigwedge_{\mathbf{S}} \nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$,*

1. *At most one sort $\underline{\mathbf{s}} \in \mathcal{S}$ may use elements from A' in $\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\underline{\mathbf{s}}}$,*
2. *For $\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\underline{\mathbf{s}}}$ we have that Π is A' -free and there is at most one element of A' in the concatenation of the lists $T_1 \cdot T_2 \cdots T_k$.*

(ii) *If φ is monotone in A (i.e., $\varphi \in \text{FOE}_1^+(A)$) then φ is completely additive in $A' \subseteq A$ iff it is equivalent to a formula in the basic form $\bigvee \bigwedge_{\mathbf{S}} \nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ where $\overline{\mathbf{T}} \in \wp(A)^k$, $\Pi \subseteq \overline{\mathbf{T}}$ and for every disjunct $\bigwedge_{\mathbf{S}} \nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$,*

1. *At most one sort $\underline{\mathbf{s}} \in \mathcal{S}$ may use elements from A' in $\nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\underline{\mathbf{s}}}$,*
2. *For $\nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\underline{\mathbf{s}}}$ we have that Π is A' -free and there is at most one element of A' in the concatenation of the lists $T_1 \cdot T_2 \cdots T_k$.*

(iii) *Over strict one-step models the normal forms hold with \mathbf{S} replaced by \mathbf{s} .*

Completely additive fragment of FO_1

5.2.30. THEOREM. *A formula of $\text{FO}_1(A, \mathcal{S})$ is completely additive in $A' \subseteq A$ iff it is equivalent to a sentence generated by the following grammar:*

$$\varphi ::= \psi \mid a(x) \mid \exists x:\mathbf{s}.\varphi \mid \varphi \vee \varphi \mid \varphi \wedge \psi$$

where $a \in A'$, $\mathbf{s} \in \mathcal{S}$ and $\psi \in \text{FO}_1(A \setminus A', \mathcal{S})$. We denote this fragment as $\text{FO}_1\text{ADD}_{A'}(A, \mathcal{S})$.

The theorem will follow from the next two lemmas.

5.2.31. LEMMA. *Every $\varphi \in \text{FO}_1\text{ADD}_{A'}(A, \mathcal{S})$ is completely additive in A' .*

Proof. As $\text{FO}_1\text{ADD}_{A'}(A, \mathcal{S})$ is included in $\text{FOE}_1\text{ADD}_{A'}(A, \mathcal{S})$ we can simply use Lemma 5.2.27 for $\text{FOE}_1\text{ADD}_{A'}(A, \mathcal{S})$ and conclude what we want. \square

As we did in the single-sorted case, we will obtain a characterization for $\text{FO}_1(A, \mathcal{S})$ using the characterization for $\text{FOE}_1(A, \mathcal{S})$.

5.2.32. DEFINITION. The translation $(-)^{\bullet} : \text{FOE}_1(A, \mathcal{S}) \rightarrow \text{FO}_1(A, \mathcal{S})$ on formulas of $\text{FOE}_1(A, \mathcal{S})$ which are in *strict normal form*, is defined as follows:

$$(\nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}})^{\bullet} := \nabla_{\text{FO}}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$$

and for $\alpha = \bigvee \bigwedge_{\mathbf{s}} \alpha_{\mathbf{s}}$ we define $(\alpha)^{\bullet} := \bigvee \bigwedge_{\mathbf{s}} (\alpha_{\mathbf{s}})^{\bullet}$. We extend this translation to the monotone and positive fragments as expected, that is, we define $(\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}})^{\bullet} := \nabla_{\text{FO}}^{A'}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ and $(\nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}})^{\bullet} := \nabla_{\text{FO}}^+(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$.

Observe that on the right hand side of the above definition, $\overline{\mathbf{T}}$ is seen as a set. The key property of this translation is the following.

5.2.33. PROPOSITION. *For every strict one-step model (D_1, \dots, D_n, V) and every formula $\alpha \in \text{FOE}_1(A)$ in strict normal form we have:*

$$(D_1, \dots, D_n, V) \models \alpha^{\bullet} \quad \text{iff} \quad (D_1 \times \omega, \dots, D_n \times \omega, V_{\pi}) \models \alpha,$$

where the valuation V_{π} is given by $V_{\pi}^{\natural}((d, k)) := V^{\natural}(d)$.

We call these (strict) one-step models \mathbf{D} and \mathbf{D}_{ω} respectively. Observe that the model \mathbf{D}_{ω} has ω -many copies of each element of \mathbf{D} .

Proof. Formulas of $\text{FOE}_1(A)$ which are in strict normal form are of the shape $\bigvee \bigwedge_{\mathbf{s}} \nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$. Therefore it will be enough to prove that $\mathbf{D} \models \nabla_{\text{FO}}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ iff $\mathbf{D}_{\omega} \models \nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$. This is done exactly as in the single-sorted version of this proposition, that is, Proposition 5.1.52. \square

We are now ready to state the lemma which provides a translation for $\text{FO}_1(A, \mathcal{S})$.

5.2.34. LEMMA. *There is an effective translation $(-)^{\oplus} : \text{FO}_1\text{MON}_{A'}(A, \mathcal{S}) \rightarrow \text{FO}_1\text{ADD}_{A'}(A, \mathcal{S})$ such that $\varphi \in \text{FO}_1\text{MON}_{A'}(A)$ is completely additive in A' if and only if $\varphi \equiv \varphi^{\oplus}$.*

Proof. To define the translation $(-)^{\oplus} : \text{FO}_1\text{MON}_{A'}(A, \mathcal{S}) \rightarrow \text{FO}_1\text{ADD}_{A'}(A, \mathcal{S})$ we will use the translation for FOE_1 given in Lemma 5.2.28. To avoid confusion, we call it $(-)^{\oplus}_{\text{FOE}} : \text{FOE}_1\text{MON}_{A'}(A, \mathcal{S}) \rightarrow \text{FOE}_1\text{ADD}_{A'}(A, \mathcal{S})$. We define, for every formula $\varphi \in \text{FO}_1\text{MON}_{A'}(A, \mathcal{S})$:

$$\varphi^{\oplus} := (\varphi^{\oplus}_{\text{FOE}})^{\bullet}.$$

$\boxed{\Leftarrow}$ A short argument reveals that indeed $\varphi^{\oplus} \in \text{FO}_1\text{ADD}_{A'}(A, \mathcal{S})$, and therefore by Lemma 5.2.31 the formula φ^{\oplus} is completely additive in A' . As φ is equivalent to φ^{\oplus} , it is also completely additive in A' .

$\boxed{\Rightarrow}$ For this direction we assume that $\varphi \in \text{FO}_1\text{MON}_{A'}(A, \mathcal{S})$ is completely additive in A' and we prove that for every one-step model we have $\mathbf{D} \models \varphi$ iff $\mathbf{D} \models \varphi^{\oplus}$.

$$\begin{aligned} \mathbf{D} \models \varphi & \text{ iff } (\mathbf{D}^n)_{\omega} \models \varphi && \text{(Properties of FO}_1 \text{ and Remark 5.2.4)} \\ & \text{ iff } (\mathbf{D}^n)_{\omega} \models \varphi^{\oplus}_{\text{FOE}} && \text{(Lemma 5.2.28)} \\ & \text{ iff } \mathbf{D}^n \models (\varphi^{\oplus}_{\text{FOE}})^{\bullet} && \text{(Proposition 5.2.33)} \\ & \text{ iff } \mathbf{D}^n \models \varphi^{\oplus} && \text{(Definition of } \varphi^{\oplus} \text{)} \\ & \text{ iff } \mathbf{D} \models \varphi^{\oplus}. && \text{(Properties of FO}_1 \text{ and Remark 5.2.4)} \end{aligned}$$

Observe that, because Proposition 5.2.33 requires a *strict* one-step model, we make a slight detour through \mathbf{D}^n (as defined in Remark 5.2.4). \square

Putting together the above lemmas we obtain Theorem 5.2.30. Moreover, a careful analysis of the translation gives us normal forms for the completely additive fragment of FO_1 .

5.2.35. COROLLARY. *Let $\varphi \in \text{FO}_1(A, \mathcal{S})$, given $\Sigma \subseteq \wp(A)$, let $L_{\Sigma} \in A^*$ be the list with repetitions of elements of A in Σ . The following hold:*

(i) *The formula φ is completely additive in $A' \subseteq A$ iff it is equivalent to a formula in the basic form $\bigvee \bigwedge_{\mathbf{s}} \nabla_{\text{FO}}^{A'}(\Sigma, \Pi)_{\mathbf{s}}$ where $\Sigma, \Pi \subseteq \wp(A)$ and for every disjunct $\bigwedge_{\mathbf{s}} \nabla_{\text{FO}}^{A'}(\Sigma, \Pi)_{\mathbf{s}}$,*

1. *At most one sort $\mathbf{s} \in \mathcal{S}$ may use elements from A' in $\nabla_{\text{FO}}^{A'}(\Sigma, \Pi)_{\mathbf{s}}$,*
2. *For $\nabla_{\text{FO}}^{A'}(\Sigma, \Pi)_{\mathbf{s}}$ we have that Π is A' -free and there is at most one element of A' in L_{Σ} .*

(ii) *If φ is monotone in A (i.e., $\varphi \in \text{FO}_1^+(A, \mathcal{S})$) then φ is completely additive in $A' \subseteq A$ iff it is equivalent to a formula of the form $\bigvee \bigwedge_{\mathbf{s}} \nabla_{\text{FO}}^+(\Sigma, \Pi)_{\mathbf{s}}$ where $\Sigma, \Pi \subseteq \wp(A)$ and for every disjunct $\bigwedge_{\mathbf{s}} \nabla_{\text{FO}}^+(\Sigma, \Pi)_{\mathbf{s}}$,*

1. *At most one sort $\mathbf{s} \in \mathcal{S}$ may use elements from A' in $\nabla_{\text{FO}}^+(\Sigma, \Pi)_{\mathbf{s}}$,*
2. *For $\nabla_{\text{FO}}^+(\Sigma, \Pi)_{\mathbf{s}}$ we have that Π is A' -free and there is at most one element of A' in L_{Σ} .*

5.2.4 One-step multiplicativity

Most of the discussion on dual fragments of Section 5.1.5 was done for an arbitrary one step language $\mathcal{L}(A)$ and it is simple to show that it also applies for sorted languages $\mathcal{L}(A, \mathcal{S})$. Therefore, we can define the following fragments and prove that they characterize complete multiplicativity.

5.2.36. DEFINITION. The fragment $\text{FOE}_1\text{MUL}_{A'}(A, \mathcal{S})$ is given by the sentences generated by:

$$\varphi ::= \psi \mid a(x) \mid \forall x:\mathbf{s}.\varphi \mid \varphi \wedge \varphi \mid \varphi \vee \psi$$

where $a \in A'$, $\mathbf{s} \in \mathcal{S}$ and $\psi \in \text{FOE}_1(A \setminus A', \mathcal{S})$. Observe that the equality is included in ψ . The fragment $\text{FO}_1\text{MUL}_{A'}(A, \mathcal{S})$ is defined as $\text{FOE}_1\text{MUL}_{A'}(A, \mathcal{S})$ but with $\psi \in \text{FO}_1(A \setminus A', \mathcal{S})$.

The following proposition states that the above fragments are actually the duals of the fragments defined earlier in this chapter.

5.2.37. PROPOSITION. *The following hold:*

$$\begin{aligned} \text{FO}_1\text{MUL}_{A'}(A, \mathcal{S}) &= \{\varphi \mid \varphi^\delta \in \text{FO}_1\text{ADD}_{A'}(A, \mathcal{S})\} \\ \text{FOE}_1\text{MUL}_{A'}(A, \mathcal{S}) &= \{\varphi \mid \varphi^\delta \in \text{FOE}_1\text{ADD}_{A'}(A, \mathcal{S})\}. \end{aligned}$$

As a corollary, we get a characterization for complete multiplicativity.

5.2.38. PROPOSITION. *A formula $\varphi \in \text{FOE}_1(A, \mathcal{S})$ is completely multiplicative in $A' \subseteq A$ if and only if it is equivalent to some $\varphi' \in \text{FOE}_1\text{MUL}_{A'}(A, \mathcal{S})$.*

Proof. Consequence of Proposition 5.2.37 and a multi-sorted analogue of Proposition 5.1.57. \square

5.3 Selected modal languages

In this section we define the one-step modal language $\text{ML}_1(A, \mathbf{P}, \mathbf{D})$ and give a normal form and characterizations for its continuous, completely additive and dual fragments. Most of the results will be obtained using the results for FO_1 , via establishing a connection between FO_1 and ML_1 .

5.3.1. DEFINITION. The set $\text{ML}_1(A, \mathbf{P}, \mathbf{D})$ of one-step modal formulas is given by the following clauses:

$$\begin{aligned} \varphi &::= \top \mid \perp \mid p \mid \neg\varphi \mid \varphi \vee \varphi \mid \langle \ell \rangle \alpha \\ \alpha &::= a \mid \neg\alpha \mid \alpha \vee \alpha \end{aligned}$$

where $p \in \mathbf{P}$, $a \in A$ and $\ell \in \mathbf{D}$. In general we will assume that \mathbf{D} is given by context and only write $\text{ML}_1(A, \mathbf{P})$.

The formulas of $\text{ML}_1(A, \mathbf{P}, \mathbf{D})$ are interpreted over labeled transition systems, together with an additional valuation $V : A \rightarrow \wp(S)$ which gives meaning to the names in A . A key observation is that, as the formulas of $\text{ML}_1(A, \mathbf{P}, \mathbf{D})$ have at most *one level* of modalities, then we can restrict the valuation V to the successors of s_I , in the following sense:

5.3.2. PROPOSITION. *For every formula $\varphi \in \text{ML}_1(A, \mathbf{P})$ such that $\mathbb{S}, V \Vdash \varphi$ with $V : A \rightarrow \wp(S)$ there is $V' : A \rightarrow \wp(R[s_I])$ such that $\mathbb{S}, V' \Vdash \varphi$.*

In Section 4.5 we already observed that the formulas

$$\exists x.(b(x) \wedge c(x)) \vee \forall y.d(y) \quad \text{and} \quad \diamond(b \wedge c) \vee \square d$$

basically describe the same requirements over the set $R[s]$. In the following proposition we make this intuition precise. Before doing it, we introduce a notation which will be useful later. Given $c \in \wp \mathbf{P}$ and $Q \subseteq \mathbf{P}$, we define

$$\varpi_c := \bigwedge_{p \in c} p \wedge \bigwedge_{p \in \mathbf{P} \setminus c} \neg p \quad \varpi_c^Q := \bigwedge_{p \in c} p \wedge \bigwedge_{p \in \mathbf{P} \setminus c, p \notin Q} \neg p \quad \varpi_c^+ := \bigwedge_{p \in c} p$$

5.3.3. PROPOSITION. *There are translations $(-)^t : \text{ML}_1(A, \emptyset, \mathbf{D}) \rightarrow \text{FO}_1(A, \mathbf{D})$ and $(-)_t : \text{FO}_1(A, \mathbf{D}) \rightarrow \text{ML}_1(A, \emptyset, \mathbf{D})$ such that for every labeled transition system \mathbb{S} , valuation $V : A \rightarrow \wp(R[s_I])$, and formulas $\varphi \in \text{ML}_1(A, \emptyset, \mathbf{D})$ and $\psi \in \text{FO}_1(A, \mathbf{D})$ we have:*

$$\mathbb{S}, V \Vdash \varphi \quad \text{iff} \quad (R[s_I], V) \models \varphi^t \quad (5.2)$$

$$\mathbb{S}, V \Vdash \psi_t \quad \text{iff} \quad (R[s_I], V) \models \psi. \quad (5.3)$$

Proof. For $(-)^t : \text{ML}_1(A, \emptyset, \mathbf{D}) \rightarrow \text{FO}_1(A, \mathbf{D})$ we define $a^t := a(x)$ for the names, $(\langle \ell \rangle \alpha)^t := \exists x:\ell.\alpha^t$ for the modalities, and the Boolean connectives as expected.

The other translation is slightly more complicated, and we use the normal form for FO_1 . Assume, by Proposition 5.2.5 that $\psi = \bigvee \bigwedge_{\mathbf{s}} \nabla_{\text{FO}}(\Sigma)_{\mathbf{s}}$ and recall that $\nabla_{\text{FO}}(\Sigma)_{\mathbf{s}} = \bigwedge_{S \in \Sigma} \exists x:\mathbf{s}.\tau_S(x) \wedge \forall x:\mathbf{s}.\bigvee_{S \in \Sigma} \tau_S(x)$. We define the translation $(-)_t : \text{FO}_1(A, \mathbf{D}) \rightarrow \text{ML}_1(A, \emptyset, \mathbf{D})$ homomorphically on Boolean operators and

$$(\nabla_{\text{FO}}(\Sigma)_{\mathbf{s}})_t := \bigwedge_{S \in \Sigma} \langle \mathbf{s} \rangle \varpi_c \wedge [\mathbf{s}] \bigvee_{S \in \Sigma} \varpi_c.$$

It is not difficult to see that these translations satisfy the above equations. \square

5.3.1 Normal forms

5.3.4. DEFINITION. We say that a formula $\varphi \in \text{ML}_1(A, \mathbf{P}, \mathbf{D})$ is in *basic form* if $\varphi = \bigvee_{c \in \wp \mathbf{P}} (\varpi_c \wedge \bigvee \bigwedge_{\ell \in \mathbf{D}} \nabla_{\text{ML}}(\Sigma)_{\ell})$ where in each conjunct

$$\nabla_{\text{ML}}(\Sigma)_{\ell} := \bigwedge_{S \in \Sigma} \langle \ell \rangle \varpi_c \wedge [\ell] \bigvee_{S \in \Sigma} \varpi_c.$$

for some set of types $\Sigma \subseteq \wp(A)$.

The following proposition states that every formula of $\text{ML}_1(A, \mathbf{P}, \mathbf{D})$ can be separated in (disjuncts containing) two parts: one which specifies the propositions that should hold, and a part which contains modalities, but does not use propositions at all.

5.3.5. PROPOSITION. *Every $\varphi \in \text{ML}_1(A, \mathbf{P}, \mathbf{D})$ is equivalent to $\bigvee_{c \in \wp \mathbf{P}} \varpi_c \wedge \varphi_c$ where $\varphi_c \in \text{ML}_1(A, \emptyset, \mathbf{D})$.*

Proof. Define the formula $\varphi' := \bigvee_{c \in \wp \mathbf{P}} (\varpi_c \wedge \varphi[p \mapsto \text{belongs}(p, c) \mid p \in \mathbf{P}])$ where

$$\text{belongs}(p, c) := \begin{cases} \top & \text{if } p \in c, \\ \perp & \text{otherwise.} \end{cases}$$

It is easy to see that φ' is of the right shape, and that $\varphi \equiv \varphi'$. □

This proposition will be crucial, since it allows us to first focus on getting a normal form for formulas of $\text{ML}_1(A, \emptyset, \mathbf{D})$ and then work on top of it to get a normal form for the full $\text{ML}_1(A, \mathbf{P}, \mathbf{D})$.

5.3.6. THEOREM. *Every $\varphi \in \text{ML}_1(A, \mathbf{P}, \mathbf{D})$ is equivalent to a formula in basic form.*

Proof. By Proposition 5.3.5 it will be enough to give a normal form for formulas $\varphi \in \text{ML}_1(A, \emptyset, \mathbf{D})$. To do it, we take such a formula φ and translate it to $\text{FO}_1(A, \mathbf{D})$ using the translation $(-)^t$ of Proposition 5.3.3. Using Proposition 5.2.5 we will now assume that φ^t is in basic form (or otherwise convert it). We now apply the translation in the other direction and obtain the formula $(\varphi^t)_t \in \text{ML}_1(A, \emptyset, \mathbf{D})$. An inspection of the translation $(-)_t$, as defined in Proposition 5.3.3, reveals that $(\varphi^t)_t$ is in the basic form $\bigvee \bigwedge_{d \in \mathbf{D}} \nabla_{\text{ML}}(\Sigma)_\ell$. We prove that these formulas are equivalent using Proposition 5.3.3, as follows:

$$\mathbb{S}, V \Vdash \varphi \quad \text{iff} \quad (R[s_I], V) \models \varphi^t \tag{5.2}$$

$$\text{iff} \quad \mathbb{S}, V \Vdash (\varphi^t)_t \tag{5.3}$$

This finishes the proof. □

5.3.2 One-step monotonicity

5.3.7. THEOREM. *A formula of $\text{ML}_1(A, \mathbf{P}, \mathbf{D})$ is monotone in $A' \subseteq A$ iff it is equivalent to a formula φ , given by the following grammar:*

$$\begin{aligned} \varphi &::= \psi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle \ell \rangle \alpha \mid [\ell] \alpha \\ \alpha &::= a \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \end{aligned}$$

where $\psi \in \text{ML}_1(A \setminus A', \mathbf{P}, \mathbf{D})$, $p \in \mathbf{P}$, $a \in A'$ and $\ell \in \mathbf{D}$. Observe that the propositions are considered in the clause ψ . We call this fragment $\text{ML}_1\text{MON}_{A'}(A, \mathbf{P}, \mathbf{D})$.

The theorem will follow from the next two lemmas.

5.3.8. LEMMA. *Every $\varphi \in \text{ML}_1\text{MON}_{A'}(A, \mathbf{P}, \mathbf{D})$ is monotone in A' .*

Proof. This can be easily proved by induction. \square

Before going on we need to introduce a bit of new notation.

5.3.9. DEFINITION. Let $A' \subseteq A$ be a finite set of names. The monotone and positive variants of $\nabla_{\text{ML}}(\Sigma)_\ell$ are given as follows:

$$\nabla_{\text{ML}}^{A'}(\Sigma)_\ell := \bigwedge_{S \in \Sigma} \langle \ell \rangle \pi_S^{A'} \wedge [\ell] \bigvee_{S \in \Sigma} \pi_S^{A'} \quad \nabla_{\text{ML}}^+(\Sigma)_\ell := \bigwedge_{S \in \Sigma} \langle \ell \rangle \pi_S^+ \wedge [\ell] \bigvee_{S \in \Sigma} \pi_S^+.$$

We also introduce the following generalized forms of the above notation:

$$\nabla_{\text{ML}}^{A'}(\Sigma, \Pi)_\ell := \bigwedge_{S \in \Sigma} \langle \ell \rangle \pi_S^{A'} \wedge [\ell] \bigvee_{S \in \Pi} \pi_S^{A'} \quad \nabla_{\text{ML}}^+(\Sigma, \Pi)_\ell := \bigwedge_{S \in \Sigma} \langle \ell \rangle \pi_S^+ \wedge [\ell] \bigvee_{S \in \Pi} \pi_S^+.$$

When the set A' is a singleton $\{a\}$ we will write a instead of A' .

5.3.10. LEMMA. *A formula $\varphi \in \text{ML}_1(A, \mathbf{P}, \mathbf{D})$ is monotone in $A' \subseteq A$ iff it is equivalent to a formula in the basic form*

$$\bigvee_{c \in \wp \mathbf{P}} (\varpi_c \wedge \bigvee_\ell \bigwedge \nabla_{\text{ML}}^{A'}(\Sigma)_\ell)$$

for some types $\Sigma \subseteq \wp A$.

Proof. The right-to-left direction is trivial by observing that every formula in the above basic form belongs to $\text{ML}_1\text{MON}_{A'}(A, \mathbf{P}, \mathbf{D})$ and using Lemma 5.3.8. For the left-to-right direction we do as we did in Section 5.3.1: we can easily prove this via FO_1 using the translations $(-)^t$ and $(-)_t$. \square

5.3.11. COROLLARY. *A formula $\varphi \in \text{ML}_1(A, \mathbf{P}, \mathbf{D})$ is monotone in all $a \in A$ (i.e., $\varphi \in \text{ML}_1^+(A, \mathbf{P}, \mathbf{D})$) iff φ is equivalent to a formula in the basic form $\bigvee_{c \in \wp \mathbf{P}} (\varpi_c \wedge \bigvee_\ell \bigwedge \nabla_{\text{ML}}^+(\Sigma)_\ell)$ for some types $\Sigma \subseteq \wp A$.*

5.3.3 One-step continuity

5.3.12. THEOREM. *A formula of $\text{ML}_1(A, \mathbf{P}, \mathbf{D})$ is continuous in $A' \subseteq A$ iff it is equivalent to a formula given by the following grammar:*

$$\begin{aligned} \varphi &::= \psi \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle \ell \rangle \alpha \\ \alpha &::= a \mid \alpha \wedge \alpha \mid \alpha \vee \alpha \end{aligned}$$

where $\psi \in \text{ML}_1(A \setminus A', \mathbf{P}, \mathbf{D})$, $p \in \mathbf{P}$, $a \in A'$ and $\ell \in \mathbf{D}$. Observe that the propositions are considered in the clause ψ . We call this fragment $\text{ML}_1\text{CON}_{A'}(A, \mathbf{P}, \mathbf{D})$.

The theorem will follow from the next two lemmas.

5.3.13. LEMMA. *Every $\varphi \in \text{ML}_1\text{CON}_{A'}(A, \mathbf{P}, \mathbf{D})$ is continuous in A' .*

Proof. This can be easily proved by induction. \square

5.3.14. LEMMA. *A formula $\varphi \in \text{ML}_1(A, \mathbf{P}, \mathbf{D})$ is continuous in $A' \subseteq A$ iff it is equivalent to a formula in the basic form*

$$\bigvee_{c \in \wp \mathbf{P}} (\varpi_c \wedge \bigvee \bigwedge_{\ell} \nabla_{\text{ML}}^{A'}(\Sigma, \Pi)_{\ell})$$

for some types $\Sigma, \Pi \subseteq \wp A$ such that $A' \cap \bigcup \Pi = \emptyset$.

Proof. The right-to-left direction is trivial by observing that every formula in the above basic form belongs to $\text{ML}_1\text{CON}_{A'}(A, \mathbf{P}, \mathbf{D})$ and using Lemma 5.3.13. For the left-to-right direction we do as we did in Section 5.3.1: we can easily prove this via FO_1 using the translations $(-)^t$ and $(-)_t$. \square

5.3.15. COROLLARY. *A formula $\varphi \in \text{ML}_1(A, \mathbf{P}, \mathbf{D})$ which is monotone in all $a \in A$ (i.e., $\varphi \in \text{ML}_1^+(A, \mathbf{P}, \mathbf{D})$) is continuous in A' iff φ is equivalent to a formula in the basic form $\bigvee_{c \in \wp \mathbf{P}} (\varpi_c \wedge \bigvee \bigwedge_{\ell} \nabla_{\text{ML}}^+(\Sigma, \Pi)_{\ell})$ for some types $\Sigma, \Pi \subseteq \wp A$ such that $A' \cap \bigcup \Pi = \emptyset$.*

5.3.4 One-step additivity

5.3.16. THEOREM. *A formula of $\text{ML}_1(A, \mathbf{P}, \mathbf{D})$ is completely additive in $A' \subseteq A$ iff it is equivalent to a formula given by the following grammar:*

$$\begin{aligned} \varphi &::= \psi \mid \varphi \vee \varphi \mid \varphi \wedge \psi \mid \langle \ell \rangle \alpha \\ \alpha &::= a \mid \alpha \wedge \alpha' \mid \alpha \vee \alpha \end{aligned}$$

where $\psi \in \text{ML}_1(A \setminus A', \mathbf{P}, \mathbf{D})$, α' is A' -free, $p \in \mathbf{P}$, $a \in A'$ and $\ell \in \mathbf{D}$. Observe that the propositions are considered in the clause ψ . We denote this fragment as $\text{ML}_1\text{ADD}_{A'}(A, \mathbf{P}, \mathbf{D})$.

The theorem will follow from the next two lemmas.

5.3.17. LEMMA. *Every $\varphi \in \text{ML}_1\text{ADD}_{A'}(A, \mathbf{P}, \mathbf{D})$ is completely additive in A' .*

Proof. This can be easily proved by induction. \square

5.3.18. LEMMA. *A formula $\varphi \in \text{ML}_1(A, \mathbf{P}, \mathbf{D})$ is completely additive in $A' \subseteq A$ iff it is equivalent to a formula in the basic form*

$$\bigvee_{c \in \wp \mathbf{P}} (\varpi_c \wedge \bigvee \bigwedge_{\ell} \nabla_{\text{ML}}^{A'}(\Sigma, \Pi)_{\ell})$$

where $\Sigma, \Pi \subseteq \wp(A)$ and for every disjunct $\bigwedge_{\ell} \nabla_{\text{ML}}^{A'}(\Sigma, \Pi)_{\ell}$,

1. At most one action $\underline{\ell} \in \mathbf{D}$ may use elements from A' in $\nabla_{\text{ML}}^{A'}(\Sigma, \Pi)_{\underline{\ell}}$,
2. For $\nabla_{\text{ML}}^{A'}(\Sigma, \Pi)_{\underline{\ell}}$ we have that Π is A' -free and there is at most one element of A' in L_Σ .

Proof. The right-to-left direction is trivial by observing that every formula in the above basic form belongs to $\text{ML}_1\text{ADD}_{A'}(A, \mathbf{P}, \mathbf{D})$ and using Lemma 5.3.17. For the left-to-right direction we do as we did in Section 5.3.1: we can easily prove this via FO_1 using the translations $(-)^t$ and $(-)_t$. \square

5.3.19. COROLLARY. A formula $\varphi \in \text{ML}_1(A, \mathbf{P}, \mathbf{D})$ which is monotone in all $a \in A$ (i.e., $\varphi \in \text{ML}_1^+(A, \mathbf{P}, \mathbf{D})$) is completely additive in $A' \subseteq A$ iff it is equivalent to a formula in the basic form

$$\bigvee_{c \in \wp \mathbf{P}} (\varpi_c \wedge \bigvee_{\ell} \bigwedge \nabla_{\text{ML}}^+(\Sigma, \Pi)_{\ell})$$

where $\Sigma, \Pi \subseteq \wp(A)$ and for every disjunct $\bigwedge_{\ell} \nabla_{\text{ML}}^+(\Sigma, \Pi)_{\ell}$,

1. At most one action $\underline{\ell} \in \mathbf{D}$ may use elements from A' in $\nabla_{\text{ML}}^+(\Sigma, \Pi)_{\underline{\ell}}$,
2. For $\nabla_{\text{ML}}^+(\Sigma, \Pi)_{\underline{\ell}}$ we have that Π is A' -free and there is at most one element of A' in L_Σ .

5.3.5 Dual fragments

In this section we give syntactic characterizations of the co-continuous and completely multiplicative fragments of ML_1 . We first define the notion of Boolean dual for one-step formulas interpreted over transition systems.

5.3.20. DEFINITION. Two formulas $\varphi, \psi \in \text{ML}_1(A, \mathbf{P}, \mathbf{D})$ are each other's *Boolean dual* (w.r.t. A) if for every labeled transition system \mathbb{S} and $V : A \rightarrow \wp(S)$ we have:

$$\mathbb{S}, V \Vdash \varphi \quad \text{iff} \quad \mathbb{S}, V^c \not\Vdash \psi,$$

where V^c is the valuation given by $V^c(a) := S \setminus V(a)$, for all a .

To define syntactic fragments for the dual notions we first give a concrete definition of the dualization operator of Definition 5.3.20 and then show that the one-step language ML_1 is closed under Boolean duals.

5.3.21. DEFINITION. The *dual* $\varphi^\delta \in \text{ML}_1(A, \mathbf{P})$ of $\varphi \in \text{ML}_1(A, \mathbf{P})$ is given by:

$$\begin{array}{ll} (a)^\delta := a & (\neg a)^\delta := \neg a \\ (\top)^\delta := \perp & (\perp)^\delta := \top \\ (p)^\delta := p & (\neg p)^\delta := \neg p \\ (\varphi \wedge \psi)^\delta := \varphi^\delta \vee \psi^\delta & (\varphi \vee \psi)^\delta := \varphi^\delta \wedge \psi^\delta \\ (\langle \ell \rangle \psi)^\delta := [\ell] \psi^\delta & ([\ell] \psi)^\delta := \langle \ell \rangle \psi^\delta \end{array}$$

The proof of the following Proposition is a routine check.

5.3.22. PROPOSITION. *For every $\varphi \in \text{ML}_1(A, \mathbf{P}, \mathbf{D})$, φ and φ^δ are Boolean duals.*

We are now ready to give the syntactic definition of the dual fragments.

5.3.23. DEFINITION. The fragment $\text{ML}_1\overline{\text{CON}}_{A'}(A, \mathbf{P}, \mathbf{D})$ is given by the formulas φ of the following grammar:

$$\begin{aligned}\varphi &::= \psi \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid [\ell]\alpha \\ \alpha &::= a \mid \alpha \vee \alpha \mid \alpha \wedge \alpha\end{aligned}$$

where $\psi \in \text{ML}_1(A \setminus A', \mathbf{P}, \mathbf{D})$, $p \in \mathbf{P}$, $a \in A'$ and $\ell \in \mathbf{D}$. Observe that the propositions are considered in the clause ψ .

The fragment $\text{ML}_1\text{MUL}_{A'}(A, \mathbf{P}, \mathbf{D})$ is given by the formulas φ of the following grammar:

$$\begin{aligned}\varphi &::= \psi \mid \varphi \wedge \varphi \mid \varphi \vee \psi \mid [\ell]\alpha \\ \alpha &::= a \mid \alpha \vee \alpha' \mid \alpha \wedge \alpha\end{aligned}$$

where $\psi \in \text{ML}_1(A \setminus A', \mathbf{P}, \mathbf{D})$, α' is A' -free, $p \in \mathbf{P}$, $a \in A'$ and $\ell \in \mathbf{D}$. Observe that the propositions are considered in the clause ψ .

The following proposition states that the above fragments are actually the duals of the fragments defined earlier in this chapter.

5.3.24. PROPOSITION. *The following hold:*

$$\begin{aligned}\text{ML}_1\overline{\text{CON}}_{A'}(A, \mathbf{P}, \mathbf{D}) &= \{\varphi \mid \varphi^\delta \in \text{ML}_1\text{CON}_{A'}(A, \mathbf{P}, \mathbf{D})\} \\ \text{ML}_1\text{MUL}_{A'}(A, \mathbf{P}, \mathbf{D}) &= \{\varphi \mid \varphi^\delta \in \text{ML}_1\text{ADD}_{A'}(A, \mathbf{P}, \mathbf{D})\}\end{aligned}$$

As a corollary, we get a characterization for co-continuity and multiplicativity.

5.3.25. COROLLARY.

- (i) *A formula $\varphi \in \text{ML}_1(A, \mathbf{P}, \mathbf{D})$ is co-continuous in $A' \subseteq A$ if and only if it is equivalent to some $\varphi' \in \text{ML}_1\overline{\text{CON}}_{A'}(A, \mathbf{P}, \mathbf{D})$.*
- (ii) *A formula $\varphi \in \text{ML}_1(A, \mathbf{P}, \mathbf{D})$ is completely multiplicative in $A' \subseteq A$ if and only if it is equivalent to some $\varphi' \in \text{ML}_1\text{MUL}_{A'}(A, \mathbf{P}, \mathbf{D})$.*

5.4 Effectiveness of the normal forms

In this section we briefly discuss the computability of the normal forms of this chapter. This includes the monotone, completely additive, completely multiplicative and continuous normal forms.

5.4.1. PROPOSITION. *The normal forms given in Table 5.1 are effective.*

The ingredients needed to calculate the normal forms for FO_1 , FOE_1 and FOE_1^∞ basically boil down to the following:

- (1) Decidability of the satisfiability problem for FO_1 , FOE_1 and FOE_1^∞ ,
- (2) Bound on the size of the normal forms.

The first item is proved in [Beh22, Löw15] for FO_1 and FOE_1 ; in [Mos57, Theorem 10] for FOE_1^∞ ; and, among others, in [BRV01, Chapter 6] for ML_1 (actually, for the full language ML). For the second item, the existence of such a bound can be induced from each normal form theorem.

As an example, we show how to calculate the normal form of arbitrary multi-sorted formulas of FOE_1 and, as well, the normal form for monotone formulas of multi-sorted FOE_1 . The other cases are similar left to the reader.

According to Theorem 5.2.17, every $\varphi \in \text{FOE}_1(A, \mathcal{S})$ is equivalent to a formula of the form $\bigvee \bigwedge_{\mathcal{S}} \nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)_{\mathcal{S}}$ where for each conjunct $\overline{\mathbf{T}} \in \wp(A)^k$ for some k and $\Pi \subseteq \overline{\mathbf{T}}$. We non-deterministically guess the number of disjuncts and parameters k , Π and $\overline{\mathbf{T}}$ for each conjunct and repeatedly check whether the formulas φ and $\bigvee \bigwedge_{\mathcal{S}} \nabla_{\text{FOE}}(\overline{\mathbf{T}}, \Pi)_{\mathcal{S}}$ are equivalent. This check can be done because FOE_1 is decidable: in [Beh22, Löw15] it is proved that *unsorted* FOE_1 is decidable. Multi-sorted FOE_1 can be reduced to unsorted FOE_1 by introducing new predicates for the sorts (a standard trick).

Next, suppose that we have some $\varphi \in \text{FOE}_1(A, \mathcal{S})$ and we want to obtain an equivalent formula in the corresponding monotone normal form given by Corollary 5.2.25; that is, belonging to $\text{FOE}_1\text{MON}_{A'}(A, \mathcal{S})$. First, we calculate the normal form for φ , as we did in the last paragraph. After that, to calculate the *monotone* normal form, we want to take the biggest possible set $A' \subseteq A$. Observe that the number of such sets is bounded (since A is finite) so we can non-deterministically guess A' . For each potential A' , we apply the translation $(-)^{\circlearrowleft}_{A'}$ of Lemma 5.2.24 to φ , and keep $\varphi_{A'}^{\circlearrowleft}$ only if $\varphi_{A'}^{\circlearrowleft} \equiv \varphi^{\circlearrowleft}$.

5.4.2. REMARK. It is worth observing that the application of the translation $(-)^{\circlearrowleft}_{A'}$ is clearly effective, because of the simplicity of its definition; this is also the case for all the other translations that we use, i.e., for additivity, continuity, etc. Hence, the crucial point (regarding effectiveness) is the use of the satisfiability procedure to check the equivalence $\varphi_{A'}^{\circlearrowleft} \equiv \varphi^{\circlearrowleft}$.

5.5 Conclusions and open problems

In this chapter we defined and studied several first-order and modal one-step languages. We gave normal forms and characterized their monotone, continuous, completely additive and dual fragments. A detailed summary of the results is given in Table 5.1.

Open problems.

1. “*Fill in the gaps*”: an obvious open problem for this chapter is to give the characterizations that are missing in Table 5.1. In particular, we have not characterized the continuous fragment of FOE_1 nor the completely additive fragment of FOE_1^∞ . Knowing these one-step languages better would be the first step to later studying automata that use them. For example, in this dissertation we study $\text{Aut}_{wc}(\text{FOE}_1^\infty)$ in connection with WMSO but we do not know how these formalisms relate to $\text{Aut}_{wc}(\text{FOE}_1)$.

Chapter 6

Concrete modal automata

The main objective of this chapter is to give automata characterizations for $\mu_c\text{ML}$, test-free PDL and full PDL. We do it by showing that these languages precisely correspond to the classes $\text{Aut}_{wc}(\text{ML}_1)$, $\text{Aut}_{wa}^-(\text{ML}_1)$ and $\text{Aut}_{wa}(\text{ML}_1)$, respectively. These results are obtained via effective transformations from formulas to automata and vice-versa.

All the aforementioned classes of automata are based on the one-step language $\text{ML}_1(A)$, which allows Boolean combinations of elements from a to be under the modalities (*cf.* Definition 5.3.1) As a byproduct, we will also prove that such Boolean combinations under the modalities are not needed, and that it suffices to have $\heartsuit a$ for $a \in A$.

6.0.1. DEFINITION. The set $\text{ML}_1^b(A, \mathbf{P}, \mathbf{D})$ of *flat* one-step modal formulas is given by the following clauses:

$$\varphi ::= \top \mid \perp \mid p \mid \neg p \mid \varphi \vee \varphi \mid \varphi \wedge \varphi \mid \langle \ell \rangle a \mid [\ell] a$$

where $p \in \mathbf{P}$, $a \in A$ and $\ell \in \mathbf{D}$. As $\text{ML}_1^b(A, \mathbf{P}) \subseteq \text{ML}_1(A, \mathbf{P})$ we assume the positive, continuous, completely additive and dual fragments defined as expected.

It is known that, for μML , we have $\mu\text{ML} \equiv \text{Aut}(\text{ML}_1) \equiv \text{Aut}(\text{ML}_1^b)$, see for example [KV09, Section 5.3] and [Ven11]. In this chapter, we will prove that the same relationship holds for the subclasses of automata under consideration.

The final section of this chapter regards the connection between automata based on ML_1 and automata based on FO_1 . As we observed in Chapter 4, we would like to use both kinds of automata interchangeably, depending on the task. It is folklore that $\text{Aut}(\text{FO}_1) \equiv \text{Aut}(\text{ML}_1)$, since it can be recovered from various results present in the literature. We start by giving a self-contained proof of this fact and discuss how to transfer this result to subclasses of parity automata.

6.1 Automata for test-free PDL

In this section we will give a class of automata corresponding to test-free PDL. This logic will not play a role further in this dissertation but, as it is a simpler version of PDL, it will be useful to introduce the main ingredients and proof methods.

We saw in Section 3.1.2 that PDL is strongly related to the notion of complete additivity. Later in this chapter we will prove that the concrete automata $Aut_{wa}(ML_1)$ correspond to PDL. In order to give automata for the test-free fragment of PDL, we will consider a more constrained version of $Aut_{wa}(ML_1)$, which we promptly introduce.

6.1.1. DEFINITION. The fragment $ML_1ADD_{A'}^-(A, P, D)$ of $ML_1(A, P, D)$ is given by the following grammar:

$$\varphi ::= \psi \mid \varphi \vee \varphi \mid \langle \ell \rangle \alpha \qquad \alpha ::= a \mid \alpha \vee \alpha,$$

where $\psi \in ML_1(A \setminus A', P, D)$, $p \in P$, $a \in A'$ and $\ell \in D$. Observe that the propositions are considered in the clause ψ . The dual fragment $ML_1MUL_{A'}^-(A, P, D)$ is defined as expected. The positive counterpart of these fragments are given by $ML_1^+ADD_{A'}^- := ML_1^+ \cap ML_1ADD_{A'}^-$ and $ML_1^+MUL_{A'}^- := ML_1^+ \cap ML_1MUL_{A'}^-$.

In other words, the fragment $ML_1ADD_{A'}^-$ is obtained from $ML_1ADD_{A'}$ by disallowing the use of conjunction (outside ψ). On the PDL side, this will mean precisely that we cannot use tests. The leading intuition is obtained from the equivalence $\langle \alpha? \rangle \psi \equiv \alpha \wedge \psi$.

6.1.2. DEFINITION. The class $Aut_{wa}^-(\mathcal{L})$ is given by the parity automata \mathbb{A} from $Aut(\mathcal{L})$ such that for every maximal strongly connected component $C \subseteq A$ and states $a, b \in C$ the following conditions hold:

(weakness) $\Omega(a) = \Omega(b)$,

(tf-additivity)

If $\Omega(a)$ is odd then $\Delta(a) \in \mathcal{L}^+ADD_C^-(A, P, D)$.

If $\Omega(a)$ is even then $\Delta(a) \in \mathcal{L}^+MUL_C^-(A, P, D)$.

A PDL^{tf}-automaton is an automaton from $Aut_{wa}^-(ML_1)$.

The main theorem of this section states that PDL^{tf}-automata characterize PDL^{tf}. It will be proved in the two following subsections.

6.1.3. THEOREM. *The following formalisms are effectively equivalent:*

(i) PDL^{tf},

(ii) $Aut_{wa}^-(ML_1^b)$,

(iii) $Aut_{wa}^-(ML_1)$.

The implication from (i) to (ii) will be proved in Lemma 6.1.4 and the implication from (iii) to (i) will be proved in Lemma 6.1.9. The remaining implication from (ii) to (iii) is trivial since $ML_1^b \subseteq ML_1$.

6.1.1 From formulas to automata

In this section we will transform formulas of PDL^{tf} into equivalent automata.

6.1.4. LEMMA. *Given a formula $\varphi \in PDL^{tf}$ we can effectively construct an equivalent automaton $\mathbb{A}_\varphi \in Aut_{wa}^-(ML_1^b)$.*

To begin, we shall consider the case of formulas of the form $\varphi = \langle \pi \rangle \alpha$ and, for the moment, assume that we already have an automaton $\mathbb{A}_\alpha \equiv \alpha$. It is in our interest to understand how the operation $\langle \pi \rangle$ changes \mathbb{A} to get an automaton for φ itself. First, we analyze how to represent π itself as a PDL^{tf} -automaton and then we will see how to combine it with \mathbb{A}_α .

