Fixed-Point Logics on Trees

Amélie Gheerbrant

Fixed-Point Logics on Trees

ILLC Dissertation Series DS-2010-08

INSTITUTE FOR LOGIC, LANGUAGE AND COMPUTATION

For further information about ILLC-publications, please contact

Institute for Logic, Language and Computation
Universiteit van Amsterdam
Science Park 904
1098 XH Amsterdam
phone: +31-20-525 6051
fax: +31-20-525 5206
e-mail: illc@uva.nl
homepage: http://www.illc.uva.nl/

Fixed-Point Logics on Trees

Academisch Proefschrift

ter verkrijging van de graad van doctor aan de
Universiteit van Amsterdam
op gezag van de Rector Magnificus
prof. dr. D.C. van den Boom
ten overstaan van een door het college voor
promoties ingestelde commissie, in het openbaar
te verdedigen in de Agnietenkapel
op donderdag 9 december 2010, te 12.00 uur

door

Amélie Pauline Gheerbrant

geboren te Parijs, Frankrijk.

Promotiecommissie:

Promotor: Prof.dr. J. F. A. K. van Benthem
Co-promotor: Dr. B. D. ten Cate

Overige leden:

Prof. dr. G. Gottlob
Prof. dr. E. Grädel
Prof. dr. D. H. J. de Jongh
Prof. dr. J. Väänänen
Prof. dr. Y. Venema
Dr. D. Berwanger

Faculteit der Natuurwetenschappen, Wiskunde en Informatica

The investigations were supported by a GLoRiClass fellowship of the European Commission (Research Training Fellowship MEST-CT-2005-020841).

Cover design by Emy (http://arrachetoiunoeil.free.fr/).
Printed and bound by http://www.printondemand-worldwide.com.

ISBN: 978–90–55776–216–1

To the O.C.C.I.I.

Contents

Acknowledgments

I began as a Philosophy student, wrote a PhD in Logic and just started as a research associate in theoretical Computer Science. Hence I went "From Philosophical to Industrial Logics" (taking some sort of wild path slightly reminiscent of [131]), which was not always easy. Some people helped me and I want to thank them. Back when I was studying Philosophy, I especially remember Susana Berestovoy, Marcin Mostowski and Gabriel Sandu. I discovered logic through Susana and she advertised model theory so well that I fell under the spell of the topic. Marcin then convinced me that at some point, I should also try to understand finite model theory and complexity theory. Finally, Gabriel advised me to go for a PhD in Amsterdam, where I could happily learn more about Logic.

Indeed I learnt a lot in Amsterdam. I am especially indebted to my supervisors, Johan van Benthem and Balder ten Cate. Balder introduced me to some very nice topics and spent countless hours pointing out mistakes in my wanna be proofs. Turning me into a competent logician was not an easy task and I feel lucky that he invested so much time and energy into the (still ongoing) transformation. As a friend and beyond work, I also want to thank him for cheering me up and having been a friendly ear in difficult moments. Johan became my supervisor only later. Johan is not only a very good logician, he is also curious of many other things and rich in stimulating new ideas. It was both instructive and refreshing to start looking at game theory through his eyes. Wherever I had only blurry intuitions or feelings, he was always able to offer me a glimpse into some higher and harder to reach standpoint.

I am also grateful to the GloRiClass coordinators (Krzysztof Apt, Johan van Benthem, Paul Dekker, Ulle Endriss and Benedikt Löwe) for making me part of the adventure and to the whole extended family of the ILLC staff and students (a special word is in order for Karin Gigengack, Tanja Kassenaar, Ingrid van Loon, Peter van Ormondt and Marjan Veldhuisen, whose help with Dutch bureaucracy is deeply acknowledged). Among these, I remember nice parties and discussions with (to name only a few) Alexandru Baltag, Nick Bezhanishvili, Cédric