In this subsection we will briefly use non-deterministic finite-state automata (NFA). Recall that an NFA is a tuple $\underline{A} = \langle A, \delta, F, a_I \rangle$ where $\delta : A \times D \rightarrow \wp A$ is the transition map, $F \subseteq A$ are the final states and $a_I \in A$ is the initial state. Given a model \mathbb{S} , an NFA denotes a set of *paths* through \mathbb{S} . We formalize the acceptance of a path with the following game.

6.1.5. DEFINITION. Given a transition system \mathbb{S} and an NFA $\underline{A} = \langle A, \delta, F, a_I \rangle$ we define the rules for the *acceptance game* $\mathcal{A}(\underline{A}, \mathbb{S})$ having as basic positions pairs $(a, s) \in A \times S$.

Position	Pl'r	Admissible moves
$(a, s) \in (A \setminus F) \times S$	\exists	$\{(b, t) \mid b \in \delta(a, \ell) \text{ and } R_\ell(s, t) \text{ for some } \ell \in D\}$
$(f, s) \in F \times S$	\exists	$\{\text{end}\} \cup \{(b, t) \mid b \in \delta(f, \ell) \text{ and } R_\ell(f, t) \text{ for some } \ell \in D\}$
end	\forall	\emptyset

Finite matches are lost by the player who gets stuck. Infinite matches are won by \forall . A path $\bar{s} \in S^+$ is accepted by \underline{A} iff \exists has a winning strategy f for the initialized game $\mathcal{A}(\underline{A}, \mathbb{S})@ (a_I, s)$ such that every f -guided match visits precisely (and in order) the states of \bar{s} .

6.1.6. LEMMA. *For every $\pi \in PDL^{tf}$ there is an automaton $\mathbb{P}_\pi \in Aut_{wa}^-(ML_1^b)$ and a set of states $F \subseteq P_\pi$ such that for all transitions systems \mathbb{S} and $s, t \in S$ the following are equivalent:*

- (i) $R_\pi(s, t)$,
- (ii) \exists has a surviving strategy in the game $\mathcal{A}(\mathbb{P}_\pi, \mathbb{S})@ (a_\pi, s)$ taking her to position (f, t) for some state $f \in F$.

Proof. A program $\pi \in \text{PDL}^{tf}$ is nothing but a regular expression over \mathbf{D} . Regular expressions over \mathbf{D} can be given semantics over labeled transition systems such that they denote a set of paths. Using this approach, we know that there is a non-deterministic finite-state automaton (NFA) which recognizes the same language as π . Let $\underline{A}_\pi = \langle A, \delta, F, a_I \rangle$ be such an automaton.

CLAIM 1. \underline{A}_π accepts the path s, \dots, t iff $R_\pi(s, t)$.

PROOF OF CLAIM. This is straightforward from \underline{A}_π accepting the language denoted by the regular expression π . ◀

Next we define $\mathbb{P}_\pi = \langle P_\pi, \Delta, \Omega, a_\pi \rangle$ where for all $a \in P_\pi$ the transition map is given by:

$$\Delta(a) := \bigvee_{\ell \in \mathbf{D}, b \in \delta(a, \ell)} \langle \ell \rangle b,$$

and the parity map is $\Omega(a) := 1$ for every element.

CLAIM 2. \mathbb{P}_π is a well-defined PDL^{tf} -automaton.

PROOF OF CLAIM. Note that every state appears under a diamond and there are only disjunctions in the transition map. Clearly, this satisfies the additivity restrictions for cycles in \mathbb{P}_π . The weakness condition is trivially satisfied. ◀

CLAIM 3. Let $(a, s) \in A \times S$. The following are equivalent:

1. (b, t) is an admissible move for \exists in $\underline{\mathcal{A}}(\underline{A}_\pi, \mathbb{S})@ (a, s)$,
2. $\{(b, t)\}$ is an admissible move for \exists in $\mathcal{A}(\mathbb{P}_\pi, \mathbb{S})@ (a, s)$.

PROOF OF CLAIM. The claim is clear from the definition of \mathbb{P}_π . ◀

Consider now the tuple (\mathbb{P}_π, F) consisting of the automaton \mathbb{P}_π and the set of final states F . Combining the claims we get that $R_\pi(s, t)$ iff \exists has a surviving strategy in $\mathcal{A}(\mathbb{P}_\pi, \mathbb{S})@ (a_\pi, s)$ taking her to (f, t) for some $f \in F$. This finishes the proof of the lemma. ◻

The above lemma gives us a tuple $(\mathbb{P}_\pi, F) = \langle P_\pi, \Delta_\pi, \Omega_\pi, a_\pi, F \rangle$ which works as a representation of π . We now combine this automaton with the representation of α given by $\mathbb{A}_\alpha = \langle A_\alpha, \Delta_\alpha, \Omega_\alpha, a_\alpha \rangle$, finally yielding an automaton \mathbb{A}_φ for $\varphi = \langle \pi \rangle \alpha$. Define $\mathbb{A}_\varphi := \langle A_\alpha \uplus P_\pi, \Delta, \Omega, a_\varphi \rangle$ where $a_\varphi := a_\pi$, $\Omega := \Omega_\alpha \cup \Omega_\pi$, and the transition map is defined as

$$\Delta(e) := \begin{cases} \Delta_\alpha(e) & \text{if } e \in A_\alpha \\ \Delta_\pi(e) & \text{if } e \in P_\pi \setminus F \\ \Delta_\pi(e) \vee \Delta(a_\alpha) & \text{if } e \in F. \end{cases}$$

6.1.7. REMARK. Observe that the construction has the following properties

- (i) \mathbb{A}_φ is a well-defined automaton belonging to $Aut_{wa}^-(ML_1^b)$,
- (ii) You can only go from the \mathbb{P}_π part to the \mathbb{A} part from a state in $F \subseteq P$,
- (iii) Once you leave the \mathbb{P}_π part you cannot come back.

Now we prove that \mathbb{A}_φ is actually an automaton representation of $\varphi = \langle \pi \rangle \alpha$.

6.1.8. PROPOSITION. $\mathbb{S} \Vdash \langle \pi \rangle \alpha$ iff \exists has a winning strategy in $\mathcal{A}(\mathbb{A}_\varphi, \mathbb{S}) @ (a_\varphi, s_I)$.

Proof. \Rightarrow Suppose $\mathbb{S}, s_I \Vdash \langle \pi \rangle \alpha$. By definition there is $t \in S$ such that $R_\pi(s_I, t)$ and $\mathbb{S}, t \Vdash \alpha$. Using Lemma 6.1.6 we know that therefore \exists has a surviving strategy in $\mathcal{A}(\mathbb{P}_\pi, \mathbb{S}) @ (a_\pi, s_I)$ taking her to (f, t) for some $f \in F$. Now \exists can use that strategy to play a match in $\mathcal{A}(\mathbb{A}_\varphi, \mathbb{S}) @ (a_\varphi, s_I)$ and get to the same position (f, t) . By inductive hypothesis (as $\mathbb{S}, t \Vdash \alpha$) we know that \exists has a winning strategy in $\mathcal{A}(\mathbb{A}, \mathbb{S}) @ (a_\alpha, t)$. Because of the way the transition map Δ is defined, she can use that same strategy to win $\mathcal{A}(\mathbb{A}_\varphi, \mathbb{S}) @ (f, t)$.

\Leftarrow Suppose that \exists has a winning strategy in $\mathcal{A}(\mathbb{A}_\varphi, \mathbb{S}) @ (a_\varphi, s_I)$. As the parity of \mathbb{P}_π is 1 for every element this means that \exists plays finitely many moves in \mathbb{P}_π which get her to some position (f, t) and then makes a move which takes her to the \mathbb{A} part of the automaton. Observe that this can only happen if $f \in F$. Using Lemma 6.1.6 we get that $R_\pi(s_I, t)$. As \exists has a winning strategy in $\mathcal{A}(\mathbb{A}_\varphi, \mathbb{S}) @ (f, t)$ and because of how Δ is defined, she can use that same strategy to win the game $\mathcal{A}(\mathbb{A}, \mathbb{S}) @ (f, t)$ and thus by inductive hypothesis we get that $\mathbb{S}, t \Vdash \alpha$. By definition, this means that $\mathbb{S}, s_I \Vdash \langle \pi \rangle \alpha$. \square

This finishes the proof of Lemma 6.1.4 for the particular case of $\varphi = \langle \pi \rangle \alpha$ when we already have an automaton for α . We now turn to the general case. We shall prove that for every $\varphi \in \text{PDL}^{tf}$ we can give an equivalent automaton $\mathbb{A}_\varphi \in \text{Aut}_{wa}^-(ML_1^b)$, by induction on φ . If φ is a propositional variable (or its negation) then we can easily give a trivial one-state automaton $\mathbb{A}_\varphi := \langle \{a_\varphi\}, \Delta, \Omega, a_\varphi \rangle$ with $\Delta(a_\varphi) := \varphi$ and any Ω . If $\varphi = \alpha \wedge \beta$ let \mathbb{A}_α and \mathbb{A}_β be automata for α and β . It is straightforward to check that the obvious automaton $\mathbb{A}_\varphi := \mathbb{A}_\alpha \uplus \mathbb{A}_\beta \uplus \{a_\varphi\}$ with initial state a_φ and $\Delta(a_\varphi) := \Delta_\alpha(a_\alpha) \wedge \Delta_\beta(a_\beta)$ is a well-defined automaton for $\alpha \wedge \beta$. The case for the disjunction is analogous. If $\varphi = \langle \pi \rangle \alpha$, Proposition 6.1.8 gives us the required automaton. If $\varphi = [\pi] \alpha$ the construction and proofs are dual to the diamond case.

6.1.2 From automata to formulas

In this section we will transform PDL^{tf} -automata into equivalent formulas.

6.1.9. LEMMA. *Given an automaton $\mathbb{A} \in \text{Aut}_{wa}^-(ML_1)$ we can effectively construct an equivalent formula $\varphi_{\mathbb{A}} \in \text{PDL}^{tf}$.*

The key objects of this proof will be the maximal strongly connected components of \mathbb{A} . As we previously observed, these components (or more specifically the cycles) naturally encode the concept of repetition which, in the case of PDL, corresponds to the iteration of programs. This will be the most difficult (and interesting) case when converting automata to formulas.

We start as follows: for every MSCC C and state $b \in C$ we will show how to get an equivalent formula $\varphi_{C,b} \in \text{PDL}^{tf}(\mathbf{P} \uplus O, \mathbf{D})$ where the propositional variables $O := A \setminus C$ correspond to the states outside C .

We do this by encoding the MCC as a set of equations of *extended* PDL^{tf} -formulas and showing that this set can be “solved inside PDL^{tf} ” through a process reminiscent of Gaussian elimination.

6.1.10. DEFINITION. A set of *B-incomplete* σ -equations is a tuple $\mathbf{E} = (E, \xi, \sigma)$ where E is a non-empty, finite set of equations specified by the map $\xi : E \rightarrow \text{PDL}^{tf}(\mathbf{P} \uplus B \uplus E, \mathbf{D})$ and σ is the type of the equations, which can be either μ (corresponding to a least fixpoint interpretation) or ν (corresponding to a greatest fixpoint interpretation). We sometimes specify a set of equations using the notation $\mathbf{E} := \{e_1 \approx \psi_1, \dots, e_n \approx \psi_n\}_\sigma$.

6.1.11. DEFINITION. Given a transition system \mathbb{S} and a set of *B-incomplete* σ -equations $\mathbf{E} = (E, \xi, \sigma)$ we define the rules for the *solution game* $\mathcal{S}(\mathbf{E}, \mathbb{S})$ having as basic positions pairs $(x, s) \in (E \cup B) \times S$.

Position	Player	Admissible moves
$(e, s) \in E \times S$	\exists	$\{V : (E \cup B) \rightarrow \wp S \mid \mathbb{S}, V, s \Vdash \xi(e)\}$
$V : (E \cup B) \rightarrow \wp S$	\forall	$\{(x, s) \mid x \in E \cup B, s \in V(x)\}$

Whenever a position of the form $(b, s) \in B \times S$ is reached, the match is declared a tie; finite matches not ending in a tie are lost by the player that got stuck and infinite matches are won by \exists if $\sigma = \nu$, and by \forall if $\sigma = \mu$.

6.1.12. REMARK. In general, parity automata and *systems* of equations can be seen as two presentations of the same information. However, we consider it of conceptual help to use sets of equations for this specific part.

Let C be an MSCC of \mathbb{A} ; first we consider the case where the parity of C is 1. We turn the information of C into a set of *O-incomplete* μ -equations $\mathbf{C} = (C, \xi, \mu)$ given by $\xi(c) := \Delta(c)$ for all $c \in C$.

6.1.13. PROPOSITION. *The set of equations \mathbf{C} can be assumed to be of the form*

$$\xi(c) = \alpha \vee \bigvee_{u \in U} \langle \pi_u \rangle u \quad \text{for } U \subseteq C \text{ and } \alpha, \pi_u \in \text{PDL}^{tf}(\mathbf{P} \uplus O, \mathbf{D}). \quad (*)$$

Proof. First observe that formulas in $\Delta(c)$ originally belong to $\text{ML}_1\text{ADD}_C^-(\text{OUC})$. These formulas allow disjunctions of elements of C under the diamonds; however, using the equivalence $\langle \ell \rangle(a \vee b) \equiv \langle \ell \rangle a \vee \langle \ell \rangle b$ we can pull the disjunctions outside, and using $\langle \pi_1 \rangle u \vee \langle \pi_2 \rangle u \equiv \langle \pi_1 \oplus \pi_2 \rangle u$ we can merge the programs and finally get the required form. \square

This set of equations is equivalent to C in the following sense:

6.1.14. PROPOSITION. *Let $b \in C$ be a state in the component, $o \in O$ be a state outside the component, and $s, t \in S$. The following are equivalent:*

- (i) \exists has a surviving strategy in $\mathcal{A}(\mathbb{A}, \mathbb{S})@(b, s)$ taking her to (o, t) ,
- (ii) \exists has a surviving strategy in $\mathcal{S}(\mathbb{C}, \mathbb{S})@(b, s)$ taking her to (o, t) .

Proof. Straightforward from the definition of the games. \square

For a moment, we forget about MSCCs and focus on sets of equations. Next, we show that if a set of equations satisfies $(*)$ we can solve it inside PDL^{tf} . The proof is basically a game-theoretic version of the one found in [BI08], which is also reminiscent of the transformation of linear grammars into regular expressions.

6.1.15. LEMMA. *Let $\mathbf{E} = (E, \xi, \mu)$ be a set of B -incomplete μ -equations satisfying $(*)$. For all $e \in E$ there exists $\varphi_{E,e} \in \text{PDL}^{tf}(\mathbb{P} \uplus B, \mathbb{D})$ such that for all $b \in B$ and $s, t \in S$ the following are equivalent:*

- (i) \exists has a surviving strategy in $\mathcal{S}(\mathbf{E}, \mathbb{S})@(e, s)$ taking her to (b, t) ,
- (ii) $\mathbb{S}, V, s \Vdash \varphi_{E,e}$ where $V : B \rightarrow \wp S$ is such that $V = \{(b, t)\}$.

Proof. By induction on $|E|$, we show that we can solve this set of equations while preserving $(*)$ and finally getting a formula in $\text{PDL}^{tf}(\mathbb{P} \uplus B, \mathbb{D})$.

For the base case let $E = \{e\}$. We have to consider two cases: if $e \notin \xi(e)$ then $\xi(e) = \alpha$ with $\alpha \in \text{PDL}^{tf}(\mathbb{P} \uplus B, \mathbb{D})$ and we are done. Otherwise, the equation should be of the form $\xi(e) = \alpha \vee \langle \pi \rangle e$. Let $\varphi_{E,e} := \langle \pi^* \rangle \alpha$, it is easy to see that the formula belongs to the right fragment. We now prove that $\varphi_{E,e} \equiv \mathbf{E}$.

CLAIM 1. The following are equivalent:

- \exists has a surviving strategy in $\mathcal{S}(\{e \approx \alpha \vee \langle \pi \rangle e\}_\mu, \mathbb{S})@(e, s)$ taking her to (b, t) .
- $\mathbb{S}, V, s \Vdash \langle \pi^* \rangle \alpha$ where $V : B \rightarrow \wp S$ is such that $V = \{(b, t)\}$.

PROOF OF CLAIM. $\boxed{\Rightarrow}$ As the set of equations is of type μ this means that \exists plays only a finite number of moves, otherwise she would lose. Because of the shape of the set of equations she has to play valuations V_1, \dots, V_k such that in each turn \forall chooses (e, s_i) for some $s_i \in V_i(e)$. After that \exists plays a marking V

such that $t \in V(e)$ and \forall must choose (b, t) . It is clear to observe that the first k rounds induce a π^* -path s, s_1, \dots, s_k and the last round implies that $\mathbb{S}, V, s_k \Vdash \alpha$. It only remains to observe that as \exists can *force* \forall to choose (b, t) then it must be the case that $V = \{(b, t)\}$.

\squareleftarrow Assume $\mathbb{S}, V, s \Vdash \langle \pi^* \rangle \alpha$, then by definition there is an s_k such that $R_{\pi^*}(s, s_k)$ and $\mathbb{S}, m, s_k \Vdash \alpha$. Moreover this means that there are s_1, \dots, s_k such that $R_{\pi}(s_i, s_{i+1})$. We can give a surviving strategy for \exists as follows: first she plays, in order, valuations V_1, \dots, V_k such that $V_i = \{(e, s_i)\}$. These valuations constitute legitimate moves for \exists and constrain \forall to follow the path s, s_1, \dots, s_k . Finally, she plays the valuation V which by hypothesis makes α true at s_k and leaves \forall only one choice, namely (b, t) . \blacktriangleleft

For the inductive case let $E = \{e, e_1, \dots, e_n\}$ with $n > 0$. If $e \notin \xi(e)$ we skip to the next step, otherwise we need to treat this equation first. Let $\xi(e) = \alpha \vee \langle \pi \rangle e \vee \bigvee_{u \in U} \langle \pi_u \rangle u$ be such that $e \notin U$. In order to eliminate e from $\xi(e)$ we create a slightly modified version of \mathbf{E} .

CLAIM 2. Let $\mathbf{E}' := (E, \xi', \mu)$ with $\xi'(e) := \langle \pi^* \rangle \alpha \vee \bigvee_{u \in U} \langle \pi^*; \pi_u \rangle u$ and $\xi'(e_i) := \xi(e_i)$ for all i . For all $s, t \in S$ and $b \in B$, the following are equivalent,

- (i) \exists has a surviving strategy in $\mathcal{S}(\mathbf{E}, \mathbb{S})@(e, s)$ taking her to (b, t) ,
- (ii) \exists has a surviving strategy in $\mathcal{S}(\mathbf{E}', \mathbb{S})@(e, s)$ taking her to (b, t) .

PROOF OF CLAIM. As the two sets only differ on e , it will be enough to show that, given a strategy for \exists , we can simulate the moves made by \exists (when standing at e) in one set of equations using the other set of equations.

\Rightarrow The type of \mathbf{E} is μ , therefore \exists will only play a finite amount of moves. Assume \exists plays, in order, valuations V_1, \dots, V_k such that \forall chooses (e, s_i) on each round and finally plays a valuation V such that \forall chooses (x, s') with $x \neq e$. It is straightforward to check that in \mathbf{E}' she can play V and will also get to (x, s') .

\Leftarrow Suppose \exists plays a valuation V such that it actually makes $\langle \pi^* \rangle \alpha$ true (the case for $\langle \pi^*; \pi_u \rangle u$ is analogous) and \forall chooses (x, s') . This means that there is an R_{π^*} path s, s_1, \dots, s_k and a valuation V_α with $\mathbb{S}, V_\alpha, s_k \Vdash \alpha$. She can simulate this play in \mathbf{E} by playing as follows: first she plays, in order, valuations V_i such that $V_i = \{(e, s_i)\}$; after that she plays V_α . \blacktriangleleft

Having removed e from $\xi(e)$, we still have a formula where other elements of E may occur. We first substitute $\xi'(e)$ into the other equations, setting $\xi'(e_i) := \xi(e_i)[e \mapsto \xi'(e)]$ for all i . It is easy to see that this substitution preserves the behaviour of \mathbf{E}' . Using the distribution laws of the diamond and PDL^{tf} identities, the new formulas can be taken to the normal form in (*). To illustrate the process suppose $\xi(e_i) = \alpha \vee \langle \pi_e \rangle e \vee \bigvee_{u \in U} \langle \pi_u \rangle u$ with $e \notin U$ and $\xi'(e) = \alpha' \vee \bigvee_{u \in U} \langle \pi'_u \rangle u$.¹

¹To simplify the presentation we assume that U is the same in $\xi(e_i)$ and $\xi(e)$. This need not be this way but the process can be easily adjusted to work for the general case.

The formula $\xi'(e_i)$ is then obtained as follows:

$$\begin{aligned}
& \alpha \vee \langle \pi_e \rangle e \vee \bigvee_{u \in U} \langle \pi_u \rangle u && \text{(before replacement)} \\
& \alpha \vee \langle \pi_e \rangle (\alpha' \vee \bigvee_{u \in U} \langle \pi'_u \rangle u) \vee \bigvee_{u \in U} \langle \pi_u \rangle u && \text{(after replacement)} \\
& (\alpha \vee \langle \pi_e \rangle \alpha') \vee \bigvee_{u \in U} \langle \pi_e \rangle \langle \pi'_u \rangle u \vee \langle \pi_u \rangle u && \text{(distribution of diamonds, regrouping)} \\
& (\alpha \vee \langle \pi_e \rangle \alpha') \vee \bigvee_{u \in U} \langle \pi_e; \pi'_u \oplus \pi_u \rangle u && \text{(program identities)}
\end{aligned}$$

We inductively solve the smaller set of equations $\mathbf{E}'' := (E \setminus \{e\}, \xi', \mu)$ and get formulas ψ_u for every $u \in E \setminus \{e\}$. Finally we give a solution for e setting $\varphi_{E,e} := \xi'(e)[u \mapsto \psi_u \mid u \in E \setminus \{e\}]$. Observe that $\varphi_{E,e} \in \text{PDL}^{tf}(\mathbf{P} \uplus B, \mathbf{D})$ because it is of the form $\alpha \vee \bigvee_{u \in U} \langle \pi_u \rangle \psi_u$ where (by induction and hypothesis) we have $\alpha, \psi_u \in \text{PDL}^{tf}(\mathbf{P} \uplus B, \mathbf{D})$. \square

It only remains to apply the above results to \mathbf{C} to get the required formula.

6.1.16. COROLLARY. *For every MSCC C and $b \in C$ there is a formula $\varphi_{C,b} \in \text{PDL}^{tf}(\mathbf{P} \uplus O, \mathbf{D})$ such that for all $o \in O$ and $s, t \in S$ the following are equivalent:*

- (i) \exists has a surviving strategy in $\mathcal{A}(\mathbb{A}, \mathbb{S})@(b, s)$ taking her to (o, t) ,
- (ii) $\mathbb{S}, V, s \Vdash \varphi_{C,b}$ where $V : O \rightarrow \wp S$ is such that $V = \{(o, t)\}$.

Proof. Direct from the combination of Proposition 6.1.14 and Lemma 6.1.15 applied to \mathbf{C} . \square

The above Corollary gives us a formula when the connected component has parity 1. The case where the parity of C is 0 is solved in a similar way. First we create a set of ν -equations $\mathbf{C} = (C, \xi, \nu)$ with formulas of the form $\xi(c) = \alpha \wedge \bigwedge_{u \in U} [\pi_u]u$ with $U \subseteq C$ and $\alpha, \pi_u \in \text{PDL}^{tf}(\mathbf{P} \uplus O, \mathbf{D})$. The key identity in this case is that a system with only equation $b \approx \alpha \wedge [\pi]b$ is equivalent to $b \approx [\pi^*]\alpha$.

Now that we can get a formula for every point of an MSCC we turn to the general case. In order to create a formula from an initialized automaton we introduce the following concept.

6.1.17. DEFINITION. Given a parity automaton \mathbb{A} , the *DAG of connected components* of \mathbb{A} is the pair $\text{DCC}(\mathbb{A}) = (G, E)$ where G is the set of maximal strongly connected \preceq -components of \mathbb{A} and $(C_1, C_2) \in E$ iff there are $a \in C_1, b \in C_2$ such that $a \rightsquigarrow b$ and $C_1 \neq C_2$.

6.1.18. REMARK. Observe that this definition considers the connected components given by \preceq and not by \prec . This will result in a DAG where each node is either a \prec -connected component or a single element $a \in A$ which does not belong to any \prec -cycle. Another way to see this DAG is as the quotient of A by the equivalence relation induced by \preceq .

Another observation is that, even though $\text{DCC}(\mathbb{A})$ may not be a tree, it certainly contains no loops. Therefore E is well-founded and, given $C \in G$, we can associate a notion of height to the subgraph generated by C .

We are now ready to prove the main theorem of this section.

Proof of Lemma 6.1.9. For every initialized automaton $\mathbb{A} \in \text{Aut}_{wa}^-(\text{ML}_1)$ we give an equivalent formula $\varphi_{\mathbb{A}} \in \text{PDL}^{tf}$. The proof will be done by induction on the height of the subgraph of $\text{DCC}(\mathbb{A})$ generated by a_I .

If the height of the subgraph is 1, then it is composed of a single MSCC C and $a_I \in C$. By Corollary 6.1.16 we get a formula $\varphi_{C,a_I} \in \text{PDL}^{tf}(\mathbf{P} \uplus O, \mathbf{D})$. We only have to observe that, because C is not connected to any other MSCC then $O = \emptyset$. Hence, we have a formula $\varphi_{C,a_I} \in \text{PDL}^{tf}(\mathbf{P}, \mathbf{D})$ which is equivalent to \mathbb{A} .

Suppose the height of the subgraph is $n > 1$ and $a_I \in C$ for some MSCC C . Again by Corollary 6.1.16 we get a formula $\varphi_{C,a_I} \in \text{PDL}^{tf}(\mathbf{P} \uplus O, \mathbf{D})$ where $O = \{o_1, \dots, o_k\}$ and $o_i \in C_i$ for some MSCCs C_i . By inductive hypothesis we get formulas $\varphi_{\mathbb{A},o_i} \in \text{PDL}^{tf}(\mathbf{P}, \mathbf{D})$. It is straightforward to check that the formula $\varphi_{\mathbb{A},a_I} := \varphi_{C,a_I}[o_i \mapsto \varphi_{\mathbb{A},o_i} \mid i \leq k]$ is equivalent to \mathbb{A} . \square

6.2 Automata for PDL

In this section we give automata corresponding to full PDL.

6.2.1. DEFINITION. A PDL-automaton is an automaton from $\text{Aut}_{wa}(\text{ML}_1)$.

The main theorem of this section states that PDL-automata characterize PDL and, additionally, that these automata can be assumed to be based on a flat modal language, without loss of generality.

6.2.2. THEOREM. *The following formalisms are effectively equivalent:*

- (i) PDL,
- (ii) $\text{Aut}_{wa}(\text{ML}_1^b)$,
- (iii) $\text{Aut}_{wa}(\text{ML}_1)$.

This theorem will be proved in the two following subsections. The implication from (i) to (ii) will be proved in Lemma 6.2.3 and the implication from (iii) to (i) will be proved in Lemma 6.2.9. The remaining implication is trivial.

6.2.1 From formulas to automata

In this section we transform PDL-formulas to equivalent automata.

6.2.3. LEMMA. *Given a formula $\varphi \in \text{PDL}$ we can effectively construct an equivalent automaton $\mathbb{A}_\varphi \in \text{Aut}_{wa}(\text{ML}_1^b)$.*

We give a proof by induction on φ . If φ is a test-free formula (i.e., $\varphi \in \text{PDL}^{tf}$) we can get the corresponding automaton using Theorem 6.1.3 and observing that every $\text{Aut}_{wa}^-(\text{ML}_1^b) \subseteq \text{Aut}_{wa}(\text{ML}_1^b)$. It is also easy to check that the class $\text{Aut}_{wa}(\text{ML}_1^b)$ is closed by the Boolean operators.

The interesting case, therefore, is $\varphi = \langle \pi \rangle \alpha$ where $\alpha \equiv \mathbb{A}_\alpha \in \text{Aut}_{wa}(\text{ML}_1^b)$ and π makes use of tests. To prove this case we use the following strategy: first we will consider tests as additional atomic actions and get an NFA for π , similar to what we did in Section 6.1.1; after that, we merge it with the automata for the tested formulas to get an automaton \mathbb{P}_π for π . To finish, we combine \mathbb{P}_π and \mathbb{A}_α to get an automaton for φ .

In the process of creating an automaton for π we encounter new complexities because of the presence of tests. To be able to properly define a merging operation we need to introduce the following concepts.

6.2.4. DEFINITION. Let B be a finite set of names such that $A \cap B = \emptyset$ and $\text{P} \cap B = \emptyset$. A *B-incomplete automaton* is an automaton based on the set of propositions $\text{P} \cup B$ such that the elements of B occur only positively in the transition map of \mathbb{A} .

The acceptance games of Definition 2.3.3 are extended for *B-incomplete* automata with the intention to interpret the elements of B as names (as opposed to propositions). Basic positions are then taken from $(A \cup B) \times S$ and valuations are of the type $V : (A \cup B) \rightarrow \wp S$. Whenever a position from $B \times S$ is reached, the match is declared a tie.

6.2.5. DEFINITION. The *completion of a B-incomplete automaton \mathbb{A} with an automaton $\mathbb{A}' = \langle A', \Delta', \Omega' \rangle$* is defined as $(\mathbb{A} \times \mathbb{A}') = \langle C, \Delta_C, \Omega_C, a_I \rangle$ where $C := A \cup A'$, $\Omega_C := \Omega \cup \Omega'$ and the transition map is given by:

$$\Delta_C(c) := \begin{cases} \Delta'(c) & \text{if } c \in A', \\ \Delta(c)[\mathbf{b} \mapsto \Delta'(\mathbf{b}) \mid \mathbf{b} \in B \cap A'] & \text{if } c \in A. \end{cases}$$

Note that the completion can be partial if $B \not\subseteq A'$, in this case the outcome will be $(B \setminus A')$ -incomplete. If $B \subseteq A'$, the outcome will be a complete automaton. Also observe that a completion cannot generate new cycles.

6.2.6. DEFINITION. Given $\pi \in \text{PDL}(\mathbf{P}, \mathbf{D})$ we use $\pi^b \in \text{PDL}^{bf}(\mathbf{P}, \mathbf{D} \cup \mathbf{T})$ to denote the version of π where its top-level tests \mathbf{T} are considered as atomic actions. The \mathbf{T} -extension of a labeled transition system $\mathbb{S} = \langle S, R_{\ell \in \mathbf{D}}, \kappa, s_I \rangle$ is defined as $\mathbb{S}^{\mathbf{T}} := \langle S, R_{\ell \in \mathbf{D}}, R_{\chi \in \mathbf{T}}, \kappa, s_I \rangle$ where $R_{\chi} := \{(s, s) \in S \times S \mid \mathbb{S}, s \Vdash \chi\}$.

6.2.7. LEMMA. For every $\pi \in \text{PDL}$ there is an \mathbf{x} -incomplete $\mathbb{P}_{\pi} \in \text{Aut}_{wa}(\text{ML}_1^b)$ such that for all transition systems \mathbb{S} and $s, t \in S$ the following are equivalent

- (i) $R_{\pi}^{\mathbb{S}}(s, t)$,
- (ii) \exists has a surviving strategy in $\mathcal{A}(\mathbb{P}_{\pi}, \mathbb{S}) @ (a_{\pi}, s)$ taking her to (\mathbf{x}, t) .

Proof. Let \mathbf{T} be the top-level tests appearing in π . The following claim is easy to check.

CLAIM 1. For every LTS \mathbb{S} and $s, t \in S$ we have $R_{\pi}^{\mathbb{S}}(s, t)$ iff $R_{\pi^b}^{\mathbb{S}^{\mathbf{T}}}(s, t)$.

As in Section 6.1.1, we can construct an NFA $\underline{A}_{\pi} = \langle A, \delta, F, a_I \rangle$ which recognizes π^b . By definition of \underline{A}_{π} recognizing π^b we have the following claim.

CLAIM 2. For every transition system \mathbb{S} and $s, t \in S$, the automaton \underline{A}_{π} accepts some path s, \dots, t in $\mathbb{S}^{\mathbf{T}}$ if and only if $R_{\pi^b}^{\mathbb{S}^{\mathbf{T}}}(s, t)$.

CLAIM 3. Without loss of generality we can assume the following on \underline{A}_{π} :

1. Every state has either exiting action transitions or test transitions, but not both. A state is called an *action state* or a *test state* in those circumstances, respectively.
2. Every cycle contains at least one action state.
3. The initial state has no incoming transitions (in particular it does not belong to a cycle).
4. Test transitions always arrive into an action state.

PROOF OF CLAIM. We prove the items as follows:

1. Suppose that there is a (mixed) state a with both test and action transitions. Take an action transition d from a to some state c . We can get an equivalent automaton (from the point of view of the initial state) by creating an intermediate action state c' such that c' has exactly one transition d to c and then transforming the action transition d of a into a dummy test transition $\top?$ from a to c' . Repeating this process on a transforms it into a pure test state.

2. It is not difficult to show that cycles without action states can only be generated by a program of the form $\pi' = \pi^*$, where π is built up from tests only (that is, π contains no atomic actions outside the scope of a top-level test). Therefore, to avoid such cycles, we will show that we can assume that our PDL-formulas do not contain such programs. Note that if π contains no atomic actions then $R_\pi \subseteq S \times S$ can be expressed as the union and composition of relations $R_{\psi_i?}$ corresponding to the rests in π . As $R_{\psi_i?} \subseteq Id_S = \{(s, s) \mid s \in S\}$ it is easy to see that $R_\pi \subseteq Id_S$. From this last observation we get that $R_{\pi^*} = R_\pi^* = Id_S = R_{\top?}$ and hence we can conclude that π' is equivalent to the program $\top?$. Therefore, we can replace any such program π' with $\top?$ and avoid cycles without action states.
3. This item is easily proved by creating a new state a'_I which has the same transitions as a_I and set it as the new initial state.
4. Assume (1–3) and suppose that there is a test state a which has a φ -transition into another test state b with transitions ψ_1, \dots, ψ_n to states c_1, \dots, c_n . We can get an equivalent automaton (from the point of view of the initial state) by replacing the φ -transition of a with n new transitions to c_1, \dots, c_n labeled by $\varphi \wedge \psi_i$ respectively. Repeating this step (and using item 2) ensures item 4.

This finishes the proof of Claim 3. ◀

Let $T = \{\mathbf{a}_\chi \mid \chi \in \mathbb{T}\} \cup \{\mathbf{x}\}$ be a finite set of names. From \underline{A}_π we define an automaton $\mathbb{A}_\pi \in Aut_{wa}(\mathbb{ML}_1^b)$ as $\mathbb{A}_\pi := \langle A_\pi, \Delta_\pi, a_\pi \rangle$ by setting $A_\pi := A$, the parity map is $\Omega(a) := 1$ for all $a \in A$, and the transition map is given by:

$$\Delta_\pi(a) := \begin{cases} \bigvee \{ \langle \ell \rangle b \mid \ell \in \mathbb{D}, b \in \delta(a, d) \} & \text{if } a \in A \setminus F \text{ is an action state,} \\ \mathbf{x} \vee \bigvee \{ \langle \ell \rangle b \mid \ell \in \mathbb{D}, b \in \delta(a, d) \} & \text{if } a \in A \cap F \text{ is an action state,} \\ \bigvee \{ \mathbf{a}_\chi \wedge \Delta_\pi(b) \mid \chi \in \mathbb{T}, b \in \delta(a, \chi) \} & \text{if } a \in A \setminus F \text{ is a test state,} \\ \mathbf{x} \vee \bigvee \{ \mathbf{a}_\chi \wedge \Delta_\pi(b) \mid \chi \in \mathbb{T}, b \in \delta(a, \chi) \} & \text{if } a \in A \cap F \text{ is a test state.} \end{cases}$$

Observe that, although the last two cases use Δ_π recursively in their own definition, they are well-defined because test states only have transitions into actions states (see Claim 3 item 4) and the transition map has already been defined for action states (in the first two cases).

CLAIM 4. $\mathbb{A}_\pi \in Aut_{wa}(\mathbb{ML}_1^b)$ is a well-defined T -incomplete automaton.

PROOF OF CLAIM. By inspecting Δ_π it is simple to see that the transition map lands in the right fragments. ◀

Let $(\mathbb{A}_\chi, a_\chi)_{\chi \in \mathbb{T}}$ be the family of automata for $\mathbb{T} = \{\chi_1, \dots, \chi_k\}$, provided by the inductive hypothesis. To finish the construction let $\mathbb{P}_\pi := \mathbb{A}_\pi \times \mathbb{A}_{\chi_1} \times \dots \times \mathbb{A}_{\chi_k}$, where \times was defined in Definition 6.2.5.

CLAIM 5. For every transition system \mathbb{S} and $s, t \in S$, the following are equivalent.

- (i) \exists has a surviving strategy in $\underline{\mathcal{A}}(\underline{\mathbb{A}}_\pi, \mathbb{S}^\top)@(a_\pi, s)$ taking her to (f, t) with $f \in F$.
- (ii) \exists has a surviving strategy in $\mathcal{A}(\mathbb{P}_\pi, \mathbb{S})@(a_\pi, s)$ taking her to (\mathbf{x}, t) .

PROOF OF CLAIM. To prove this claim, we will show that every move made by \exists in one of the acceptance games can be simulated by one or more moves in the other game.

\Rightarrow Dividing by cases, it is enough to prove that for all $a, b \in A, s, t \in S$,

- (a) If a is a final state then \exists has a surviving strategy in $\mathcal{A}(\mathbb{P}_\pi, \mathbb{S})@(a, s)$ taking her to (\mathbf{x}, s) ,
- (b) If a is an action state and (b, t) is an admissible move for \exists in $\underline{\mathcal{A}}(\underline{\mathbb{A}}_\pi, \mathbb{S}^\top)@(a, s)$ then \exists has a surviving strategy in $\mathcal{A}(\mathbb{P}_\pi, \mathbb{S})@(a, s)$ taking her to (b, t) ,
- (c) If a is a test state and (b, s) is an admissible move for \exists in $\underline{\mathcal{A}}(\underline{\mathbb{A}}_\pi, \mathbb{S}^\top)@(a, s)$ then \exists has a surviving strategy in $\mathcal{A}_s(\mathbb{P}_\pi, \mathbb{S})@(a, s)$ leading to $(\Delta_\pi(b), s)$, which is equivalent to (b, s) .

Item (a) is clear from the definition of Δ_π for final states. That is, \exists can always choose $\{(\mathbf{x}, s)\}$ as a move. In the same way, for item (b), it is also clear that $\{(b, t)\}$ is an admissible move for \exists .

For item (c) we will consider a symmetric definition of the acceptance game (*cf.* Definition 2.3.3) to reason about strategies [Ven11], for the special case of parity automata based on the one-step language ML_1 . We denote this parity game by $\mathcal{A}_s(\mathbb{A}, \mathbb{S})$ and define it as follows²:

Position	Player	Admissible moves	Parity
$(a, s) \in A \times S$	–	$\{(\Delta(a, \kappa(s)), s)\}$	$\Omega(a)$
$(\bigwedge \Phi, s)$	\forall	$\{(\varphi_i, s) \mid \varphi \in \Phi\}$	$\max(\Omega[A])$
$(\bigvee \Phi, s)$	\exists	$\{(\varphi_i, s) \mid \varphi \in \Phi\}$	$\max(\Omega[A])$
$([\ell]b, s)$	\forall	$\{(b, t) \mid R_\ell(s, t)\}$	$\max(\Omega[A])$
$(\langle \ell \rangle b, s)$	\exists	$\{(b, t) \mid R_\ell(s, t)\}$	$\max(\Omega[A])$
(p, s) with $s \in V(p)$	\forall	\emptyset	$\max(\Omega[A])$
(p, s) with $s \notin V(p)$	\exists	\emptyset	$\max(\Omega[A])$
$(\neg p, s)$ with $s \notin V(p)$	\forall	\emptyset	$\max(\Omega[A])$
$(\neg p, s)$ with $s \in V(p)$	\exists	\emptyset	$\max(\Omega[A])$

Suppose a is a test state and (b, s) is an admissible move for \exists in $\underline{\mathcal{A}}(\underline{\mathbb{A}}_\pi, \mathbb{S}^\top)@(a, s)$ going through test χ . In particular, this means that $\mathbb{S}, s \Vdash \chi$ and hence $\mathbb{S}, s \Vdash \mathbb{A}_\chi$.

²As defined in Section 2.3.3, we only care about the *minimum* parity occurring in a match. Therefore, $\max(\Omega[A])$ is used as a “dummy” parity when we want the parity of a particular round to have no effect in the overall match.

Observe that in $\mathcal{A}_s(\mathbb{P}_\pi, \mathbb{S})@(a, s)$ it is \exists 's turn to move and she can always choose the disjunct that corresponds to χ . After that it is \forall 's choice and he is forced to choose $\Delta_\pi(b)$ because if he chose the conjunct corresponding to χ he would lose. Therefore we arrive at position $(\Delta_\pi(b), s)$ which is what we wanted.

$\boxed{\Leftarrow}$ It is enough to prove that for all $a, b \in A, s, t \in S$,

- (a) If $\{(x, s)\}$ is admissible for \exists in $\mathcal{A}(\mathbb{P}_\pi, \mathbb{S})@(a, s)$ then a is a final state in \underline{A}_π ,
- (b) If a is an action state and $\{(b, t)\}$ is an admissible move for \exists in $\mathcal{A}(\mathbb{P}_\pi, \mathbb{S})@(a, s)$ then (b, t) is an admissible move for \exists in $\underline{\mathcal{A}}(\underline{A}_\pi, \mathbb{S}^\top)@(a, s)$,
- (c) If a is a test state and \exists has a surviving strategy in $\mathcal{A}_s(\mathbb{P}_\pi, \mathbb{S})@(a, s)$ leading her to $(\Delta_\pi(b), s)$ then (b, s) is an admissible move for \exists in $\underline{\mathcal{A}}(\underline{A}_\pi, \mathbb{S}^\top)@(a, s)$.

We only prove item (c). Suppose that at position (a, s) player \exists chose a disjunct which (sloppily formulated) corresponds to $\mathbf{a}_\chi \wedge \Delta_\pi(b)$ for some $\chi \in \mathbb{T}$. If her strategy leads \exists to $(\Delta_\pi(b), s)$ it means that \forall did not choose \mathbf{a}_χ . From this we can conclude that $\mathbb{S}, s \Vdash \chi$ since otherwise \forall would have chosen that conjunct and won. Therefore $R_\chi(s, s)$ holds in \mathbb{S}^\top . Moreover, as $\mathbf{a}_\chi \wedge \Delta_\pi(b)$ occurs in $\Delta_\pi(a)$, by definition of Δ_π (see “test” cases) we know that in \underline{A}_π there is a χ -transition to b . Putting these observations together, we can conclude that (b, s) is an admissible move for \exists in $\underline{\mathcal{A}}(\underline{A}_\pi, \mathbb{S}^\top)@(a, s)$. \blacktriangleleft

The lemma is then a corollary of the claims, chained as follows

$$\begin{aligned}
 R_\pi^{\mathbb{S}}(s, t) &\iff R_{\pi^b}^{\mathbb{S}^\top}(s, t) && \text{(Claim 1)} \\
 &\iff \underline{A}_\pi \text{ accepts the path } s, \dots, t \text{ in } \mathbb{S}^\top && \text{(Claim 2)} \\
 &\iff \exists \text{ has a surviving strategy in} && \\
 &\quad \mathcal{A}(\mathbb{P}_\pi, \mathbb{S})@(a_\pi, s) \text{ leading to } (x, t) && \text{(Claim 5)}
 \end{aligned}$$

Done. \square

To finish, we give an automaton for $\varphi = \langle \pi \rangle \alpha$. Let \mathbb{A}_α be the automaton for α , given by the inductive hypothesis, and assume its initial state is called \mathbf{x} . Define $\mathbb{A}_\varphi := \mathbb{P}_\pi \times \mathbb{A}_\alpha$. We prove that \mathbb{A}_φ is an automaton representation of φ .

6.2.8. PROPOSITION. $\mathbb{S} \Vdash \langle \pi \rangle \alpha$ iff \exists has a winning strategy in $\mathcal{A}(\mathbb{A}_\varphi, \mathbb{S})@(a_\varphi, s_I)$.

Proof. The proof is the same as in Proposition 6.1.8 but using Lemma 6.2.7. \square

6.2.2 From automata to formulas

In this section we will transform PDL-automata into equivalent formulas.

6.2.9. LEMMA. *Given an automaton $\mathbb{A} \in \text{Aut}_{wa}(\text{ML}_1)$ we can effectively construct an equivalent formula $\varphi_{\mathbb{A}} \in \text{PDL}$.*

The proof is basically the same as for PDL^{tf}. The crucial difference lies in showing that when we want to solve the system of equations (i.e., an analogue of Lemma 6.1.15) we can still provide a normal form like (*) in Proposition 6.1.13.

6.2.10. PROPOSITION. *Every $\varphi \in \text{ML}_1\text{ADD}_C(O \uplus C)$ is equivalent to a formula of the form:*

$$\gamma \vee \bigvee_{u \in U} \langle \pi_u \rangle u \quad \text{with} \quad U \subseteq C; \gamma, \pi_u \in \text{PDL}(\text{P} \cup O, \text{D})$$

Proof. We prove this by induction on the construction of the formula. In case $\varphi = \psi \in \text{ML}_1(O)$ we just choose $U := \emptyset$ and $\gamma := \psi$.

Suppose $\varphi = \langle \ell \rangle \alpha$ with α a Boolean expression over $O \cup C$ (cf. Theorem 5.3.16). The key observation is that, given the definition of α in this additive fragment, it is not difficult to show that α can be expressed in the disjunctive normal form $\alpha \equiv \bigvee_i (u_i \wedge \psi_i)$ with $\psi_i \in \text{ML}_1(O)$. Therefore, $\varphi \equiv \perp \vee \bigvee_i \langle \psi_i?; \ell \rangle u_i$.

If $\varphi = \varphi_1 \vee \varphi_2$ first assume (by inductive hypothesis) that φ_1, φ_2 are in normal form; it is easy to see that some regrouping and joining of programs converts φ to the required normal form. If $\varphi = \varphi' \wedge \psi$ with $\psi \in \text{ML}_1(O)$ we do as follows. Let $\varphi' \equiv (\gamma \vee \bigvee_{u \in U} \langle \pi_u \rangle u)$ by inductive hypothesis.

$$\begin{aligned} \varphi' \wedge \psi &\equiv (\gamma \vee \bigvee_{u \in U} \langle \pi_u \rangle u) \wedge \psi \\ &\equiv (\gamma \wedge \psi) \vee (\psi \wedge \bigvee_{u \in U} \langle \pi_u \rangle u) && \text{(distribution of } \psi) \\ &\equiv (\gamma \wedge \psi) \vee (\bigvee_{u \in U} \psi \wedge \langle \pi_u \rangle u) && \text{(distribution of } \psi) \\ &\equiv (\gamma \wedge \psi) \vee (\bigvee_{u \in U} \langle \psi?; \pi_u \rangle u) && \text{(conjunction as test)} \end{aligned}$$

This finishes the proof. □

6.3 Automata for $\mu_c\text{ML}$

In this section we give modal parity automata corresponding to $\mu_c\text{ML}$.

6.3.1. THEOREM. *The following formalisms are effectively equivalent:*

- (i) $\mu_c\text{ML}$,
- (ii) $\text{Aut}_{wc}(\text{ML}_1^b)$,
- (iii) $\text{Aut}_{wc}(\text{ML}_1)$.

The implication from (i) to (ii) will be proved in Lemma 6.3.2 and the implication from (iii) to (i) will be proved in Lemma 6.3.10. The remaining implication is trivial.

6.3.1 From formulas to automata

In this section we show how to convert a formula $\varphi \in \mu_c\text{ML}$ into a continuous-weak automata. We will adapt a technique introduced in [Ven11, Section 6.3] for μML and prove the following result.

6.3.2. LEMMA. *Given a formula $\varphi \in \mu_c\text{ML}$ we can effectively construct an equivalent automaton $\mathbb{A}_\varphi \in \text{Aut}_{wc}(\text{ML}_1^b)$.*

While formulas of basic modal logic can be seen as trees, a formula of μML can be seen as tree with back-edges (*cf.* Definition 4.4.1). These back edges go from each bound variable p to its binding definition $\sigma_p p.\delta_p$. The main idea is to start by defining a new kind of automaton based on the tree structure of φ and then massage its structure and transition map to obtain an automaton in $\text{Aut}_{wc}(\text{ML}_1^b)$.

When translating formulas of the μ -calculus to automata it will be useful to assume a special form. Without loss of generality, we will assume that they are in negation normal form and clean. Additionally, we assume that they are guarded, and proceed to show that this can be done without loss of generality.

6.3.3. DEFINITION. An occurrence of a bound variable is called *guarded* if there is a modal operator between its binding definition and the variable itself. An occurrence is *weakly guarded* if there is another fixpoint quantifier between its binding definition and the variable itself. A formula $\varphi \in \mu\text{ML}$ is called *guarded* if every occurrence of every bound variable is guarded.

For instance, consider the following formula (example adapted from [BFL15]): $\mu p.a \vee (\mu q.(a \wedge p) \vee (\neg a \wedge q) \vee \langle \ell \rangle q)$. In this formula, the variable q has both a guarded and an unguarded occurrence, while the only occurrence of p is not guarded but it is weakly guarded.

It is possible to convert any formula of μML into an equivalent formula which is guarded. Although this may induce an exponential blowup in size, this is not a problem for our expressiveness concerns in this section. For an extensive historical overview of guarding methods and a proof of the exponential blowup we refer to [BFL15].

The procedure given by Kupferman et. al [KVW00] uses the following facts to bring the formulas into guarded form. The first is the fixpoint unfolding rule.

6.3.4. FACT. *For every formula $\sigma_p p.\varphi \in \mu\text{ML}$ we have $\sigma_p p.\varphi \equiv \varphi[p \mapsto \sigma_p p.\varphi]$.*

The second rule helps to eliminate spurious unguarded variables under fixpoints. For example, consider the formula $\mu p.p \vee \gamma$. It is easy to see that in the evaluation game, it will never be useful for \exists to choose the left disjunct and therefore we have the equivalences $\mu p.p \vee \gamma \equiv \mu p.\perp \vee \gamma \equiv \mu p.\gamma$.

6.3.5. FACT ([KVV00, MAT02]). *For every formula $\sigma_p p.\varphi \in \mu\text{ML}$ we have that $\sigma_p p.\varphi \equiv \sigma_p p.\varphi[p \mapsto \hat{\sigma}_p \mid p \text{ is not weakly guarded}]$ where $\hat{\sigma}_p := \top$ if $\sigma_p = \nu$ and $\hat{\sigma}_p := \perp$ if $\sigma_p = \mu$.*

The following is obtained by repeatedly applying the above facts starting from the innermost fixpoint.

6.3.6. THEOREM ([KVV00, THEOREM 2.1]). *For every $\varphi \in \mu\text{ML}$ we can effectively construct a guarded formula $\varphi^\sharp \in \mu\text{ML}$ such that $\varphi \equiv \varphi^\sharp$.*

Additionally, it should be observed that this transformation preserves the continuous fragment of the μ -calculus.

6.3.7. PROPOSITION. *If $\varphi \in \mu_c\text{ML}$ then $\varphi^\sharp \in \mu_c\text{ML}$. Hence for every $\varphi \in \mu_c\text{ML}$ we can effectively construct a guarded formula $\varphi^\sharp \in \mu_c\text{ML}$ such that $\varphi \equiv \varphi^\sharp$.*

Proof. If φ belongs to $\mu_c\text{ML}$, a syntactic inspection reveals that the application of any of the above transformations results in a formula which again belongs to $\mu_c\text{ML}$. In the case of Theorem 6.3.6 it is possible that the transformation requires part of the formula to be in conjunctive normal form, but this can also be done inside $\mu_c\text{ML}$, since this fragment allows all Boolean connectives. \square

Now that we have preprocessed our formula $\varphi \in \mu_c\text{ML}$ to make it guarded, we will start the transformation into automata, which will be done in two stages: first we create a continuous-weak automaton \mathbb{A} having as states the subformulas of φ . That is, for every subformula $\varphi \trianglelefteq \alpha$ we will have a state $\hat{\alpha}$ in A . In order to give an easy and direct construction of such an automaton, we will first allow formulas of the transition map to be in the language $\text{ML}_1^b(A, \mathbf{P} \cup A, \mathbf{D})$. The idea is to allow states to occur at the level of propositions (i.e., not necessarily under a modality).³

6.3.8. PROPOSITION. *For every $\varphi \in \mu_c\text{ML}$ we can effectively construct an equivalent automaton $\mathbb{A}_\varphi \in \text{Aut}_{wc}(\text{ML}_1^b(A, \mathbf{P} \cup A, \mathbf{D}))$.*

Proof. Define the automaton as $\mathbb{A}_\varphi := \langle A, \Delta, \Omega, a_I \rangle$ where

$$A := \{\hat{\alpha} \mid \alpha \text{ is a subformula of } \varphi\}.$$

The initial state is $a_I := \hat{\varphi}$, and the transition map is given by:

$$\begin{aligned} \Delta(\widehat{\alpha \vee \beta}) &:= \hat{\alpha} \vee \hat{\beta} & \Delta(\widehat{\alpha \wedge \beta}) &:= \hat{\alpha} \wedge \hat{\beta} \\ \Delta(\widehat{\heartsuit \alpha}) &:= \heartsuit \hat{\alpha} & \Delta(\widehat{\sigma_p p.\alpha}) &:= \hat{\alpha} \end{aligned}$$

³However, states will always be required to have positive polarity.

$$\begin{aligned}\Delta(\widehat{\chi}) &:= \chi \quad \text{for } \chi \in \{\top, \perp, p, \neg p\} \text{ with } \textit{unbound } p, \\ \Delta(\widehat{p}) &:= \widehat{\delta}_p \quad \text{for } \textit{bound } p.\end{aligned}$$

Observe that this definitions is *not* inductive nor recursive and that there are no Boolean operators under modalities.

In order to define the parity map, we introduce the following definition: we say that a subformula α of φ is a μ -subformula (resp. ν -subformula) if there is a subformula $\mu p.\alpha' \sqsubseteq \alpha$ (resp. $\nu p.\alpha' \sqsubseteq \alpha$) of φ and p is free in α . Example: consider the formula $\psi \wedge \mu p.(\langle \ell \rangle p \vee \nu q.[\ell]q)$; in this case $\langle \ell \rangle p$ and p are μ -subformulas, $[\ell]q$ and q are ν -subformulas, and ψ and $\nu q.[\ell]q$ are neither μ - nor ν -formulas.

CLAIM 1. Every subformula α of φ belongs to exactly one of the following categories: (a) no free variable of α is bound in φ ; or (b) α is a μ -subformula; or (c) α is a ν -subformula.

PROOF OF CLAIM. The claim follows from φ being an alternation-free formula (*cf.* Definition 2.4.4), since it belongs to $\mu_c\text{ML}$. \blacktriangleleft

For the parity we set $\Omega(\widehat{\alpha}) := 0$ if α is a ν -subformula of φ and 1 otherwise. The following claim will be useful in proving that the automaton is weak. To state the claim, recall that $a \preceq_{\mathbb{A}} b$ in an automaton \mathbb{A} if b can be reached from a in the graph structure induced by the transition map of \mathbb{A} (*cf.* Definition 2.3.4).

CLAIM 2. If $\widehat{\alpha} \preceq \widehat{\beta} \preceq \widehat{\alpha}$ we can assume, modulo switching the order of the states, that one of the following cases holds (see Fig. 6.1):

$$\begin{array}{lll} \text{(a) } \sigma_p p.\delta_p \sqsubseteq \alpha \sqsubseteq \beta \sqsubseteq p & \text{(b) } \sigma_p p.\delta_p \sqsubseteq \alpha \sqsubseteq p & \text{(c) } \sigma_q q.\delta_q \sqsubseteq \sigma_p p.\delta_p \sqsubseteq \alpha \sqsubseteq q \\ & \sqsubseteq \beta \sqsubseteq p & \sqsubseteq \beta \sqsubseteq p, \end{array}$$

where $p \in \text{FV}(\alpha) \cap \text{FV}(\beta)$ in cases (a) and (b); and $q \in \text{FV}(\alpha)$, $p \in \text{FV}(\beta)$ in (c). In particular, both α and β are σ -subformulas of φ of the same type σ .

PROOF OF CLAIM. Observe that because φ is alternation-free then we must have $\sigma_p = \sigma_q$ in (c). \blacktriangleleft

We prove that the automaton is weak.

CLAIM 3. \mathbb{A}_φ is a weak automaton.

PROOF OF CLAIM. Let $\widehat{\alpha}$ and $\widehat{\beta}$ belong to some maximal connected component, we have to show that $\Omega(\widehat{\alpha}) = \Omega(\widehat{\beta})$. Using Claim 2 we can see that both α and β will be σ -subformulas of the same type and hence have the same parity. \blacktriangleleft

We prove that the automaton satisfies the continuity condition.

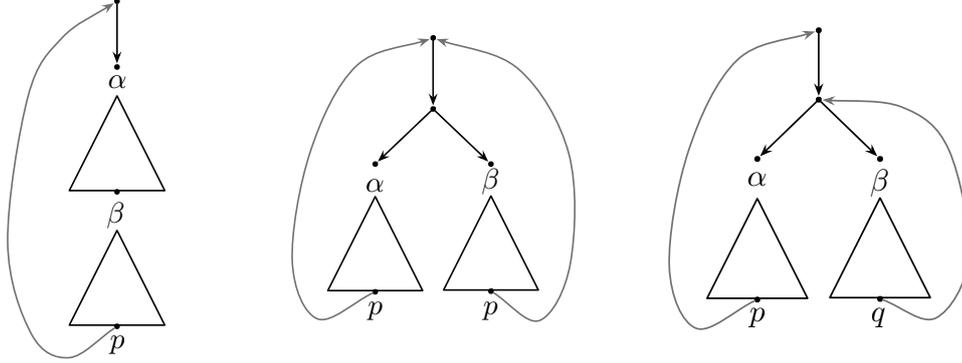


Figure 6.1: Relative positioning of α and β if $\hat{\alpha} \preceq \hat{\beta} \preceq \hat{\alpha}$.

CLAIM 4. \mathbb{A}_φ satisfies the continuity condition.

PROOF OF CLAIM. We only prove the continuity condition for the case of a maximal connected component with parity 1 (corresponding to a least fixpoint). The case of parity 0 (corresponding to a greatest fixpoint) is dual. Let $\hat{\alpha}$ and $\hat{\beta}$ belong to some maximal connected component C with parity 1, we have to show that $\Delta(\hat{\alpha})$ is continuous in C . For continuity, this is equivalent to proving that $\Delta(\hat{\alpha})$ is continuous in all $\hat{\beta} \in C$.

We consider the relative positioning of α and β in light of Claim 2, assuming that $\sigma_p = \sigma_q = \mu$. In cases (b) and (c) it is obvious that $\Delta(\hat{\alpha})$ is continuous in $\hat{\beta}$, since $\hat{\beta}$ does not occur in $\Delta(\hat{\alpha})$. For case (a) we do as follows: note that $\delta_p \in \mu\text{MLCON}_p$ by definition of $\mu_c\text{ML}$. This implies that in the formula tree of φ , if we go from p to δ_p we will never go through a box operator. As a consequence, it is not difficult to see but rather cumbersome to prove, that $\Delta(\hat{\alpha})$ is continuous in $\hat{\beta}$. ◀

Finally, a straightforward argument shows that $\varphi \equiv \mathbb{A}_\varphi$. ◻

We have now built an automaton which is equivalent to φ , but we still have to massage it further to get the current one-step language $\text{ML}_1(A, P \cup A, D)$ to the right form, i.e, $\text{ML}_1(A, P, D)$. The crucial difference between these languages, is the unguarded occurrence of states in the former. Automata with unguarded occurrences are called “silent-step” automata in [Ven11], because there can be rounds in the acceptance game where the state in the automaton changes, but the current element in the transition system does not. While there are many applications where silent-step automata are not a problem, it is usually useful to have this coordination in the moves. Moreover, none of the automata in this dissertation are silent. In the following proposition, we show how to convert the silent-step automaton \mathbb{A}_φ , that we just obtained, to an automaton in $\text{Aut}_{wc}(\text{ML}_1(A, P, D))$.

6.3.9. PROPOSITION. *We can effectively transform $\mathbb{A}_\varphi \in \text{Aut}_{wc}(\text{ML}_1(A, P \cup A))$ into an equivalent automaton $\mathbb{A}'_\varphi \in \text{Aut}_{wc}(\text{ML}_1)$.*

Proof. We first define, for every $\widehat{\alpha} \in A$, a new term $\underline{\Delta}(\widehat{\alpha})$. To begin with, we set $\underline{\Delta}(\widehat{\alpha}) := \Delta(\widehat{\alpha})$. Next, we replace every non-guarded occurrence of every $\widehat{\beta} \in A$ in $\underline{\Delta}(\widehat{\alpha})$ by $\Delta(\widehat{\beta})$. We repeat this process until every occurrence of every $\widehat{\beta} \in A$ is of the form $\heartsuit\widehat{\beta}$. It is crucial to observe that this process will eventually converge because φ was originally *guarded*. This means that while running this process, for every branch of $\underline{\Delta}(\widehat{\alpha})$ we will always go through some modality before unfolding some state that was already unfolded.

Observe that $\underline{\Delta}(\widehat{\alpha})$ belongs to $\text{ML}_1(A, P, D)$. Guardedness is clear by construction, and a short argument reveals that given the form of Δ , we do not go beyond modal depth one.

We define the new automaton as $\mathbb{A}'_\varphi := \langle A, \underline{\Delta}, \Omega, \widehat{\varphi} \rangle$. From the following claim it is easy to see that \mathbb{A}'_φ is weak and continuous and therefore $\mathbb{A}'_\varphi \in \text{Aut}_{wc}(\text{ML}_1^b)$.

CLAIM 1. If $a \preceq_{\mathbb{A}'_\varphi} b$ then $a \preceq_{\mathbb{A}_\varphi} b$.

To prove that \mathbb{A}'_φ is equivalent to \mathbb{A}_φ first observe that the acceptance games of these two automata are pretty much the same. The main difference is that \mathbb{A}'_φ could process many rounds (or steps) of the silent-step automata \mathbb{A}_φ at the same time. The only possible issue would be if \mathbb{A}'_φ processes in one round with a parity k what \mathbb{A}_φ processes in many rounds with distinct parities k, k'_1, \dots, k'_n . This is not a problem in our setting of weak automata since, in the relevant cases (i.e. cycles), all those parities would be the same because of the weakness condition. Using these observations, it is not difficult to prove that \mathbb{A}'_φ is equivalent to \mathbb{A}_φ and therefore $\mathbb{A}'_\varphi \equiv \varphi$. \square

6.3.2 From automata to formulas

In this section we show how to convert an automaton $\mathbb{A} \in \text{Aut}_{wc}(\text{ML}_1)$ into a formula $\varphi_{\mathbb{A}} \in \mu_c\text{ML}$. We have already developed the necessary tools to make this conversion quite straightforward. First, we turn the shape of \mathbb{A} into a tree with back-edges. After that, the final formula is obtained, intuitively, by (1) composing the transition maps of the nodes in the tree and (2) adding fixpoint quantifiers at the target of the back-edges.

6.3.10. LEMMA. *For every automaton $\mathbb{A} \in \text{Aut}_{wc}(\text{ML}_1)$ we can effectively construct an equivalent formula $\varphi_{\mathbb{A}} \in \mu_c\text{ML}$.*

Proof. Because of Lemma 4.4.4 we assume that \mathbb{A} can be decomposed as a tree with back edges (A, E, B) . We define auxiliary formulas $\chi_{a \in A}$ by induction on

the tree (A, E) .

$$\chi_a := \begin{cases} \mu a.\Delta(a)[a' \mapsto \chi_{a'} \mid (a, a') \in E] & \text{if } \Omega(a) \text{ is odd} \\ \nu a.\Delta(a)[a' \mapsto \chi_{a'} \mid (a, a') \in E] & \text{if } \Omega(a) \text{ is even.} \end{cases}$$

Observe that if a is a leaf then $\{(a, a') \in E\}$ is empty; therefore the induction is well-defined. Finally, we set $\varphi_{\mathbb{A}} := \chi_{a_I}$. The proof of the equivalence of \mathbb{A} and $\varphi_{\mathbb{A}}$ is given in [Jan06, Lemma 3.2.3.2–3]. However, we still have to prove that $\varphi_{\mathbb{A}}$ lands in the appropriate fragment, i.e., that $\varphi_{\mathbb{A}} \in \mu_c\text{ML}$.

CLAIM 1. $\varphi_{\mathbb{A}} \in \mu_c\text{ML}$.

PROOF OF CLAIM. It is not difficult to show, inductively, that if $a \in A$ belongs to a maximal strongly connected component $C \subseteq A$ of parity 1 (resp. 0) then β_a will be continuous (resp. co-continuous) in $C \subseteq A$: if $\Omega(a) = 1$ and a is a leaf then $\chi_a := \mu a.\Delta(a)$. First observe that $\Delta(a)$ is continuous on C by hypothesis (by the continuity constraint on \mathbb{A}); second, the fixpoint does not bind any variable in this case. Therefore $\chi_a \in \mu\text{MLCON}_C(A)$. For the inductive case the key observation is that if both $\Delta(a)$ and $\chi_{a'}$ belong to $\mu\text{MLCON}_C(A)$ then $\Delta(a)[b \mapsto \chi_{a'}]$ belongs to $\mu\text{MLCON}_C(A)$ for $b \in C$. ◀

This finishes the proof. □

6.3.11. REMARK. The proof of Lemma 6.3.10 adds fixpoint quantifiers, not only at the target of back-edges, but at every node of the tree. From these quantifiers, the only ones that actually bind a variable are those added at the target of back-edges. The other fixpoints are spurious but harmless, and could have been avoided by giving a more complex formulation of χ_a .

6.4 Modal automata versus first-order automata

In this section we briefly discuss why the classes $\text{Aut}(\text{ML}_1)$ and $\text{Aut}(\text{FO}_1)$ are equivalent, and why this relationship also holds for all the subclasses defined in Chapter 4. In other words, for all classes of automata that we have been considering, we can equivalently use the achromatic modal automata model or the chromatic first-order automata model.

6.4.1. PROPOSITION. *Let $\mathcal{C} \in \{\text{Aut}, \text{Aut}_{wc}, \text{Aut}_{wa}, \text{Aut}_{wa}^-\}$ be a class of parity automata, then $\mathcal{C}(\text{ML}_1) \equiv \mathcal{C}(\text{FO}_1)$.*

Proof. We prove this proposition by giving automata translations in both directions. These translations will be completely determined at the one-step level. That is, the automata structure will stay the same, and we will apply translations on the transition map. The main tool for this task was already developed

in Proposition 5.3.3, which gives translations $(-)^t : \text{ML}_1(A, \emptyset, \mathbf{D}) \rightarrow \text{FO}_1(A, \mathbf{D})$ and $(-)_t : \text{FO}_1(A, \mathbf{D}) \rightarrow \text{ML}_1(A, \emptyset, \mathbf{D})$ such that for every transition system \mathbb{S} , valuation $V : A \rightarrow \wp(R[s_I])$ we have:

$$\begin{aligned} \mathbb{S}, V \Vdash \varphi & \text{ iff } (R[s_I], V) \models \varphi^t \\ \mathbb{S}, V \Vdash \psi_t & \text{ iff } (R[s_I], V) \models \psi. \end{aligned}$$

\Rightarrow Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ with $\Delta : A \rightarrow \text{ML}_1(A, \mathbf{P}, \mathbf{D})$ belong to $\mathcal{C}(\text{ML}_1)$, we define $\mathbb{A}' = \langle A, \Delta', \Omega, a_I \rangle$ with $\Delta' : A \times \wp\mathbf{P} \rightarrow \text{FO}_1(A, \mathbf{D})$ as follows: using Theorem 5.3.6 we assume that $\Delta(a)$ is in the normal form $\bigvee_{c \in \wp\mathbf{P}} (\varpi_c \wedge \bigvee_i \varphi_i)$. We then define, for every $c \in \wp\mathbf{P}$ the new transition map as $\Delta'(a, c) := \bigvee \varphi_i^t$. The equivalence $\mathbb{A} \equiv \mathbb{A}'$ is easily proved by a routine argument, using Proposition 5.3.3 and the acceptance games of both automata. Also, it is not difficult to see that the weakness, continuity and additivity conditions are preserved, since there are no structural changes and $(-)^t$ preserves the required fragments.

\Leftarrow Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ with $\Delta : A \times \wp\mathbf{P} \rightarrow \text{FO}_1(A, \mathbf{D})$ belong to $\mathcal{C}(\text{FO}_1)$, we define $\mathbb{A}' = \langle A, \Delta', \Omega, a_I \rangle$ with $\Delta' : A \rightarrow \text{ML}_1(A, \mathbf{P}, \mathbf{D})$. The new transition map is given by $\Delta'(a) := \bigvee_{c \in \wp\mathbf{P}} \varpi_c \wedge \Delta(a, c)_t$. Once more, the equivalence $\mathbb{A} \equiv \mathbb{A}'$ and the preservation of the subclass conditions is easily proved. \square

Given the above proposition, from now on we will use these two kinds of automata interchangeably and without explicit mention.

6.5 Conclusions and open problems

In this chapter we gave automata characterizations for PDL, test-free PDL and $\mu_c\text{ML}$. For the first two, we used an approach involving finite-state automata and solving of equations, while for $\mu_c\text{ML}$ we used silent-step automata and an unraveling technique. The reader may remark that, for (test-free) PDL we could have used the equivalence $\text{PDL} \equiv \mu_a\text{ML}$ of Theorem 3.1.27 and then develop automata for $\mu_a\text{ML}$ just as we did for $\mu_c\text{ML}$. We acknowledge that this is another option, however, we think that treating PDL in its own right, and particularly PDL-programs, gives more insight on their automata-theoretic nature.

Open problems.

1. *Automata for CPDL:* Even though we now know that PDL corresponds to $\text{Aut}_{wa}(\text{ML}_1)$, it is not clear which conditions should be relaxed, in order to get automata for CPDL. It is clear that the additivity condition should be change to something closer to the continuity condition, since the \otimes operator of CPDL translates to a more general conjunction. On the other hand, it is not clear how to mimic the separation of variables condition of $\mu_c\text{ML}^\vee$ (cf. Definition 3.2.26) on the automata side.

2. *Automata for GL*: These automata should be naturally more complex than the ones considered in this dissertation. For example, since GL goes through all the fixpoint alternation hierarchy of μML , the weakness condition should not be imposed. We think that the starting point for these automata should be $\text{Aut}(\text{ML}_1)$, and the key objective would be to understand what the separation of variables means for automata.
3. *More natural automata*: The classes of automata defined in this chapter have at least three nice features (1) they are subclasses of $\text{Aut}(\text{ML}_1)$, and therefore give a clear picture on how the logics that they represent fit inside the μ -calculus; (2) they are *logical* automata (i.e., the transition map contains logical formulas), which will allow us to obtain results focusing on a one-step analysis; (3) they are precise characterizations. However, these automata have a clear downside: they are complex. For example, NFA are a nice representation of regular expressions because they are simple. It would be nice to have automata, for the logics of this chapter, which still satisfy (2) and (3), but with a simpler definition.
4. *Bisimulation quantifiers*: It is known that μML is equivalent to $\text{PDL} + \tilde{\exists}$ where $\tilde{\exists}q.\varphi$ is a bisimulation quantifier [DH00, Fre06]. The meaning of this quantifier is the following

$$\mathbb{S} \Vdash \tilde{\exists}q.\varphi \quad \text{iff} \quad \mathbb{S}' \Vdash \varphi \quad \text{and} \quad \mathbb{S} \xleftrightarrow{q} \mathbb{S}',$$

where $\mathbb{S} \xleftrightarrow{q} \mathbb{S}'$ means that the models are bisimilar, if we disregard the extension of the propositional variable q . Actually, a closer look at the proof of this result in [DH00] reveals that μML is already equivalent to $\text{ML} + \square^* + \tilde{\exists}$ where $\square^* := [(d_1 \oplus \dots \oplus d_n)^*]$.

We conjecture that a similar characterization can be given for PDL. Namely, we think that PDL is equivalent to $\text{ML} + \square^* + \tilde{\exists}_{\text{wc}}$ where the bisimulation quantifier $\tilde{\exists}_{\text{wc}}q.\varphi$ holds when φ is true in a \xleftrightarrow{q} -bisimilar model \mathbb{S}' such that the extension of q in \mathbb{S}' is a *finite chain*. To show this, the difficult step would be to prove the closure of $\text{Aut}_{\text{wa}}(\text{FO}_1)$ under finite chain projection.

This new perspective can help to obtain new results for PDL and, for example, give another way to look at (non-uniform) interpolation for PDL.

Chapter 7

Concrete first-order automata

The main objective of this chapter is to give automata characterizations for WMSO and WCL, on the class of tree models. We do it by showing that these languages precisely correspond to the classes $Aut_{wc}(\text{FOE}_1^\infty)$ and $Aut_{wa}(\text{FOE}_1)$, respectively. These results are obtained via effective transformations from formulas to automata and vice-versa. As a byproduct of these transformations we will also obtain characterizations for the mentioned automata (and second-order logics) as fixpoint logics (like μFOE).

In the first section, as an introduction, we review and discuss the necessary techniques to prove the equivalence $Aut(\text{FOE}_1) \equiv \text{MSO}$ (due to Walukiewicz). In the following sections we prove the results for WMSO and WCL and in the final section we discuss the (open) question of parity automata for $\text{FO}(\text{TC}^1)$.

7.1 Automata for MSO

In this section we review the automata characterization for monadic second-order logic (MSO) given by Walukiewicz [Wal96, Wal02] and also studied by Janin and Walukiewicz in [JW96]. The objective of this section is to introduce the basic techniques to give logical characterizations for parity automata. Most of the results of this section are not original, unless otherwise stated. In general, we follow the main ideas used in the aforementioned papers, but give a presentation which is rephrased in the terms and notation used in this dissertation.

Originally, Walukiewicz introduced parity automata for MSO in the article [Wal96]. These automata are shown to be equivalent to MSO on trees. However, the transition map of these automata is not explicitly based on formulas, but on functions. On the other hand, the functions in consideration are those ‘induced’ by formulas of MSO. The explicit formulation of these automata with a logical transition map is given in [JW96]. In our terminology, automata for MSO are defined as follows.

7.1.1. DEFINITION. A MSO-*automaton* is a parity automaton from $Aut(\text{FOE}_1)$.

In order to show the equivalence between MSO and $Aut(\text{FOE}_1)$, we can divide the process in two obvious parts: (1) from formulas to automata and (2) from automata to formulas.

For the first part, it is customary to give a proof by induction on the complexity of the formula. That is, for every formula of MSO, we construct an equivalent automaton in $Aut(\text{FOE}_1)$. The most interesting cases are the negation and the existential quantifier $\exists p.\varphi$. Negation is paralleled on the automata side by closure under complementation (*cf.* Proposition 2.3.7). In general automata theory this is quite an interesting and difficult topic. In our case, it is fortunately easy to prove the closure under complementation, since we work with *alternating* parity automata whose one-step language is closed under Boolean duals. The main challenge for us will be the existential quantifier. We will discuss how this construction is paralleled by the closure under projection of $Aut(\text{FOE}_1)$ and how to prove it.

In a more general context, in the direction from automata to formulas, there are a handful of techniques which can be used. One possibility is to express, in the target logic, the existence of a winning strategy in the acceptance game for a given automaton (as done, e.g., in [Wal96, Lemma 44]). Another possibility, is to pre-process the original automaton into a tree with back-edges (as done in Section 4.4) and then write an equivalent formula (as done in Section 6.3). When translating parity automata, it is sometimes more straightforward to give a translation when the target logic is a fixpoint logic (as opposed to a second-order logic). For the case of MSO-automata, this means that the translation from $Aut(\text{FOE}_1)$ to MSO can be done in two steps: an automaton from $Aut(\text{FOE}_1)$ is translated to μFOE and then this formula is translated to MSO.

After proving these two directions we obtain that on trees, the formalisms $Aut(\text{FOE}_1)$, μFOE and MSO are effectively equivalent. As a byproduct we get a characterization of MSO and $Aut(\text{FOE}_1)$ as a fixpoint logic.

7.1.1 From MSO to $Aut(\text{FOE}_1)$

In this subsection we discuss the transformation of formulas of MSO into automata. More precisely, we discuss the proof of the following theorem.

7.1.2. THEOREM ([WAL96, WAL02]). *For every formula $\varphi \in \text{MSO}(\mathbf{P})$ with free variables $\mathbf{F} \subseteq \mathbf{P}$ we can effectively construct an automaton $\mathbb{A}_\varphi \in \text{Aut}(\text{FOE}_1, \mathbf{F})$ such that for every \mathbf{F} -tree \mathbb{T} we have $\mathbb{T} \models \varphi$ iff $\mathbb{T} \models \mathbb{A}_\varphi$.*

This proposition is proved inductively, on the complexity of the formula. We start by giving automata for the atomic formulas and after that we discuss the closure under Boolean operations and existential quantification (projection).

Atomic formulas. For the atomic formulas $p \sqsubseteq q$ and $R(p, q)$ we give the following MSO-automata from $Aut(\text{FOE}_1, \{p, q\})$.

$\mathbb{A}_{p \sqsubseteq q} := \langle \{a_0\}, \Delta, \Omega, a_0 \rangle$ where $\Omega(a_0) = 0$ and

$$\Delta(a_0, c) := \begin{cases} \forall x. a_0(x) & \text{if } q \in c \text{ or } p \notin c, \\ \perp & \text{otherwise} \end{cases}$$

$\mathbb{A}_{R(p, q)} := \langle \{a_0, a_1\}, \Delta, \Omega, a_0 \rangle$ where $\Omega(a_0) = \Omega(a_1) = 0$ and

$$\Delta(a_0, c) := \begin{cases} \exists x. a_1(x) \wedge \forall y. a_0(y) & \text{if } p \in c, \\ \forall x. a_0(x) & \text{otherwise.} \end{cases}$$

$$\Delta(a_1, c) := \begin{cases} \top & \text{if } q \in c, \\ \perp & \text{if } q \notin c. \end{cases}$$

7.1.3. REMARK. A nice observation is that, modally, these automata correspond to the formulas $\Box^*(p \rightarrow q)$ and $\Box^*(p \rightarrow \Diamond q)$ respectively. Also, none of the following automata constructions (i.e., Booleans and projection) create cycles on the automata. This shows that all the “iterative power” of these automata boils down to the \Box^* construction.

Boolean operations. To prove the inductive steps of the Boolean operators and the negation it will be enough to prove that the class of automata is closed under complementation and union.

7.1.4. DEFINITION. Given an automaton \mathbb{A} , we define the *tree language recognized by \mathbb{A}* as the class of P-labeled trees $\mathcal{T}(\mathbb{A})$ given by:

$$\mathcal{T}(\mathbb{A}) := \{\mathbb{T} \mid \mathbb{A} \text{ accepts } \mathbb{T}\}.$$

Starting with the closure under union, we just mention the following result, without providing the (completely routine) proof.

7.1.5. PROPOSITION. *Let \mathbb{A} and \mathbb{A}' belong to $Aut(\text{FOE}_1)$. There is an automaton $\mathbb{U} \in Aut(\text{FOE}_1)$ such that $\mathcal{T}(\mathbb{U}) = \mathcal{T}(\mathbb{A}) \cup \mathcal{T}(\mathbb{A}')$.*

Proof. The automaton \mathbb{U} is defined as the disjoint union of \mathbb{A} and \mathbb{A}' plus a new initial state u_I . The transition map of u_I is then given, for every c , as $\Delta_U(u_I, c) := \Delta(a_I, c) \vee \Delta(a'_I, c)$. \square

In order to prove closure under complementation, we crucially use that the one-step language FOE_1 is closed under Boolean duals (cf. Proposition 5.1.57).

7.1.6. PROPOSITION. *If $\mathbb{A} \in Aut(\text{FOE}_1)$ then the automaton \mathbb{A}^δ defined in Definition 2.3.6 recognizes the complement of $\mathcal{T}(\mathbb{A})$.*

Proof. The automaton is well-defined by Proposition 5.1.57, and accepts exactly the transition systems that are rejected by \mathbb{A} , by Proposition 2.3.7. \square

Projection. To prove the inductive step of the existential quantification we want to show that for every automaton $\mathbb{A} \in \text{Aut}(\text{FOE}_1, \mathbf{P}' \uplus \{p\})$ we can give an automaton $\exists p.\mathbb{A} \in \text{Aut}(\text{FOE}_1, \mathbf{P}')$ which works as follows: for every \mathbf{P}' -tree \mathbb{T} we should have that $\mathbb{T} \models \exists p.\mathbb{A}$ iff there is some set of $X_p \subseteq T$ such that $\mathbb{T}[p \mapsto X_p] \models \mathbb{A}$. That is, there is a way to colour the nodes of \mathbb{T} with p to make \mathbb{A} accept this newly colored tree. This property can be rephrased as closure under projection of the class of languages recognized by $\text{Aut}(\text{FOE}_1)$.

7.1.7. DEFINITION. Let $p \notin \mathbf{P}'$ and L be a tree language of $(\mathbf{P}' \uplus \{p\})$ -labeled trees. The *projection* of L over p is the language of \mathbf{P}' -labeled trees defined as

$$\exists p.L := \{\mathbb{T} \mid \mathbb{T}[p \mapsto X_p] \in L \text{ for some } X_p \subseteq T\}.$$

In the following definition we give a concrete definition for the projection of an automaton. Shortly after, we analyze the objective of this construction and why it is still not sufficient.

7.1.8. DEFINITION. Let \mathbb{A} belong to $\text{Aut}(\text{FOE}_1, \mathbf{P}' \uplus \{p\})$. We define the *projection of \mathbb{A} over p* as the automaton $\exists p.\mathbb{A} := \langle A, \Delta^\exists, \Omega, a_I \rangle \in \text{Aut}(\text{FOE}_1, \mathbf{P}')$ given as follows, for every $c \in \wp(\mathbf{P}')$:

$$\Delta^\exists(a, c) := \Delta(a, c) \vee \Delta(a, c \cup \{p\})$$

The intuition behind this construction is the following: suppose that we are playing a fixed match π^\exists of the acceptance game $\mathcal{A}(\exists p.\mathbb{A}, \mathbb{T})$ and we are at a basic position $(a, s) \in A \times T$. At this point, \exists will play a valuation $V_{a,s}$ such that $(R[s], V_{a,s}) \models \Delta^\exists(a, c)$. The key observation is that this play gives us a hint on how to color (or not) the node s , and build up the set X_p . If $(R[s], V_{a,s}) \models \Delta(a, c)$, we can safely decide that $s \notin X_p$. On the other hand, if $(R[s], V_{a,s}) \models \Delta(a, c \cup \{p\})$, we have to set $s \in X_p$.

From the local perspective of the match π^\exists , this procedure gives us a coloring of \mathbb{T} . If we would play a match π of $\mathcal{A}(\mathbb{A}, \mathbb{T})$ going through the same basic positions as π^\exists , it is easy to see that we can give a winning strategy for \exists . However, the problem that we must face, is that the acceptance game $\mathcal{A}(\exists p.\mathbb{A}, \mathbb{T})$ comprises many concurrent matches, depending on the choices of \forall , and each match may suggest a *different* set X_p . More precisely, one of the matches could suggest that $s \in X_p$ and other that $s \notin X_p$ and we could have a situation where both conditions are crucial to win each given match. Luckily, there is a way to pre-process \mathbb{A} to ensure that this situation can be avoided. The key notion is that of *non-determinism*.

Non-determinism and the Simulation Theorem. One of the main technical results for parity automata is the so-called ‘‘Simulation Theorem’’. In a

nutshell, it says that every automaton \mathbb{A} can be converted to an equivalent automaton \mathbb{A}^\forall for which we can always avoid the situation described in the last paragraph. Specifically, what we want is that every winning strategy for \exists in $\mathcal{A}(\mathbb{A}, \mathbb{T})$ can be assumed to be *functional*. A strategy f for \exists is called functional, if whenever \forall can choose to play both (a, s) and (b, s) at a given moment, then $a = b$. That is, \forall 's power boils down to being a *pathfinder* in \mathbb{T} . He chooses the elements of \mathbb{T} whereas the state of \mathbb{A} is ‘fixed’ by the valuation played by \exists , for every given s .

To get a better picture of what a functional strategy means, it is good to do the following: first observe that if we fix a strategy f for \exists for the game $\mathcal{A}(\mathbb{A}, \mathbb{T})$ then the whole game can be represented by a tree, whose nodes are the different admissible moves for \forall . We assume that the automaton is clear from context and denote such a tree by \mathbb{T}_f . Fig. 7.1 shows the move-tree for \forall for some fixed strategy f for \exists . Each branch of \mathbb{T}_f represents a possible f -guided match.

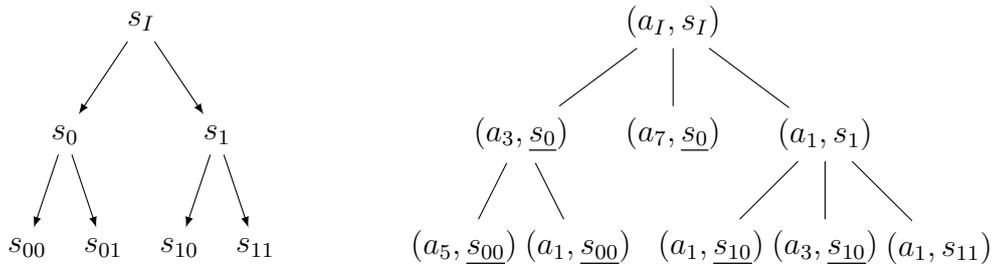


Figure 7.1: A tree \mathbb{T} and \mathbb{T}_f for a fixed strategy for \exists .

Observe that in this figure the chosen strategy for \exists is *not* functional. The admissible moves which violate this condition are underlined. On trees, the notion of functional strategy can be rephrased as “every element $s \in T$ occurs at most once as an admissible move for \forall .” The main property of functional strategies is that for every element $s \in T$ occurring in \mathbb{T}_f we can assign a *unique* state $a_s \in A$ such that $(a_s, s) \in \mathbb{T}_f$.

The automata for which we can assume that every winning strategy for \exists is functional are called *non-deterministic*.

7.1.9. THEOREM ([WAL96, WAL02]). *Every $\mathbb{A} \in \text{Aut}(\text{FOE}_1)$ is effectively equivalent (over all models) to a non-deterministic automaton $\mathbb{A}^\forall \in \text{Aut}(\text{FOE}_1)$.*

7.1.10. REMARK. The terminology “non-deterministic” may seem confusing at first, given that \mathbb{A}^\forall is certainly more “deterministic” than \mathbb{A} (from the point of view of \forall). However, the terminology is sensible when seen from the following perspective: we say that a finite state automaton (on words) is *deterministic* when the next state is uniquely determined by the current state (and the input); on the other hand, they are called *non-deterministic* when \exists can choose between different

transitions, leading to the next state; finally, *alternating* finite state automata are a generalization where the next state is chosen by a complex interaction of \exists and \forall . Going back to parity automata, the above theorem then says that every alternating parity automaton is equivalent to a non-deterministic automaton. In light of our brief discussion, it should be clear that non-deterministic automata are “more deterministic” than alternating automata.

We will not go into the details of the definition of the construction of non-deterministic automata. We only observe that the construction is a variant of the usual powerset construction, and refer the reader to [Zan12, Chapter 2] for a nice exposition of the details.

Now that we have non-deterministic automata at our disposal we can properly define an automata construction which works for the existential quantification. The construction is a straightforward combination of the non-deterministic transformation and the projection.

7.1.11. LEMMA. *For each $\mathbb{A} \in \text{Aut}(\text{FOE}_1, \mathcal{P}'\uplus\{p\})$ we have $\mathcal{T}(\exists p.\mathbb{A}^\forall) = \exists p.\mathcal{T}(\mathbb{A})$.*

Proof. For the difficult direction we have to prove that if $\mathbb{T} \models \exists p.\mathbb{A}^\forall$ then there is some set $X_p \subseteq T$ such that $\mathbb{T}[p \mapsto X_p] \models \mathbb{A}$. We only sketch how to properly define the set X_p . The first observation is the following.

CLAIM 1. $\exists p.\mathbb{A}^\forall$ is non-deterministic.