Dégremont, Gaëlle Fontaine, Nina Gierasimczuk, Florian Horn, Daisuke Ikegami, Olivia Ladinig, Tadeusz Litak, Salvador Mascarenhas, Olivier Roy, Sonja Smets, Maria Spychalska, Jakub Szymanik, Jouko Väänänen, Yde Venema, Jakob Vosmaer, Andreas Witzel and Jonathan Zvesper. I am especially grateful to Jouko for illuminating discussions about Ehrenfeucht Fraïssé games and for letting me be his teaching assistant for the Model Theory course. Thank you also to my friend Alexandru for his hospitality in Oxford, where heavenly cheese and wine accompanied our endless discussions ranging from Philosophy and Mathematics to the private life of robots. I am also grateful to Maarten Marx, Maarten de Rijke and the ILPS people for welcoming me warmly and giving me an office when Balder was still there. At least as far as work is concerned, Gaëlle also deserves a special acknowledgment. We talked for hours about life and logic in our own strange mix of French and English, often while seriously partying. I subscribe both to her good mathematical taste and to her vision of life and I express all my respect for her ability to discuss my broken proofs at completely indecent times of the night. I also want to thank our common friends in Amsterdam: in particular Charlotte Caldwell, Luc van Welden, Olivia Ladinig, Sjoerd Stolk and Teun Doekes, but also the whole O.C.C.I.I. crowd. The O.C.C.I.I. is a wonderful place for independent music and it was my (our) second home, I miss it badly. I also loved living with the three outstanding girls that are Charlotte, Gaëlle and Olivia (thank you Charlie for welcoming me in your flat when I was homeless, boring and struggling to finish the thesis). I am also grateful to the people of the LSV in Cachan. I visited the place extensively in the past year and a half and I have been welcomed in a very friendly way. A warm thank you to Dietmar Berwanger, Diego Figueira, Thomas Place, Luc Ségoufin and Camille Vacher for at various times providing me with a cheerful office space. Thank you also to the people of the IHPST (in particular my friend Alexandra Arapinis) that I kept visiting after I left for Amsterdam. Another person in Paris to whom I want to express my gratitude is Claire David. She introduced me both to the mysteries of XML and to Leonid Libkin, who ended up offering me a great job in Edinburgh. I address here an especially enthusiastic thank you to Leonid for making me part of his group. I am also grateful to the members of my committee, who are all top researchers and did me the great honor of accepting to read my thesis.

Finally, I want to mention my family and friends in Paris. I owe a lot to my mother, who raised me among books with unconditional love. I am also grateful to my father, grand-parents, brother and friends for being there. A special thank you for those from musiquesincongrues, musikmekanikcirkus and arrachetoiunoeil, who contribute in making Paris a great place to return. The last three words are for Julien Daigremont a.k.a. Computer Truck: I LOVE YOU!

Edinburgh Amélie
October, 2010.

Chapter 1

Introduction

Tree structures are an appealing way of representing complex hierarchical information in a graphical form. Tree pictures are ubiquitous, from family histories to internet documents, and from medical records to blueprints for plans. One could make a very rich inventaire à la Prévert out of all the different species of tree structures encountered in human activities.

Not surprisingly then, trees are also prominent in the world of academic research. They are a major tool in disciplines as diverse as linguistics, mathematics, philosophy, game theory, or computer science. And in addition to being a favorite tool for research, finite and infinite trees are themselves the object of explicit study, for instance, in mathematics and set theory. Specific literature references to such areas will be made later on, when relevant to our investigation.

Trees in Logic

This thesis is about trees from the perspective of logic. Logicians use trees in central notions of their discipline, witness the tree structure of formulas, natural deduction proofs, or semantic tableaux. But tree structures being prominent in many other disciplines, logicians have also found various motivations to study trees theoretically, and the languages and logics most appropriate to describing them. Our investigation is in the latter mode.

One striking feature, then, is that there is not one single logic of trees, there is a whole family of approaches. Trees have been described in first-order logic, or stronger second-order logics like MSO (first-order logic with monadic set quantifiers). But they have also been studied extensively with an array of temporal logics, in both "linear time" versions (LTL) and "branching time" versions (CTL). And in yet another mode, they have also been studied using modal logics with additional fixed-point operators to describe recursive notions naturally associated with trees, such as "reachability". In addition, a source of diversity is that trees in practice usually have relevant additional structure, such as "sibling orders" in linguistic parse trees, or "preference orders" in game trees.