Suppose that $\mathbb{T} \models \exists p.\mathbb{A}^\forall$. As the automaton is non-deterministic, we can assume that the given winning strategy f_\exists for \exists in $\mathcal{A}(\exists p.\mathbb{A}^\forall, \mathbb{T})@(\mathbf{a}_I, \mathbf{s}_I)$ is functional. We now want to isolate the nodes that f_\exists treats “as if they were labeled with p .” For this purpose, let V_s be the valuation suggested by f_\exists at a position $(a_s, s) \in A \times T$. As f_\exists is winning, the suggested valuation is admissible. In other words, we have $(R[s], V_s) \models \Delta^\exists(a_s, c) = \Delta(a_s, c) \vee \Delta(a_s, c \cup \{p\})$ for $c = \kappa(s)$. We define:

$$X_p := \{s \in T \mid a_s \text{ is defined and } (R[s], V_s) \models \Delta(a_s, c \cup \{p\})\}.$$

The fact that f_\exists is functional guarantees that X_p is well-defined, because for every $s \in T$ there is a unique $a_s \in A$.

We can prove that $\mathbb{T}[p \mapsto X_p] \models \mathbb{A}$ by providing a winning strategy for \exists in the game $\mathcal{A}(\mathbb{A}, \mathbb{T}[p \mapsto X_p])@(\mathbf{a}_I, \mathbf{s}_I)$. This strategy is built based on the (functional) strategy f_\exists that \exists has for $\mathcal{A}(\exists p.\mathbb{A}^\forall, \mathbb{T})@(\mathbf{a}_I, \mathbf{s}_I)$. The reader is referred to [Zan12, Proposition 2.29] for details. \square

7.1.2 From $\text{Aut}(\text{FOE}_1)$ to $\text{FO}(\text{LFP}^1)$

In this section we discuss how to translate automata from $\text{Aut}(\text{FOE}_1)$ to the fixpoint logic μFOE . To translate the automata we will use a technique similar

to that of Section 6.3.2: given an automaton we will first unravel it and obtain a tree with back-edges, then we translate this automaton to a fixpoint formula.

We also want to use this section to discuss the peculiarities of the target language of the translation. We will see that if we allow the target formula to have a free variable x , then we can fall inside a very particular fragment μFOE^\gg of μFOE which we call the *forward-looking fragment of μFOE* . This fragment can be seen as a kind of ‘modal’ fragment of μFOE , whose formulas are invariant under generated submodels, and guarded (in a sense that will be made precise). Our first translation will then transform an automaton $\mathbb{A} \in \text{Aut}(\text{FOE}_1)$ into a formula $\varphi_{\mathbb{A}}^\gg(x) \in \mu\text{FOE}^\gg$ such that

$$\exists \text{ wins } \mathcal{A}(\mathbb{A}, \mathbb{T})@(a_I, s) \quad \text{iff} \quad \mathbb{T} \models \varphi_{\mathbb{A}}^\gg(s),$$

for every tree \mathbb{T} and $s \in T$. Observe that in this equivalence, the interpretation of x should be the element where the automaton starts running. As a corollary of this translation, we can construct a *sentence* $\varphi_{\mathbb{A}} \in \mu\text{FOE}$ such that

$$\mathbb{T} \models \mathbb{A} \quad \text{iff} \quad \mathbb{T} \models \varphi_{\mathbb{A}},$$

by setting $\varphi_{\mathbb{A}} := \exists x.\text{isroot}(x) \wedge \varphi_{\mathbb{A}}^\gg(x)$. While the predicate $\text{isroot}(x)$ can be easily defined in μFOE (on trees) as $\text{isroot}(x) := \forall y.\neg R(y, x)$ it will be worth observing that it cannot be defined in μFOE^\gg , since we anticipated that this logic is invariant under generated submodels.

The technique that we use here is an adaptation of the transformations given in [Jan06] for modal automata. The detailed development of this technique for first-order automata and the focus on the forward-looking fragment are original, as far as we know.

Forward-looking fragment. It is easy to see that parity automata ‘restrict to descendants.’ That is, whenever the game $\mathcal{A}(\mathbb{A}, \mathbb{T})$ is at some basic position (a, s) , the match can only continue to positions of the form (b, t) where $t \in R^*[s]$. Moreover, the game can never go back towards the root of the tree. Therefore, it is to be expected that formulas that correspond to parity automata also ‘restrict to descendants.’ This concept is formalized as follows.

7.1.12. DEFINITION. The *forward-looking fragment* μFOE^\gg of μFOE is defined as the smallest collection of formulas such that:

- It contains the atomic formulas $p(x)$, $x \approx y$ for all $p \in \text{P}$ and $x, y \in \text{iVar}$,
- It is closed under Boolean connectives,
- If $\bar{x} = (x_1, \dots, x_m)$ are individual variables and $\varphi(\bar{x}, y)$ is a μFOE^\gg -formula whose free variables are among $\{\bar{x}, y\}$ then the formulas

$$\exists y.(R(x_j, y) \wedge \varphi(\bar{x}, y)) \quad \text{and} \quad \forall y.(R(x_j, y) \rightarrow \varphi(\bar{x}, y))$$

are in μFOE^\gg for all $1 \leq j \leq m$.

- If $\varphi(q, y)$ is a μFOE^\gg -formula which is positive in q and whose only free individual variable is y then $[\text{LFP}_{q,y}.\varphi(q, y)](x)$ is in μFOE^\gg for all $q \in \mathbf{P}$.

Observe that binary relation R only occurs as guard, and that the fixpoints of μFOE^\gg are parameter-free.

7.1.13. DEFINITION. Let $\varphi \in \mu\text{FOE}^\gg$ be such that $FV(\varphi) \subseteq \{\bar{z}\}$. We say that φ *restricts to descendants* if for every model \mathbb{M} , assignment g and $p \in \mathbf{P}$ the following holds:

$$\mathbb{M}, g \models \varphi \quad \text{iff} \quad \mathbb{M}[p \uparrow R^*[\bar{z}]], g \models \varphi$$

where $R^*[\bar{z}] := \bigcup_i R^*[g(z_i)]$.

7.1.14. REMARK. The reader may have expected an alternative definition which requires that $\mathbb{M}, g \models \varphi$ iff $\mathbb{M}[\mathbf{P} \uparrow R^*[\bar{z}]], g \models \varphi$ or even that $\mathbb{M}, g \models \varphi$ if and only if $\mathbb{M}[\mathbf{P} \uparrow R^*[FV(\varphi)]], g \models \varphi$.¹ All these definitions can be proved to be equivalent, and we keep the above version because it will simplify our inductive proofs.

7.1.15. REMARK. Restriction to descendants is a weak kind of invariance under generated submodels. Suppose that for formulas $\varphi \in \mu\text{FOE}$ whose free variables are among \bar{x} we say that φ is *invariant under generated submodels* if for every model \mathbb{M} and assignment g we have:

$$\mathbb{M}, g \models \varphi \quad \text{iff} \quad \mathbb{M}_{\bar{x}}^\downarrow, g \models \varphi$$

where $\mathbb{M}_{\bar{x}}^\downarrow$ is the submodel of \mathbb{M} generated by $g(x_1), \dots, g(x_m)$. As an example, the formula $\varphi(x) := \exists y. \neg R(x, y)$ is not invariant under generated submodels but, as no p occurs in it, it trivially restricts to descendants of x . The fragment μFOE^\gg can be proved to be invariant under generated submodels, but we don't do it in this dissertation because we will not need it.

In order to prove that every formula of μFOE^\gg restricts to descendants we will first define an analogous notion for maps, and study the fixpoints of such maps. This analysis will be instrumental to prove that if $\psi(q, y)$ restricts to descendants then $[\text{LFP}_{q,y}.\psi(q, y)](z)$ restricts to descendants as well.

7.1.16. DEFINITION. A map $G : \wp(M)^n \rightarrow \wp(M)$ on a model \mathbb{M} is said to *restrict to descendants* if for every $s \in M$ and $\bar{X} \in \wp(M)^n$ we have that

$$s \in G(\bar{X}) \quad \text{iff} \quad s \in G(\bar{X} \cap R^*[s]).$$

7.1.17. THEOREM. *If $G(X, \bar{Y})$ is monotone and restricts to descendants then the map $H(\bar{Y}) := \text{LFP}_X.G(X, \bar{Y})$ also restricts to descendants.*

¹Recall that $\mathbb{M}[\mathbf{P} \uparrow X]$ is defined as $\mathbb{M}[p \mapsto \kappa^{\mathbf{h}}(p) \cap X \mid p \in \mathbf{P}]$.

Proof. Define the abbreviations $F(X) := G(X, \overline{\mathbf{Y}})$ and $F_s(X) := G(X, \overline{\mathbf{Y}} \cap R^*[s]) \cap R^*[s]$. We first prove the following claim linking F and F_s .

CLAIM 1. For every $t \in R^*[s]$ we have that $t \in F(X)$ iff $t \in F_s(X)$.

PROOF OF CLAIM. Direct using restriction to descendants and monotonicity, together with the observation that $R^*[t] \subseteq R^*[s]$. ◀

This connection also lifts to the approximants of the least fixpoints of F and F_s .

CLAIM 2. For every $t \in R^*[s]$ we have that $t \in F^\alpha(\emptyset)$ iff $t \in F_s^\alpha(\emptyset)$.

PROOF OF CLAIM. We prove it by transfinite induction. For the base case it is clear that $F^0(\emptyset) = \emptyset = F_s^0(\emptyset)$. For the inductive case of a successor ordinal $\alpha + 1$ let t belong to $R^*[s]$. we have:

$$\begin{aligned} t \in F^{\alpha+1}(\emptyset) & \text{ iff } t \in F(F^\alpha(\emptyset)) && \text{(by definition)} \\ & \text{ iff } t \in F_s(F^\alpha(\emptyset)) && \text{(by Claim 1)} \\ & \text{ iff } t \in F_s(F_s^\alpha(\emptyset)) && \text{(by IH)} \\ & \text{ iff } t \in F_s^{\alpha+1}(\emptyset). && \text{(by definition)} \end{aligned}$$

The case of limit ordinals is left to the reader. ◀

The following claim is direct by the definition of F_s as $G(X, \overline{\mathbf{Y}} \cap R^*[s]) \cap R^*[s]$.

CLAIM 3. $\text{LFP}_X.F_s(X) \subseteq \text{LFP}_X.G(X, \overline{\mathbf{Y}} \cap R^*[s])$.

Finally, we use the claims and prove that $s \in \text{LFP}_X.G(X, \overline{\mathbf{Y}})$ if and only if $s \in \text{LFP}_X.G(X, \overline{\mathbf{Y}} \cap R^*[s])$ which means that $H(\overline{\mathbf{Y}})$ restricts to descendants.

◀ The key observation for this direction is that $G(X, \overline{\mathbf{Y}} \cap R^*[s]) \subseteq G(X, \overline{\mathbf{Y}})$ by monotonicity of G . Therefore $\text{LFP}_X.G(X, \overline{\mathbf{Y}} \cap R^*[s]) \subseteq \text{LFP}_X.G(X, \overline{\mathbf{Y}})$.

⇒ If $s \in \text{LFP}_X.G(X, \overline{\mathbf{Y}})$ then there is an ordinal β such that $s \in F^\beta(\emptyset)$. By Claim 2, we then have that $s \in F_s^\beta(\emptyset)$ and hence $s \in \text{LFP}_X(F_s(X))$. Using Claim 3 we can conclude that $s \in \text{LFP}_X.G(X, \overline{\mathbf{Y}} \cap R^*[s])$. ◻

The connection between the notions of restriction to descendants for formulas and maps is given in the following proposition.

7.1.18. PROPOSITION. *Let $\varphi \in \mu\text{FOE}$ restrict to descendants and be such that $FV(\varphi) \subseteq \{x\}$. For every model \mathbb{M} , assignment g and predicates $\mathbf{Q} \subseteq \mathbf{P}$ the map $G_x : \wp(M)^n \rightarrow \wp(M)$ given by:*

$$G_x(\overline{\mathbf{Z}}) := \{t \in M \mid \mathbb{M}[\mathbf{Q} \mapsto \overline{\mathbf{Z}}], g[x \mapsto t] \models \varphi\}$$

restricts to descendants.

Proof. An element t belongs to $G_x(\bar{\mathbf{Z}})$ iff $\mathbb{M}[\mathbf{Q} \mapsto \bar{\mathbf{Z}}], g[x \mapsto t] \models \varphi$. As φ restricts to descendants, this occurs iff $\mathbb{M}[\mathbf{Q} \mapsto \bar{\mathbf{Z}} \cap R^*[t]], g[x \mapsto t] \models \varphi$. By definition of G_x , this is equivalent to saying that $t \in G_x(\bar{\mathbf{Z}} \cap R^*[t])$. That is, the map G_x restricts to descendants. \square

We are now ready to prove the main lemma about μFOE^\gg .

7.1.19. LEMMA. *Every $\varphi \in \mu\text{FOE}^\gg$ restricts to descendants.*

Proof. Fix $p \in \mathbf{P}$, we prove the statement (of Definition 7.1.13) by induction.

- If φ does not include p or $\varphi = p(x)$ the statement is clear.
- Let $\varphi(p, \bar{\mathbf{x}}, \bar{\mathbf{y}}) = \psi_1(p, \bar{\mathbf{x}}) \vee \psi_2(p, \bar{\mathbf{y}})$; and consider $\bar{\mathbf{z}}$ such that $\{\bar{\mathbf{x}}, \bar{\mathbf{y}}\} \subseteq \{\bar{\mathbf{z}}\}$.
 - \Rightarrow Without loss of generality suppose $\mathbb{M}, g \models \psi_1$, then by inductive hypothesis we know that $\mathbb{M}[p \upharpoonright R^*[\bar{\mathbf{z}}]], g \models \psi_1$. From this we conclude $\mathbb{M}[p \upharpoonright R^*[\bar{\mathbf{z}}]], g \models \varphi$.
 - \Leftarrow Without loss of generality suppose $\mathbb{M}[p \upharpoonright R^*[\bar{\mathbf{z}}]], g \models \psi_1$. By inductive hypothesis we get $\mathbb{M}, g \models \psi_1$ which clearly implies $\mathbb{M}, g \models \varphi$.
- Negation is handled by the inductive hypothesis.
- Let $\varphi(p, \bar{\mathbf{x}}) = \exists y.(R(x_j, y) \wedge \psi(\bar{\mathbf{x}}, y))$; and consider $\bar{\mathbf{z}}$ such that $\{\bar{\mathbf{x}}\} \subseteq \{\bar{\mathbf{z}}\}$.
 - \Rightarrow Suppose $\mathbb{M}, g \models \varphi$. Then there is $s_y \in R[g(x_j)]$ such that $\mathbb{M}, g[y \mapsto s_y] \models \psi(\bar{\mathbf{x}}, y)$. By inductive hypothesis we get $\mathbb{M}[p \upharpoonright R^*[\bar{\mathbf{z}}, y]], g[y \mapsto s_y] \models \psi(\bar{\mathbf{x}}, y)$ and as $s_y \in R[g(x_j)]$ and $x_j \in \bar{\mathbf{z}}$ we get that $\mathbb{M}[p \upharpoonright R^*[\bar{\mathbf{z}}]], g[y \mapsto s_y] \models \psi(\bar{\mathbf{x}}, y)$. From this, we can conclude that $\mathbb{M}[p \upharpoonright R^*[\bar{\mathbf{z}}]], g \models \exists y.(R(x_j, y) \wedge \psi(\bar{\mathbf{x}}, y))$.
 - \Leftarrow Suppose $\mathbb{M}[p \upharpoonright R^*[\bar{\mathbf{z}}]], g \models \varphi$. Then there exists an element $s_y \in R[g(x_j)]$ such that $\mathbb{M}[p \upharpoonright R^*[\bar{\mathbf{z}}]], g[y \mapsto s_y] \models \psi(\bar{\mathbf{x}}, y)$. As $s_y \in R[g(x_j)]$ and $x_j \in \bar{\mathbf{z}}$ we know that $R^*[\bar{\mathbf{z}}] = R^*[\bar{\mathbf{z}}, y]$. Therefore we also have that $\mathbb{M}[p \upharpoonright R^*[\bar{\mathbf{z}}, y]], g[y \mapsto s_y] \models \psi(\bar{\mathbf{x}}, y)$. By inductive hypothesis we get $\mathbb{M}, g[y \mapsto s_y] \models \psi(\bar{\mathbf{x}}, y)$. From this, we can conclude $\mathbb{M}, g \models \exists y.(R(x_j, y) \wedge \psi(\bar{\mathbf{x}}, y))$.
- Let $\varphi = [\text{LFP}_{q:x}.\psi(q, x)](z)$. Observe that by definition of the fragment, we have $FV(\varphi) = \{z\}$, q is positive in ψ and $FV(\psi) \subseteq \{x\}$. Consider $\bar{\mathbf{z}}$ such that $z \in \{\bar{\mathbf{z}}\}$, we have to prove that

$$\mathbb{M}, g \models \varphi \quad \text{iff} \quad \mathbb{M}[p \upharpoonright R^*[\bar{\mathbf{z}}]], g \models \varphi.$$

By the semantics of the fixpoint operator $\mathbb{M}, g \models \varphi$ iff $g(z) \in \text{LFP}(F_{q:x}^{\mathbb{M}})$ where

$$F_{q:x}^{\mathbb{M}}(Q) := \{t \in M \mid \mathbb{M}[q \mapsto Q], g[x \mapsto t] \models \psi\}.$$

It will be useful to take a slightly more general definition: consider the map

$$G_{q:x}^\psi(Q, P) := \{t \in M \mid \mathbb{M}[q \mapsto Q; p \mapsto P], g[x \mapsto t] \models \psi\}$$

and observe that $F_{q:x}^{\mathbb{M}}(Q) = G_{q:x}^{\psi}(Q, \kappa^{\natural}(p))$ and therefore their least fixpoints will be the same. By inductive hypothesis and Proposition 7.1.18, we know that $G_{q:x}^{\psi}(Q, P)$ restricts to descendants. Using Theorem 7.1.17 we get that $\text{LFP}_Q.G_{q:x}^{\psi}(Q, \kappa^{\natural}(p))$ restricts to descendants as well. That is,

$$g(z) \in \text{LFP}_Q.G_{q:x}^{\psi}(Q, \kappa^{\natural}(p)) \quad \text{iff} \quad g(z) \in \text{LFP}_Q.G_{q:x}^{\psi}(Q, \kappa^{\natural}(p) \cap R^*[g(z)]).$$

Because $z \in \{\bar{z}\}$ and the monotonicity of $G_{q:x}^{\psi}$, we also get that

$$g(z) \in \text{LFP}_Q.G_{q:x}^{\psi}(Q, \kappa^{\natural}(p) \cap R^*[g(z)]) \quad \text{iff} \quad g(z) \in \text{LFP}_Q.G_{q:x}^{\psi}(Q, \kappa^{\natural}(p) \cap R^*[\bar{z}]).$$

Using the definition of $F_{q:x}^{\mathbb{M}}$ and the above equations we can conclude that

$$g(z) \in \text{LFP}(F_{q:x}^{\mathbb{M}}) \quad \text{iff} \quad g(z) \in \text{LFP}(F_{q:x}^{\mathbb{M}}[p \upharpoonright R^*[\bar{z}]]).$$

From this, we finally get $\mathbb{M}, g \models \varphi$ iff $\mathbb{M}[p \upharpoonright R^*[\bar{z}]], g \models \varphi$.

This finishes the proof of the lemma. \square

The translations. First we show that for every $\mathbb{A} \in \text{Aut}(\text{FOE}_1)$ it is possible to give a formula $\varphi_{\mathbb{A}}^{\gg}(x) \in \mu\text{FOE}^{\gg}$ which is equivalent on all transition systems.

7.1.20. PROPOSITION. *For every automaton $\mathbb{A} \in \text{Aut}(\text{FOE}_1, \mathbb{P})$ we can effectively construct a formula $\varphi_{\mathbb{A}}^{\gg}(x) \in \mu\text{FOE}^{\gg}(\mathbb{P})$ with exactly one free variable x , such that for every transition system \mathbb{S} , and $s \in S$*

$$\exists \text{ wins } \mathcal{A}(\mathbb{A}, \mathbb{S}) @ (a_I, s) \quad \text{iff} \quad \mathbb{S} \models \varphi_{\mathbb{A}}(s).$$

Proof. Because of Lemma 4.4.4 we may assume that \mathbb{A} can be decomposed as a tree with back edges (A, E, B) . First we need the following definitions:

$$\varpi_c(x) := \bigwedge_{p \in c} p(x) \wedge \bigwedge_{p \in \mathbb{P} \setminus c} \neg p(x) \quad \beta_a(x) := \bigvee_{c \in C} (\varpi_c(x) \wedge \Delta_{a,c}^g(x))$$

where $\Delta_{a,c}^g$ is a guarded version of $\Delta(a, c)$, defined as

$$\Delta_{a,c}^g(x) := \Delta(a, c)[\exists y. \alpha \mapsto \exists y. (R(x, y) \wedge \alpha); \forall y. \alpha \mapsto \forall y. (R(x, y) \rightarrow \alpha)].$$

Now we define auxiliary formulas $\{\chi_a(x)\}_{a \in A}$ by induction, on the tree (A, E) .

$$\chi_a(x) := \begin{cases} [\text{LFP}_{a:z}. \beta_a(z)[a'(y) \mapsto \chi_{a'}(y) \mid (a, a') \in E]](x) & \text{if } \Omega(a) \text{ is odd} \\ [\text{GFP}_{a:z}. \beta_a(z)[a'(y) \mapsto \chi_{a'}(y) \mid (a, a') \in E]](x) & \text{if } \Omega(a) \text{ is even.} \end{cases}$$

Observe that if a is a leaf then $\{(a, a') \in E\}$ is empty; therefore the induction is well-defined. Finally, we set $\varphi_{\mathbb{A}}^{\gg}(x) := \chi_{a_I}(x)$. For an arbitrary $a \in A$, the formula

$\chi_a(x)$ may have unbound (free) predicates from A . However, in the formula $\varphi_{\mathbb{A}}^{\gg}(x)$ every such predicate is bound by a fixpoint operator.

The equivalence of \mathbb{A} and $\varphi_{\mathbb{A}}^{\gg}(x)$ is a corollary of the following claim. First we introduce the following definition: for every $a \in A$, the automaton $\mathbb{A}_a^{\blacktriangle}$ is obtained by \mathbb{A} by restricting the domain of \mathbb{A} to the states that are below (of equal) to a in the tree (A, E) . That is, the subtree generated by a . This does not include any predecessor of a even if there is a back-edge to it.

CLAIM 1. Fix $a \in A$ and let $V : \text{FV}(\chi_a) \rightarrow \wp(S)$ be a valuation for the free names of A in χ_a . For every transition system \mathbb{S} we have:

$$\mathbb{S}^V \models \chi_a(s) \quad \text{iff} \quad \exists \text{ wins } \mathcal{A}(\mathbb{A}_a^{\blacktriangle}, \mathbb{S}^V) @ (a, s),$$

where $\mathbb{S}^V := \mathbb{S}[b \mapsto V(b) \mid b \in \text{FV}(\chi_a)]$.

PROOF OF CLAIM. This proof is very similar in spirit to what it is done in [Ven11, Theorem 3.14 and 3.27]; we only give a sketch of the case where $\Omega(a)$ is odd.

\Rightarrow Suppose that a is a leaf, then $\chi_a(x) = [\text{LFP}_{a.z} \beta_a(z)](x)$ and $\mathbb{A}_a^{\blacktriangle}$ is an automaton of one state a , possibly with a back edge to itself. Suppose that $\mathbb{S}^V \models \chi_a(s)$ and let $X \subseteq S$ be the least fixpoint of the map $F_a^{\beta_a}$. By definition we have that $s \in X$ and that for all $t \in X$ we have $\mathbb{S}^V \models \beta_a(t)$. Now we give a strategy for \exists in the acceptance game $\mathcal{A}(\mathbb{A}_a^{\blacktriangle}, \mathbb{S}^V) @ (a, s)$: whenever \exists is at a basic position (a, t') she plays the valuation $V' := V[a \mapsto X] \upharpoonright R[t']$. That is, V' is the extension of V with $V(a) := X$ and afterwards restricted to the successors of t' . It is not difficult to see that, as X is a least fixpoint, this strategy is winning for \exists in finitely many steps. The case where a is not a leaf is proved in a similar way, composing the strategies obtained by inductive hypothesis.

\Leftarrow Suppose that a is a leaf. We prove the contrapositive: if $s \notin \text{LFP}(F_a^{\beta_a})$ then \forall wins $\mathcal{A}(\mathbb{A}_a^{\blacktriangle}, \mathbb{S}^V) @ (a, s)$. We give a winning strategy for \forall . Suppose that \exists plays an admissible valuation U ; a brief inspection reveals that $\mathbb{S}^V, U \models \beta_a(s)$ or, in other words, $s \in F_a^{\beta_a}(U(a))$. Now, the key observation is that $U(a) \not\subseteq \text{LFP}(F_a^{\beta_a})$ because otherwise $s \in F_a^{\beta_a}(U(a)) \subseteq F(\text{LFP}(F_a^{\beta_a})) \subseteq \text{LFP}(F_a^{\beta_a})$ which would contradict that $s \notin \text{LFP}(F_a^{\beta_a})$. Therefore, \forall can choose some position $(a, t') \in Z_U$ such that $t' \notin \text{LFP}(F_a^{\beta_a})$. In this position, we are again where we started. Consequently, either \exists loses the match because she gets stuck, or the match continues indefinitely, going through state a infinitely many times. The parity of a is odd and it is the lowest parity in the (sub)automaton, by construction of the unraveling (*cf.* Definition 4.4.2(5)). This means that \exists loses one way or another. The inductive case is left to the reader. \blacktriangleleft

CLAIM 2. $\varphi_{\mathbb{A}}^{\gg}(x) \in \mu\text{FOE}^{\gg}$.

PROOF OF CLAIM. The formula $\varphi_{\mathbb{A}}^{\gg}(x)$ can be seen to belong to μFOE^{\gg} by a simple inspection of the construction: more specifically, the definition of $\Delta_{a,c}^g(x)$ guards every quantifier, and the fixpoint operators introduced in every $\chi_a(x)$ are exactly of the form required by the fragment μFOE^{\gg} . ◀

It is worth observing that, as a consequence of the last claim, the fixpoints operators of $\varphi_{\mathbb{A}}^{\gg}(x)$ do not use parameters. ◻

As a corollary, we get the following translation on trees.

7.1.21. COROLLARY. *For every automaton $\mathbb{A} \in \text{Aut}(\text{FOE}_1, \mathbb{P})$ we can effectively construct a sentence $\varphi_{\mathbb{A}} \in \mu\text{FOE}(\mathbb{P})$ such that for every tree \mathbb{T} ,*

$$\mathbb{T} \models \mathbb{A} \quad \text{iff} \quad \mathbb{T} \models \varphi_{\mathbb{A}}.$$

Proof. Simply set $\varphi_{\mathbb{A}} := \exists x.\text{isroot}(x) \wedge \varphi_{\mathbb{A}}^{\gg}(x)$. ◻

Historical remarks and related results. The fragment μFOE^{\gg} defined here is similar in spirit to the bounded and guarded fragments defined in [ABM99, GW99, ANB98]. The most natural perspective is to see μFOE^{\gg} as an extension of the bounded fragment of first-order logic given in [ABM99] to first-order logic with fixpoints. In [GW99] the authors introduce a guarded fragment of μFOE , however, they aim to make it as big as possible. For example, their formalism can define the mu-calculus with backward-looking modalities, and therefore is not invariant under generated submodels.

7.1.3 From $\text{FO}(\text{LFP}^1)$ to MSO

This section contains a discussion about the translation of formulas of μFOE to MSO. The first observation to be made is that, as μFOE contains individual variables, it will be easier to use the two-sorted 2MSO as a target language.

7.1.22. PROPOSITION. *There exists an effective translation $\text{ST} : \mu\text{FOE}(\mathbb{P}) \rightarrow 2\text{MSO}(\mathbb{P})$ such that for every model \mathbb{M} , assignment g and formula $\varphi \in \mu\text{FOE}(\mathbb{P})$ we have $\mathbb{M}, g \models \varphi$ iff $\mathbb{M}, g \models \text{ST}(\varphi)$.*

Proof. Most of the translation is just given homomorphically as expected:

- $\text{ST}(p(x)) := p(x)$,
- $\text{ST}(R(x, y)) := R(x, y)$
- $\text{ST}(x \approx y) := x \approx y$
- $\text{ST}(\varphi \vee \psi) := \text{ST}(\varphi) \vee \text{ST}(\psi)$,
- $\text{ST}(\neg\varphi) := \neg\text{ST}(\varphi)$,

- $\text{ST}(\exists x.\varphi) := \exists x.\text{ST}(\varphi)$,

and the translation of the fixpoint $[\text{LFP}_{p:y}.\psi(p, y, \bar{z})](x)$ is given by:

$$\forall W.(W \in \text{PRE}(F_{p:y}^\psi) \rightarrow x \in W)$$

where the predicate $W \in \text{PRE}(F_{p:y}^\psi)$ expresses that W is a *prefixpoint* of $F_{p:y}^\psi$. Namely, that $F_{p:y}^\psi(W) \subseteq W$. This predicate can be defined in first-order logic as

$$W \in \text{PRE}(F_{p:y}^\psi) := \forall x.(\psi(p, x, \bar{z}) \rightarrow x \in W).$$

This translation is justified by the following fact about monotone maps:

$$s \in \text{LFP}(F_{p:y}^\psi) \quad \text{iff} \quad s \in \bigcap \{W \subseteq M \mid W \in \text{PRE}(F_{p:y}^\psi)\}.$$

That is, an element belongs to the least fixpoint of a monotone map if and only if it belongs to every prefixpoint of that map. \square

Observe that the crucial element of this proof is that MSO can easily encode the least fixpoint of a map using second-order quantification. We will see in the following sections that this becomes increasingly more difficult when we restrict the quantification to finite sets (i.e., WMSO) and finite chains (i.e., WCL).

7.1.4 Subtleties of the obtained translations

In the previous sections we have given translations between many formalisms. Namely, we can write the translations as the following chain:

$$\text{MSO} \rightarrow \text{Aut}(\text{FOE}_1) \rightarrow \mu\text{FOE}^\gg \leftrightarrow \mu\text{FOE} \rightarrow 2\text{MSO} \rightarrow \text{MSO}$$

As the chain starts and ends with MSO, can we say that all of the above formalisms are *equivalent*? Well, not exactly; it depends on what we mean by ‘equivalent’. This apparent equivalence hides some subtleties which are explicit in the precise statement of each translation.

For example, Proposition 7.1.20 tells us that we can go from $\text{Aut}(\text{FOE}_1)$ to μFOE^\gg but the target formula crucially has a free variable x which must be evaluated at the root of the tree. Corollary 7.1.21 tells us, however, that we can go from $\text{Aut}(\text{FOE}_1)$ to μFOE and obtain an equivalent sentence. It therefore makes more sense to say that $\text{Aut}(\text{FOE}_1) \subseteq \mu\text{FOE}$ instead of the same statement with μFOE^\gg . Moreover, we already observed that in μFOE we can give a formula $\text{isroot}(x)$ expressing that x is the root, while we cannot do that in μFOE^\gg .

Suppose now that we consider μFOE and MSO. Proposition 7.1.22 translates $\mu\text{FOE}(\text{P})$ to $2\text{MSO}(\text{P})$ in a very direct way; if the original formula was a sentence, the target formula will also be a sentence. We would now be tempted to use the translation of Proposition 2.8.3 from 2MSO to MSO to get that $\mu\text{FOE}(\text{P})$ is

at most as expressive as $\text{MSO}(\mathbf{P})$. Here, again, there is a subtle point to be taken into account: the mentioned translation adds extra propositional variables $\mathbf{P}_{\text{iVar}} := \{p_x \mid x \in \text{iVar}\}$ for each individual (first-order) variable. Hence, the relationship we get is then $\mu\text{FOE}(\mathbf{P}) \subseteq \text{MSO}(\mathbf{P} \cup \mathbf{P}_{\text{iVar}})$.

These observations are consistent with the following: suppose that we start with a formula $\varphi(\bar{z}) \in \mu\text{FOE}$ with free variables \bar{z} . If we go through the above diagram to MSO , and then from MSO to μFOE again, we obtain a *sentence* $\varphi' \in \mu\text{FOE}$. It would be strange to have such an equivalence, if we did not change the set of propositional variables.

7.2 Automata for WMSO

In this section we give an automata characterization for WMSO, on trees. Namely,

7.2.1. THEOREM. *WMSO and $\text{Aut}_{wc}(\text{FOE}_1^\infty)$ are effectively equivalent on trees.*

We start by defining what a WMSO-automaton is.

7.2.2. DEFINITION. A WMSO-*automaton* is an automaton from $\text{Aut}_{wc}(\text{FOE}_1^\infty)$.

The main result of this section is that the formalisms $\text{Aut}_{wc}(\text{FOE}_1^\infty)$, $\mu_c\text{FOE}^\infty$ and WMSO are effectively equivalent on trees. Before going into details, it is worth discussing the rationale behind choosing $\text{Aut}_{wc}(\text{FOE}_1^\infty)$ as the automata for WMSO, and why the more standard automata $\text{Aut}_{wc}(\text{FOE}_1)$ based on the one-step language FOE_1 do not work.

Why $\text{Aut}_{wc}(\text{FOE}_1^\infty)$ and not $\text{Aut}_{wc}(\text{FOE}_1)$? In Section 4.2 we discussed extra constraints on parity automata driven by the will to characterize WMSO, and we defined continuous-weak automata for an arbitrary one-step language. However, if we turn to concrete automata (as we do in this chapter) then only adding constraints will not suffice to characterize WMSO, at least not if we start from the class of automata $\text{Aut}(\text{FOE}_1)$.

Since we know that $\text{Aut}(\text{FOE}_1)$ characterizes MSO on trees, then every constraint on $\text{Aut}(\text{FOE}_1)$ will give a logic which is at most as expressive as MSO . However, even on trees of arbitrary branching degree, the logics MSO and WMSO are incomparable. Therefore, we cannot get WMSO from simply constraining $\text{Aut}(\text{FOE}_1)$, even if the continuity condition sounds reasonable.

As a starting motivation towards using FOE_1^∞ as a one-step language, we can consider the contents of Theorem 3.2.28. This theorem states that, on one-step models, WMSO and FOE_1^∞ are equivalent. This is a good sign since, if we want our automata to be equivalent to WMSO on trees, in particular we want the formalisms to be equivalent on one-step models.

To prove the equivalence of $Aut_{wc}(\text{FOE}_1^\infty)$ and WMSO we proceed similarly to the case of MSO (i.e., Section 7.1). We first prove a simulation theorem in full detail, which will shortly after be used to prove closure of $Aut_{wc}(\text{FOE}_1^\infty)$ under projection. However, it is crucial to observe that $Aut_{wc}(\text{FOE}_1^\infty)$ will not be closed under arbitrary set projection (i.e., MSO quantification) but it will be closed under *finite set* projection (i.e., WMSO quantification).

7.2.1 Simulation theorem

In Section 7.1.1 we saw that every MSO-automaton is equivalent to a non-deterministic automaton. Unfortunately, the transformation performed by Theorem 7.1.9 does not preserve the weakness and continuity conditions (see [Zan12, Remark 3.5]), and therefore it does not provide non-deterministic automata for the class $Aut_{wc}(\text{FOE}_1^\infty)$.

Before developing an alternative solution for $Aut_{wc}(\text{FOE}_1^\infty)$, we would like to make a point about the alphabet of the automata in this section. As shown in the case of MSO, the goal of this section is to introduce the necessary tools (in particular a simulation theorem) to obtain a projection lemma (e.g., Lemma 7.1.11 for MSO). Until now we have used a fixed context set of propositions P for our automata. However, when stating and proving projection theorems it is natural to divert from this set of propositions: given an automaton \mathbb{A} based on some set of propositions $P' \uplus \{p\}$, its projection $\exists p.\mathbb{A}$ should be based on the set P' . Therefore, in this section we will generally work with an arbitrary set of propositions $P' \subseteq P$.

In this section we provide, for every automaton $\mathbb{A} \in Aut_{wc}(\text{FOE}_1^\infty, P')$ and $p \in P'$, an automaton $\mathbb{A}_p^\dagger \in Aut_{wc}(\text{FOE}_1^\infty, P')$ which, although not being fully non-deterministic, is specially tailored to prove the closure of $Aut_{wc}(\text{FOE}_1^\infty)$ under finite set projection. The key property of \mathbb{A}_p^\dagger is that it is equivalent to \mathbb{A} for a certain class of trees. Namely, the class of P' -trees \mathbb{T} such that $\kappa^{\natural}(p)$ is *finite*.

$$\text{For all } \mathbb{T} \text{ with } \kappa^{\natural}(p) \text{ finite we have: } \mathbb{T} \models \mathbb{A}_p^\dagger \quad \text{iff} \quad \mathbb{T} \models \mathbb{A}.$$

Recall that $\kappa^{\natural} : P' \rightarrow \wp(T)$ is the valuation associated to \mathbb{T} .

7.2.3. CONVENTION. To reduce the amount of parentheses in the notation, we sometimes use A^\wp to denote the set $\wp(A)$.

The state space of \mathbb{A}_p^\dagger will be the disjoint union of two parts:

- A *non-deterministic* part based on A^\wp ; and
- An *alternating* part based on A .

The non-deterministic part will basically be a powerset construction of \mathbb{A} , and contain the initial state of the automaton. It will have very nice properties enforced by construction:

- It will behave non-deterministically,
- The parity of every state will be 1 (trivially satisfying the weakness condition); and
- The transition map of every state will be continuous in A^\wp .

The alternating part is a copy of \mathbb{A} modified such that it cannot be used to read nodes colored with the propositional variable p . The automaton \mathbb{A}_p^\ddagger is therefore based both on states from A and on “macro-states” from A^\wp . Moreover, the transition map of \mathbb{A}_p^\ddagger is defined such that if a match goes from the non-deterministic part to the alternating part, then it cannot come back. Successful runs of \mathbb{A}_p^\ddagger will have the property of processing only a *finite* amount of the input being in a macro-state and all the rest behaving exactly as \mathbb{A} (but without reading any node colored with p).

Fig. 7.2 shows an schematic view of the two-part construction: on the right side it shows a copy of the original automaton, now as the alternating part of the construction. On the left side it shows the non-deterministic part of the construction, where macro-states consist of many former states of the original automaton. Besides the internal transitions of each part, there are also transitions from the non-deterministic to the alternating part.

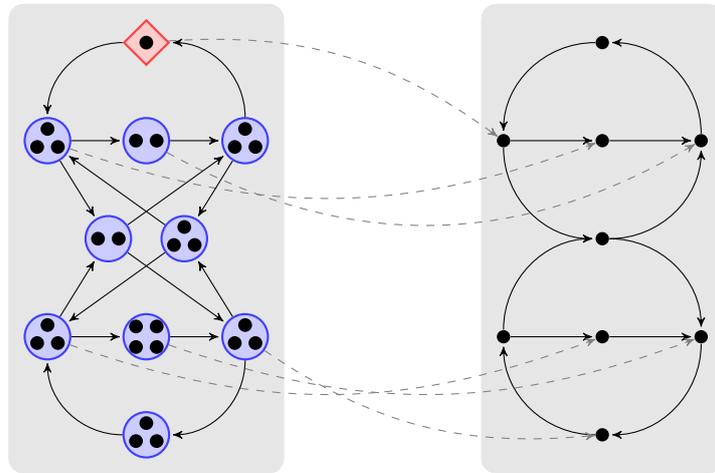


Figure 7.2: Two-part construction illustration, initial state marked as a diamond.

The next definition introduces some notions of strategies that are closely related to our desiderata on the two-part construct.

7.2.4. DEFINITION. Given an automaton $\mathbb{A} \in \text{Aut}(\mathcal{L})$, a subset $B \subseteq A$ of the states of \mathbb{A} , and a tree \mathbb{T} ; a strategy f for \exists in $\mathcal{A}(\mathbb{A}, \mathbb{T})$ is called:

- *Functional in B* if for each node $s \in \mathbb{T}$ there is at most one $b \in B$ such that (b, s) belongs to \mathbb{T}_f .

- *Finitely branching in B* if every node of \mathbb{T}_f with a state from B only has finitely many successors with a state from B .
- *Well-founded in B* if the set of nodes of \mathbb{T}_f with a state from B are all contained in a well-founded subtree of \mathbb{T}_f .

We will construct \mathbb{A}_p^\dagger such that it satisfies the properties of the above definition (for every tree and strategy) for the set of states A^\wp . That is, for the non-deterministic part.

In order to define the non-deterministic part of \mathbb{A}_p^\dagger we need to do two things: (1) first, lift the state space A of \mathbb{A} to A^\wp and adapt the formulas in the transition map Δ of \mathbb{A} accordingly; (2) ensure that the formulas of the non-deterministic part are continuous in A^\wp .

The first step is standard: for $Q \in A^\wp$ and $c \in \wp(\mathbf{P})$ one first considers the formula $\Phi_{Q,c} = \bigwedge_{a \in Q} \Delta(a, c)$. Since the macro-state Q is supposed to encode concurrent matches $\pi_{a \in Q}$ of the acceptance game, it is natural that the transition map of Q should contain the moves for every state in Q . The next step is to lift $\Phi_{Q,c}$ to use the set of names A^\wp instead of A . For this matter, we assume that it is in the normal form $\bigvee \nabla_{\text{FOE}^\infty}^+(\overline{\mathbf{T}}, \Pi, \Sigma)$ and define a lifting for the disjuncts. For example, suppose that $\Phi_{Q,c}$ contains a disjunct $\alpha = \nabla_{\text{FOE}^\infty}^+(\{a, b\} \cdot \{a\}, \{a\}, \{a\})$ belonging to $\text{FOE}_1^{\infty+}(A)$. That is, in this case $\overline{\mathbf{T}}$ contains the two sets $T_1 := \{a, b\}$ and $T_2 := \{a\}$; while Π and Σ only contain the set $\{a\}$. The usual approach is to define the lifted version of α as $\alpha' := \nabla_{\text{FOE}^\infty}^+(\{\{a, b\}\} \cdot \{\{a\}\}, \{\{a\}\}, \{\{a\}\})$ which now belongs to $\text{FOE}_1^{\infty+}(A^\wp)$. The problem with this formula, is that it is not continuous in A^\wp . The reason is that $\{\{a\}\} \subseteq A^\wp$ belongs to the third argument of $\nabla_{\text{FOE}^\infty}^+$, and this is not allowed for formulas continuous in A^\wp –see the normal forms in Corollary 5.1.44.

To overcome this problem, we will only perform a partial lifting on α . That is, we lift α in such a way that we obtain a formula which is continuous in A^\wp . For α , there are eight ways to do this (changes are underlined):

$$\begin{aligned}
\alpha'_1 &:= \nabla_{\text{FOE}^\infty}^+(\{a, b\} \cdot \{a\}, \{a\}, \{a\}) & \alpha'_2 &:= \nabla_{\text{FOE}^\infty}^+(\{a, b\} \cdot \{a\}, \underline{\{\{a\}\}}, \{a\}) \\
\alpha'_3 &:= \nabla_{\text{FOE}^\infty}^+(\underline{\{\{a, b\}\}} \cdot \{a\}, \{a\}, \{a\}) & \alpha'_4 &:= \nabla_{\text{FOE}^\infty}^+(\underline{\{\{a, b\}\}} \cdot \{a\}, \underline{\{\{a\}\}}, \{a\}) \\
\alpha'_5 &:= \nabla_{\text{FOE}^\infty}^+(\{a, b\} \cdot \underline{\{\{a\}\}}, \{a\}, \{a\}) & \alpha'_6 &:= \nabla_{\text{FOE}^\infty}^+(\{a, b\} \cdot \underline{\{\{a\}\}}, \underline{\{\{a\}\}}, \{a\}) \\
\alpha'_7 &:= \nabla_{\text{FOE}^\infty}^+(\underline{\{\{a, b\}\}} \cdot \underline{\{\{a\}\}}, \{a\}, \{a\}) & \alpha'_8 &:= \nabla_{\text{FOE}^\infty}^+(\underline{\{\{a, b\}\}} \cdot \underline{\{\{a\}\}}, \underline{\{\{a\}\}}, \{a\})
\end{aligned}$$

We call these liftings ‘continuous liftings’ of α and then define $\alpha' := \bigvee_i \alpha'_i$.

7.2.5. REMARK. We can define the following order among these liftings: we say that $\alpha'_i \leq_\wp \alpha'_j$ if every element which is lifted in α'_i is also lifted in α'_j . Observe that for our example, α'_1 is a minimal element (and a minimum) of this order and α'_8 is a maximal element (and a maximum). Later, it will become evident in Lemma 7.2.10 that it would have been enough to take $\alpha' := \alpha'_8$ or, more generally, define α' as the disjunction of the maximal elements of \leq_\wp . The maximal elements will play a more important role in the definition of liftings for WCL.