In this broad area of logic on trees, the interests underlying this thesis are several. First, we design and study some detailed tree logics for mainly computational purposes, such as data structures for XML documents. Usually this means that trees have to be enriched with appropriate additional structure, where one of our guising interests is recursion and reachability. But next, such applications are best understood in the light of theory. We also study basic properties of the logics that we propose, including complete axiomatizations. A view from logical theory also has another virtue, namely, bringing to light connections between the sometimes quite different logics proposed in practice, and maintaining an overview of the area. We will give several examples of this coherence, in terms of new connections between logics on trees, interpolation properties singling out especially well-behaved systems, and notions of semantic invariance. We will be mainly using techniques from classical model theory, also from abstract model theory and we will sometimes rely on automata-theory, which is a common tool in the domain. While the main thrust of this work is directed toward computer science (database theory, formal verification), at the end of the thesis, we take a look at a new interface, namely, game theory. The encounter between logic and games was in fact a major theme of the "Gloriclass" Marie Curie Center at the ILLC in Amsterdam (http://www.illc.uva.nl/GLoRiClass/), of which this thesis project was a part. We take methods developed in fixed-point logics of computation, joint with dynamic logics of knowledge and belief, to the analysis of game solution procedures, and find new links between logic, computation and games.

Chapter Overview

Chapter 2 In the preliminary chapter of the thesis, we list the main logical notions, tools and results that will be used later on. These include basics of modal logic, temporal logic, fixed-point logics, and some first-order and higher-order logics of tree structure.

Chapter 3 In Chapter 3, we consider a specific class of tree structures that can represent basic structures in linguistics and computer science such as XML documents, parse trees, and treebanks, namely, finite node-labelled sibling-ordered trees. We present axiomatizations of the monadic second-order logic (MSO), monadic transitive closure logic ($\mathsf{FO}(\mathsf{TC}^1)$) and monadic least fixed-point logic ($\mathsf{FO}(\mathsf{LFP}^1)$) theories of this class of structures. These logics can express important properties such as reachability. Using model-theoretic techniques, we show by a uniform argument that these axiomatizations are complete, i.e., each formula which is valid on all finite trees is provable using our axioms. As a backdrop to our positive results, on arbitrary structures, the logics that we study are known to be non-recursively axiomatizable.

Chapter 4 Next, we take a more general abstract look at temporal logics of tree structure through their system properties. In Chapter 4, using techniques from abstract model-theory, we propose a general viewpoint on temporal logics in term of their Craig interpolation properties. We consider various fragments and extensions of propositional linear temporal logic (LTL), obtained by restricting the set of temporal connectives or by adding a least fixed-point construct to the language. For each of these logics, we identify its smallest extension that has Craig interpolation. Depending on the underlying set of temporal operators, this framework turns out to be one of the following three logics: the fragment of LTL having only the Next operator; the extension of LTL with a least fixed-point operator μ (known as linear time μ-calculus); and the least fixed-point extension of the "Until-only" fragment of LTL.

Chapter 5 Our next perspective on tree logics is through fixed-point operations and semantic invariance. We focus on the logic $\mu\mathsf{TL}(\mathsf{U})$, the least fixed-point extension of the "Until"-only fragment of linear-time temporal logic. In the previous chapter, we identified $\mu\mathsf{TL}(\mathsf{U})$ as the stutter-invariant fragment of the linear-time μ-calculus $\mu\mathsf{TL}$. We also identified this logic as one of the three only temporal fragments of $\mu\mathsf{TL}$ that satisfy Craig interpolation. Complete axiom systems were known for the two other fragments, but this was not the case for $\mu\mathsf{TL}(\mathsf{U})$. We provide complete axiomatizations of $\mu\mathsf{TL}(\mathsf{U})$ on the class of finite words and on the class of ω-words. For this purpose, we introduce a new logic $\mu\mathsf{TL}(\Diamond_\Gamma)$, a variation of $\mu\mathsf{TL}$ where the "Next time" operator is replaced by the family of its stutter-invariant counterparts. This logic has exactly the same expressive power as $\mu\mathsf{TL}(\mathsf{U})$. Using known results for $\mu\mathsf{TL}$, we first prove completeness for $\mu\mathsf{TL}(\Diamond_\Gamma)$, which then allows us to obtain completeness for $\mu\mathsf{TL}(\mathsf{U})$.

Chapter 6 Finally, we take our style of analysis via modal and temporal fixed-point logics to games. Current methods for solving games embody a form of "procedural rationality" that invites logical analysis in its own right. This chapter is a case study of Backward Induction for extensive games. We consider a number of analyses from recent years in terms of knowledge and belief update in logics that also involve preference structure, and we prove that they are all mathematically equivalent in the perspective of fixed-point logics of trees. We then generalize our perspective on games to an exploration of fixed-point logics on finite trees that best fit game-theoretic equilibria. We end with a broader program for merging computational logics to the area of game theory.

Origins of the Material

Chapter 3 is based on joint work with Balder ten Cate, a short version of which was published in [70]. Chapter 4 is also based on joint work with Balder ten Cate, which was published in [71]. The material in Chapter 5 was published in [69]. Finally, Chapter 6 is based on joint work with Johan van Benthem, a large part of which was published in [22].

Chapter 2

Preliminaries

2.1 Trees as Relational Structures

2.1.1 Different Sorts of Trees

By a relational vocabulary[1], or *signature* σ, we mean a finite set of relational symbols, or *predicates* $\{R_1, \ldots, R_n\}$ with associated arities $ar(R_i) \in \mathbb{N}$. A *relational structure* $\mathfrak{M}$ over σ, or σ-structure, is a tuple $(dom(\mathfrak{M}), R_1^{\mathfrak{M}}, \ldots, R_n^{\mathfrak{M}})$ where $dom(\mathfrak{M})$, the domain of $\mathfrak{M}$, is a set and for each R_i, $R_i^{\mathfrak{M}} \subseteq dom(\mathfrak{M})^{ar(R_i)}$. In this thesis, we will focus on particular relational structures that are called trees and we will be using appropriate relational symbols in order to talk about them. A tree is a partially ordered set with a unique smallest element called *the root* and such that apart from the root, each element (or *node*) has one unique immediate predecessor. Often, the underlying partial order, or *descendent* order, is explicitly represented by means of a binary predicate $\leq$, but sometimes, only the corresponding immediate successor relation, or descending edge relation in the graph, is represented using some binary predicate $<_{ch}$. Note that $<_{ch}$ is not a partial order (in particular, it is not transitive). We say that *x is an ancestor of y* and *y is a descendant of x* whenever $x \leq y$ and we say that *x is the parent of y* and *y is the child of x* whenever $x <_{ch} y$. A minimal vocabulary to deal with trees involves at least one of these two predicates. For instance, assuming $dom(\mathfrak{M})$ is some finite set and $\leq$ has a suitable interpretation in $\mathfrak{M}$, $(dom(\mathfrak{M}), \leq^{\mathfrak{M}})$ is a (finite) tree.

Starting from such basic tree structures, different classes of trees are relevant in different areas and accordingly, many different optional features come into play. First of all, from a mathematical point of view, it makes a big difference if one assumes that $dom(\mathfrak{M})$ is *finite* or *infinite*. An other important distinction, both mathematically and historically speaking (see [104]), is the distinction *ranked*

[1] In the remaining of the thesis, we will work exclusively with *purely* relational vocabularies, i.e., with no individual constant or function symbols.

versus unranked trees. In ranked trees, all non-terminal nodes have the same fixed number of children, [2], whereas no such constraint is put in the case of unranked trees. The distinction can sometimes be considered as a mere technical convenience, as any countable tree can be encoded as a binary tree. Indeed, given some unranked tree, one can always find a binary tree which can be put in one to one correspondence with it in the following way: each node x in the unranked tree will correspond to a node x' in the binary tree, the left child of x' will correspond to the first child of x, and the right child of x' will correspond to x's next sibling, this way the n^{th} sibling of x will correspond to some n^{th} descendant of x'. Another distinction related to the one ranked versus unranked is the one *bounded versus unbounded branching.* All ranked trees and finite unranked trees are of (some) bounded branching, and among infinite unranked trees, some are also of bounded branching. An interesting case of trees which are of bounded branching is those where every node has at most one successor. Such trees are called *linear orders* and they are also ranked trees of "degree" 1. We will focus respectively on finite unranked trees (chapter 3 and 6) and on finite linear orders and linear orders of order type ω (chapter 4 and 5).