The main intuition behind this definition is the ‘finitely branching’ condition of Definition 7.2.4. Recall that the branching of \mathbb{T}_f is given by the choices of \forall to continue the current match. What we want, is that at any given point of a match of the acceptance game, at most finitely many of the choices of \forall can stay in the non-deterministic part, and the rest of the branching matches should go to the alternating part. That is, \exists should never be required to colour more than finitely many elements with a state of A^\wp . We obtain this through the continuity of α' .

This construction can be done effectively for every α , and by extension for every $\Phi_{Q,c}$ by considering it in normal form and processing each part. We will denote the resulting formula as $\Psi_{Q,c} \in \text{FOE}_1^{\infty+}(A^\wp \cup A)$. This finishes the intuitive explanations and we now turn to the necessary definitions to prove the results.

7.2.6. DEFINITION. Let $\alpha \in \text{FOE}_1^{\infty+}(A)$ be of the shape $\nabla_{\text{FOE}^\infty}^+(\bar{\mathbf{T}}, \Pi, \Sigma)$ for some $\bar{\mathbf{T}} \in \wp(A)^k$ and $\Sigma \subseteq \Pi \subseteq A^\wp$. We say that $\alpha' \in \text{FOE}_1^{\infty+}(A^\wp \cup A)$ is a *continuous lifting* of α if

- (i) $\alpha' = \nabla_{\text{FOE}^\infty}^+(\bar{\mathbf{R}}, \Pi', \Sigma)$ for some $\bar{\mathbf{R}} \in \wp(A^\wp \cup A)^k$ and $\Pi' \in \wp(A^\wp)$,
- (ii) For every i , either: (a) $R_i = T_i$, or (b) $T_i \neq \emptyset$ and $R_i = \{T_i\}$.
- (iii) For every $S \in \Pi$, either: (a) $S \in \Pi'$, or (b) $S \neq \emptyset$ and $\{S\} \in \Pi'$.
- (iv) For every $S' \in \Pi'$, either: (a) $S' \in \Pi$, or (b) $S' = \{S\}$ for some $\emptyset \neq S \in \Pi$.

Observe that every such α' is continuous in A^\wp since $A^\wp \cap \bigcup \Sigma = \emptyset$, as required by Corollary 5.1.44(ii).

7.2.7. DEFINITION. Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle \in \text{Aut}_{wc}(\text{FOE}_1^\infty)$. Let $c \in \wp(\mathbf{P}')$ be a color and $Q \in A^\wp$ be a macro-state. First consider the formula $\bigwedge_{a \in Q} \Delta(a, c)$. By Corollary 5.1.35 there is $\Phi_{Q,c} \in \text{FOE}_1^{\infty+}(A)$ such that $\Phi_{Q,c} \equiv \bigwedge_{a \in Q} \Delta(a, c)$ and $\Phi_{Q,c}$ is in the basic form $\bigvee_j \varphi_j$ where each $\varphi_j = \nabla_{\text{FOE}^\infty}^+(\bar{\mathbf{T}}, \Pi, \Sigma)$ for some $\bar{\mathbf{T}}$ and $\Sigma \subseteq \Pi$. We define:

$$\Psi_{Q,c} := \bigvee_j \bigvee \{\psi \mid \psi \text{ is a continuous lifting of } \varphi_j\}.$$

Observe that $\Psi_{Q,c}$ belongs to $\text{FOE}_1^{\infty+}(A^\wp \cup A)$ and is continuous in A^\wp . The latter is because the continuous liftings have this property, which is preserved by disjunction.

We are finally ready to define the two-part construct.

7.2.8. DEFINITION. Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ belong to $\text{Aut}_{wc}(\text{FOE}_1^\infty, \mathbf{P}')$ and let p be a propositional variable. We define the *two-part construct* of \mathbb{A} with respect to

p as the automaton $\mathbb{A}_p^\dagger = \langle A^F, \Delta^F, \Omega^F, a_I^F \rangle \in \text{Aut}_{wc}(\text{FOE}_1^\infty, \mathbf{P}')$ given by:

$$\begin{aligned} A^F &:= A \cup A^\wp & \Delta^F(Q, c) &:= \Psi_{Q,c} \\ a_I^F &:= \{a_I\} \\ \Omega^F(Q) &:= 1 & \Delta^F(a, c) &:= \begin{cases} \perp & \text{if } p \in c, \\ \Delta(a, c) & \text{otherwise.} \end{cases} \\ \Omega^F(a) &:= \Omega(a) \end{aligned}$$

Before proving that the two-part construct satisfies nice properties we need a few propositions. The following two lemmas show how to go from admissible moves in the two-part construct to the original automaton and vice-versa.

7.2.9. LEMMA (ND TO ALT). *Given an automaton \mathbb{A} from $\text{Aut}_{wc}(\text{FOE}_1^\infty, \mathbf{P}')$, a macro-state $Q \in A^\wp$ and a color $c \in \wp(\mathbf{P}')$ such that $(D, V_{Q,c}) \models \Psi_{Q,c}$ for some valuation $V_{Q,c} : A^\wp \cup A \rightarrow \wp(D)$; there is a valuation $U : A \rightarrow \wp(D)$ such that*

- (i) $(D, U) \models \Delta(a, c)$ for all $a \in Q$,
- (ii) If $d \in U(b)$ then either

- (a) $d \in V_{Q,c}(b)$, or
- (b) $d \in V_{Q,c}(Q')$ for some $Q' \in A^\wp$ such that $b \in Q'$.

Proof. Define $U : A \rightarrow \wp(D)$ as

$$U(b) := V_{Q,c}(b) \cup \bigcup_{b \in Q'} V_{Q,c}(Q').$$

Recall that $\Psi_{Q,c} := \bigvee_j \bigvee \{\psi \mid \psi \text{ is a continuous lifting of } \varphi_j\}$. As a first step, let $(D, V_{Q,c}) \models \psi$ where ψ is a continuous lifting of some φ_j . Recall that by definition of continuous lifting (cf. Definition 7.2.6) the difference between ψ and φ_j is that for every $Q'(x)$ in ψ there is $\bigwedge \{b(x) \mid b \in Q'\}$ in φ_j at the same position. Given a subformula β of ψ , we use $\beta^\Downarrow := \beta[Q'(x) \mapsto \bigwedge \{b(x) \mid b \in Q'\} \mid Q' \in A^\wp]$ to denote the *lowering* of β .

CLAIM 1. For every $\beta \geq \psi$ we have $(D, V_{Q,c}), g \models \beta$ iff $(D, U), g \models \beta^\Downarrow$.

PROOF OF CLAIM. This claim is proved by induction on β . If $\beta = a(x)$ for some $a \in A$ then the lowering does not change β , and the equivalence is clear because $d \in U(a)$ iff $d \in V_{Q,c}(a)$ by definition of U .

If $\beta = Q'(x)$ then $\beta^\Downarrow = \bigwedge \{b(x) \mid b \in Q'\}$. From left to right suppose that $g(x) \in V_{Q,c}(Q')$. By definition of U we have $g(x) \in U(b)$ for all $b \in Q'$. Therefore we conclude that $(D, U), g \models \beta^\Downarrow$.

All the other cases are direct by induction hypothesis, since the lowering operation works homomorphically on them; for example $(\gamma_1 \vee \gamma_2)^\Downarrow = \gamma_1^\Downarrow \vee \gamma_2^\Downarrow$. ◀

As $(D, V_{Q,c}) \models \psi$, it is direct from this claim that $(D, U) \models \psi^\downarrow = \varphi_j$. Hence, as $\bigvee_j \varphi_j \equiv \bigwedge_{a \in Q} \Delta(a, c)$, we get that $(D, U) \models \Delta(a, c)$ for all $a \in Q$. \square

7.2.10. LEMMA (ALT TO ND). *Let \mathbb{A} belong to $\text{Aut}_{wc}(\text{FOE}_1^\infty, \mathbf{P}')$, $Q \in A^\wp$ be a macro-state, and $c \in \wp(\mathbf{P}')$ be a color. Let $\{V_{a,c} : A \rightarrow \wp(D) \mid a \in Q\}$ be a family of valuations such that $(D, V_{a,c}) \models \Delta(a, c)$ for each $a \in Q$. Then, for every finite set $P \subseteq_\omega D$ there is a valuation $V_{Q,c} : A \cup A^\wp \rightarrow \wp(D)$ such that*

- (i) $(D, V_{Q,c}) \models \Psi_{Q,c}$,
- (ii) If $d \in V_{Q,c}(b)$ for $b \in A$, then $d \in V_{a,c}(b)$ for some $a \in Q$.
- (iii) If $d \in V_{Q,c}(Q')$ for $Q' \in A^\wp$, then $d \in V_{a,c}(b)$ for some $a \in Q$, $b \in Q'$.
- (iv) If $d \in P \cap V_{Q,c}(q)$ then $q \in A^\wp$.

Proof. We first define an auxiliary valuation $V_t : A \rightarrow \wp(D)$ which gathers all the valuations from the hypothesis, that is $V_t(b) := \bigcup_{a \in Q} V_{a,c}(b)$.

CLAIM 1. $(D, V_t) \models \bigwedge_{a \in Q} \Delta(a, c)$.

PROOF OF CLAIM. Observe that for every $a \in Q$, $b \in A$ we have $V_{a,c}(b) \subseteq V_t(b)$; then by monotonicity we get that $(D, V_t) \models \Delta(a, c)$ for every $a \in Q$. \blacktriangleleft

Recall that $\bigwedge_{a \in Q} \Delta(a, c) \equiv \bigvee_j \nabla_{\text{FOE}^\infty}^+(\overline{\mathbf{T}}, \Pi, \Sigma)_j$ and

$$\Psi_{Q,c} := \bigvee_j \bigvee \{ \psi \mid \psi \text{ is a continuous lifting of } \nabla_{\text{FOE}^\infty}^+(\overline{\mathbf{T}}, \Pi, \Sigma)_j \}.$$

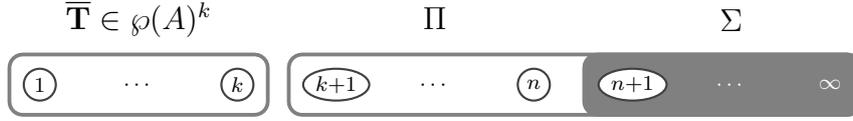
Assume that $(D, V_t) \models \nabla_{\text{FOE}^\infty}^+(\overline{\mathbf{T}}, \Pi, \Sigma)_j$. Observe that this formula gives a full description of the elements of D (see Fig. 7.3). Namely,

1. There are distinct $d_1, \dots, d_k \in D$ such that d_i has type T_i ,
2. Every $d' \in D$ which is not among d_1, \dots, d_k satisfies some type in $\Pi \cup \Sigma$,
3. There are infinitely many elements of each type $S \in \Sigma$,
4. Only finitely many elements do not witness a type in Σ , therefore, there are finitely many elements witnessing $\Pi \setminus \Sigma$.

Let $W := \{d_1, \dots, d_k\} \cup \{d \in D \mid d \text{ witnesses some } S \in \Pi \setminus \Sigma\}$ be the elements in the white part of Fig. 7.3. It is important to observe that W will always be *finite* (see point (4) above). Also note that the set P need not be included in W , since there could be some element $d' \in P$ which realizes a type from Σ .

We will now define a valuation $V_{Q,c} : A \cup A^\wp \rightarrow \wp(D)$ which lifts the types of the elements of W and P . We define it using the alternative marking representation $V_{Q,c}^\natural : D \rightarrow \wp(A \cup A^\wp)$, as follows:

$$V_{Q,c}^\natural(d) := \begin{cases} \{V_t^\natural(d)\} & \text{if } d \in W \cup P \\ V_t^\natural(d) & \text{otherwise.} \end{cases}$$

Figure 7.3: Partitioning of D .

We still need to show that $(D, V_{Q,c}) \models \psi$ where ψ is some continuous lifting of $\nabla_{\text{FOE}\infty}^+(\bar{\mathbf{T}}, \Pi, \Sigma)_j$. We will prove that $(D, V_{Q,c}) \models \nabla_{\text{FOE}\infty}^+(\bar{\mathbf{T}}^\uparrow, \Pi^\uparrow, \Sigma)$ where $T_i^\uparrow := \{T_i\}$ and $\Pi^\uparrow := \{\{S\} \mid S \in \Pi, S \neq \emptyset\} \cup \{\emptyset \mid \emptyset \in \Pi\}$.

- It is easy to see that every d_i realizes the type T_i^\uparrow by definition of $V_{Q,c}$.
- The difficult part is to prove that every $d \in D \setminus \{d_1, \dots, d_k\}$ realizes some type in $\Pi^\uparrow \cup \Sigma$. We divide in cases again.
 - (i) Suppose that d was originally a witness for some $S \in \Pi \setminus \Sigma$ in (D, V_t) . In this case, it is clear that d will realize $\{S\} \in \Pi^\uparrow$ in $(D, V_{Q,c})$.
 - (ii) Other possible case is that d was originally a witness for some $S \in \Sigma$ (therefore standing in the gray area of Fig. 7.3). We have two more subcases: if $d \in P$ we crucially use that $\Sigma \subseteq \Pi$ and therefore $S \in \Pi$. From this we can conclude that d now realizes $\{S\} \in \Pi^\uparrow$. If on the other hand $d \notin P$ then $V_t^\natural(d) = V_{Q,c}^\natural(d)$. It is then clear that d realizes in $(D, V_{Q,c})$ the same type S which it realized in (D, V_t) . By our assumption, this type belongs to Σ (and hence to $\Pi^\uparrow \cup \Sigma$).
- Finally, we have to prove that (i) there are infinitely many elements satisfying each $S \in \Sigma$, and (ii) only finitely many elements do not satisfy a type from Σ . This is easy to prove recalling that $W \cup P$ is finite.

We finish the proof by observing that $\nabla_{\text{FOE}\infty}^+(\bar{\mathbf{T}}^\uparrow, \Pi^\uparrow, \Sigma)$ is indeed a continuous lifting of $\nabla_{\text{FOE}\infty}^+(\bar{\mathbf{T}}, \Pi, \Sigma)_j$. \square

7.2.11. REMARK. Observe that, without loss of generality, \exists can always choose to play *minimal* valuations. That is, in a basic position (a, s) she plays a valuation $V : A \rightarrow \wp(R[s])$ such that for every $t \in R[s]$ and $a \in A$, the element t belongs to $V(b)$ only if it is strictly needed to make $\Delta(a, \kappa(s))$ true. That is, she plays valuations V such that

$$\text{If } (D, V) \models \Delta(a, \kappa(s)) \text{ then } (D, V[b \mapsto V(b) \setminus \{t\}]) \not\models \Delta(a, \kappa(s))$$

for all $a, b \in A$ and $t \in V(b)$. In what follows we assume that \exists plays minimal valuations and we call such strategies minimal. For more detail we refer the reader to [Zan12, Proposition 2.13].

Finally we can state and prove the properties of the two-part construct.

7.2.12. THEOREM. *Let $\mathbb{A} \in \text{Aut}_{wc}(\text{FOE}_1^\infty, \mathbf{P}')$, and let $p \in \mathbf{P}'$ be a propositional variable, and \mathbb{T} be a \mathbf{P}' -tree. The following holds:*

1. $\mathbb{A}_p^\dagger \in \text{Aut}_{wc}(\text{FOE}_1^\infty, \mathbf{P}')$.
2. *If \exists wins $\mathcal{A}(\mathbb{A}_p^\dagger, \mathbb{T}) @ (a_I^F, s_I)$ then she has a winning strategy which is functional, continuous and well-founded in A^\wp .*
3. *If $\kappa^{\mathbb{h}}(p)$ is finite then: \mathbb{A}_p^\dagger accepts \mathbb{T} iff \mathbb{A} accepts \mathbb{T} .*

Proof. (1) The key observation is that $\Psi_{Q,c}$ is continuous in A^\wp .

(2) We treat the properties separately:

- *Functional in A^\wp :* Suppose that (a, s) is a position of an f -guided match where the proposed valuation $V : A \rightarrow \wp(R[s])$ is such that $t \in V(Q)$ and $t \in V(Q')$ for distinct $Q, Q' \in A^\wp$ and some $t \in R[s]$. Let ψ be a disjunct of $\Psi_{Q,c}$ witnessing $(R[s], V) \models \Psi_{Q,c}$. As ψ is a continuous lifting, the element t has to be witness for exactly one type $T_i = \{Q''\}$ with $Q'' \in A^\wp$. As we assume that \exists plays minimal strategies then we can assume that $t \in V(Q'')$ only, among A^\wp . Therefore, t cannot be required to be a witness for both Q and Q' at the same time.
- *Finitely branching in A^\wp :* This is direct from the syntactical form of $\Psi_{Q,c}$; that is, $\Psi_{Q,c}$ is continuous in A^\wp . Assuming that \exists plays minimal strategies then she always proposes a valuation V where $V(A^\wp)$ is finite.
- *Well-founded in A^\wp :* The game starts in A^\wp and, as the parity of A^\wp is 1, it can only stay there for finitely many rounds. This means that, as f is winning, every branch of \mathbb{T}_f has to leave A^\wp at some finite stage.

(3) $\boxed{\Leftarrow}$ Given a winning strategy f for \exists in $\mathcal{G} = \mathcal{A}(\mathbb{A}, \mathbb{T}) @ (a_I, s_I)$ we construct a winning strategy f^F for \exists in $\mathcal{G}^F = \mathcal{A}(\mathbb{A}_p^\dagger, \mathbb{T}) @ (a_I^F, s_I)$. We define it inductively for a match π^F of \mathcal{G}^F . While playing π^F we maintain a bundle (set) \mathcal{B} of f -guided shadow matches. We use \mathcal{B}_i to denote the bundle at round i . We maintain the following condition (\dagger) for every round along the play:

- \dagger 1. If the current basic position in π^F is of the form $(Q, s) \in A^\wp \times T$, then for every $a \in Q$ there is an f -guided shadow match $\pi_a \in \mathcal{B}$ such that the current basic position is $(a, s) \in A \times T$;
- \dagger 2. Otherwise, (a) $\mathcal{B} = \{\pi\}$ for some π , and the position in both π^F and π is of the form $(a, s) \in A \times T$; and moreover, (b) $\mathbb{T}.s$ is p -free.

Intuitively, in order to simulate \mathbb{A} with \mathbb{A}_p^\dagger we have to keep two things in mind: (1) \mathbb{A}_p^\dagger can only read p while it is in the non-deterministic part; and (2) every choice of \forall in \mathcal{G} corresponds to a match that has to be won by \exists ; these parallel matches are kept track of in \mathcal{B} and represented as a macro state in \mathbb{A}_p^\dagger . Therefore, we want the simulation to stay in the non-deterministic part as long as we could

potentially read some p (condition †1). Whenever there are no more p 's to be read, we can relax and behave exactly as \mathbb{A} (condition †2).

At round 0 we initialize the bundle $\mathcal{B} = \{\pi_{a_I}\}$ with the f -guided match π_{a_I} at basic position (a_I, s_I) . It is clear that (†1) holds. For the inductive step we divide in cases:

- If (†2) holds we are given a bundle $\mathcal{B}_i = \{\pi\}$ such that both π^F and π are in position $(a, s) \in A \times T$. We define f^F as f for this position. To see that this gives an admissible move in π^F observe that as $\mathbb{T}.s$ is p -free then $p \notin \kappa(s)$ and hence $\Delta^F(a, \kappa(s)) = \Delta(a, \kappa(s))$. Now it is \forall 's turn to make a move in π^F . By definition of Δ^F , the formula $\Delta^F(a, \kappa(s))$ belongs to $\text{FOE}_1^{\infty+}(A)$ and hence the next position in π^F will be of the form $(a', s') \in A \times T$ with $\mathbb{T}.s$ p -free. We replicate the move in the shadow match π and hence (†2) is preserved.
- If (†1) holds we are given f -guided matches $\mathcal{B}_i = \{\pi_{a_1}, \dots, \pi_{a_k}\}$ such that for the current position $(Q, s) \in A^\wp \times T$ and for each $a \in Q$ we have $\pi_a \in \mathcal{B}_i$. For every match π_a , the strategy f provides a valuation V_a which is an admissible move in this match. Define $P := \{t \in R[s] \mid \mathbb{T}.t \text{ is not } p\text{-free}\}$ and observe that, as $\kappa^\sharp(p)$ is finite, P will be finite as well. Using Lemma 7.2.10 with P and $\{V_a\}_{a \in Q}$ we can combine these valuations into an admissible move V^F in π^F .

To prove that (†) is preserved we distinguish cases as to \forall 's move: first suppose that \forall chooses a position of the form $(b, t) \in A \times T$. Because of Lemma 7.2.10(ii) we know that $t \in V_{a_j}(b)$ for some $a_j \in Q$. That is, we can replicate this move in one of the shadow matches π_{a_j} . We do that and set $\mathcal{B}_{i+1} := \{\pi_{a_j}\}$ hence validating (†2a). To see that (†2b) is also satisfied observe that Lemma 7.2.10(iv) ensures that $\mathbb{T}.t$ is p -free.

For the other case, suppose that \forall chooses a position $(Q', t) \in A^\wp \times T$. Similar to the previous case, this time using Lemma 7.2.10(iii), we can trace every $b \in Q'$ back to some match $\pi_b \in \mathcal{B}_i$. We define $\pi_b \cdot (b, t)$ as the match π_b extended with \forall 's move (b, t) . Finally we let $\mathcal{B}_{i+1} := \{\pi_b \cdot (b, t) \mid b \in Q'\}$, which validates (†1).

Now we prove that f^F is actually winning. It is clear that \exists wins every finite full f^F -guided match (because the moves are admissible). Now suppose that an f^F -guided match is infinite. By hypothesis $\kappa^\sharp(p)$ is a finite set, so after a finite amount of rounds we arrive to an element s such that $\mathbb{T}.s$ is p -free. This means –because of (†)– that the automaton stays in A^\wp only for a finite amount of steps and then moves to A , at a position (a, s) which is *winning* for \exists . From there on the match π^F and π are exactly the same and, as \exists wins π (which is f -guided for a winning strategy f), she also wins π^F .

$\boxed{\Rightarrow}$ Given a winning strategy f^F for \exists in $\mathcal{G}^F = \mathcal{A}(\mathbb{A}_p^\sharp, \mathbb{T}) @ (a_I^F, s_I)$ we construct a winning strategy f for \exists in $\mathcal{G} = \mathcal{A}(\mathbb{A}, \mathbb{T}) @ (a_I, s_I)$. We define the strategy inductively for a match π of \mathcal{G} . While playing π we maintain an f^F -guided

shadow match π^F . We maintain the following condition (\ddagger) for every round along the play: let $(a, s) \in A \times T$ be the current position in π , then one of the following conditions holds:

- \ddagger 1. The current basic position in π^F is of the form $(Q, s) \in A^\wp \times T$ with $a \in Q$,
- \ddagger 2. The current basic position in π^F is also $(a, s) \in A \times T$.

At round 0 the matches π and π^F are in position (a_I, s_I) and $(\{a_I\}, s_I)$ respectively, therefore (\ddagger 1) holds. For the inductive step we divide in cases:

- If (\ddagger 2) holds, the match π is in position (a, s) . For this position, we let f be defined as f^F . Observe that it must be the case that $p \notin \kappa(s)$, otherwise \exists wouldn't have an admissible move V^F in π^F . Given this, and assuming that \exists plays minimal strategies, \exists can use the same V^F in π . It is easy to see that we can replicate \forall 's next move in the shadow match.
- If (\ddagger 1) holds, the matches π and π^F are respectively in position (a, s) and (Q, s) with $a \in Q$. The strategy f^F provides a valuation V^F which is admissible in π^F ; that is $(R[s], V_s) \models \Delta^F(Q, \kappa(s))$. Using Lemma 7.2.9 we can get a valuation U which is admissible in π —see item (i); that is, $(R[s], U) \models \Delta(a, \kappa(s))$. Suppose now that \forall chooses (b, t) as a next position in π . Using Lemma 7.2.9(ii) we know that either (a) $t \in V^F(b)$ or, (b) there is some $Q' \in A^\wp$ with $b \in Q'$ and $t \in V^F(Q')$. In both cases we have a way to replicate \forall 's move in π^F and preserve (\ddagger).

To see that f is winning we proceed similar to the other direction. □

Historical remarks and related results. The idea of a Simulation Theorem goes back to (at least) Safra [Saf88] and Muller and Schupp [MS95]. In the first case, Safra used an augmented state space to convert non-deterministic Büchi automata into deterministic automata. In the latter, Muller and Schupp also use an augmented state space to convert alternating tree automata to non-deterministic tree automata.

The idea to use a two-part automata to preserve the weakness condition was introduced in [Zan12, FVZ13], although the authors claim that some concepts were already present in [MSS92]. In [Zan12, FVZ13] the authors use an automaton based on $\wp(A \times A)$ and A with a non-parity acceptance condition. This automaton is then converted to a parity automaton with a standard trick. Using $\wp(A \times A)$ as the state space instead of A^\wp is necessary to correctly keep track of *infinite* runs of the automata. The first explicit use of $\wp(A \times A)$ as the state space of such automata seems to be in [AN01, Section 9.6.2].

Observe, however, that in the non-deterministic part of our constructions the parity is uniformly 1 and therefore any infinite run which stays in that part will be a rejecting run. Using this observation, we give a slightly simpler

construction based on A^\wp and A . In this respect, the proofs are cleaner and we avoid a non-parity acceptance condition.

7.2.2 From WMSO to $Aut_{wc}(\text{FOE}_1^\infty)$

In this section we give an effective transformation from formulas of WMSO to automata of $Aut_{wc}(\text{FOE}_1^\infty)$. As we observed, this class is not closed under the existential quantification of MSO (that is, under ‘projection.’) We start by proving that this class is closed under a ‘finite set projection’.

Finite set projection.

7.2.13. DEFINITION. Let $p \notin P'$ and L be a tree language of $(P' \uplus \{p\})$ -labeled trees. The *finite set projection* of L over p is the language $\exists_{\text{fin}} p.L$ of P' -labeled trees defined as

$$\exists_{\text{fin}} p.L := \{\mathbb{T} \mid \mathbb{T}[p \mapsto X_p] \in L \text{ for some finite set } X_p \subseteq T\}.$$

In the following definition we give, for every $\mathbb{A} \in Aut_{wc}(\text{FOE}_1^\infty, P' \uplus \{p\})$, the finite set projection over p , denoted $\exists_{\text{fin}} p.\mathbb{A} \in Aut_{wc}(\text{FOE}_1^\infty, P')$. The domain and transition function of the automaton $\exists_{\text{fin}} p.\mathbb{A}$ will be based on \mathbb{A}_p^\ddagger .

7.2.14. DEFINITION. Let \mathbb{A} belong to $Aut_{wc}(\text{FOE}_1^\infty, P' \uplus \{p\})$. We define the *finite set projection of \mathbb{A} over p* as the automaton $\exists_{\text{fin}} p.\mathbb{A} := \langle A \cup A^\wp, \Delta^\exists, \Omega^\exists, \{a_I\} \rangle$ from $Aut_{wc}(\text{FOE}_1^\infty, P')$ given as follows, for every $c \in \wp(P')$:

$$\begin{aligned} \Omega^\exists(a) &:= \Omega(a) & \Delta^\exists(a, c) &:= \Delta(a, c) \\ \Omega^\exists(Q) &:= 1 & \Delta^\exists(Q, c) &:= \Psi_{Q,c} \vee \Psi_{Q,c \cup \{p\}} \end{aligned}$$

where $\Psi_{Q,c}$ is as in Definition 7.2.7.

The key observation to be made about the above definition is that $\exists_{\text{fin}} p.\mathbb{A}$ is actually defined based on the two-part construction \mathbb{A}_p^\ddagger (see Definition 7.2.8). The main change is that the non-deterministic part (A^\wp) has been projected with respect to p . This can be observed in the definition of $\Delta^\exists(Q, c)$.

7.2.15. LEMMA. For each automaton $\mathbb{A} \in Aut_{wc}(\text{FOE}_1^\infty, P' \uplus \{p\})$ we have that $\mathcal{T}(\exists_{\text{fin}} p.\mathbb{A}) = \exists_{\text{fin}} p.\mathcal{T}(\mathbb{A})$.

Proof. What we need to show is that for any P' -tree \mathbb{T} :

$$\begin{aligned} \exists_{\text{fin}} p.\mathbb{A} \text{ accepts } \mathbb{T} &\text{ iff } && \text{there is a finite set } X_p \subseteq T \\ &&& \text{such that } \mathbb{A} \text{ accepts } \mathbb{T}[p \mapsto X_p]. \end{aligned}$$

However, we will show that the following statement holds:

$$\begin{aligned} \exists_{\text{fin}} p. \mathbb{A} \text{ accepts } \mathbb{T} \quad \text{iff} \quad & \text{there is a finite set } X_p \subseteq T \\ & \text{such that } \mathbb{A}_p^\ddagger \text{ accepts } \mathbb{T}[p \mapsto X_p]. \end{aligned}$$

Since X_p is finite, Theorem 7.2.12(3) tells us that that \mathbb{A}_p^\ddagger accepts $\mathbb{T}[p \mapsto X_p]$ iff \mathbb{A} accepts $\mathbb{T}[p \mapsto X_p]$. Therefore, the two statements above are equivalent.

\Rightarrow It is not difficult to prove that properties (1,2) in Theorem 7.2.12 hold for $\exists_{\text{fin}} p. \mathbb{A}$ as well, since the latter is defined in terms of \mathbb{A}_p^\ddagger . Therefore we can assume that the given winning strategy f_\exists for \exists in $\mathcal{G}_\exists = \mathcal{A}(\exists_{\text{fin}} p. \mathbb{A}, \mathbb{T}) @ (a_I^F, s_I)$ is functional, finitely branching and well-founded in A^\wp . Functionality allows us to associate with each node s either none or a unique state $Q_s \in A^\wp$ (cf. [Zan12, Prop. 3.12]). We now want to isolate the nodes that f_\exists treats “as if they were labeled with p ”. For this purpose, let V_s be the valuation suggested by f_\exists at a position $(Q_s, s) \in A^\wp \times T$. As f_\exists is winning, the one-step model $(R[s], V_s)$ makes the formula $\Delta^\exists(Q_s, \kappa(s)) = \Psi_{Q, \kappa(s)} \vee \Psi_{Q, \kappa(s) \cup \{p\}}$ true. We define

$$X_p := \{s \in T \mid Q_s \text{ is defined and } (R[s], V_s) \models \Psi_{Q, \kappa(s) \cup \{p\}}\}.$$

The fact that f_\exists is functional in A^\wp guarantees that X_p is well-defined; as the strategy is finitely branching and well-founded in A^\wp we get that X_p is finite. Let $\mathbb{T}' := \mathbb{T}[p \mapsto X_p]$, we show that we can give a winning strategy f_\ddagger for \exists in the game $\mathcal{G}_\ddagger = \mathcal{A}(\mathbb{A}_p^\ddagger, \mathbb{T}') @ (a_I^F, s_I)$. Actually, we show that $f_\ddagger := f_\exists$ works. We do it by induction for a match π_\ddagger of \mathcal{G}_\ddagger . We keep a shadow match π_\exists in \mathcal{G}_\exists such that the following condition holds at each round:

$$\text{Both matches } \pi_\ddagger \text{ and } \pi_\exists \text{ are in the same position } (q, s) \in A \cup A^\wp \times T. \quad (\ddagger)$$

This condition obviously holds at the beginning of the games. For the inductive step let $\kappa' = \kappa[p \mapsto X_p]$ be an abbreviation for the coloring of \mathbb{T}' and consider the following cases:

- If the current basic position in π_\ddagger is of the form $(a, s) \in A \times T$, then by definition of X_p we know that $s \notin X_p$, so $p \notin \kappa'(s)$ and hence $\kappa'(s) = \kappa(s)$. As f_\exists is winning in \mathcal{G}_\exists we know that the suggested valuation $V_{a,s}$ is admissible in π_\exists , that is, $(R[s], V_{a,s}) \models \Delta(a, \kappa(s))$. As $\kappa'(s) = \kappa(s)$, we can conclude that $(R[s], V_{a,s}) \models \Delta(a, \kappa'(s))$ and thus is also an admissible move in π_\ddagger .
- If the current basic position in π_\ddagger is of the form $(Q, s) \in A^\wp \times T$ we let $V_{Q,s}$ be the valuation suggested by f_\exists and consider the following cases:
 1. If $p \in \kappa'(s)$: then by definition of X_p we have that $(R[s], V_{Q,s}) \models \Psi_{Q, \kappa(s) \cup \{p\}}$. As $\kappa'(s) = \kappa(s) \uplus \{p\}$ we have that $(R[s], V_{Q,s}) \models \Psi_{Q, \kappa'(s)}$. This is, by definition of \mathbb{A}_p^\ddagger , equivalent to $(R[s], V_{Q,s}) \models \Delta^F(Q, \kappa'(s))$ and therefore $V_{Q,s}$ is admissible in π_\ddagger .

2. If $p \notin \kappa'(s)$: then $(R[s], V_{Q,s}) \models \Psi_{Q,\kappa(s)} \vee \Psi_{Q,\kappa(s) \cup \{p\}}$ but $(R[s], V_{Q,s}) \not\models \Psi_{Q,\kappa(s) \cup \{p\}}$ hence it must be the case that $(R[s], V_{Q,s}) \models \Psi_{Q,\kappa(s)}$. As $\kappa'(s) = \kappa(s)$, then $(R[s], V_{Q,s}) \models \Psi_{Q,\kappa'(s)} = \Delta^F(Q, \kappa'(s))$ and therefore $V_{Q,s}$ is admissible in $\pi_{\dot{\exists}}$.

As the move by \exists is the same in both matches it is clear that we can mimic in the shadow match π_{\exists} the choice of \forall in $\pi_{\dot{\exists}}$, therefore preserving (\dagger) .

It is only left to show that this strategy is winning for \exists . It is enough to observe that $\pi_{\dot{\exists}}$ and π_{\exists} go through the same basic positions and, as \exists wins π_{\exists} , she also wins $\pi_{\dot{\exists}}$.

\Leftarrow Given a winning strategy $f_{\dot{\exists}}$ for \exists in $\mathcal{G}_{\dot{\exists}} := \mathcal{A}(\mathbb{A}_p^{\dot{\exists}}, \mathbb{T}') @ (a_I^F, s_I)$ it is not difficult to see that the same strategy is winning for \exists in $\mathcal{G}_{\exists} := \mathcal{A}(\exists_{\text{fin}} p. \mathbb{A}, \mathbb{T}) @ (a_I^F, s_I)$. As before, we can maintain the following invariant between a match π_{\exists} of \mathcal{G}_{\exists} and a shadow match $\pi_{\dot{\exists}}$ of $\mathcal{G}_{\dot{\exists}}$:

The matches $\pi_{\dot{\exists}}$ and π_{\exists} are in the same position $(q, s) \in A \cup A^{\wp} \times T$.

The key observation in this case is that whenever the match $\pi_{\dot{\exists}}$ is in a position (a, s) then $p \notin \kappa'(s)$. This is because $\Delta^F(a, c) = \perp$ if $p \in c$ and that would contradict that $f_{\dot{\exists}}$ is winning. As a consequence, $\Delta^{\exists}(a, \kappa(s)) = \Delta^F(a, \kappa'(s))$ and therefore the move suggested by $f_{\dot{\exists}}$ in $\mathcal{G}_{\dot{\exists}}$ will also be admissible in \mathcal{G}_{\exists} . \square

The translation. We are now ready to prove the main result of this section.

7.2.16. PROPOSITION. *For every formula $\varphi \in \text{WMSO}(\mathbb{P})$ with free variables $F \subseteq \mathbb{P}$ we can effectively construct an automaton $\mathbb{A}_{\varphi} \in \text{Aut}_{wc}(\text{FOE}_1^{\infty}, F)$ such that for every F-tree \mathbb{T} we have $\mathbb{T} \models \varphi$ iff $\mathbb{T} \models \mathbb{A}_{\varphi}$.*

Proof. The proof is by induction on φ .

- For the base cases $p \sqsubseteq q$ and $R_{\ell}(p, q)$ we use the same automata as in Section 7.1.1. It is easy to syntactically check that these automata are continuous-weak.
- For the Boolean cases we only discuss the closure under complementation. In order to prove closure under complementation, we crucially use that the one-step language FOE_1^{∞} is closed under Boolean duals (cf. Proposition 5.1.57).

CLAIM 1. For every $\mathbb{A} \in \text{Aut}_{wc}(\text{FOE}_1^{\infty})$ the automaton \mathbb{A}^{δ} defined in Definition 2.3.6 belongs to $\text{Aut}_{wc}(\text{FOE}_1^{\infty})$ and recognizes the complement of $\mathcal{T}(\mathbb{A})$.

PROOF OF CLAIM. Since we already know that \mathbb{A}^{δ} accepts exactly the transition systems that are rejected by \mathbb{A} , we only need to check that \mathbb{A}^{δ} indeed belongs to $\text{Aut}_{wc}(\text{FOE}_1^{\infty})$. But this is straightforward: for instance, the (co-)continuity constraint can be checked by observing the dual nature of continuity and co-continuity. \blacktriangleleft

- For the case $\varphi = \exists_{\text{fin}} p. \psi$ let F be the set of free variables of φ . We only consider the case where p is free in ψ as otherwise $\varphi \equiv \psi$ and by induction hypothesis we already have an automaton \mathbb{A}_ψ which we can use as \mathbb{A}_φ .

Let $\mathbb{A}_\psi \in \text{Aut}_{wc}(\text{FOE}_1^\infty, F \uplus \{p\})$ be given by the inductive hypothesis. We define $\mathbb{A}_\varphi := \exists_{\text{fin}} p. \mathbb{A}_\psi$ using the construction given in Definition 7.2.14. Observe that \mathbb{A}_φ is an automaton over $\varphi(F)$ and that:

$$\begin{aligned} \mathbb{T} \models \exists_{\text{fin}} p. \mathbb{A}_\psi & \quad \text{iff} \quad \mathbb{T}[p \mapsto X_p] \models \mathbb{A}_\psi \text{ for a finite set } X_p \subseteq T & \quad (\text{Lem. 7.2.15}) \\ & \quad \text{iff} \quad \mathbb{T}[p \mapsto X_p] \models \psi \text{ for a finite set } X_p \subseteq T & \quad (\text{IH}) \\ & \quad \text{iff} \quad \mathbb{T} \models \exists_{\text{fin}} p. \psi & \quad (\text{semantics of WMSO}) \end{aligned}$$

This finishes the proof of the proposition. \square

7.2.3 From $\text{Aut}_{wc}(\text{FOE}_1)$ to $\text{FO}(\text{LFP}^1)$

From Section 7.1.2 we know that we can translate parity automata to the forward-looking fragment μFOE^\gg of μFOE . In this section we show that we can translate an automaton $\mathbb{A} \in \text{Aut}_{wc}(\text{FOE}_1^\infty)$ to a formula $\varphi_{\mathbb{A}}^\gg(x)$ of the fragment $\mu_c\text{FOE}^{\infty\gg}$ which is defined as follows.

7.2.17. DEFINITION. The forward-looking fragment $\mu\text{FOE}^{\infty\gg}$ of μFOE^∞ is defined like μFOE^\gg (cf. Definition 7.1.12) with the additional restriction that the generalized quantifier $\exists^\infty x. \varphi$ is also guarded. The forward-looking fragment $\mu_c\text{FOE}^{\infty\gg}$ of $\mu_c\text{FOE}^\infty$ is defined as $\mu_c\text{FOE}^{\infty\gg} := \mu_c\text{FOE}^\infty \cap \mu\text{FOE}^{\infty\gg}$.

We are now ready to prove the main result of this section.

7.2.18. PROPOSITION. *For every automaton $\mathbb{A} \in \text{Aut}_{wc}(\text{FOE}_1^\infty, P)$ we can effectively construct a formula $\varphi_{\mathbb{A}}^\gg(x) \in \mu_c\text{FOE}^{\infty\gg}(P)$ with exactly one free variable x , such that for every transition system \mathbb{S} , and $s \in S$*

$$\exists \text{ wins } \mathcal{A}(\mathbb{A}, \mathbb{S}) @ (a_I, s) \quad \text{iff} \quad \mathbb{S} \models \varphi_{\mathbb{A}}(s).$$

Proof. The same proof as Proposition 7.1.20 works almost unchanged. It is easy to see that the procedure gives us a formula $\varphi_{\mathbb{A}}(x) \in \mu\text{FOE}^{\infty\gg}$. Additionally, we also have to check that the resulting formula belongs to the right fragment.

CLAIM 1. $\varphi_{\mathbb{A}}(x) \in \mu_c\text{FOE}^\infty$.

PROOF OF CLAIM. It is not difficult to show, inductively, that if $a \in A$ belongs to a maximal strongly connected component $C \subseteq A$ of odd (resp. even) parity then $\beta_a^\dagger(z)$ will be continuous (resp. co-continuous) in $C \subseteq A$. This is enough, because then the fixpoint operators bind formulas of the right kind. \blacktriangleleft

Hence we can conclude that $\varphi_{\mathbb{A}}(x) \in \mu_c\text{FOE}^{\infty\gg}$. \square

As before, we get the following corollary on trees.

7.2.19. COROLLARY. *For every automaton $\mathbb{A} \in \text{Aut}_{wc}(\text{FOE}_1^\infty, \mathbf{P})$ we can effectively construct a sentence $\varphi_{\mathbb{A}} \in \mu_c\text{FOE}^\infty(\mathbf{P})$ such that for every tree \mathbb{T} ,*

$$\mathbb{T} \models \mathbb{A} \quad \text{iff} \quad \mathbb{T} \models \varphi_{\mathbb{A}}.$$

Proof. Simply set $\varphi_{\mathbb{A}} := \exists x.\text{isroot}(x) \wedge \varphi_{\mathbb{A}}^\gg(x)$. \square

7.2.4 From $\mu_c\text{FOE}^\infty$ to WMSO

In Proposition 3.2.37 of Chapter 3 we already proved that there is an effective translation $(-)^t$ from $\mu_c\text{FOE}^\infty$ to 2WMSO such that for every model \mathbb{M} , assignment g and $\varphi \in \mu_c\text{FOE}^\infty$ we have: $\mathbb{M}, g \models \varphi$ if and only if $\mathbb{M}, g \models \varphi^t$.

In order to prove that proposition it was crucial to characterize the least fixpoints of continuous maps. That is, we proved that for every continuous map $F : \wp(M) \rightarrow \wp(M)$, we have $s \in \text{LFP}(F)$ iff $s \in \text{LFP}(F|_Y)$ for some finite set $Y \subseteq M$. Using this characterization, we were able to encode the fixpoint operator of $\mu_c\text{FOE}^\infty$ using the finite set quantification of 2WMSO.

7.3 Automata for WCL

In this section we give an automata characterization for WCL, on trees. Namely,

7.3.1. THEOREM. *WCL and $\text{Aut}_{wa}(\text{FOE}_1)$ are effectively equivalent on trees.*

We start by defining what a WCL-automaton is.

7.3.2. DEFINITION. A WCL-automaton is an automaton from $\text{Aut}_{wa}(\text{FOE}_1)$.

The main result of this section is that the formalisms $\text{Aut}_{wa}(\text{FOE}_1)$ and WCL are effectively equivalent on trees. As we did for MSO and WMSO we will go through a fixpoint logic to prove this result. If we look at our previous results, it would seem like $\mu_a\text{FOE}$ is a good candidate for such a fixpoint logic. However, this will not be the case. We defer a discussion on this to Section 7.3.4 and 7.4.

7.3.1 Simulation theorem

In this section we provide, for every automaton $\mathbb{A} \in \text{Aut}_{wa}(\text{FOE}_1, \mathcal{P}')$ and proposition $p \in \mathcal{P}'$, an automaton $\mathbb{A}_p^\ddagger \in \text{Aut}_{wa}(\text{FOE}_1, \mathcal{P}')$ which, although not being fully non-deterministic, is specially tailored to prove the closure of $\text{Aut}_{wa}(\text{FOE}_1)$ under finite chain projection. The key property of \mathbb{A}_p^\ddagger is that it is equivalent to \mathbb{A} for a certain class of trees.

For all \mathbb{T} where $\kappa^\sharp(p)$ is a finite chain we have: $\mathbb{T} \models \mathbb{A}_p^\ddagger$ iff $\mathbb{T} \models \mathbb{A}$.

Recall that $\kappa^\sharp : \mathcal{P} \rightarrow \wp(T)$ is the valuation associated to \mathbb{T} . The construction will be similar to what we did in Section 7.2.1, when we gave a simulation theorem for WMSO. However, in this case we will care that the transition map of the non-deterministic part of the automaton is *completely additive* in A^\wp .

7.3.3. DEFINITION. Given an automaton $\mathbb{A} \in \text{Aut}(\mathcal{L})$, a subset $B \subseteq A$ of the states of \mathbb{A} , and a tree \mathbb{T} ; a strategy f for \exists in $\mathcal{A}(\mathbb{A}, \mathbb{T})$ is called *non-branching in B* if all the nodes of \mathbb{T}_f with a state from B belong to the same branch of \mathbb{T}_f .