Many other features can also be represented by means of additional predicates. Adding unary predicates allows for instance to label nodes in the tree, so that different trees do not differ only by the cardinality of their domains and by their underlying descendent order, but also by the way they are labelled. Such trees are called *node-labelled.* Unary predicates can for instance be introduced to label special nodes like the root or the leaves. In the following, we will always deal with node-labelled trees. Often incorporated in the language is another feature, which is implicit in the graphic representation of a tree: the *sibling-order* linearly ordering the children of each node, which can be represented by means of an additional binary predicate $\preceq$. The unranked finite trees in Chapter 3 will be considered together with such an additional feature. Many other options are of course possible. In the case of finite trees, one could also want to consider one or more linear orders over the leaves. We will see in the last chapter that finite extensive games in perfect information can be represented in this way. We will also see that such "trees" are much more complex objects than the other trees described above. Hence, in the remaining, whenever we will say *on trees*, without any other precision, we will mean on the sort of simple trees described above, but not on game trees.

Finally, one reason for which trees are nice mathematical objects is that they are really easy to picture. It follows that in practise, reasoning about them often involves drawing interesting samples and trying to modify or compose them: cut-

[2]Sometimes, whenever trees are *node labelled*, the number of children can also be determined by the label of that node. Consider for instance a tree where all non-terminal nodes are labelled by "AND", "OR" or "NOT", while leaves are labelled by propositional letters. Whenever the nodes labelled by "AND" or "OR" are all binary branching and the nodes labelled by "NOT" are all unary branching, we say that the tree is ranked.

ting and pasting some parts of the graph, coloring or adding edges, considering disjoint unions of trees or even more complex generalized products like forests (as we will do in Chapter 3)... In this context, some important part of the mathematical thinking really takes place at the picture level. Hence, let us illustrate our quick tree typology with the following simple graphic example of a tree:

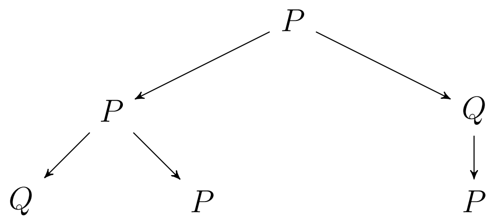


Figure 2.1: A finite unranked node-labelled tree

Obviously the tree pictured here is finite (there are exactly six nodes in its domain), node-labelled, unranked and its branching is bounded by 2. What is less clear is the exact vocabulary of the structure. It could be $\{\leq, P, Q\}$, $\{<_{ch}, P, Q\}$, $\{\leq, \preceq, P, Q\}$, $\{\leq, \prec_{ch}, P, Q\}$, or some other variation involving $\leq$, $\prec_{ch}$ and $\preceq$.

2.1.2 Fixed-Point Extensions of First-Order Logic

In order to talk about relational structures (and, hence, about trees), one typically uses first-order logic (FO). We will shortly recall basics notions and facts about FO, but we will introduce also five of its extensions: monadic second-order logic MSO, monadic transitive closure logic FO(TC^1), least fixed-point logic FO(LFP), inflationary fixed-point logic FO(IFP) and partial fixed-point logic FO(PFP). In the remaining of the thesis (unless explicitly stated otherwise), we will always be working with a fixed purely relational vocabulary σ and hence, with σ-structures. We assume as usual that we have a countably infinite set of first-order variables. In the case of MSO, FO(LFP), FO(IFP) and FO(PFP) we also assume that we have a countably infinite set of second-order variables (only of arity 1 in the case of MSO, of arbitrary arity otherwise). Given the fact that we will introduce in Chapter 3 alternative semantics for some of these logics, the semantics defined in this section we will refer to as *standard semantics* and the associated structures, as *standard structures*.