We will construct \mathbb{A}_p^\ddagger such that it satisfies the properties of the above definition (for every tree and strategy) for the set of states A^\wp . That is, for the non-deterministic part.

In order to define the non-deterministic part of \mathbb{A}_p^\ddagger we need to adapt the formulas in the transition map Δ of \mathbb{A} accordingly, and ensure that the formulas of the non-deterministic part are completely additive in A^\wp .

For example, consider the formula $\alpha = \nabla_{\text{FOE}}^+(\{a, b\} \cdot \{a\}, \{a\})_s$ belonging to $\text{FOE}_1^+(A)$. The usual approach is to define the lifted version of α as the formula $\alpha' := \nabla_{\text{FOE}}^+(\{\{a, b\}\} \cdot \{\{a\}\}, \{\{a\}\})_s$ which now belongs to $\text{FOE}_1^+(A^\wp)$. The problem with this formula, is that it is not completely additive in A^\wp .

To overcome this problem, we will only perform a partial lifting on α . That is, we lift α in such a way that we obtain a formula which is completely additive in A^\wp . For α , there are three ways to do this (changes are underlined):

- $\alpha'_1 := \nabla_{\text{FOE}}^+(\{a, b\} \cdot \{a\}, \{a\})_s$
- $\alpha'_2 := \nabla_{\text{FOE}}^+(\{\underline{\{a, b\}}\} \cdot \{a\}, \{a\})_s$
- $\alpha'_3 := \nabla_{\text{FOE}}^+(\{a, b\} \cdot \{\underline{\{a\}}\}, \{a\})_s$

We call these liftings ‘additive liftings’ of α and then define $\alpha' := \bigvee_i \alpha'_i$. The main intuition behind this definition is the ‘non-branching’ condition of Definition 7.3.3. What we want, is that at any given point of a match of the acceptance game, at most one of the choices of \forall can stay in the non-deterministic part, and the rest of the branching matches should go to the alternating part. That is, \exists should never be required to colour more than one element with a state of A^\wp . We obtain this through the complete additivity of α' .

This finishes the intuitive explanations and we now turn to the necessary definitions to prove the results.

7.3.4. DEFINITION. Let $\alpha \in \text{FOE}_1^+(A, \mathcal{S})$ be of the shape $\nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathcal{S}}$ for some $\overline{\mathbf{T}} \in \wp(A)^k$ and $\Pi \subseteq \overline{\mathbf{T}}$. We say that $\alpha' \in \text{FOE}_1^+(A^\wp \cup A, \mathcal{S})$ is an *additive lifting* of α if the following conditions are satisfied:

- (i) $\alpha' = \nabla_{\text{FOE}}^+(\overline{\mathbf{R}}, \Pi)_{\mathcal{S}}$ for some $\overline{\mathbf{R}} \in \wp(A^\wp \cup A)^k$,
- (ii) For every i , either: (a) $R_i = T_i$, or (b) $T_i \neq \emptyset$ and $R_i = \{T_i\}$.
- (iii) Case (ii.b) occurs at most once.

Consider now $\psi \in \text{FOE}_1^+(A, \mathcal{S})$ of the shape $\bigwedge_{\mathcal{S}} \alpha_{\mathcal{S}}$. We say that a formula $\psi' \in \text{FOE}_1^+(A^\wp \cup A, \mathcal{S})$ is a *additive lifting* of ψ if $\psi' = \bigwedge_{\mathcal{S}} \alpha'_{\mathcal{S}}$ and

- (i) For every $\mathcal{S} \subseteq \mathcal{S}$, either:
 - (a) $\alpha'_{\mathcal{S}} = \alpha_{\mathcal{S}}$, or
 - (b) $\alpha'_{\mathcal{S}}$ is a additive lifting of $\alpha_{\mathcal{S}}$.
- (ii) Case (i.b) occurs at most once.

Observe that every such α' is completely additive in A^\wp .

7.3.5. DEFINITION. Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle \in \text{Aut}_{wa}(\text{FOE}_1, \mathcal{P}')$. Let $c \in \wp(\mathcal{P}')$ be a color and $Q \in A^\wp$ be a macro-state. First consider the formula $\bigwedge_{a \in Q} \Delta(a, c)$. By Corollary 5.2.25 there is a formula $\Phi_{Q,c} \in \text{FOE}_1^+(A)$ such that $\Phi_{Q,c} \equiv \bigwedge_{a \in Q} \Delta(a, c)$ and $\Phi_{Q,c}$ is in the basic form $\bigvee_j \varphi_j$ where $\varphi_j = \bigwedge_{\mathcal{S}} \nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathcal{S}}$. We define

$$\Psi_{Q,c} := \bigvee_j \bigvee \{\psi \mid \psi \text{ is an additive lifting of } \varphi_j\}.$$

Observe that $\Psi_{Q,c} \in \text{FOE}_1^+(A^\wp \cup A)$ and $\Psi_{Q,c}$ is completely additive in A^\wp . The latter is because the additive liftings have this property, which is preserved by disjunction.

We are finally ready to define the two-part construct.

7.3.6. DEFINITION. Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ belong to $\text{Aut}_{wa}(\text{FOE}_1, \mathcal{P}')$ and let p be a propositional variable. We define the *two-part construct* of \mathbb{A} with respect to p as the automaton $\mathbb{A}_p^\ddagger = \langle A^F, \Delta^F, \Omega^F, a_I^F \rangle \in \text{Aut}_{wa}(\text{FOE}_1, \mathcal{P}')$ given by:

$$\begin{aligned} A^F &:= A \cup A^\wp & \Delta^F(Q, c) &:= \Psi_{Q,c} \\ a_I^F &:= \{a_I\} \\ \Omega^F(Q) &:= 1 & \Delta^F(a, c) &:= \begin{cases} \perp & \text{if } p \in c, \\ \Delta(a, c) & \text{otherwise.} \end{cases} \\ \Omega^F(a) &:= \Omega(a) \end{aligned}$$

Before proving that the two-part construct satisfies nice properties we need a few propositions. The following two lemmas show how to go from admissible moves in the two-part construct to the original automaton and vice-versa.

7.3.7. LEMMA (ND TO ALT). *Given an automaton \mathbb{A} from $\text{Aut}_{wa}(\text{FOE}_1, \mathbf{P}')$, a macro-state $Q \in A^\wp$ and a color $c \in \wp(\mathbf{P}')$ such that $(D, V_{Q,c}) \models \Psi_{Q,c}$ for some valuation $V_{Q,c} : A^\wp \cup A \rightarrow \wp(D)$; there is a valuation $U : A \rightarrow \wp(D)$ such that*

- (i) $(D, U) \models \Delta(a, c)$ for all $a \in Q$,
- (ii) If $d \in U(b)$ then either
 - (a) $d \in V_{Q,c}(b)$, or
 - (b) $d \in V_{Q,c}(Q')$ for some $Q' \in A^\wp$ such that $b \in Q'$.

Proof. This lemma is proved exactly as Lemma 7.2.9 but using the notion of additive lifting instead of continuous lifting. \square

7.3.8. LEMMA (ALT TO ND). *Let \mathbb{A} belong to $\text{Aut}_{wa}(\text{FOE}_1, \mathbf{P}')$, $Q \in A^\wp$ be a macro-state, and $c \in \wp(\mathbf{P}')$ be a color. Let $\{V_{a,c} : A \rightarrow \wp(D) \mid a \in Q\}$ be a family of valuations such that $(D, V_{a,c}) \models \Delta(a, c)$ for each $a \in Q$. Then, for every $P \subseteq D$ with $|P| \leq 1$ there is a valuation $V_{Q,c} : A \cup A^\wp \rightarrow \wp(D)$ such that*

- (i) $(D, V_{Q,c}) \models \Psi_{Q,c}$,
- (ii) If $d \in V_{Q,c}(b)$ then $d \in V_{a,c}(b)$ for some $a \in Q$.
- (iii) If $d \in V_{Q,c}(Q')$ then $d \in V_{a,c}(b)$ for some $a \in Q$, $b \in Q'$.
- (iv) If $d \in P \cap V_{Q,c}(q)$ then $q \in A^\wp$.

Proof. We first define an auxiliary valuation $V_t : A \rightarrow \wp(D)$ which gathers all the valuations from the hypothesis, that is $V_t(b) := \bigcup_{a \in Q} V_{a,c}(b)$.

CLAIM 1. $(D, V_t) \models \bigwedge_{a \in Q} \Delta(a, c)$.

PROOF OF CLAIM. Observe that for every $a \in Q$, $b \in A$ we have $V_{a,c}(b) \subseteq V_t(b)$; then by monotonicity we get that $(D, V_t) \models \Delta(a, c)$ for every $a \in Q$. \blacktriangleleft

Define the valuation $V_{Q,c} : A \cup A^\wp \rightarrow \wp(D)$, using the alternative marking representation $V_{Q,c}^\sharp : D \rightarrow \wp(A \cup A^\wp)$, as follows:

$$V_{Q,c}^\sharp(d) := \begin{cases} V_t^\sharp(d) & \text{if } d \notin P, \\ \{V_t^\sharp(d)\} & \text{if } d \in P. \end{cases}$$

and recall that $\bigwedge_{a \in Q} \Delta(a, c) \equiv \bigvee_i \varphi_i$ and

$$\Psi_{Q,c} := \bigvee_j \bigvee \{\psi \mid \psi \text{ is an additive lifting of } \varphi_j\}.$$

Assume that $(D, V_t) \models \varphi_j$. We show that $(D, V_{Q,c}) \models \psi$ for some additive lifting of φ_j . If P is empty then $V_{Q,c} = V_t$ and as φ_j is itself a additive lifting of φ_j and $(D, V_t) \models \varphi_j$, we can conclude that $(D, V_{Q,c}) \models \varphi_j$ and we are done.

If $P = \{d\}$ we proceed as follows: first recall that the shape of φ_j is $\varphi_j = \bigwedge_{\mathbf{s}} \nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$. Let $\mathbf{S}_d \subseteq \mathcal{S}$ be the set of sorts to which d belongs. We show that $(D, V_{Q,c}) \models \psi_{\mathbf{S}_d}$ for some additive lifting $\psi_{\mathbf{S}_d}$ of $\nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathbf{S}_d}$. This will be enough, since it is easy to show that in that case $\psi := \psi_{\mathbf{S}_d} \wedge \bigwedge_{\mathbf{s} \neq \mathbf{S}_d} \nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathbf{s}}$ is a additive lifting of φ_j (by Definition 7.3.4) and $(D, V_{Q,c}) \models \psi$.

Our hypothesis is that $(D, V_t) \models \nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathbf{S}_d}$. Also, observe that the formula $\nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathbf{S}_d}$ gives a full description of the elements of D with sorts \mathbf{S}_d . Namely, if we restrict to the elements of sorts \mathbf{S}_d , then:

- There are distinct $d_1, \dots, d_k \in D$ such that d_i has type T_i ,
- Every $d' \in D$ which is not among d_1, \dots, d_k satisfies some type in Π .

We consider the following two cases:

- (1) Suppose that $d = d_i$ for some i ; without loss of generality assume that $i = 1$. In this case it is easy to see that $(D, V_{Q,c}) \models \nabla_{\text{FOE}}^+(\{T_1\} \cdot T_2 \cdots T_k, \Pi)_{\mathbf{S}_d}$, which is an additive lifting of $\nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathbf{S}_d}$.
- (2) Suppose that $d \neq d_i$ for all i . Then, d must have some type $S_d \in \Pi$. The key observation is that $\Pi \subseteq \overline{\mathbf{T}}$. Hence, there is some T_i such that $T_i = S_d$. Observe now that if we ‘switch’ the elements d and d_i we end up in case (1).

This finishes the proof of the lemma. \square

Finally we can state and prove the properties of the two-part construct.

7.3.9. THEOREM. *Let $\mathbb{A} \in \text{Aut}_{wa}(\text{FOE}_1, \mathbf{P}')$, $p \in \mathbf{P}'$ be a propositional variable, and \mathbb{T} be a \mathbf{P}' -tree. The following holds:*

1. $\mathbb{A}_p^\dagger \in \text{Aut}_{wa}(\text{FOE}_1, \mathbf{P}')$.
2. If \exists wins $\mathcal{A}(\mathbb{A}_p^\dagger, \mathbb{T}) @ (a_I^F, s_I)$ then she has a winning strategy which is functional, non-branching and well-founded in A^\wp .
3. If $\kappa^{\mathbb{A}}(p)$ is a finite chain then: \mathbb{A}_p^\dagger accepts \mathbb{T} iff \mathbb{A} accepts \mathbb{T} .

Proof. (1) The key observation is that $\Psi_{Q,c}$ is completely additive in A^\wp .

(2) The functional and well-founded parts are shown exactly as in Theorem 7.2.12. For the non-branching property we do as follows: This is direct from the syntactical form of $\Psi_{Q,c}$. Observe that in each disjunct, at most one element of A^\wp can occur. Assuming that \exists plays minimal strategies then she always proposes a valuation V where $V(A^\wp)$ is a quasi-atom.

(3) This point is proved as Theorem 7.2.12(3) but using Lemma 7.3.7 and 7.3.8 to perform the transformation between alternating and functional strategies. \square

7.3.2 From WCL to $Aut_{wa}(\text{FOE}_1)$

In this section we give an effective transformation from formulas of WCL to automata of $Aut_{wa}(\text{FOE}_1)$. As we observed, this class is not closed under the existential quantification of MSO (that is, under ‘projection.’) We start by proving that this class is closed under a ‘finite chain projection’.

Finite chain projection.

7.3.10. DEFINITION. Let $p \notin P'$ and let L be a tree language of $(P' \uplus \{p\})$ -labeled trees. The *finite chain projection* of L over p is the language $\exists_{\text{fch}} p.L$ of P' -labeled trees defined as

$$\exists_{\text{fch}} p.L := \{\mathbb{T} \mid \mathbb{T}[p \mapsto X_p] \in L \text{ for some finite chain } X_p \subseteq T\}.$$

In the following definition we give, for every $\mathbb{A} \in Aut_{wa}(\text{FOE}_1, P' \uplus \{p\})$, the finite chain projection over p , denoted $\exists_{\text{fch}} p.\mathbb{A} \in Aut_{wa}(\text{FOE}_1, P')$. The domain and transition function of the automaton $\exists_{\text{fch}} p.\mathbb{A}$ will be based on \mathbb{A}_p^\ddagger .

7.3.11. DEFINITION. Let \mathbb{A} belong to $Aut_{wa}(\text{FOE}_1, P' \uplus \{p\})$. We define the *finite chain projection of \mathbb{A} over p* as the automaton $\exists_{\text{fch}} p.\mathbb{A} := \langle A \cup A^\varrho, \Delta^\exists, \Omega^\exists, \{a_I\} \rangle$ from $Aut_{wa}(\text{FOE}_1, P')$ given as follows, for every $c \in \wp(P')$:

$$\begin{aligned} \Omega^\exists(a) &:= \Omega(a) & \Delta^\exists(a, c) &:= \Delta(a, c) \\ \Omega^\exists(Q) &:= 1 & \Delta^\exists(Q, c) &:= \Psi_{Q,c} \vee \Psi_{Q,c \cup \{p\}} \end{aligned}$$

where $\Psi_{Q,c}$ is as in Definition 7.3.5.

The key observation to be made about the above definition is that $\exists_{\text{fch}} p.\mathbb{A}$ is actually based on the two-part construction \mathbb{A}_p^\ddagger (see Definition 7.3.6). The main change is that the non-deterministic part (A^ϱ) has been projected with respect to p . This can be observed in the definition of $\Delta^\exists(Q, c)$.

7.3.12. LEMMA. *For each automaton $\mathbb{A} \in Aut_{wa}(\text{FOE}_1, P' \uplus \{p\})$ we have that $\mathcal{T}(\exists_{\text{fch}} p.\mathbb{A}) = \exists_{\text{fch}} p.\mathcal{T}(\mathbb{A})$.*

Proof. Same as Lemma 7.2.15 but using the simulation theorem for WCL-automata, i.e, Theorem 7.3.9. \square

The translation. We are now ready to prove the main result of this section.

7.3.13. PROPOSITION. *For every formula $\varphi \in \text{WCL}(P)$ with free variables $F \subseteq P$ we can effectively construct an automaton $\mathbb{A}_\varphi \in Aut_{wa}(\text{FOE}_1, F)$ such that for every F -tree \mathbb{T} we have $\mathbb{T} \models \varphi$ iff $\mathbb{T} \models \mathbb{A}_\varphi$.*

Proof. The proof is by induction on φ .

- For the base cases $p \sqsubseteq q$ and $R_\ell(p, q)$ we give the following automata, which are a generalization of the ones given before, to the context of many relations:
 $\mathbb{A}_{p \sqsubseteq q} := \langle \{a_0\}, \Delta, \Omega, a_0 \rangle$ where $\Omega(a_0) = 0$ and

$$\Delta(a_0, c) := \begin{cases} \bigwedge_{\mathbf{s}} \forall x:\mathbf{s}. a_0(x) & \text{if } q \in c \text{ or } p \notin c, \\ \perp & \text{otherwise} \end{cases}$$

$\mathbb{A}_{R_\ell(p, q)} := \langle \{a_0, a_1\}, \Delta, \Omega, a_0 \rangle$ where $\Omega(a_0) = \Omega(a_1) = 0$ and

$$\Delta(a_0, c) := \begin{cases} \exists x:\ell. a_1(x) \wedge \bigwedge_{\mathbf{s}} (\forall y:\mathbf{s}. a_0(y)) & \text{if } p \in c, \\ \bigwedge_{\mathbf{s}} \forall x:\mathbf{s}. a_0(x) & \text{otherwise.} \end{cases}$$

$$\Delta(a_1, c) := \begin{cases} \top & \text{if } q \in c, \\ \perp & \text{if } q \notin c. \end{cases}$$

It is easy to syntactically check that these automata are additive-weak and also it is not too difficult to see that they do what they should.

- For the Boolean cases we only discuss the closure under complementation. In order to prove closure under complementation, we crucially use that the one-step language FOE_1 is closed under Boolean duals (cf. Proposition 5.1.57).

CLAIM 1. For every $\mathbb{A} \in \text{Aut}_{wa}(\text{FOE}_1)$ the automaton \mathbb{A}^δ defined in Definition 2.3.6 belongs to $\text{Aut}_{wa}(\text{FOE}_1)$ and recognizes the complement of $\mathcal{T}(\mathbb{A})$.

PROOF OF CLAIM. Since we already know that \mathbb{A}^δ accepts exactly the transition systems that are rejected by \mathbb{A} , we only need to check that \mathbb{A}^δ indeed belongs to $\text{Aut}_{wa}(\text{FOE}_1)$. But this is straightforward: for instance, the additivity and multiplicativity constraints can be checked by observing the dual nature of these properties. ◀

- For the case $\varphi = \exists_{\text{fch}} p. \psi$ let F be the set of free variables of φ . We only consider the case where p is free in ψ as otherwise $\varphi \equiv \psi$ and by induction hypothesis we already have an automaton \mathbb{A}_ψ which we can use as \mathbb{A}_φ .

Let $\mathbb{A}_\psi \in \text{Aut}_{wa}(\text{FOE}_1, F \uplus \{p\})$ be given by the inductive hypothesis. We define $\mathbb{A}_\varphi := \exists_{\text{fch}} p. \mathbb{A}_\psi$ using the construction given in Definition 7.3.11. Observe that \mathbb{A}_φ is an automaton over $\wp(F)$ and that:

$$\begin{aligned} \mathbb{T} \models \exists_{\text{fch}} p. \mathbb{A}_\psi & \text{ iff } \mathbb{T}[p \mapsto X_p] \models \mathbb{A}_\psi \text{ for a finite chain } X_p \subseteq T \text{ (Lem. 7.3.12)} \\ & \text{ iff } \mathbb{T}[p \mapsto X_p] \models \psi \text{ for a finite chain } X_p \subseteq T \quad \text{(IH)} \\ & \text{ iff } \mathbb{T} \models \exists_{\text{fch}} p. \psi \quad \text{(semantics of WCL)} \end{aligned}$$

This finishes the proof of the proposition. \square

7.3.3 From $Aut_{wa}(\text{FOE}_1)$ to $\text{FO}(\text{LFP}^1)$

From Section 7.1.2 we know that we can translate parity automata to the forward-looking fragment μFOE^\gg of μFOE . In this section we show that if the automaton is additive-weak then we can translate $\mathbb{A} \in Aut_{wa}(\text{FOE}_1)$ to a formula $\varphi_{\mathbb{A}}^\gg(x)$ of the fragment $\mu_a\text{FOE}^\gg$ which is defined as follows.

7.3.14. DEFINITION. The forward-looking fragment $\mu_a\text{FOE}^\gg$ of $\mu_a\text{FOE}$ is defined as $\mu_a\text{FOE}^\gg := \mu_a\text{FOE} \cap \mu\text{FOE}^\gg$.

We are now ready to prove the main result of this section.

7.3.15. PROPOSITION. *For every automaton $\mathbb{A} \in Aut_{wa}(\text{FOE}_1, \mathbf{P})$ we can effectively construct a formula $\varphi_{\mathbb{A}}^\gg(x) \in \mu_a\text{FOE}^\gg(\mathbf{P})$ with exactly one free variable x , such that for every transition system \mathbb{S} , and $s \in S$*

$$\exists \text{ wins } \mathcal{A}(\mathbb{A}, \mathbb{S}) @ (a_I, s) \quad \text{iff} \quad \mathbb{S} \models \varphi_{\mathbb{A}}(s).$$

Proof. The same proof as Proposition 7.1.20 works almost unchanged. We need to modify the definition of the guarding formula to deal with the sorts, as follows:

$$\Delta_{a,c}^g(x) := \Delta(a, c) [\exists y: \mathbf{s}. \alpha \mapsto \exists y. R_{\mathbf{s}}(x, y) \wedge \alpha; \forall y: \mathbf{s}. \alpha \mapsto \forall y. R_{\mathbf{s}}(x, y) \rightarrow \alpha].$$

Additionally, we also have to check that the resulting formula belongs to the right fragment. We already know that $\varphi_{\mathbb{A}}(x) \in \mu\text{FOE}^\gg$, we need to check the following.

CLAIM 1. $\varphi_{\mathbb{A}}(x) \in \mu_a\text{FOE}$.

PROOF OF CLAIM. It is not difficult to show, inductively, that if $a \in A$ belongs to a maximal strongly connected component $C \subseteq A$ of odd parity (resp. even) then $\beta_a^\dagger(z)$ will be completely additive (resp. multiplicative) in $C \subseteq A$. This is enough, because then the fixpoint operators bind formulas of the right kind. \blacktriangleleft

Hence we can conclude that $\varphi_{\mathbb{A}}(x) \in \mu_a\text{FOE}^\gg$. \square

As before, we get the following corollary on trees.

7.3.16. COROLLARY. *For every automaton $\mathbb{A} \in Aut_{wa}(\text{FOE}_1, \mathbf{P})$ we can effectively construct a sentence $\varphi_{\mathbb{A}} \in \mu_a\text{FOE}(\mathbf{P})$ such that for every tree \mathbb{T} ,*

$$\mathbb{T} \models \mathbb{A} \quad \text{iff} \quad \mathbb{T} \models \varphi_{\mathbb{A}}.$$

Proof. Simply set $\varphi_{\mathbb{A}} := \exists x. \text{isroot}(x) \wedge \varphi_{\mathbb{A}}^\gg(x)$. \square

7.3.4 From $\mu_a\text{FOE}^\gg$ to WCL

In this case we will make use of the correspondence between single-sorted WCL and the two-sorted version 2WCL given in Section 2.8 and give a translation from $\mu_a\text{FOE}^\gg$ to 2WCL. It is important to observe that our source logic is $\mu_a\text{FOE}^\gg$ and not $\mu_a\text{FOE}$. The reasoning behind this restriction will be discussed in Section 7.4, after the necessary elements are introduced in the current section.

The translation will be given inductively, but before giving the translation, we focus for a moment on the fixpoint case. Let $\varphi = [\text{LFP}_{p,y}.\psi(p,y)](x) \in \mu_a\text{FOE}^\gg$ be such that $FV(\psi) \subseteq \{y\}$. Observe that, because of the definition of μFOE^\gg , this formula *does not* have parameters in the fixpoint. In consequence, the only variable free in φ is x .

7.3.17. PROPOSITION. *The formula $\psi(p,y)$ restricts to descendants and is completely additive in p .*

Proof. Both claims are almost direct from the definition of the fragment $\mu_a\text{FOE}^\gg$. As $\psi(p,y) \in \mu_a\text{FOE}^\gg\text{ADD}_p$ we know that it is completely additive in p , and by Lemma 7.1.19 we know that formulas of μFOE^\gg restrict to descendants. \square

We can now use Proposition 7.1.18 and obtain that the functional F^ψ induced by ψ is completely additive and restricts to descendants as well. The following theorem, which will be critical for our translation, combines the content of Theorem 3.1.10 and Theorem 7.1.17 and gives a characterization of the fixpoint of maps that, at the same time, are completely additive and restrict to descendants.

7.3.18. THEOREM. *Let $F : \wp(M) \rightarrow \wp(M)$ be completely additive and restrict to descendants. For every $s \in M$ we have that $s \in \text{LFP}(F)$ iff $s \in \text{LFP}(F|_Y)$ for some finite chain Y .*

Proof. This theorem follows from a minor modification of Lemma 3.1.11 for completely additive maps, which provides elements $t_1, \dots, t_k = s$ such that $t_i \in F^i(\emptyset)$. The key observation is that, because F restricts to descendants, we can choose the elements in a way that $t_i R^* t_{i+1}$ for all i .

The main change to the proof of Lemma 3.1.11 is when we want to define u_i in terms of $u_{i+1} \in F^{i+1}(\emptyset)$. By definition we have that $u_{i+1} \in G(F^i(\emptyset), \bar{\mathbf{Y}})$. Now we can use that G restricts to descendants and get that $u_{i+1} \in G(F^i(\emptyset) \cap R^*[u_{i+1}], \bar{\mathbf{Y}} \cap R^*[u_{i+1}])$. By complete additivity of G there is a quasi-atom $(T, \bar{\mathbf{Q}}')$ of $(F^i(\emptyset) \cap R^*[u_{i+1}], \bar{\mathbf{Y}} \cap R^*[u_{i+1}])$ such that $u_{i+1} \in G(T, \bar{\mathbf{Q}}')$. This means that the element chosen from T will be a descendant of u_{i+1} and therefore we will get a (finite) chain. \square

We are now ready to give the translation.

7.3.19. PROPOSITION. *There is an effective translation $ST : \mu_a\text{FOE}^\gg(\mathbf{P}) \rightarrow 2\text{WCL}(\mathbf{P})$ such that for every model \mathbb{M} and assignment g we have $\mathbb{M}, g \models \varphi$ if and only if $\mathbb{M}, g \models ST(\varphi)$.*

Proof. Clearly the interesting case is that of the fixpoint operator. Let $\varphi = [\text{LFP}_{p,y}.\psi(p, y)](x) \in \mu_a\text{FOE}^\gg$ be such that $FV(\psi) \subseteq \{y\}$. By induction hypothesis we know that there is a formula $\psi'(p, y) \in 2\text{WCL}$, which is equivalent to $\psi(p, y)$. We define the translation of the fixpoint as follows:

$$\begin{aligned} ST([\text{LFP}_{p,y}.\psi(p, y)](x)) &:= \exists_{\text{fch}} Y. (\forall_{\text{fch}} W \subseteq Y. W \in \text{PRE}(F_{|Y}^\psi) \rightarrow x \in W) \\ W \in \text{PRE}(F_{|Y}^\psi) &:= \forall v. \psi'(W, v) \wedge v \in Y \rightarrow v \in W. \end{aligned}$$

Observe that we use ψ' in the translation, since $F_{|Y}^\psi = F_{|Y}^{\psi'}$. To justify the translation first recall from Section 7.1.3 that the translation of $[\text{LFP}_{p,y}.\psi(p, y)](x)$ into MSO is given by:

$$\forall W. (W \in \text{PRE}(F^\psi) \rightarrow x \in W)$$

where $W \in \text{PRE}(F^\psi)$ expresses that W is a prefixpoint of $F^\psi : \wp(M) \rightarrow \wp(M)$. This translation is based on the following fact about fixpoints of monotone maps:

$$s \in \text{LFP}(F^\psi) \quad \text{iff} \quad s \in \bigcap \{W \subseteq M \mid W \in \text{PRE}(F^\psi)\}. \quad (\text{PRE})$$

In our translation, however, we cannot make use of the arbitrary set quantifier $\exists W$, since we are dealing with WCL. The crucial observation is that, as F^ψ is completely additive and restricts to descendants, then we can restrict ourselves to finite chains, in the following sense:

$$\begin{aligned} s \in \text{LFP}(F^\psi) \quad \text{iff} \quad s \in \text{LFP}(F_{|Y}^\psi) \text{ for some f.c. } Y & \quad (\text{Theorem 7.3.18}) \\ \text{iff} \quad s \in \bigcap \{W \subseteq Y \mid W \in \text{PRE}(F_{|Y}^\psi)\} \text{ for some f.c. } Y. & \quad (\text{PRE}) \end{aligned}$$

The crucial observation in the last step is that $W \subseteq Y$ because the domain and range of $F_{|Y}^\psi$ is Y , and not M as in the original map F^ψ . Therefore, the translation basically expresses the same as the MSO case, but relativized to a finite chain Y . The correctness of the translation is then justified by the above equations. \square

As a corollary, we get that we can express the sentences of $\mu_a\text{FOE}$ corresponding to $\text{Aut}_{wa}(\text{FOE}_1)$, in 2WCL.

7.3.20. COROLLARY. *For every $\mathbb{A} \in \text{Aut}_{wa}(\text{FOE}_1)$, the formula $\varphi_{\mathbb{A}} \in \mu_a\text{FOE}$ of Corollary 7.3.16 is (effectively) expressible in 2WCL.*

Proof. The formula $\varphi_{\mathbb{A}}$ is defined as $\exists x. \text{isroot}(x) \wedge \varphi_{\mathbb{A}}^\gg(x)$ where $\varphi_{\mathbb{A}}^\gg(x) \in \mu_a\text{FOE}^\gg$. As the predicate $\text{isroot}(x)$ can be expressed in first-order logic, the formula $\exists x. \text{isroot}(x) \wedge ST(\varphi_{\mathbb{A}}^\gg)(x)$ belongs to 2WCL and is equivalent to $\varphi_{\mathbb{A}}$. \square

7.4 The question of automata for FO(TC¹)

In this chapter we analyzed several concrete first-order automata, and we gave logical characterizations for them, both as a second-order logic and as a fixpoint logic. The results of this chapter are summarized in the following table, where each row represents equivalent formalisms. We use $\mathcal{L}_{\text{root}}$ to denote the language \mathcal{L} extended with $\exists x.\text{isroot}(x)$.

<i>Automata</i>	<i>Second-order</i>	<i>Fixpoint</i>	<i>Forward-looking</i>
$Aut(\text{FOE}_1)$	MSO	μFOE	$\mu\text{FOE}_{\text{root}}^{\gg}$
$Aut_{wc}(\text{FOE}_1^\infty)$	WMSO	$\mu_c\text{FOE}^\infty$	$\mu_c\text{FOE}_{\text{root}}^{\infty\gg}$
$Aut_{wa}(\text{FOE}_1)$	WCL	–	$\mu_a\text{FOE}_{\text{root}}^{\gg}$

If we look at the first two cases, a pattern seems to emerge: the corresponding fixpoint logic for $Aut(\text{FOE}_1)$ is FOE extended with a fixpoint operator, i.e., μFOE . For $Aut_{wc}(\text{FOE}_1^\infty)$ we have FOE_1^∞ as the one-step language and an extra continuity condition on cycles. Therefore, we get $\mu_c\text{FOE}^\infty$, that extends FOE^∞ with a fixpoint operator which is restricted to continuous formulas. Following this pattern, a natural candidate for the third row is $\mu_a\text{FOE}$, since it is the natural extension of FOE with completely additive fixpoints. Using Theorem 3.1.44, this would give us automata for FO(TC¹).

However, in Section 7.3.4 we saw that when translating a completely additive least fixpoint $[\text{LFP}_{p:y}.\psi(p,y)](x)$ to WCL, we crucially used that ψ restricts to descendants. This condition was necessary to ensure that the least fixpoint of such a formula forms (on trees) a finite *chain*, which is the kind of quantification that we have available in WCL.² Unfortunately, the least fixpoint of an arbitrary completely additive formula of $\mu_a\text{FOE}$ need not be a chain. Therefore, there does not seem to be a clear way to express these fixpoints with WCL. That is, unless $\mu_a\text{FOE}$ is equivalent to $\mu_a\text{FOE}_{\text{root}}^{\gg}$, or equivalently, to WCL. We think this is not the case.

7.4.1. CONJECTURE. $\mu_a\text{FOE} \not\subseteq \text{WCL}$, *even on trees.*

The failure of such an equivalence can also be looked at under the light of the following intuitions, which attempt to explain the connection among the mentioned formalisms: the projection theorem for $Aut(\text{FOE}_1)$ basically runs the automaton $\exists p.\mathbb{A}$ in such a way that it is possible to extract a coloring X for p . This set X does not have any particular form, and this is paralleled by the arbitrary quantification of MSO. On the fixpoint side, we have μFOE , whose fixpoints are also arbitrary sets, which can be emulated with the quantification of MSO.

²Actually, the fixpoint of maps that are completely additive and restrict to descendants is not necessarily a chain, but we can restrict to such a chain, as stated in Theorem 7.3.18. For this section, when we say that a fixpoint is a chain or finite set we mean that we can restrict to such a set.

For $Aut_{wc}(\text{FOE}_1^\infty)$ things start to change. The restrictions on the cycles have a direct impact on the projection theorem. The weakness condition makes the extracted coloring to be included in a well-founded tree. The continuity condition makes this coloring to be finitely branching. Incidentally, on trees the notion of finite set and “set embed in a finitely branching and well-founded tree” coincide. This is the key reason why these automata correspond to WMSO. On the fixpoint side, we have $\mu_c\text{FOE}^\infty$, whose fixpoints are finite sets, which can be simulated with WMSO.

The case of $Aut_{wa}(\text{FOE}_1)$ turns out to be quite different, because of the strong restriction entailed by the additivity constraint. The weakness condition, again, makes the extracted coloring well-founded. The additivity condition, makes this coloring a chain. Therefore, we get finite chain projection. As we saw, this projection coincides with the quantification of WCL. However, contrary to the other cases, WCL imposes a clear *connectivity* condition on the quantified set. Namely, it has to be a chain; in other words, every element has to belong to the same branch. This brings immediate problems when looking at the associated fixpoint logics. If we look at the least fixpoint of a formula $\varphi \in \mu_a\text{FOE}^\gg$ we know that $s \in F_\varphi^{i+1}(\emptyset)$ only depends on (at most) one element of $F_\varphi^i(\emptyset)$ by complete additivity and, moreover, that such an element can be taken among the descendants of s , by restriction to descendants. This means that all the elements can be taken to be in a chain. However, if we look at a formula $\psi \in \mu_a\text{FOE}$, we only have the additivity condition. Pictorially, this means that, even though we can express the least fixpoint as a set of elements s_1, \dots, s_k , these elements may belong to *completely different branches*. This suggests that we cannot express such a fixpoint with WCL.

In some sense, we can say that parity automata as considered in this dissertation (i.e., with one-step formulas on the transition map) are inherently *modal*, in the sense that they always travel downwards on the trees, and always do it from a node to its successors. Taking into account their iterating capability (cycles) it is quite natural that they correspond to a fixpoint logic *which restricts to descendants*. This is the most fundamental connection. The fact that they correspond to a second-order logic or to a non-forward-looking fixpoint logic can be seen as a secondary byproduct. In the case of WMSO this last correspondence holds thanks to finite sets coinciding with “well-founded and finitely branching sets.” In the case of MSO the correspondence is trivial. For $Aut_{wa}(\text{FOE}_1)$, however, an analogous correspondence does not seem to hold.

7.5 Conclusions and open problems

In this chapter we gave automata characterizations for WMSO and WCL, as the classes $Aut_{wc}(\text{FOE}_1^\infty)$ and $Aut_{wa}(\text{FOE}_1)$ respectively. We also proved that each of these formalisms corresponds, as well, to some extension of first-order logic with

fixpoint operators. In order to prove these results, we also developed new simulation theorems for continuous-weak and additive-weak automata respectively.

Open problems.

1. *Automata for FO(TC¹)*: As extensively discussed in Section 7.4, the question of parity automata for FO(TC¹) –or equivalently, for μ_a FOE– is still open. We have stated Conjecture 7.4.1 which says that $\text{WCL} \not\subseteq \mu_a\text{FOE}$ on trees. If this conjecture is true, then the corresponding automata for FO(TC¹) would have to be essentially different from $\text{Aut}_{wa}(\text{FOE}_1)$; on the other hand, if the conjecture is false, we would be able to use $\text{Aut}_{wa}(\text{FOE}_1)$ for FO(TC¹).
2. *Logical characterization without projection*: Among the key technical tools of this chapter are the simulation and projection constructions. With them, we can mimic the second-order quantification on the automata side. Moreover, in each case we then use the second-order quantification to simulate the fixpoint operator of the associated fixpoint logic. In some cases it may be interesting to go directly from a fixpoint logic to its corresponding automata (for example, for μ_a FOE) without going through any second-order logic (like WCL). For this purpose, it would be interesting to study an “inverse” operation of the transformation given in Section 7.1.2, which transforms parity automata into formulas of μ FOE using partial unraveling of automata. It is possible that the technique used in [Jan06, Theorem 3.2.2.1] for modal automata could be adapted to the setting of first-order automata.

Chapter 8

Expressiveness modulo bisimilarity

In this chapter we use the tools developed in the previous chapters to prove novel bisimulation-invariance results. Namely, our main results are:

$$\mu_c\text{ML} \equiv \text{WMSO}/\leftrightarrow \quad \text{and} \quad \text{PDL} \equiv \text{WCL}/\leftrightarrow.$$

Moreover, we show that the equivalences are effective. That is, we give effective translations in both directions.

When proving results of the form $L \equiv L'/\leftrightarrow$, the inclusion $L \subseteq L'/\leftrightarrow$ is usually given by a more or less straightforward translation from L to L' . The inclusion $L \supseteq L'/\leftrightarrow$, however, requires much more work. In the context of fixpoint logics, the use of automata is a powerful technique to prove this direction.

In the original work of Janin and Walukiewicz [JW96], the authors prove that μML is the bisimulation-invariant fragment of MSO . An important step of the proof is to define a construction $(-)^{\bullet}$ that transforms automata from $\text{Aut}(\text{FOE}_1)$ to automata of $\text{Aut}(\text{FO}_1)$, in such a way that

$$\mathbb{A}^{\bullet} \text{ accepts } \mathbb{S} \quad \text{iff} \quad \mathbb{A} \text{ accepts } \mathbb{S}^{\omega},$$

for every transition system \mathbb{S} , where \mathbb{S}^{ω} is the ω -unraveling of \mathbb{S} . With this result at hand, it is not difficult to prove that $\text{Aut}(\text{FO}_1) \equiv \text{Aut}(\text{FOE}_1)/\leftrightarrow$. Using the fact that the automata of $\text{Aut}(\text{FOE}_1)$ correspond to MSO (on trees) and the automata of $\text{Aut}(\text{FO}_1)$ correspond to μML , it is now simple to get the final result.

A key observation made by Venema in [Ven14] is that the construction $(-)^{\bullet}$ is completely determined at the one-step level, by a translation $(-)_1^{\bullet} : \text{FOE}_1 \rightarrow \text{FO}_1$ satisfying certain properties. In this chapter we show also that this technique provides a nice modular way of approaching (automata-based) bisimulation-invariance proofs and, moreover, it works for subclasses of parity automata.

Finally, we discuss the relative expressive power of PDL , WCL and $\text{FO}(\text{TC}^1)$ and the quest for the bisimulation-invariant fragment of $\text{FO}(\text{TC}^1)$.

8.1 Continuous-weak automata

In this section characterize the bisimulation-invariant fragment of WMSO.

8.1.1. THEOREM. $\mu_c\text{ML}$ is effectively equivalent to $\text{WMSO}/\leftrightarrow$.

In this section we work with the logics $\mu_c\text{ML}$ and WMSO and with the class of continuous-weak automata $\text{Aut}_{wc}(\mathcal{L})$. In particular, this means that we consider a signature of a single relation R and single-sorted one-step languages.

We start by defining a construction $(-)^{\bullet} : \text{Aut}_{wc}(\text{FOE}_1^{\infty}) \rightarrow \text{Aut}_{wc}(\text{FO}_1)$ such that for every automaton \mathbb{A} and transition system \mathbb{S} we have:

$$\mathbb{A}^{\bullet} \text{ accepts } \mathbb{S} \quad \text{iff} \quad \mathbb{A} \text{ accepts } \mathbb{S}^{\omega}.$$

As we shall see, the map $(-)^{\bullet}$ is completely determined at the one-step level, that is, by some model-theoretic connection between FOE_1^{∞} and FO_1 .

8.1.2. DEFINITION. Using Corollary 5.1.35 any formula in $\text{FOE}_1^{\infty+}(A)$ is equivalent to a disjunction of formulas of the form $\nabla_{\text{FOE}^{\infty}}^+(\overline{\mathbf{T}}, \Pi, \Sigma)$. We define the one-step translation $(-)^{\bullet}_1 : \text{FOE}_1^{\infty+}(A) \rightarrow \text{FO}_1^+(A)$ as follows:

$$(\nabla_{\text{FOE}^{\infty}}^+(\overline{\mathbf{T}}, \Pi, \Sigma))^{\bullet}_1 := \bigwedge_i \exists x_i. \tau_{T_i}^+(x_i) \wedge \forall x. \bigvee_{S \in \Sigma} \tau_S^+(x)$$

and for $\alpha = \bigvee_i \alpha_i$ we define $(\alpha)^{\bullet}_1 := \bigvee_i (\alpha_i)^{\bullet}_1$.

Observe that, as computing the normal form of a formula in $\text{FOE}_1^{\infty+}(A)$ is effective, then this one-step translation is also clearly effective. The key property of this translation is the following.

8.1.3. PROPOSITION. For every $\alpha \in \text{FOE}_1^{\infty+}(A)$ the following holds:

(1) If $(D, V) \models \alpha^{\bullet}_1$ then there is a valuation V_{π} such that

- (i) $(D \times \omega, V_{\pi}) \models \alpha$, and
- (ii) If $(d, i) \in V_{\pi}(a)$ then $d \in V(a)$.

(2) If $(D \times \omega, V) \models \alpha$ then there is a valuation V_{\bullet} such that

- (i) $(D, V_{\bullet}) \models \alpha^{\bullet}_1$, and
- (ii) If $d \in V_{\bullet}(a)$ then $(d, i) \in V(a)$ for some i .

Proof. As every formula of $\text{FOE}_1^{\infty+}(A)$ is equivalent to a disjunction of formulas of the form $\nabla_{\text{FOE}^{\infty}}^+(\overline{\mathbf{T}}, \Pi, \Sigma)$ it will be enough to prove the statement for formulas of the form $\alpha = \nabla_{\text{FOE}^{\infty}}^+(\overline{\mathbf{T}}, \Pi, \Sigma)$.

(1): Suppose $(D, V) \models \alpha_1^\bullet$ and define $V_\pi^\natural((d, k)) := V^\natural(d)$. We will show that $(D \times \omega, V_\pi) \models \nabla_{\text{FOE}^\infty}^+(\overline{\mathbf{T}}, \Pi, \Sigma)$. Let d_i be such that $\tau_{T_i}^+(d_i)$ in (D, V) . It is clear that the (d_i, i) provide *distinct* elements satisfying $\tau_{T_i}^+((d_i, i))$ in $(D \times \omega, V_\pi)$ and therefore the first-order existential part of α is satisfied. With a similar but easier argument it is straightforward that the existential generalized quantifier part of α is also satisfied. For the universal parts of $\nabla_{\text{FOE}^\infty}^+(\overline{\mathbf{T}}, \Pi, \Sigma)$ it is enough to observe that, because of the universal part of α^\bullet , *every* $d \in D$ realizes a positive type in Σ . By construction, the same applies to $(D \times \omega, V_\pi)$, therefore this takes care of both universal quantifiers.

(2): Suppose that $(D \times \omega, V) \models \nabla_{\text{FOE}^\infty}^+(\overline{\mathbf{T}}, \Pi, \Sigma)$; we first define the auxiliary valuation $U : A \rightarrow \wp(D \times \omega)$ as follows:

$$U(a) := \{(d, i) \in D \times \omega \mid (d, j) \in V(a) \text{ for some } j \in \omega\}.$$

This valuation has two nice properties, namely: (1) $V(a) \subseteq U(a)$ for all $a \in A$, and (2) $U^\natural((d, i)) = U^\natural((d, j))$ for all $i, j \in \omega$. In particular, as $\nabla_{\text{FOE}^\infty}^+(\overline{\mathbf{T}}, \Pi, \Sigma)$ is monotone in A , from (1) we can conclude that $(D \times \omega, U) \models \nabla_{\text{FOE}^\infty}^+(\overline{\mathbf{T}}, \Pi, \Sigma)$. Finally, we define $V_\bullet : A \rightarrow \wp(D)$ as $V_\bullet^\natural(d) := U^\natural((d, i))$ for an arbitrary $i \in \omega$. This is well defined because of (2).¹ We now show that $(D, V_\bullet) \models \alpha^\bullet$. The existential part of α_1^\bullet is trivial. For the universal part we have to show that every element of D realizes the positive part of a type in Σ . Suppose not, and let $d \in D$ be such that $\neg \tau_S^+(d)$ for all $S \in \Sigma$. Then we have $(D \times \omega, U) \not\models \tau_S^+((d, k))$ for all k . That is, there are infinitely many elements not realizing the positive part of any type in Σ . Hence we have $(D \times \omega, U) \not\models \forall^\infty y. \bigvee_{S \in \Sigma} \tau_S^+(y)$. Absurd, because that is part of $\nabla_{\text{FOE}^\infty}^+(\overline{\mathbf{T}}, \Pi, \Sigma)$. \square

Finally, we can give the main definition of this section.

8.1.4. DEFINITION. Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ be an automaton in $\text{Aut}(\text{FOE}_1^\infty)$. We define $\mathbb{A}^\bullet := \langle A, \Delta^\bullet, \Omega, a_I \rangle$ in $\text{Aut}(\text{FO}_1)$ by putting, for each $(a, c) \in A \times \wp(\mathbf{P})$:

$$\Delta^\bullet(a, c) := (\Delta(a, c))_1^\bullet.$$

It remains to be checked that the construction $(-)^{\bullet}$, which has been defined for arbitrary automata in $\text{Aut}(\text{FOE}_1^\infty)$, transforms continuous-weak automata of $\text{Aut}_{wc}(\text{FOE}_1^\infty)$ into automata in the right class, that is, $\text{Aut}_{wc}(\text{FO}_1)$.

8.1.5. PROPOSITION. *If $\mathbb{A} \in \text{Aut}_{wc}(\text{FOE}_1^\infty)$, then $\mathbb{A}^\bullet \in \text{Aut}_{wc}(\text{FO}_1)$.*

Proof. This proposition can be verified by a straightforward inspection, at the one-step level, that if a formula $\alpha \in \text{FOE}_1^{\infty+}(A)$ belongs to the fragment $\text{FOE}_1^{\infty+}\text{CON}_a(A)$, then its translation α_1^\bullet lands in the fragment $\text{FO}_1^+\text{CON}_a(A)$. The same relationship holds between $\text{FOE}_1^{\infty+}\overline{\text{CON}}_a(A)$ and $\text{FO}_1^+\overline{\text{CON}}_a(A)$. \square

¹In connection to item (1), observe that we are defining V_\bullet so that $(V_\bullet)_\pi = U$.

We are now ready to prove the main lemma of this section.

8.1.6. LEMMA. *The construction $(-)^{\bullet} : \text{Aut}(\text{FOE}_1) \rightarrow \text{Aut}(\text{FO}_1)$ is effective, and for every automaton \mathbb{A} and transition system \mathbb{S} we have:*

$$\mathbb{A}^{\bullet} \text{ accepts } \mathbb{S} \quad \text{iff} \quad \mathbb{A} \text{ accepts } \mathbb{S}^{\omega}.$$

Proof. The proof of this lemma is based on a fairly routine comparison of the acceptance games $\mathcal{A}(\mathbb{A}^{\bullet}, \mathbb{S})$ and $\mathcal{A}(\mathbb{A}, \mathbb{S}^{\omega})$, using Proposition 8.1.3 to transform valuations from one game to the other. We give a proof of the left-to-right direction, and leave the other direction to the reader, since it is very similar.

We show that for every move made in a match π of $\mathcal{A}(\mathbb{A}^{\bullet}, \mathbb{S})$, we can respond in the match π^{ω} of $\mathcal{A}(\mathbb{A}, \mathbb{S}^{\omega})$ while keeping the following relationship:

Match π is in the basic position $(a, s) \in A \times S$ and match π^{ω} is in the basic position $(a, t) \in A \times S^{\omega}$ for some $s \leftrightarrow t$.

In the beginning of the games both matches are in the position (a, s_I) so this relationship is clearly satisfied. Now suppose that the matches π and π^{ω} are standing in positions (a, s) and (a, t) respectively, and let $\alpha = \Delta(a, \kappa(t))$. In his turn, \forall plays some valuation V in π such that $(R[s], V) \models \alpha_1^{\bullet}$. We have to show that \exists has an admissible move in π^{ω} , that is, she has to play a valuation U such that $(R[t] \times \omega, U) \models \alpha$. Now we crucially use Proposition 8.1.3(1) which solves our one-step problem: as $(R[s], V) \models \alpha_1^{\bullet}$ and $s \leftrightarrow t$ then we get that $(R[t] \times \omega, V_{\pi}) \models \alpha$. Therefore, \exists can play V_{π} and survive.

After this, the next move is for \forall , who chooses a basic position $(a', (d, i))$ such that $(d, i) \in R[t] \times \omega$ and $(d, i) \in V_{\pi}(a')$. Using Proposition 8.1.3(1ii) we have that $d \in R[s]$ satisfies $d \in V(a')$ and therefore it is clear that \exists can choose the position (a', d) in the shadow match π . Using that $s \leftrightarrow t$ we also obtain that $d \leftrightarrow (d, i)$. This finishes the round and preserves the relationship between the games. Since both games go through basic positions with the same state in \mathbb{A} then, since \exists wins π by hypothesis, then she will also win π^{ω} . \square

8.1.7. REMARK. As a corollary of the previous two propositions we find that

- $\text{Aut}(\text{FO}_1) \equiv \text{Aut}(\text{FOE}_1^{\infty}) / \leftrightarrow$, and
- $\text{Aut}_{wc}(\text{FO}_1) \equiv \text{Aut}_{wc}(\text{FOE}_1^{\infty}) / \leftrightarrow$.

In fact, we are dealing here with an instantiation of a more general phenomenon that is essentially coalgebraic in nature. In [Ven14] it is proved that if \mathcal{L} and \mathcal{L}' are two one-step languages that are connected by a translation $(-)^{\bullet}_1 : \mathcal{L}' \rightarrow \mathcal{L}$ satisfying a condition similar to those in Proposition 8.1.3, then we find that $\text{Aut}(\mathcal{L})$ corresponds to the bisimulation-invariant fragment of $\text{Aut}(\mathcal{L}')$. This subsection can be generalized to prove similar results relating $\text{Aut}_w(\mathcal{L})$ to $\text{Aut}_w(\mathcal{L}')$, and $\text{Aut}_{wc}(\mathcal{L})$ to $\text{Aut}_{wc}(\mathcal{L}')$.

8.1.1 Bisimulation-invariant fragment of WMSO

In this section we prove the following equivalence:

$$\mu_c\text{ML} \equiv \text{WMSO}/\leftrightarrow.$$

Moreover, we prove that the equivalence is effective. One of the inclusions is given by a translation $\mu_c\text{ML} \rightarrow 2\text{WMSO}$.

8.1.8. PROPOSITION. *There is an effective translation $(-)^t : \mu_c\text{ML} \rightarrow 2\text{WMSO}$ such that for every $\varphi \in \mu_c\text{ML}$ we have that $\varphi \equiv \varphi^t$.*

Proof. Since we already proved in Proposition 3.2.37 that $\mu_c\text{FOE}^\infty \subseteq 2\text{WMSO}$, it is enough to give a translation $\text{ST}_x : \mu_c\text{ML} \rightarrow \mu_c\text{FOE}^\infty$ and obtain the desired translation $\mu_c\text{ML} \rightarrow 2\text{WMSO}$ by composition.

The only interesting case of the translation is the fixpoint operator. Let $\varphi = \mu p.\psi(p)$ where ψ is continuous in p . We define the translation as follows:

$$\text{ST}_x(\mu p.\psi) := [\text{LFP}_{p.y}.\text{ST}_y(\psi)](x).$$

The correctness of this translation is straightforward. □

For the other inclusion we prove the following stronger lemma.

8.1.9. LEMMA. *There is an effective translation $(-)_\blacktriangledown : \text{WMSO} \rightarrow \mu_c\text{ML}$ such that for every $\varphi \in \text{WMSO}$ we have that $\varphi \equiv \varphi_\blacktriangledown$ iff φ is bisimulation-invariant.*

Proof. The translation $(-)_\blacktriangledown : \text{WMSO} \rightarrow \mu_c\text{ML}$ is defined as follows: given a formula $\varphi \in \text{WMSO}$ we first construct an automaton $\mathbb{A}_\varphi \in \text{Aut}_{wc}(\text{FOE}_1^\infty)$ as done in Proposition 7.2.16. Next, we compute the automaton $\mathbb{A}_\varphi^\bullet \in \text{Aut}_{wc}(\text{FO}_1)$ using Lemma 8.1.6. To finish, we use Theorem 6.3.1 to get a formula $\varphi_\blacktriangledown \in \mu_c\text{ML}$.

CLAIM 1. $\varphi \equiv \varphi_\blacktriangledown$ iff φ is invariant under bisimulation.

The left to right direction is trivial because $\varphi_\blacktriangledown \in \mu_c\text{ML}$, therefore if $\varphi \equiv \varphi_\blacktriangledown$ it also has to be invariant under bisimulation. The opposite direction is obtained by the following chain of equivalences:

$$\begin{aligned} \mathbb{S} \models \varphi & \text{ iff } \mathbb{S}^\omega \models \varphi && (\varphi \text{ bisimulation invariant}) \\ & \text{ iff } \mathbb{S}^\omega \models \mathbb{A}_\varphi. && (\text{Theorem 7.2.1: } \text{WMSO} \equiv \text{Aut}_{wc}(\text{FOE}_1^\infty) \text{ on trees}) \\ & \text{ iff } \mathbb{S} \Vdash \mathbb{A}_\varphi^\bullet && (\text{Lemma 8.1.6}) \\ & \text{ iff } \mathbb{S} \Vdash \varphi_\blacktriangledown. && (\text{Theorem 6.3.1}) \end{aligned}$$

This finishes the proof for WMSO. □

As a corollary of this lemma we get Theorem 8.1.1.

8.2 Additive-weak automata

In this section obtain the following results which closes the open questions of bisimulation-invariant characterizations for PDL and WCL.

8.2.1. THEOREM. *PDL is effectively equivalent to WCL/ \leftrightarrow .*

In this section we work with the logics PDL and WCL and with the class of additive-weak automata $Aut_{wa}(\mathcal{L})$. In particular, we use a signature with many relations $(R_\ell)_{\ell \in \mathbb{D}}$ and many-sorted one-step languages. In this context, the difference between normal and strict trees will play a role. We refer the reader to Section 2.1 for these definitions.

We start by defining a construction $(-)^{\bullet} : Aut_{wa}(FOE_1) \rightarrow Aut_{wa}(FO_1)$ such that for every automaton \mathbb{A} and *strict* tree \mathbb{T} we have:

$$\mathbb{A}^{\bullet} \text{ accepts } \mathbb{T} \quad \text{iff} \quad \mathbb{A} \text{ accepts } \mathbb{T}^{\omega}.$$

Again, the map $(-)^{\bullet}$ is completely determined at the one-step level. This time, by some model-theoretic connection between FOE_1 and FO_1 . We will make use of the one-step translation $(-)^{\bullet}_1 : FOE_1^+(A, \mathcal{S}) \rightarrow FO_1^+(A, \mathcal{S})$ given in Definition 5.2.32 for formulas of $FOE_1^+(A, \mathcal{S})$ which are in strict normal form. Recall that:

$$(\nabla_{FOE}^+(\overline{\mathbb{T}}, \Pi)_s)^{\bullet}_1 := \nabla_{FO}^+(\overline{\mathbb{T}}, \Pi)_s$$

and for $\alpha = \bigvee \bigwedge_s \alpha_s$ we define $(\alpha)^{\bullet}_1 := \bigvee \bigwedge_s (\alpha_s)^{\bullet}_1$. In the context of this section, the key property of this translation is the following.

8.2.2. PROPOSITION. *For every $\alpha \in FOE_1^+(A, \mathcal{S})$ in strict normal form and every strict one-step model (D_1, \dots, D_n, V) the following holds:*

- (i) *If $(D, V) \models \alpha^{\bullet}_1$ then there is a valuation V_{π} such that*
 - (i) *$(D \times \omega, V_{\pi}) \models \alpha$, and*
 - (ii) *If $(d, i) \in V_{\pi}(a)$ then $d \in V(a)$.*
- (ii) *If $(D \times \omega, V) \models \alpha$ then there is a valuation U such that*
 - (i) *$(D, V_{\bullet}) \models \alpha^{\bullet}_1$, and*
 - (ii) *If $d \in V_{\bullet}(a)$ then $(d, i) \in V(a)$ for some i .*

Proof. Formulas of $FOE_1^+(A, \mathcal{S})$ which are in strict normal form are of the shape $\bigvee \bigwedge_s \nabla_{FOE}^+(\overline{\mathbb{T}}, \Pi)_s$; therefore it will be enough to prove the proposition for $\alpha = \nabla_{FOE}^+(\overline{\mathbb{T}}, \Pi)_s$. We will reuse what we proved in Proposition 5.2.33, which is almost the same as what we want. Namely, Proposition 5.2.33 states that:

$$(D, V) \models \alpha^{\bullet}_1 \quad \text{iff} \quad (D \times \omega, V_{\pi}) \models \alpha, \quad (*)$$

where the valuation V_π is given by $V_\pi^{\mathfrak{h}}((d, k)) := V^{\mathfrak{h}}(d)$.

(i): This item is direct by the left-to-right direction of (*).

(2): Suppose that $(D \times \omega, V) \models \nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathfrak{s}}$; we define the valuation $U : A \rightarrow \wp(D)$ as follows:

$$U^{\mathfrak{h}}(d) := \bigcup_k V^{\mathfrak{h}}((d, k)).$$

The following claim connects V and U through the construction $(-)_\pi$ used in (*).

CLAIM 1. $V(a) \subseteq U_\pi(a)$ for all $a \in A$.

In particular, as $\nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathfrak{s}}$ is monotone in every $a \in A$, from this claim we get $(D \times \omega, U_\pi) \models \nabla_{\text{FOE}}^+(\overline{\mathbf{T}}, \Pi)_{\mathfrak{s}}$. Now we can use the right-to-left direction of (*) and get that $(D, U) \models \alpha_1^\bullet$. \square

Finally, we can give the main definition of this section.

8.2.3. DEFINITION. Let $\mathbb{A} = \langle A, \Delta, \Omega, a_I \rangle$ be an automaton in $\text{Aut}(\text{FOE}_1)$. We use Theorem 5.2.11 and get, for every $(a, c) \in A \times \wp(\mathbf{P})$, a formula $\psi_{a,c} \equiv \Delta(a, c)$ which is in strict normal form. We define the automaton $\mathbb{A}^\bullet := \langle A, \Delta^\bullet, \Omega, a_I \rangle$ in $\text{Aut}(\text{FO}_1)$ by putting, for each $(a, c) \in A \times \wp(\mathbf{P})$:

$$\Delta^\bullet(a, c) := (\psi_{a,c})_1^\bullet.$$

First, it needs to be checked that the construction $(-)^{\bullet}$, which has been defined for arbitrary automata in $\text{Aut}(\text{FOE}_1)$, transforms the additive-weak automata of $\text{Aut}_{wa}(\text{FOE}_1)$ into automata in the right class, that is, $\text{Aut}_{wa}(\text{FO}_1)$.

8.2.4. PROPOSITION. *If $\mathbb{A} \in \text{Aut}_{wa}(\text{FOE}_1)$, then $\mathbb{A}^\bullet \in \text{Aut}_{wa}(\text{FO}_1)$.*

Proof. This proposition can be verified by a straightforward inspection, at the one-step level, that if a formula $\alpha \in \text{FOE}_1^+(A)$ belongs to the fragment $\text{FOE}_1^+\text{ADD}_{A'}(A)$, then its translation α_1^\bullet lands in the fragment $\text{FO}_1^+\text{ADD}_{A'}(A)$. The same relationship holds for $\text{FOE}_1^+\text{MUL}_{A'}(A)$ and $\text{FO}_1^+\text{MUL}_{A'}(A)$. \square

We are now ready to prove the main lemma of this section.

8.2.5. LEMMA. *The construction $(-)^{\bullet} : \text{Aut}_{wa}(\text{FOE}_1) \rightarrow \text{Aut}_{wa}(\text{FO}_1)$ is effective, and for every automaton \mathbb{A} and strict tree \mathbb{T} we have:*

$$\mathbb{A}^\bullet \text{ accepts } \mathbb{T} \quad \text{iff} \quad \mathbb{A} \text{ accepts } \mathbb{T}^\omega.$$

Proof. This lemma is proved as Lemma 8.1.6, but using Proposition 8.2.2 and, additionally, that \mathbb{T} and \mathbb{T}^ω are strict trees. \square

8.2.1 Bisimulation-invariant fragment of WCL

In this section we prove the following equivalence:

$$\text{PDL} \equiv \text{WCL} / \leftrightarrow.$$

Moreover, we prove that the equivalence is effective.

One of the inclusions is given by a translation from PDL to WCL. We prove this through a detour via the modal μ -calculus. In Section 3.1.2 it is shown that PDL is equivalent to the fragment $\mu_a\text{ML}$ where the fixpoint operator $\mu p.\varphi$ is restricted to formulas which are completely additive in p . We will therefore give a translation $\text{ST}_x : \mu_a\text{ML} \rightarrow 2\text{WCL}$ which proves that $\text{PDL} \subseteq \text{WCL}$. The idea is to use basically the same translation as in Section 7.3.4 where we prove that $\mu_a\text{FOE}^\triangleright \subseteq \text{WCL}$.

8.2.6. PROPOSITION. *There is an effective translation $\text{ST}_x : \mu_a\text{ML} \rightarrow 2\text{WCL}$ such that for every $\varphi \in \mu_a\text{ML}$ we have that $\varphi \equiv \text{ST}_x(\varphi)$.*

Proof. The only interesting case of the translation is the fixpoint operator. Let $\varphi = \mu p.\psi(p)$ where ψ is completely additive in p . We state the following claim:

CLAIM 1. The formula $\psi \in \mu_a\text{ML}$ restricts to descendants.

PROOF OF CLAIM. This is clear because the formula belongs to μML . These formulas are invariant under generated submodels, in particular, they restrict to descendants. \blacktriangleleft

To finish, define the translation of the fixpoint as follows:

$$\begin{aligned} \text{ST}_x(\mu p.\psi) &:= \exists_{\text{fch}} Y. (\forall_{\text{fch}} W \subseteq Y. W \in \text{PRE}(F_Y^\psi) \rightarrow x \in W) \\ W \in \text{PRE}(F_Y^\psi) &:= \forall v. \text{ST}_v(\psi)[p \mapsto W] \wedge v \in Y \rightarrow v \in W. \end{aligned}$$

The correctness of this translation is a simplified version of the proof of Proposition 7.3.19, using Claim 1 and Theorem 7.3.18. \square

For the other inclusion we prove the following stronger lemma.

8.2.7. LEMMA. *There is an effective translation $(-)_\blacktriangledown : \text{WCL} \rightarrow \text{PDL}$ such that for every $\varphi \in \text{WCL}$ we have that $\varphi \equiv \varphi_\blacktriangledown$ iff φ is bisimulation-invariant.*

Proof. The translation $(-)_\blacktriangledown : \text{WCL} \rightarrow \text{PDL}$ is defined as follows: given a formula $\varphi \in \text{WCL}$ we first construct an automaton $\mathbb{A}_\varphi \in \text{Aut}_{wa}(\text{FOE}_1)$ as done in Section 7.3.2. Next, we compute the automaton $\mathbb{A}_\varphi^\bullet \in \text{Aut}_{wa}(\text{FO}_1)$ using Lemma 8.2.5. To finish, we use Theorem 6.2.2 to get a formula $\varphi_\blacktriangledown \in \text{PDL}$.

CLAIM 1. $\varphi \equiv \varphi_{\blacktriangledown}$ iff φ is invariant under bisimulation.

The left to right direction is trivial because $\varphi_{\blacktriangledown} \in \text{PDL}$, therefore if $\varphi \equiv \varphi_{\blacktriangledown}$ it also has to be invariant under bisimulation. The opposite direction is obtained by the following chain of equivalences. Recall that $\hat{\mathbb{S}}$ is the unraveling of \mathbb{S} .

$$\begin{aligned}
\mathbb{S} \models \varphi & \text{ iff } \hat{\mathbb{S}}^\omega \models \varphi && (\varphi \text{ bisimulation invariant}) \\
& \text{ iff } \hat{\mathbb{S}}^\omega \models \mathbb{A}_\varphi && (\text{Theorem 7.3.1: } \text{WCL} \equiv \text{Aut}_{wa}(\text{FOE}_1) \text{ on trees}) \\
& \text{ iff } \hat{\mathbb{S}} \Vdash \mathbb{A}_\varphi^\bullet && (\text{Lemma 8.2.5}) \\
& \text{ iff } \hat{\mathbb{S}} \Vdash \varphi_{\blacktriangledown} && (\text{Theorem 6.2.2}) \\
& \text{ iff } \mathbb{S} \Vdash \varphi_{\blacktriangledown}. && (\varphi_{\blacktriangledown} \text{ bisimulation invariant})
\end{aligned}$$

As Lemma 8.2.5 requires a strict tree, we use a detour through the unraveling of \mathbb{S} , which has this property. \square

As a corollary of this lemma we get Theorem 8.2.1.

8.2.2 Relative expressive power of PDL, WCL and FO(TC^1)

In this section we prove a few results regarding the relative expressive power of PDL, WCL and FO(TC^1). Namely, we prove that:

- PDL cannot be translated to a *naive* generalization of WCL from trees to arbitrary models. This gives insight on the relationship of PDL and chains.
- WCL and FO(TC^1) are not expressively equivalent.

The letter of the law. Recall from the preliminaries (Section 2.1) that,

- A *chain* on \mathbb{S} is a set $X \subseteq S$ such that (X, R^*) is a totally ordered set.
- A *generalized chain* is a set $X \subseteq S$ such that $X \subseteq P$, for some path P of \mathbb{S} .

In the definition of WCL in Section 2.8 we chose to follow the spirit of the original definition of CL in [Tho84], and we required the quantifier to range over generalized finite chains, instead of finite chains, in the context of arbitrary models.

There is another reason for this choice: if we had followed the letter of the definition and had required the quantifier to range over (non-generalized) finite chains in the context of arbitrary models, then PDL *would not* have been translatable to the resulting logic! Suppose then that we define a variant L of MSO with the quantifier:

$$\mathbb{S} \models \tilde{\exists}p.\varphi \quad \text{iff} \quad \text{there is a finite chain } X \subseteq S \text{ such that } \mathbb{S}[p \mapsto X] \models \varphi.$$

We show that L cannot express the PDL-formula $\varphi := \langle \ell^* \rangle p$. That is, L cannot express the property “I can reach an element colored with p .” Intuitively, the problem is that chains are a lot more restricted than paths (on arbitrary models).

We will first define a class of models where the expressive power of L is reduced to that of FOE, and then prove that FOE cannot express φ on this class of models. Let \mathbb{C}_i be defined as a model with i elements laid out on a circle (see Fig. 8.1) and \mathbb{C}_i^p be as \mathbb{C}_i but with one (any) element colored with p . We define the class of models $K := \{\mathbb{C}_i \uplus \mathbb{C}_i^p \mid i \geq 3\}$.

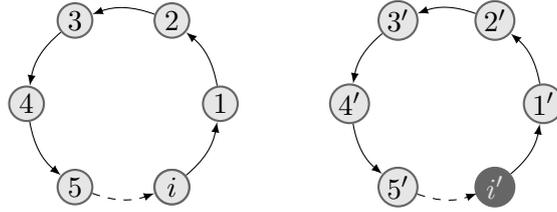


Figure 8.1: Model $\mathbb{C}_i \uplus \mathbb{C}_i^p$. The element i' is colored with p .

8.2.8. PROPOSITION. *Over the class K , we have $L \equiv \text{FOE}$.*

Proof. Every chain on a model of K is either a singleton or empty. □

Observe now that our formula $\varphi = \langle \ell^* \rangle p$ is true exactly in the elements of \mathbb{C}_i^p , and false in \mathbb{C}_i , for every i . Assume towards a contradiction that there is a formula $\psi \in L$ such that $\varphi \equiv \psi$ on all models. If we focus on K , using the above proposition, we must also have a formula $\gamma \in \text{FOE}$ such that $\psi \equiv \gamma$ (on K). We show that such a $\gamma \in \text{FOE}$ cannot exist.

To do it, we rely on the fact that first-order logic is “essentially local”, proved by Gaifman [Gai82]. Recall that the n -neighbourhood of an element e is the set of all the elements e' such that the undirected distance $\text{dist}(e, e')$ is smaller or equal than n . The following fact is a corollary of Gaifman’s theorem.

8.2.9. FACT. *For every first-order formula $\gamma(\bar{x})$ there is a number $t \in \mathbb{N}$ (which depends only on the quantifier rank of γ) such that for every model \mathbb{M} and elements $\bar{a}, \bar{a}' \in M$: if the t -neighbourhoods of a_i and a'_i are isomorphic for every i then $\mathbb{M} \models \gamma(\bar{a})$ iff $\mathbb{M} \models \gamma(\bar{a}')$.*

Let t be the number obtained by the above fact applied to $\gamma(x)$. To finish, we prove the following fact.

CLAIM 1. $\mathbb{C}_{4t} \models \gamma(2t)$ iff $\mathbb{C}_{4t} \models \gamma(2t')$.

This leads to a contradiction, since γ should be false at $2t$ and true at $2t'$.

PROOF OF CLAIM. The t -neighbourhoods of $2t$ and $2t'$ are isomorphic, since no element is colored with p with distance lower than t . Therefore by the above fact about first-order locality the two elements satisfy the same first-order formulas. A more detailed proof of a similar argument can be found in [LN99, Ex. 2]. ◀

As a consequence, we get the following proposition:

8.2.10. PROPOSITION. $\text{PDL} \not\subseteq L$.

Separating $\text{FO}(\text{TC}^1)$ and WCL on all models. We prove that $\text{FO}(\text{TC}^1) \not\subseteq \text{WCL}$ by showing that *undirected reachability* is expressible in $\text{FO}(\text{TC}^1)$ but not in WCL. First observe that in $\text{FO}(\text{TC}^1)$ the formula

$$\varphi(x, y) := [\text{TC}_{u,v}.R(u, v) \vee R(v, u)](x, y)$$

is true if and only if (1) $x = y$; or (2) there is a way to get from x to y disregarding the direction of the edges.

Consider the model shown in Fig. 8.2, which has two copies of the integers but with an alternating successor relation. The arrows denote the binary relation R which is *not* taken to be transitive.

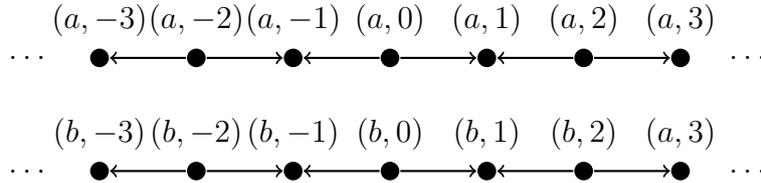


Figure 8.2: Separating example for $\text{FO}(\text{TC}^1)$ and WCL.

It was observed by Yde Venema (private communication) that on the model of Fig. 8.2 the expressive power of WCL collapses to that of plain first-order logic with equality. The reason for this is that every generalized chain (finite or not) has length at most one, and therefore the second order existential $\exists_{\text{fch}} X.\psi$ can be replaced by $\exists x_1, x_2.\psi'$ with a minor variation of ψ . Therefore, it will be enough to show that first-order logic cannot express undirected reachability over this model. Again, we will use Fact 8.2.9.

Assume that φ has an equivalent formulation $\varphi' \in \text{FOE}$. Let t be the number obtained by Fact 8.2.9. To finish, we prove that

CLAIM 1. $\mathbb{M} \models \varphi'((a, 0), (b, 0))$ iff $\mathbb{M} \models \varphi'((a, 0), (a, 2t))$.

Observe that this leads to a contradiction, since the first two elements are not connected and the second ones are.

PROOF OF CLAIM. The t -neighbourhoods of $(b, 0)$ and $(a, 2t)$ are isomorphic, therefore by Fact 8.2.9 the two elements satisfy the same first-order formulas. ◀

Observe also that in the above model \mathbb{M} we also have that non-weak chain logic (CL) collapses to first-order logic, therefore the same proof gives us that $\text{FO}(\text{TC}^1) \not\subseteq \text{CL}$. Therefore, we have proved the following proposition:

8.2.11. PROPOSITION. $\text{FO}(\text{TC}^1) \not\subseteq \text{WCL}$ and $\text{FO}(\text{TC}^1) \not\subseteq \text{CL}$.

The above results can also be proved with a finite part of \mathbb{M} , for example, restricting it to the segments $(a, \pm 4t)$ and $(b, \pm 4t)$.

8.2.3 On the bisimulation-invariant fragment of $\text{FO}(\text{TC}^1)$

It is thought in the modal logic community that, as $\text{FOE}/\leftrightarrow \equiv \text{ML}$, then it would be natural to have $\text{FO}(\text{TC}^1)/\leftrightarrow \equiv \text{PDL}$. However, we are not aware of any proof of this result. In this dissertation we have made some progress in this direction, giving different ways to look at $\text{FO}(\text{TC}^1)$ and PDL and a characterization result for PDL. Namely, the following results are relevant:

- $\text{FO}(\text{TC}^1) \equiv \mu_a \text{FOE}$ (Theorem 3.1.44)
- $\text{PDL} \equiv \text{Aut}_{wa}(\text{FO}_1) \equiv \mu_a \text{ML}$ (Theorem 6.2.2 and 3.1.27)
- $\text{WCL} \equiv \text{Aut}_{wa}(\text{FOE}_1) \equiv \mu_a \text{FOE}_{\text{root}}^{\gg}$ (Theorem 7.3.1 and Corollary 7.2.19)
- $\text{WCL}/\leftrightarrow \equiv \mu_a \text{FOE}_{\text{root}}^{\gg}/\leftrightarrow \equiv \text{PDL}$ (Theorem 8.2.1).

In Section 7.4 we discussed that $\text{Aut}_{wa}(\text{FOE}_1)$ does not seem to be a suitable class of automata for $\mu_a \text{FOE}$ and hence neither for $\text{FO}(\text{TC}^1)$. Whether these three formalisms are equivalent on trees, depends on whether $\mu_a \text{FOE}$ and $\mu_a \text{FOE}_{\text{root}}^{\gg}$ coincide on trees. However, even if the answer is negative as we conjecture, it could still be the case that $\mu_a \text{FOE}/\leftrightarrow \equiv \mu_a \text{FOE}_{\text{root}}^{\gg}/\leftrightarrow$. In fact, we conjecture that this equivalence holds.

8.2.12. CONJECTURE. $\mu_a \text{FOE}/\leftrightarrow \equiv \mu_a \text{FOE}_{\text{root}}^{\gg}/\leftrightarrow$.

This equivalence, combined with the results of this dissertation, would immediately yield the result we are looking for. That is, the following chain of equivalences would be true: $\text{PDL} \equiv \mu_a \text{FOE}_{\text{root}}^{\gg}/\leftrightarrow \equiv \mu_a \text{FOE}/\leftrightarrow \equiv \text{FO}(\text{TC}^1)/\leftrightarrow$.

8.2.13. CONJECTURE. $\text{PDL} \equiv \text{FO}(\text{TC}^1)/\leftrightarrow$.

8.3 Conclusions and open problems

In this chapter we used the automata-theoretic characterization of several logics to provide bisimulation-invariance results. The crucial step was, in all cases, to give a construction $(-)^{\bullet}$ transforming automata based on some first-order

language *with* equality into automata based on first-order *without* equality. This construction was built to satisfy the relationship

$$\mathbb{A}^\bullet \text{ accepts } \mathbb{S} \quad \text{iff} \quad \mathbb{A} \text{ accepts } \mathbb{S}^\omega.$$

Following the philosophy of [Ven14], we induced this construction via a one-step translation $(-)_1^\bullet$ which directly transformed the one-step languages, while preserving the continuity and additivity properties.

8.3.1. REMARK. It is not difficult to prove that for each automaton \mathbb{A} it is possible to give a number $k \in \mathbb{N}$ such that

$$\mathbb{A}^\bullet \text{ accepts } \mathbb{S} \quad \text{iff} \quad \mathbb{A} \text{ accepts } \mathbb{S}^k,$$

taking $k := \max\{n \mid \text{diff}(x_1, \dots, x_n) \text{ occurs in } \mathbb{A}\} + 1$. That is, the ω -unraveling can be substituted for a k -unraveling, which has k -many copies of each node different from the root. This means that the results of this chapter transfer to the class of finitely branching trees and finite trees as well.

Open problems.

1. *Bisimulation-invariance on finite models:* It would be interesting to know if the bisimulation-invariance results of this chapter hold in the class of finite models. However, it is also not known whether the more fundamental equivalence $\mu\text{ML} \equiv \text{MSO}/\leftrightarrow$ holds on finite models or not.
2. *The confusion conjecture:* In [Boj04] Bojańczyk defines a notion of ‘confusion’ and conjectures that a regular language (i.e., MSO definable) of finite trees is definable in Chain Logic iff it contains no confusion. A remarkable property of the notion of confusion is that it is *decidable* whether a language has it or not. As the results of this chapter transfer to finite trees (and $\text{CL} \equiv \text{WCL}$ in that class) the conjecture implies that a language definable in the mu-calculus (on finite trees) is definable in PDL iff it contains no confusion. It is a major open problem whether we can decide if an arbitrary formula of μML is equivalent to some formula in PDL. Therefore, it would be important to check the confusion conjecture.
3. *Bisimulation-invariant fragment of $\text{FO}(\text{TC}^1)$:* As discussed in Section 8.2.3, the bisimulation-invariant fragment of $\text{FO}(\text{TC}^1)$ is thought to be PDL, but it has not been proved. The results in this dissertation should hopefully provide support to prove such a result. In particular, it is enough to prove that $\mu_a\text{FOE}/\leftrightarrow \equiv \mu_a\text{FOE}_{\text{root}}^\bullet/\leftrightarrow$ to obtain it, as explained in Section 8.2.3.

Bibliography

- [ABM99] Carlos Areces, Patrick Blackburn, and Maarten Marx. Hybrid logic is the bounded fragment of first order logic. In R. de Queiroz and W. Carnielli, editors, *Proceedings of 6th Workshop on Logic, Language, Information and Computation, WOLLIC99*, pages 33–50, Rio de Janeiro, Brazil, 1999. Cited on page 185.
- [AN92] André Arnold and Damian Niwinski. Fixed point characterization of weak monadic logic definable sets of trees. In *Tree Automata and Languages*, pages 159–188. 1992. Cited on pages 18, 20, and 87.
- [AN01] André Arnold and Damian Niwiński. *Rudiments of μ -calculus*, volume 146 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 2001. Cited on page 197.
- [ANB98] Hajnal Andréka, István Németi, and Johan van Benthem. Modal languages and bounded fragments of predicate logic. *Journal of Philosophical Logic*, 27(3):217–274, 1998. Cited on page 185.
- [BCM⁺92] Jerry R. Burch, Edmund M. Clarke, Kenneth L. McMillan, David L. Dill, and L. J. Hwang. Symbolic model checking: 10^{20} states and beyond. *Inf. Comput.*, 98(2):142–170, 1992. Cited on page 7.
- [Beh22] Heinrich Behmann. Beiträge zur Algebra der Logik, insbesondere zum Entscheidungsproblem. *Mathematische Annalen*, 1922. Cited on page 146.
- [Ben77] Johan van Benthem. *Modal Correspondence Theory*. PhD thesis, Universiteit van Amsterdam, 1977. Cited on page 19.
- [Ben96] Johan van Benthem. *Exploring Logical Dynamics*. CSLI Publications, Stanford, California, 1996. Cited on pages 10, 20, 51, and 52.
- [Ben98] Johan van Benthem. Program constructions that are safe for bisimulation. *Studia Logica*, 60(2):311–330, 1998. Cited on pages 10 and 20.

- [Ben06] Johan van Benthem. Modal frame correspondences and fixed-points. *Studia Logica*, 83(1-3):133–155, 2006. Cited on pages 10 and 67.
- [Ber03] Dietmar Berwanger. Game logic is strong enough for parity games. *Studia Logica*, 75(2):205–219, 2003. Cited on pages 7 and 76.
- [Ber05] Dietmar Berwanger. *Games and Logical Expressiveness*. Ph.D. Thesis, Department of Computer Science, RWTH Aachen, Germany, 2005. Cited on pages 6, 11, 31, 77, 81, and 90.
- [BFL15] Florian Bruse, Oliver Friedmann, and Martin Lange. On guarded transformation in the modal μ -calculus. *Logic Journal of IGPL*, 23(2):194–216, 2015. Cited on page 165.
- [BGL05] Dietmar Berwanger, Erich Grädel, and Giacomo Lenzi. The variable hierarchy of the μ -calculus is strict. In *STACS 2005, LNCS 3404*, pages 97–109. Springer, 2005. Cited on pages 11, 31, 61, 76, and 77.
- [BI08] Johan van Benthem and Daisuke Ikegami. Modal fixed-point logic and changing models. In Arnon Avron, Nachum Dershowitz, and Alexander Rabinovich, editors, *Pillars of Computer Science*, volume 4800 of *Lecture Notes in Computer Science*, pages 146–165. Springer, 2008. Cited on page 155.
- [Boj04] Mikołaj Bojańczyk. *Decidable Properties of Tree Languages*. Ph.D. Thesis, University of Warsaw, 2004. Cited on pages 9, 13, and 227.
- [Bra96] Julian C. Bradfield. The modal μ -calculus alternation hierarchy is strict. In Ugo Montanari and Vladimiro Sassone, editors, *CONCUR '96: Concurrency Theory*, volume 1119 of *Lecture Notes in Computer Science*, pages 233–246. Springer Berlin / Heidelberg, 1996. Cited on pages 7 and 76.
- [Bra98] Julian C. Bradfield. The modal μ -calculus alternation hierarchy is strict. *Theor. Comput. Sci.*, 195(2):133–153, March 1998. Cited on pages 7 and 76.
- [BRV01] Patrick Blackburn, Maarten De Rijke, and Yde Venema. *Modal logic*. Cambridge University Press, Cambridge England New York, 2001. Cited on pages 50 and 146.
- [Büc60] J. Richard Büchi. Weak second-order arithmetic and finite automata. *Mathematical Logic Quarterly*, 6(1-6):66–92, 1960. Cited on page 13.
- [Büc62] J. Richard Büchi. On a decision method in restricted second order arithmetic. In Saunders Mac Lane and Dirk Siefkes, editors, *The Collected Works of J. Richard Büchi*, pages 425–435. Springer New York, 1962. Cited on page 13.
- [Car15] Facundo Carreiro. PDL is the bisimulation-invariant fragment of Weak Chain Logic. In *30th Annual ACM/IEEE Symposium on Logic in Computer Science, LICS 2015, Kyoto, Japan, July 6-10, 2015*, pages 341–352. IEEE, 2015. Cited on page 21.

- [CF11] Balder ten Cate and Alessandro Facchini. Characterizing EF over infinite trees and modal logic on transitive graphs. In *MFCS*, pages 290–302, 2011. Cited on pages 9, 41, and 72.
- [CFVZ14a] Facundo Carreiro, Alessandro Facchini, Yde Venema, and Fabio Zanasi. Weak MSO: Automata and expressiveness modulo bisimilarity. *CoRR*, abs/1401.4374, 2014. Online: <http://arxiv.org/abs/1401.4374>. Cited on page 21.
- [CFVZ14b] Facundo Carreiro, Alessandro Facchini, Yde Venema, and Fabio Zanasi. Weak MSO: Automata and expressiveness modulo bisimilarity. In *Joint Meeting of CSL-LICS '14*, CSL-LICS '14, pages 27:1–27:27. ACM, 2014. Cited on page 21.
- [CH82] Ashok Chandra and David Harel. Structure and complexity of relational queries. *Journal of Computer and System Sciences*, 25(1):99–128, 1982. Cited on pages 8 and 38.
- [CV14] Facundo Carreiro and Yde Venema. PDL inside the μ -calculus: a syntactic and an automata-theoretic characterization. In Rajeev Goré, Barteld P. Kooi, and Agi Kurucz, editors, *Advances in Modal Logic 10, invited and contributed papers from the tenth conference on Advances in Modal Logic, held in Groningen, The Netherlands, August 5-8, 2014*, pages 74–93. College Publications, 2014. Cited on pages 10 and 21.
- [Dan84] Ryszard Danecki. Nondeterministic propositional dynamic logic with intersection is decidable. In Andrzej Skowron, editor, *Symposium on Computation Theory*, volume 208 of *Lecture Notes in Computer Science*, pages 34–53. Springer, 1984. Cited on page 34.
- [Daw98] Anuj Dawar. A restricted second order logic for finite structures. *Information and Computation*, 143(2):154–174, 1998. Cited on page 72.
- [DH00] Giovanna D’Agostino and Marco Hollenberg. Logical questions concerning the μ -calculus: Interpolation, Lyndon and Łoś-Tarski. *The Journal of Symbolic Logic*, 65(1):310–332, 2000. Cited on pages 17, 77, and 172.
- [DO09] Anuj Dawar and Martin Otto. Modal characterisation theorems over special classes of frames. *Ann. Pure Appl. Logic*, 161(1):1–42, 2009. Cited on page 19.
- [EF95] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite model theory*. Perspectives in Mathematical Logic. Springer, 1995. Cited on pages 103 and 129.
- [Egg63] Lawrence C. Eggan. Transition graphs and the star-height of regular events. *Michigan Math. J.*, 10(4):385–397, 12 1963. Cited on page 84.
- [EJ91] Ernest Allen Emerson and Charanjit S. Jutla. Tree automata, μ -calculus and determinacy. In *FOCS*, pages 368–377, 1991. Cited on pages 15, 26, and 76.

- [EJ99] E. Allen Emerson and Charanjit S. Jutla. The complexity of tree automata and logics of programs. *SIAM J. Comput.*, 29(1):132–158, September 1999. Cited on page 6.
- [EL86] Ernest Allen Emerson and Chin-Laung Lei. Efficient model checking in fragments of the propositional μ -calculus. In *LICS*, pages 267–278, 1986. Cited on pages 7 and 9.
- [Elg61] Calvin C. Elgot. Decision problems of finite automata design and related arithmetics. *Transactions of the American Mathematical Society*, 98(1):21–51, 1961. Cited on page 13.
- [ES95] E.A. Emerson and A. P. Sistla. On model-checking for fragments of μ -calculus. In *In CAV'93, volume 697 of LNCS*, pages 385–396. Springer-Verlag, 1995. Cited on page 6.
- [Fag75] Ronald Fagin. Monadic generalized spectra. *Mathematical Logic Quarterly*, 21(1):89–96, 1975. Cited on pages 7 and 39.
- [FL79] Michael J. Fischer and Richard E. Ladner. Propositional dynamic logic of regular programs. *J. Comput. Syst. Sci.*, 18(2):194–211, 1979. Cited on pages 5, 6, and 31.
- [FLV10] Gaëlle Fontaine, Raul Leal, and Yde Venema. Automata for coalgebras: An approach using predicate liftings. In Samson Abramsky, Cyril Gavoille, Claude Kirchner, Friedhelm Meyer auf der Heide, and Paul Spirakis, editors, *Automata, Languages and Programming*, volume 6199 of *Lecture Notes in Computer Science*, pages 381–392. Springer Berlin Heidelberg, 2010. Cited on page 83.
- [Fon08] Gaëlle Fontaine. Continuous fragment of the mu-calculus. In Michael Kaminski and Simone Martini, editors, *Computer Science Logic*, volume 5213 of *Lecture Notes in Computer Science*, pages 139–153. Springer Berlin Heidelberg, 2008. Cited on pages 60, 67, 68, and 83.
- [Fon10] Gaëlle Fontaine. *Modal fixpoint logic: some model-theoretic questions*. PhD thesis, ILLC, University of Amsterdam, 2010. Cited on pages 10, 52, 53, 60, 64, 67, 68, 77, and 83.
- [Fre06] Tim French. *Bisimulation Quantifiers for Modal Logics*. Phd thesis, School of Computer Science & Software Engineering, University of Western Australia, December 2006. Cited on page 172.
- [FV12] Gaëlle Fontaine and Yde Venema. Some model theory for the modal μ -calculus: syntactic characterizations of semantic properties. Submitted, 2012. Cited on pages 10 and 52.
- [FVZ13] Alessandro Facchini, Yde Venema, and Fabio Zanasi. A characterization theorem for the alternation-free fragment of the modal μ -calculus. In *LICS*, pages 478–487, 2013. Cited on pages 86 and 197.

- [Gai82] Haim Gaifman. On local and non-local properties. Logic colloquium '81, Proc. Herbrand Symp., Marseille 1981, Stud. Logic Found. Math. 107, 105–135, 1982. Cited on page 224.
- [GHK⁺80] G. Gierz, K. H. Hofmann, K. Keimel, J. D. Lawson, M. Mislove, and D. S. Scott. *A Compendium of Continuous Lattices*. Springer-Verlag, Berlin, 1980. Cited on page 64.
- [GKL⁺05] Erich Grädel, P. G. Kolaitis, L. Libkin, M. Marx, J. Spencer, Moshe Y. Vardi, Y. Venema, and Scott Weinstein. *Finite Model Theory and Its Applications (Texts in Theoretical Computer Science. An EATCS Series)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 2005. Cited on page 39.
- [Grä02] Erich Grädel. Model checking games. *Electronic Notes in Theoretical Computer Science*, 67:15–34, 2002. WoLLIC 2002, 9th Workshop on Logic, Language, Information and Computation. Cited on page 8.
- [Gro96] Martin Grohe. Arity hierarchies. *Annals of Pure and Applied Logic*, 82(2):103–163, 1996. Cited on pages 7 and 39.
- [GTW02] Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games: A Guide to Current Research*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002. Cited on page 100.
- [GW99] Erich Grädel and Igor Walukiewicz. Guarded fixed point logic. In *LICS*, pages 45–54, 1999. Cited on page 185.
- [GW05] Dov M. Gabbay and John Woods. *Logic and the Modalities in the Twentieth Century*, volume 7 of *Handbook of the History of Logic*. Elsevier, 2005. Cited on page 33.
- [Har83] David Harel. Recurring dominoes: Making the highly undecidable highly understandable. In Marek Karpinski, editor, *FCT*, volume 158 of *Lecture Notes in Computer Science*, pages 177–194. Springer, 1983. Cited on page 34.
- [HJJ⁺95] J.G. Henriksen, J. Jensen, M. Jørgensen, N. Klarlund, B. Paige, T. Rauhe, and A. Sandholm. Mona: Monadic second-order logic in practice. In *Tools and Algorithms for the Construction and Analysis of Systems, First International Workshop, TACAS '95, LNCS 1019*, 1995. Cited on page 13.
- [Hol98] Marco Hollenberg. *Logic and Bisimulation*. PhD thesis, University of Utrecht, 1998. Cited on pages 10, 20, and 52.
- [HTK00] David Harel, Jerzy Tiuryn, and Dexter Kozen. *Dynamic Logic*. MIT Press, Cambridge, MA, USA, 2000. Cited on pages 34 and 53.
- [Imm87] Neil Immerman. Languages that capture complexity classes. *SIAM Journal of Computing*, 16:760–778, 1987. Cited on pages 7, 8, and 39.

- [Jac13] Elisabeth Jacobi. Weak monadic second-order logic on infinitely branching trees. Master's thesis, Fachbereich Mathematik, Technischen Universität Darmstadt, Germany, 2013. Cited on pages 13, 15, and 16.
- [Jan06] David Janin. Contributions to formal methods: games, logic and automata, 2006. Habilitation thesis. Cited on pages 85, 90, 91, 170, 179, 214, 247, and 249.
- [JL04] David Janin and Giacomo Lenzi. On the relationship between monadic and weak monadic second order logic on arbitrary trees, with applications to the μ -calculus. *Fundam. Inform.*, 61(3-4):247–265, 2004. Cited on page 15.
- [JT51] Bjarni Jónsson and Alfred Tarski. Boolean algebras with operators. *Amer. J. Math.*, 73:891–939, 1951. Cited on pages 45, 47, and 49.
- [JW95] David Janin and Igor Walukiewicz. Automata for the modal μ -calculus and related results. In *MFC3*, pages 552–562, 1995. Cited on pages 17 and 93.
- [JW96] David Janin and Igor Walukiewicz. On the expressive completeness of the propositional μ -calculus with respect to monadic second order logic. In *Proceedings of the 7th International Conference on Concurrency Theory, CONCUR '96*, pages 263–277, London, UK, 1996. Springer-Verlag. Cited on pages 6, 17, 19, 20, 173, and 215.
- [Koz83] Dexter Kozen. Results on the propositional mu-calculus. *Theor. Comput. Sci.*, 27:333–354, 1983. Cited on page 6.
- [Kri65] Saul A. Kripke. Semantical analysis of modal logic II. Non-normal modal propositional calculi. In J. W. Addison, A. Tarski, and L. Henkin, editors, *The Theory of Models*, number 1, pages 135–135. North Holland, 1965. Cited on page 50.
- [KV95] Phokion G. Kolaitis and Jouko A. Väänänen. Generalized quantifiers and pebble games on finite structures. *Annals of Pure and Applied Logic*, 74(1):23–75, 1995. Cited on page 105.
- [KV98] Orna Kupferman and Moshe Y. Vardi. Freedom, weakness, and determinism: From linear-time to branching-time. In *Thirteenth Annual IEEE Symposium on Logic in Computer Science, Indianapolis, Indiana, USA, June 21-24, 1998*, pages 81–92. IEEE Computer Society, 1998. Cited on pages 18 and 87.
- [KV01] Orna Kupferman and Moshe Y. Vardi. Weak alternating automata are not that weak. *ACM Trans. Comput. Logic*, 2(3):408–429, July 2001. Cited on page 86.
- [KV03] Orna Kupferman and Moshe Y. Vardi. $\Pi_2 \cap \Sigma_2 \equiv \text{AFMC}$. In *ICALP*, pages 697–713, 2003. Cited on page 87.

- [KV05] Orna Kupferman and Moshe Y. Vardi. From linear time to branching time. *ACM Trans. Comput. Log.*, 6(2):273–294, 2005. Cited on page 87.
- [KV09] Christian Kissig and Yde Venema. Complementation of coalgebra automata. In Alexander Kurz, Marina Lenisa, and Andrzej Tarlecki, editors, *Algebra and Coalgebra in Computer Science, Third International Conference, CALCO 2009, Udine, Italy, September 7-10, 2009. Proceedings*, volume 5728 of *Lecture Notes in Computer Science*, pages 81–96. Springer, 2009. Cited on pages 29 and 149.
- [KVV00] Orna Kupferman, Moshe Y. Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking. *J. ACM*, 47(2):312–360, March 2000. Cited on pages 18, 86, 165, and 166.
- [Lan06] Martin Lange. Model checking propositional dynamic logic with all extras. *Journal of Applied Logic*, 4(1):39–49, 2006. Cited on page 6.
- [Lib04] Leonid Libkin. *Elements of Finite Model Theory*. Springer, 2004. Cited on page 38.
- [LN99] Leonid Libkin and Juha Nurmonen. Counting and locality over finite structures a survey. In Jouko Väänänen, editor, *Generalized Quantifiers and Computation*, volume 1754 of *Lecture Notes in Computer Science*, pages 18–50. Springer Berlin Heidelberg, 1999. Cited on page 225.
- [Löw15] Leopold Löwenheim. Über Möglichkeiten im Relativkalkül. *Mathematische Annalen*, 76(4):447–470, 1915. Cited on page 146.
- [Mat02] Radu Mateescu. Local model-checking of modal mu-calculus on acyclic labeled transition systems. In Joost-Pieter Katoen and Perdita Stevens, editors, *Tools and Algorithms for the Construction and Analysis of Systems*, volume 2280 of *Lecture Notes in Computer Science*, pages 281–295. Springer Berlin Heidelberg, 2002. Cited on page 166.
- [Mos57] Andrzej Mostowski. On a generalization of quantifiers. *Fundamenta Mathematicae*, 44(1):12–36, 1957. Cited on pages 37 and 146.
- [Mos74] Yiannis N. Moschovakis. *Elementary induction on abstract structures*. North-Holland, 1974. Cited on pages 8 and 38.
- [Mos85] A.W. Mostowski. Regular expressions for infinite trees and a standard form of automata. In Andrzej Skowron, editor, *Computation Theory*, volume 208 of *Lecture Notes in Computer Science*, pages 157–168. Springer Berlin Heidelberg, 1985. Cited on page 15.
- [Mos91] Andrzej Mostowski. Games with forbidden positions. Technical Report 78, University of Gdansk, 1991. Cited on page 26.
- [Mos08] Yiannis N. Moschovakis. *Elementary induction on abstract structures*. Dover Publications, 2008. Cited on pages 8 and 38.

- [MS85] David Muller and Paul E. Schupp. Alternating automata on infinite objects, determinacy and rabin's theorem. In M. Nivat and D. Perrin, editors, *Automata on Infinite Words*, volume 192 of *Lecture Notes in Computer Science*, pages 99–107. Springer Berlin Heidelberg, 1985. Cited on page 15.
- [MS87] David E. Muller and Paul E. Schupp. Alternating automata on infinite trees. *Theoretical Computer Science*, 54(2-3):267–276, 1987. Cited on page 29.
- [MS95] David E. Muller and Paul E. Schupp. Simulating alternating tree automata by nondeterministic automata. *Theor. Comput. Sci.*, 141(1-2):69–107, 1995. Cited on page 197.
- [MSS88] David E. Muller, Ahmed Saoudi, and Paul E. Schupp. Weak alternating automata give a simple explanation of why most temporal and dynamic logics are decidable in exponential time. In *Proceedings of the Third Annual IEEE Symposium on Logic in Computer Science (LICS 1988)*, pages 422–427. IEEE Computer Society Press, July 1988. Cited on page 18.
- [MSS92] David E. Muller, Ahmed Saoudi, and Paul E. Schupp. Alternating automata, the weak monadic theory of trees and its complexity. *Theor. Comput. Sci.*, 97(2):233–244, 1992. Cited on pages 14, 18, 86, and 197.
- [Mul63] David E. Muller. Infinite sequences and finite machines. In *4th Annual Symposium on Switching Circuit Theory and Logical Design, Chicago, Illinois, USA, October 28-30, 1963*, pages 3–16. IEEE Computer Society, 1963. Cited on page 13.
- [Niw86] Damian Niwiński. On fixed-point clones (extended abstract). In *ICALP*, pages 464–473, 1986. Cited on pages 7, 15, 17, and 31.
- [NSW02] Jakub Neumann, Andrzej Szepietowski, and Igor Walukiewicz. Complexity of weak acceptance conditions in tree automata. *Inf. Process. Lett.*, 84(4):181–187, 2002. Cited on page 86.
- [Ott99] Martin Otto. Eliminating recursion in the μ -calculus. In Christoph Meinel and Sophie Tison, editors, *STACS 99*, volume 1563 of *Lecture Notes in Computer Science*, pages 531–540. Springer Berlin Heidelberg, 1999. Cited on page 84.
- [Par76] David Park. Finiteness is mu-ineffable. *Theoretical Computer Science*, 3(2):173–181, 1976. Cited on pages 44, 68, and 72.
- [Par80] David Park. On the semantics of fair parallelism. In *Proceedings of the Abstract Software Specifications, 1979 Copenhagen Winter School*, pages 504–526, London, UK, UK, 1980. Springer-Verlag. Cited on pages 7 and 31.
- [Par85] Rohit Parikh. The logic of games and its applications. In *Selected Papers of the International Conference on "Foundations of Computation Theory"*

- on *Topics in the Theory of Computation*, pages 111–139, New York, NY, USA, 1985. Elsevier North-Holland, Inc. Cited on pages 6 and 34.
- [Pau01] Marc Pauly. *Logic for Social Software*. PhD thesis, Institute for Logic Language and Computation, University of Amsterdam, The Netherlands, 2001. Cited on pages 34 and 35.
- [Pel85] David Peleg. Concurrent dynamic logic. In *Proceedings of the Seventeenth Annual ACM Symposium on Theory of Computing, STOC '85*, pages 232–239, New York, NY, USA, 1985. ACM. Cited on pages 6, 11, 33, 34, 68, and 84.
- [Pra80] Vaughan R. Pratt. A near-optimal method for reasoning about action. *Journal of Computer and System Sciences*, 20(2):231–254, 1980. Cited on page 6.
- [PSZng] Alessandra Palmigiano, Sumit Sourabh, and Zhiguang Zhao. Sahlqvist Theory for Impossible Worlds. *Journal of Logic and Computation*, forthcoming. Cited on page 50.
- [Rab69] Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the American Mathematical Society*, 141:1–35, jul 1969. Cited on pages 13 and 14.
- [Rab70] Michael O. Rabin. Weakly definable relations and special automata. In Yehoshua Bar-Hillel, editor, *Proceedings of the Symposium on Mathematical Logic and Foundations of Set Theory (SMLFST'70)*, pages 1–23. North-Holland, 1970. Cited on pages 14, 18, and 86.
- [Ros97] Eric Rosen. Modal logic over finite structures. *Journal of Logic, Language and Information*, 6:427–439, 1997. Cited on page 19.
- [Saf88] Shmuel Safra. On the complexity of ω -automata. In *Proceedings of the 29th Annual Symposium on Foundations of Computer Science, SFCS '88*, pages 319–327, Washington, DC, USA, 1988. IEEE Computer Society. Cited on page 197.
- [Sch06] Nicole Schweikardt. On the expressive power of monadic least fixed point logic. *Theoretical Computer Science*, 350(2-3):325–344, 2006. Automata, Languages and Programming: Logic and Semantics (ICALP-B 2004) Automata, Languages and Programming: Logic and Semantics 2004. Cited on page 72.
- [Sto74] Larry J. Stockmeyer. *The complexity of decision problems in automata theory and logic*. PhD thesis, 1974. Cited on page 8.
- [Str81] Robert S. Streett. Propositional dynamic logic of looping and converse. In *STOC*, pages 375–383. ACM, 1981. Cited on page 18.

- [Str82] Robert S. Streett. Propositional dynamic logic of looping and converse is elementarily decidable. *Information and Control*, 54(1-2):121–141, 1982. Cited on page 18.
- [SV10] Luigi Santocanale and Yde Venema. Completeness for flat modal fixpoint logics. *Ann. Pure Appl. Logic*, 162(1):55–82, 2010. Cited on pages 10, 68, and 84.
- [Tho75] Steven K. Thomason. Categories of frames for modal logic. *J. Symb. Log.*, 40(3):439–442, 1975. Cited on page 50.
- [Tho84] Wolfgang Thomas. Logical aspects in the study of tree languages. In *Proc. Of the Conference on Ninth Colloquium on Trees in Algebra and Programming*, pages 31–49, New York, NY, USA, 1984. Cambridge University Press. Cited on pages 9 and 223.
- [Tho96] Wolfgang Thomas. Languages, automata, and logic. In *Handbook of Formal Languages*, pages 389–455. Springer, 1996. Cited on pages 9, 13, and 42.
- [TK09] Hans-Jörg Tiede and Stephan Kepser. Monadic second-order logic and transitive closure logics over trees. *Research on Language and Computation*, 7(1):41–54, 2009. Cited on pages 8 and 39.
- [Tra61] Boris A. Trakthenbrot. Finite automata and the logic of monadic second order predicates. *Doklady Akademii Nauk SSSR*, 140:326–329, 1961. In Russian. Cited on page 13.
- [Vää77] Jouko Väänänen. Remarks on generalized quantifiers and second-order logics. In *Set theory and hierarchy theory*, pages 117–123. Prace Naukowe Instytutu Matematyki Politechniki Wrocławskiej, Wrocław, 1977. Cited on pages 12, 16, and 72.
- [Var82] Moshe Y. Vardi. The complexity of relational query languages. In *Proceedings of the Fourteenth Annual ACM Symposium on Theory of Computing, STOC '82*, pages 137–146, New York, NY, USA, 1982. ACM. Cited on page 8.
- [Ven08] Yde Venema. PDL as a fragment of the modal μ -calculus. Unpublished, 2008. Cited on page 10.
- [Ven11] Yde Venema. *Lecture Notes on the Modal μ -calculus*. 2011. Cited on pages 13, 41, 93, 149, 162, 165, 168, and 184.
- [Ven14] Yde Venema. Expressiveness modulo bisimilarity: A coalgebraic perspective. In Alexandru Baltag and Sonja Smets, editors, *Johan van Benthem on Logic and Information Dynamics*, volume 5 of *Outstanding Contributions to Logic*, pages 33–65. Springer International Publishing, 2014. Cited on pages 20, 100, 215, 218, and 227.

- [VW86] Moshe Y. Vardi and Pierre Wolper. Automata-theoretic techniques for modal logics of programs. *Journal of Computer and System Sciences*, 32(2):183–221, 1986. Cited on page 18.
- [Wal96] Igor Walukiewicz. Monadic second order logic on tree-like structures. In *STACS*, pages 401–413, 1996. Cited on pages 9, 14, 41, 93, 173, 174, and 177.
- [Wal02] Igor Walukiewicz. Monadic second-order logic on tree-like structures. *Theor. Comput. Sci.*, 275(1-2):311–346, 2002. Cited on pages 173, 174, and 177.
- [Wil01] Thomas Wilke. Alternating tree automata, parity games, and modal μ -calculus. *Bull. Soc. Math. Belg*, 8, 2001. Cited on page 87.
- [Zan12] Fabio Zanasi. Expressiveness of monadic second order logics on infinite trees of arbitrary branching degree. Master’s thesis, ILLC, Universiteit van Amsterdam, the Netherlands, 2012. Cited on pages 15, 86, 178, 188, 194, 197, and 199.

Index

	Symbols		
2MSO	40	
2WCL	42	
2WMSO	41	
F^α	31	
F_\blacklozenge	50	
$F_{ Y}$	47	
Qp	23	
$R[x], R^*, R^+$	23	
R_\blacklozenge	50	
Z_V	28	
$[\text{LFP}_{p.x}.\varphi(p, x)]$	38	
$[\text{TC}_{\bar{x}, \bar{y}}.\varphi(\bar{x}, \bar{y})](\bar{\mathbf{u}}, \bar{\mathbf{v}})$	39	
$\text{Aut}(\mathcal{L})$	28	
$\text{Aut}_w(\mathcal{L})$	86	
$\text{Aut}_{wa}(\mathcal{L})$	88	
$\text{Aut}_{wc}(\mathcal{L})$	87	
$\text{Aut}_{wa}^-(\mathcal{L})$	150	
FV	30, 37, 38	
GL	34	
AFMC	31	
$\mathcal{A}(\mathbb{A}, \mathbb{S})$	28, 93	
$\underline{\mathcal{A}}(\underline{\mathbb{A}}, \underline{\mathbb{S}})$	151	
\mathbb{A}	28	
\mathbb{A}^γ	177	
\mathbb{A}^u	90	
\times	159	
FO(TC ¹)	39	
\Leftrightarrow	36	
\oplus	32	
CPDL	33	
$\nabla_\infty(\Sigma)$	103	
$\nabla_{\text{FOE}^\infty}(\bar{\mathbf{T}}, \Pi, \Sigma)$	103	
$\nabla_{\text{FOE}}(\bar{\mathbf{T}}, \Pi)$	101	
$\nabla_{\text{FOE}}(\bar{\mathbf{T}}, \Pi)_s$	131	
$\nabla_{\text{FOE}}(\bar{\mathbf{T}}, \Pi)_s$	127	
$\nabla_{\text{FO}}(\Sigma)$	100	
$\nabla_{\text{FO}}(\Sigma)_s$	126, 132	
$\nabla_{\text{ML}}(\Sigma)_\ell$	140	
δ_p	30	
\forall^∞	37	
\mathbb{A}^δ	29	
π^δ	34	
φ^δ	28, 78	
$\exists p.L$	176	
$\exists p.\mathbb{A}$	176	
$\exists_{\text{fin}} p.L$	198	
$\exists_{\text{fin}} p.\mathbb{A}$	198	
$\exists_{\text{fin}}, \forall_{\text{fin}}$	41	
$\exists_{\text{fch}} p.L$	207	
$\exists_{\text{fch}} p.\mathbb{A}$	207	
$\exists_{\text{fch}}, \forall_{\text{fch}}$	42	
$[[\varphi]]^S$	30	
F_p^ψ	30	
$F_{Q:x}^\varphi$	38	
FO	36	
FOE	36	

FOE^∞	37	$\mu_c\text{FOE}^\infty$	75
\approx	36	μFOE^\gg	179
$\text{FO}(\text{TC})$	39	$\mu\text{FOE}^{\infty\gg}$	201
iVar	36	μFOE	38
A_p^\ddagger	204	ADD_Q	61
A_p^\ddagger	191	μFOE^∞	
\mathcal{G}	26	CON_Q	73
\Downarrow	40	\otimes	33
\sqsubseteq	40	$\mu_{na}\text{ML}$	53
\rightsquigarrow	28	\mathbb{M}	37
\mathcal{L}	27	FO_1	99
\mathcal{L}^+	27	$\text{ADD}_{A'}$	121
μML^\vee	77	$\overline{\text{CON}}_{A'}$	124
$\mu_c\text{ML}^\vee$	71	$\text{CON}_{A'}$	114
$\mu_{nc}\text{ML}^\vee$	68	$\text{MON}_{A'}$	108
\Vdash	28, 30, 32, 33	$\text{MUL}_{A'}$	124
$\mathbb{S}[p \mapsto X_p], \mathbb{S}[p \upharpoonright X_p]$	24	FO_1 (multi-sorted)	
\models	28, 37–40, 125	$\text{ADD}_{A'}$	136
$\nabla_\infty^{A'}(\Sigma)$	112	$\text{MON}_{A'}$	132
$\nabla_{\text{FOE}^\infty}^{A'}(\overline{\mathbf{T}}, \Pi, \Sigma)$	112	$\text{MUL}_{A'}$	139
$\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)$	110	FOE_1	99
$\nabla_{\text{FOE}}^{A'}(\overline{\mathbf{T}}, \Pi)_s$	133	$\text{ADD}_{A'}$	119
$\nabla_{\text{FO}}^{A'}(\Sigma)$	109	$\text{MON}_{A'}$	110
$\nabla_{\text{ML}}^{A'}(\Sigma)_\ell$	142	FOE_1 (multi-sorted)	125
$\nabla_{\text{FO}}^{A'}(\Sigma, \Pi)$	109	$\text{ADD}_{A'}$	134
$\nabla_{\text{FO}}^{A'}(\Sigma, \Pi)_s$	132	$\text{MON}_{A'}$	133
$\nabla_{\text{ML}}^{A'}(\Sigma, \Pi)_\ell$	142	$\text{MUL}_{A'}$	139
MSO	40	FOE_1^∞	99
μML	29	$\overline{\text{CON}}_{A'}$	124
ADD_Q	58	$\text{CON}_{A'}$	116
CON_Q	67	$\text{MON}_{A'}$	111
MON_Q^\vee	77	$\text{MUL}_{A'}$	124
nCON_Q^\vee	68, 71	ML_1	139
nADD_Q	52	$\text{ADD}_{A'}$	143
nADD_Q^-	54	$\overline{\text{CON}}_{A'}$	145
$\mu\text{ML}[k]$	31	$\text{CON}_{A'}$	142
$\mu_c\text{ML}$	68	$\text{MON}_{A'}$	141
$\mu_c\text{ML}^b$	68	$\text{MUL}_{A'}$	145
$\mu_a\text{ML}$	58	$\text{ADD}_{A'}^-$	150
$\mu_a\text{FOE}^\gg$	209	ML_1^b	149
$\mu_a\text{FOE}$	63	\preceq, \prec	28
		D	27

PDL	32	$\bar{\mathbf{T}}, \bar{\mathbf{x}}$	23
PDL^{tf}	32	WCL	42
R_{π}^{\rightarrow}	34, 35	Win	26
Ω	26	Win_{Ω}	26
$\nabla_{\text{FOE}^{\infty}}^{+}(\bar{\mathbf{T}}, \Pi, \Sigma)$	112	WMSO	41
$\nabla_{\text{FOE}}^{+}(\bar{\mathbf{T}}, \Pi)$	110	$\mathbf{W}x.(\varphi, \psi)$	73
$\nabla_{\text{FOE}}^{+}(\bar{\mathbf{T}}, \Pi)_{\mathbf{s}}$	133	$f : \bar{\mathbf{d}} \mapsto \bar{\mathbf{d}}'$	100, 126
$\nabla_{\text{FO}}^{+}(\Sigma)_{\mathbf{s}}$	132	A	
$\nabla_{\text{ML}}^{+}(\Sigma)_{\ell}$	142	additivity	
$\nabla_{\text{FO}}^{+}(\Sigma, \Pi)_{\mathbf{s}}$	132	complete	44
$\nabla_{\text{ML}}^{+}(\Sigma, \Pi)_{\ell}$	142	complete, in the	
$\pi; \pi$	32	j^{th} -coordinate	45, 46
π^{*}	32	complete, in the product	45, 46
π°	35	approximant	31
π^{\flat}	159	atom	46
\exists^{∞}	37	quasi-	46
$R_{\pi}^{\mathcal{S}}$	32	automaton	
$\mu_{na}^{-}\text{ML}$	54	B -incomplete	159
$\mathcal{S}(\mathbf{E}, \mathcal{S})$	154	completion	159
A^{\wp}	188	MSO-	173
σ_p	30	PDL-	158
\heartsuit	30	PDL^{tf} -	150
\trianglelefteq	30	WCL-	202
$\mathfrak{M}_1^{\mathcal{S}}$	27	WMSO-	187
$\tau_S^{A'}$	108	non-deterministic	177
τ_S	98	parity	28
τ_S^{+}	98	achromatic	94
Δ	28	additive-weak	88
$\mathbb{T}, \mathbb{T}.s$	24	chromatic	93
$\varpi_c, \varpi_c^Q, \varpi_c^{+}$	140	continuous-weak	87
$\mathcal{T}(\mathbb{A})$	175	modal	93
κ	23	weak	86
κ^{\sharp}	23	unraveling	90
$\kappa^{\sharp}(\bar{\mathbf{P}})$	24	B	
\mathfrak{M}_1	27	bisimulation	36
$\text{FO}(\text{LFP}^1)$	38	board	26
$\hat{\mathcal{S}}, \mathcal{S}^{\omega}$	24	C	
V	28	calculus, μ -	29
V^{\sharp}	28	chain	24
V^c	28	finite	25
$\varphi^?$	32	generalized	25
φ^{δ}	144		

- closure
 - reflexive-transitive 39
- coloring 23, 28
- component
 - maximal strongly connected 28
 - strongly connected 28
- condition
 - additivity 88
 - continuity 87
 - weakness 86
 - winning 26
- continuity 64
 - in the j^{th} -coordinate 64
 - in the product 64
- D**
- definition, binding 30
- depth
 - alternation 30
- dual
 - FOE_1^∞ 124
 - ML_1 144
 - Boolean 28, 78
- E**
- equation
 - σ -, μ -, ν - 154
 - set of B -incomplete σ - . . . 154
- F**
- fixpoint
 - least 30
- form, basic
 - FO_1 100
 - FOE_1 101
 - FOE_1 (multi-sorted) 131
 - FOE_1^∞ 103
 - ML_1 140
 - strict
 - FO_1 (multi-sorted) 126
 - FOE_1 (multi-sorted) 127
- formula
 - bisimulation-invariant 36
 - clean 30
 - completely additive 51, 61
 - continuous 67, 73
 - guarded 165
 - one-step 27
 - co-continuous 87
 - completely additive 88
 - completely multiplicative 88
 - continuous 87
 - monotone 108
 - restrict to descendants 180
- fragment
 - alternation-free 31
 - co-continuous
 - FO_1 124
 - FOE_1^∞ 124
 - ML_1 145
 - completely additive 52, 58, 61
 - FO_1 121
 - FO_1 (multi-sorted) 136
 - FOE_1 119
 - FOE_1 (multi-sorted) 134
 - ML_1 143
 - normal and 52
 - completely multiplicative
 - FO_1 124
 - FO_1 (multi-sorted) 139
 - FOE_1 (multi-sorted) 139
 - FOE_1^∞ 124
 - ML_1 145
 - continuous 67, 73
 - FO_1 114
 - FOE_1^∞ 116
 - ML_1 142
 - forward-looking 179, 201
 - monotone 77
 - FO_1 108
 - FO_1 (multi-sorted) 132
 - FOE_1 110
 - FOE_1 (multi-sorted) 133
 - FOE_1^∞ 111
 - ML_1 141
 - positive 27
- function, parity 26

- G**
- game
- acceptance 28
 - modal 93
 - NFA 151
 - board 26
 - determined 26
 - initialized board 26
 - parity 26
 - solution 154
- graph
- induced 28
- H**
- hierarchy
- arity 39
 - variable 31
- I**
- isomorphism, partial 100, 126
- L**
- language
- one-step 27
 - multi-sorted 27
- lifting
- additive 204
 - continuous 191
- logic
- chain 42
 - first-order 36
 - with generalized quantifiers 37
 - with reflexive-transitive
 - closure 39
 - with unary fixpoints 38
 - Game 34
 - monadic second-order 40
 - Propositional Dynamic 32
 - Concurrent 33
 - test-free 32
 - weak chain 42
 - weak monadic second-order 41
- M**
- map
- completely additive 44–46
 - in the j^{th} -coordinate . 45, 46
 - in the product 45, 46
 - normal and 45
 - constructive 47
 - continuous 64
 - in the j^{th} -coordinate 64
 - in the product 64
 - normal 45
 - parity 26, 28
 - restrict to descendants . . . 180
 - transition 28
- marking 23, 28
- match 26
- f -guided 26
 - partial 26
 - total 26
- model 37
- bisimilar 36
 - one-step 27
 - strict 27
- move, admissible 26
- N**
- name 27
- normal form
- dual 35
- normality 45
- P**
- parameter 38, 39
- path 24
- projection
- finite chain 207
 - finite set 198
 - of \mathbb{A} over p 176
- Q**
- quantifier, generalized 37
- quasi-atom 46
- R**
- rank, quantifier 99

- restriction
 - additivity 88
 - continuity 87
 - weakness 86
- restriction, to descendants . . . 180
- root 24
- S**
- set of B -incomplete σ -equations 154
- strategy 26
 - finitely branching 189
 - functional 177, 189
 - non-branching 203
 - positional 26
 - surviving 26
 - well-founded 189
 - winning 26
- T**
- transition system
 - bisimilar 36
 - labeled 23
- tree 24
 - class of 24
 - language 24
 - strict 24
 - unraveling 24
 - with back-edges 90
- type 98
 - A - 98
 - A' -positive 108
 - positive 98
- U**
- unraveling 24
 - ω - 24
 - finite 90
 - partial 90
 - tree 24
- V**
- valuation 23, 28
- variable
 - bound 30
 - guarded 165
 - weakly 165
 - individual 37
 - separation of 54

Samenvatting

Dit proefschrift, met de titel *Fragmenten van Dekpuntlogica's: Automaten en Expressiviteit*, bestudeert de relatieve uitdrukingskracht en de eigenschappen van een aantal dekpunt- en tweede orde-logica's. De term dekpuntlogica wordt in dit manuscript in de brede zin gebruikt; er wordt mee gerefereerd aan iedere logica waarmee recursie, iteratie of repetitie kan worden uitgedrukt. Ons belangrijkste doel is om op systematische wijze verschillende belangrijke logica's als precieze fragmenten van andere bekende logica's te identificeren. Om deze taak te kunnen bewerkstelligen hebben we automaten-theoretische middelen ontwikkeld om deze fragmenten te kunnen analyseren. De resultaten van dit proefschrift geven nieuw inzicht in de relatie tussen dekpunt- en tweede-orde-logica en leveren verder bewijs voor de succesvolle samenhang van logica en automaten.

In Hoofdstuk 3 definiëren en analyseren we fragmenten van zowel modale als eerste-orde-dekpuntlogica's. Om deze fragmenten te definiëren gebruiken we hoofdzakelijk de methode van restrictie van de toepassing van de dekpuntoperator $\mu p.\varphi$ (en zijn eerste-orde equivalent) op formules φ met een specifieke eigenschap. De voornaamste eigenschappen die we beschouwen zijn *volledige additiviteit* en *continuïteit*, maar ook andere syntactische restricties en hun effecten worden bestudeerd. Aan de modale kant geven we precieze en semantische karakteriseringen van PDL, Concurrent PDL en GL binnen de μ -calculus. Voor de kant van de eerste-orde geven we een analoge karakterisering van FO(TC^1) binnen FO(LFP¹).

In Hoofdstuk 4 introduceren we verschillende subklassen van pariteitsautomaten en bespreken we de intuïties en motivaties voor deze definities. De subklassen zijn geïnspireerd op de fragmenten van Hoofdstuk 3 en proberen een parallel te vinden –aan de kant van de automaat– voor de beperkingen van additiviteit en continuïteit van de syntactische fragmenten. In het laatste deel van dit hoofdstuk introduceren we een algemene techniek (te danken aan Janin [Jan06]) om pariteitsautomaten in een boomvorm te verkrijgen. Deze structuur heeft het voordeel dat zij een “bijna (dekpunt) formule” is en daardoor makkelijker te ver-

talen is naar een passende dekpuntlogica. Als afsluiting introduceren we andere mogelijke equivalente definities van pariteitsautomaten (d.w.z. modale, eerste-orde, chromatische en achromatische) en bespreken we de voor- en nadelen van elk van deze perspectieven.

Een van de voordelen van de automaten-aanpak voor dekpuntlogica's is dat hun complexiteit onderverdeeld kan worden in twee simpelere en duidelijk gedefinieerde delen: een graafstructuur die de repetities (d.w.z. de toestanden van de automaat) representeert en een afbeelding van de transitie met een simpele één-stap-logica. In Hoofdstuk 5 zal de nadruk liggen op dit laatste deel. We introduceren de één-stap-logica's die in dit proefschrift gebruikt zullen worden en vervolgen met uitdieping van deze logica's. Ons streven is om normale vormen te kunnen geven en om een aantal fragmenten van deze logica's (continu, volledig additief, etc.) te kunnen karakteriseren. De resultaten van deze analyse zullen cruciaal zijn voor de volgende hoofdstukken, waarin we gebaseerd op deze talen de eigenschappen van automaten bewijzen.

In Hoofdstuk 6 geven we automaten-karakterisering voor een aantal modale logica's. De resultaten van dit hoofdstuk worden verkregen door het gebruik van pariteitsautomaten gebaseerd op modale één-stap-logica's. We laten zien dat (1) test-free PDL en volledig PDL corresponderen met concrete klassen van additief-zwakke pariteitsautomaten; en (2) de continue restrictie van μ -calculus correspondeert met een concrete klasse van continu-zwakke pariteitsautomaten. Deze resultaten worden verkregen door effectieve transformaties van formules naar automaten en vice-versa.

In Hoofdstuk 7 geven we een automaten-karakterisering voor WMSO (zwakke monadische tweede-orde-logica) en WCL (zwakke kettinglogica), voor de klasse van boommodellen. In dit geval gebruiken we pariteitsautomaten gebaseerd op (uitbreidingen van) eerste-orde-logica (met identiteit). De belangrijkste uitdaging van dit hoofdstuk is om simulatie- en projectiestellingen te geven voor de klassen van continu-zwakke en additief-zwakke pariteitsautomaten. Een bijproduct hiervan is dat we karakterisering voor de genoemde automaten (en tweede-orde-logica's) als fragmenten van dekpuntlogica's verkrijgen.

In Hoofdstuk 8 gebruiken we de middelen die in de voorgaande hoofdstukken ontwikkeld zijn om nieuwe bisimulatie-invariantie resultaten te bewijzen. We bewijzen namelijk dat het bisimulatie-invariante fragment van WCL PDL is, en dat het bisimulatie-invariante fragment van WMSO equivalent is aan de continue restrictie van de μ -calculus.

Abstract

This dissertation, entitled *Fragments of Fixpoint Logics: Automata and Expressiveness*, studies the relative expressive power and properties of several fixpoint and second-order logics. We use the term fixpoint logic in a broad sense, referring to any logic which can encode some type of recursion, iteration or repetition. Our main objective is to systematically identify several important logics as precise fragments of other well-known logics. In order to accomplish this task, we develop automata-theoretic tools to analyze these fragments. The results of this dissertation provide new insight on the relationship of fixpoint and second-order logic and provides further evidence of the successful logic-automata connection.

In Chapter 3 we define and analyze fragments of both modal and first-order fixpoint logics. The main method that we use to define these fragments is the restriction of the application of the fixpoint operator $\mu p.\varphi$ (and the first-order equivalent) to formulas φ having a special property. The main properties that we consider are *complete additivity* and *continuity*, but we also consider other syntactic restrictions and their effects. On the modal side, we give precise syntactic and semantic characterizations of PDL, Concurrent PDL and GL inside the μ -calculus. On the first-order side, we give an analogous characterization of FO(TC^1) inside FO(LFP^1).

In Chapter 4 we introduce several subclasses of parity automata and discuss the intuitions and motivations behind these definitions. The subclasses are inspired by the fragments of Chapter 3 and try to parallel, on the automata side, the additivity and continuity constraints of the syntactic fragments. In the last part of this chapter we introduce a general technique (due to Janin [Jan06]) to bring parity automata into a tree-like shape. This structure has the advantage of being ‘almost a (fixpoint) formula’ and therefore it is easy to translate it to an appropriate fixpoint language. To finish, we introduce other possible equivalent definitions of parity automata (i.e., modal, first-order, chromatic and achromatic) and discuss the (dis)advantages of each perspective.

One of the advantages of taking an automata approach to fixpoint logics is

that their complexity can be divided in two simpler and clearly defined parts: a graph structure representing the repetitions (i.e., the states of the automata) and a transition map with a simple one-step logic. In Chapter 5 we focus on the latter part. We introduce the one-step logics that we use in this dissertation and carry on an in-depth study of them. Our objective is to provide normal forms and characterize several fragments of this logics (continuous, completely additive, etc.) The results of this analysis will be crucial in later chapters, when we prove properties of automata based on these languages.

In Chapter 6 we give automata characterizations for a number of modal logics. The results of this chapter are obtained using parity automata based on modal one-step languages. We show that (1) test-free PDL and full PDL correspond to concrete classes of additive-weak parity automata; and (2) the continuous restriction of the μ -calculus corresponds to a concrete class of continuous-weak parity automata. These results are obtained via effective transformations from formulas to automata and vice-versa.

In Chapter 7 we give automata characterizations for WMSO (weak monadic second-order logic) and WCL (weak chain logic), on the class of tree models. In this case, we use parity automata based on (extensions of) first-order logic (with equality). The main challenge of this chapter is to give simulation and projection theorems for the classes of continuous-weak and additive-weak parity automata. As a byproduct, we also obtain characterizations for the mentioned automata (and second-order logics) as fragments fixpoint logics.

In Chapter 8 we use the tools developed in the previous chapters to prove novel bisimulation-invariance results. Namely, we prove that the bisimulation invariant fragment of WCL is PDL, and that the bisimulation-invariant fragment of WMSO is equivalent to the continuous restriction of the μ -calculus.

Titles in the ILLC Dissertation Series:

ILLC DS-2010-01: **Reut Tsarfaty**

Relational-Realizational Parsing

ILLC DS-2010-02: **Jonathan Zvesper**

Playing with Information

ILLC DS-2010-03: **Cédric Dégrement**

The Temporal Mind. Observations on the logic of belief change in interactive systems

ILLC DS-2010-04: **Daisuke Ikegami**

Games in Set Theory and Logic

ILLC DS-2010-05: **Jarmo Kontinen**

Coherence and Complexity in Fragments of Dependence Logic

ILLC DS-2010-06: **Yanjing Wang**

Epistemic Modelling and Protocol Dynamics

ILLC DS-2010-07: **Marc Staudacher**

Use theories of meaning between conventions and social norms

ILLC DS-2010-08: **Amélie Gheerbrant**

Fixed-Point Logics on Trees

ILLC DS-2010-09: **Gaëlle Fontaine**

Modal Fixpoint Logic: Some Model Theoretic Questions

ILLC DS-2010-10: **Jacob Vosmaer**

Logic, Algebra and Topology. Investigations into canonical extensions, duality theory and point-free topology.

ILLC DS-2010-11: **Nina Gierasimczuk**

Knowing One's Limits. Logical Analysis of Inductive Inference

ILLC DS-2010-12: **Martin Mose Bentzen**

Stit, It, and Deontic Logic for Action Types

ILLC DS-2011-01: **Wouter M. Koolen**

Combining Strategies Efficiently: High-Quality Decisions from Conflicting Advice

ILLC DS-2011-02: **Fernando Raymundo Velazquez-Quesada**

Small steps in dynamics of information

- ILLC DS-2011-03: **Marijn Koolen**
The Meaning of Structure: the Value of Link Evidence for Information Retrieval
- ILLC DS-2011-04: **Junte Zhang**
System Evaluation of Archival Description and Access
- ILLC DS-2011-05: **Lauri Keskinen**
Characterizing All Models in Infinite Cardinalities
- ILLC DS-2011-06: **Rianne Kaptein**
Effective Focused Retrieval by Exploiting Query Context and Document Structure
- ILLC DS-2011-07: **Jop Briët**
Grothendieck Inequalities, Nonlocal Games and Optimization
- ILLC DS-2011-08: **Stefan Minica**
Dynamic Logic of Questions
- ILLC DS-2011-09: **Raul Andres Leal**
Modalities Through the Looking Glass: A study on coalgebraic modal logic and their applications
- ILLC DS-2011-10: **Lena Kurzen**
Complexity in Interaction
- ILLC DS-2011-11: **Gideon Borensztajn**
The neural basis of structure in language
- ILLC DS-2012-01: **Federico Sangati**
Decomposing and Regenerating Syntactic Trees
- ILLC DS-2012-02: **Markos Mylonakis**
Learning the Latent Structure of Translation
- ILLC DS-2012-03: **Edgar José Andrade Lotero**
Models of Language: Towards a practice-based account of information in natural language
- ILLC DS-2012-04: **Yurii Khomskii**
Regularity Properties and Definability in the Real Number Continuum: idealized forcing, polarized partitions, Hausdorff gaps and mad families in the projective hierarchy.
- ILLC DS-2012-05: **David García Soriano**
Query-Efficient Computation in Property Testing and Learning Theory

- ILLC DS-2012-06: **Dimitris Gakis**
Contextual Metaphilosophy - The Case of Wittgenstein
- ILLC DS-2012-07: **Pietro Galliani**
The Dynamics of Imperfect Information
- ILLC DS-2012-08: **Umberto Grandi**
Binary Aggregation with Integrity Constraints
- ILLC DS-2012-09: **Wesley Halcrow Holliday**
Knowing What Follows: Epistemic Closure and Epistemic Logic
- ILLC DS-2012-10: **Jeremy Meyers**
Locations, Bodies, and Sets: A model theoretic investigation into nominalistic mereologies
- ILLC DS-2012-11: **Floor Sietsma**
Logics of Communication and Knowledge
- ILLC DS-2012-12: **Joris Dormans**
Engineering emergence: applied theory for game design
- ILLC DS-2013-01: **Simon Pauw**
Size Matters: Grounding Quantifiers in Spatial Perception
- ILLC DS-2013-02: **Virginie Fiutek**
Playing with Knowledge and Belief
- ILLC DS-2013-03: **Giannicola Scarpa**
Quantum entanglement in non-local games, graph parameters and zero-error information theory
- ILLC DS-2014-01: **Machiel Keestra**
Sculpting the Space of Actions. Explaining Human Action by Integrating Intentions and Mechanisms
- ILLC DS-2014-02: **Thomas Icard**
The Algorithmic Mind: A Study of Inference in Action
- ILLC DS-2014-03: **Harald A. Bastiaanse**
Very, Many, Small, Penguins
- ILLC DS-2014-04: **Ben Rodenhäuser**
A Matter of Trust: Dynamic Attitudes in Epistemic Logic
- ILLC DS-2015-01: **María Inés Crespo**
Affecting Meaning. Subjectivity and evaluativity in gradable adjectives.

ILLC DS-2015-02: **Mathias Winther Madsen**

The Kid, the Clerk, and the Gambler - Critical Studies in Statistics and Cognitive Science

ILLC DS-2015-03: **Shengyang Zhong**

Orthogonality and Quantum Geometry: Towards a Relational Reconstruction of Quantum Theory

ILLC DS-2015-04: **Sumit Sourabh**

Correspondence and Canonicity in Non-Classical Logic