First-Order Logic

Let us first recall the syntax and semantics of FO:

Definition 2.1.1 (Syntax and semantics of FO)**.** Let $P \in \sigma$ be a predicate of arity n and $x, x_1, \ldots, x_n$ be first-order variables. We let At be a formula of either the form $x_i = x_j$, or $P(x_1, \dots x_n)$ and call it a first-order atomic formula. We inductively define the set of FO formulas in the following way:

$$\varphi := At \mid \varphi \vee \psi \mid \neg\varphi \mid \exists x\ \varphi$$

We use $\forall x\varphi$, $\varphi \wedge \psi$, $\varphi \rightarrow \psi$ as shorthand for, respectively, $\neg\exists x\neg\varphi$, $\neg(\neg\varphi \vee \neg\psi)$ and $\neg(\varphi \wedge \neg\psi)$. We define the *quantifier depth* of a FO formula as the maximal number of nested first-order quantifiers. We interpret FO formulas in relational structures. The truth of a FO-formula in a relational structure $\mathfrak{M}$ is defined modulo a valuation g of variables as objects. We let $g[a/x]$ be the assignment which differs from g only in assigning a to x. We inductively define the truth of FO-formulas as follows:

$\mathfrak{M}, g \models x_i = x_j$	iff	$g(x_i) = g(x_j)$
$\mathfrak{M}, g \models P(x_1, \dots, x_n)$	iff	$(g(x_1), \dots, g(x_n)) \in P^{\mathfrak{M}}$
$\mathfrak{M}, g \models \neg\phi$	iff	$\mathfrak{M}, w \not\models \phi$
$\mathfrak{M}, g \models \phi \vee \psi$	iff	$\mathfrak{M}, w \models \phi$ or $\mathfrak{M} \models \psi$
$\mathfrak{M}, g \models \exists x\phi$	iff	there exists $a \in dom(\mathfrak{M})$ such that $\mathfrak{M}, g[a/x] \models \phi$

We presented the semantics of FO in a general way and only assumed that $\mathfrak{M}$ was “some” relational structure. Still, in the remaining we want to focus on trees. A natural question is hence the following. With the FO language at hands, what are the interesting things that one can say about trees? Moreover, how fine grained can one be using FO? Looking at the simple tree drawing in Figure 2.1 and assuming it corresponds to a $\{\leq, \preceq, P, Q\}$-structure, we can observe for instance that this structure satisfies the following FO-formulas:

- $\exists x(Q(x) \wedge \forall y(x \leq y \rightarrow x = y))$
- $\exists x\exists y(P(x) \wedge Q(y) \wedge x \leq y \wedge \forall z((x \leq z \wedge z \leq y) \rightarrow (z = x \vee z = y)))$

The first formula says that there is a leaf labelled by Q. The second formula says that there is a node labelled by P which has a child labelled by Q. Something else that these formulas show is that the predicates $Leaf$ and $<_{ch}$ are *definable in FO* whenever one takes as basis vocabulary $\{\leq\}$.[3] One only needs to stipulate:

- $Leaf(x) := \forall y(x \leq y \rightarrow x = y)$
- $x <_{ch} y := x \leq y \wedge \forall z((x \leq z \wedge z \leq y) \rightarrow z = x \vee z = y))$

A relation $\prec_{ns}$ of *next sibling* is similarly definable in FO using the relation $\preceq$. On the other hand, it is known that $\leq$ and $\preceq$ are *not definable* in FO by means of $<_{ch}$ and $\prec_{ns}$, whereas both are definable in the more expressive logics that we are about to introduce. A classical way to show this relies on a model-theoretic tool called the FO *Ehrenfeucht-Fraïssé game* (see [104] or [58]). We will present this game, but first, we need the following notion:

[3]Note that we treated = as a logical constant, already present in the apparatus of FO.

Definition 2.1.2 (Elementary Equivalence)**.** Given two relational structures $\mathfrak{M}$ and $\mathfrak{N}$, we write $\mathfrak{M} \equiv \mathfrak{N}$ and say that $\mathfrak{M}$ and $\mathfrak{N}$ are *elementary-equivalent* if they satisfy the same FO-sentences. Also, for any natural number n, we write $\mathfrak{M} \equiv^n \mathfrak{N}$ and say that $\mathfrak{M}$ and $\mathfrak{N}$ are *n-elementary equivalent* if $\mathfrak{M}$ and $\mathfrak{N}$ satisfy the same FO-sentences of quantifier depth at most n. In particular, $\mathfrak{M} \equiv \mathfrak{N}$ holds iff, for all n, $\mathfrak{M} \equiv^n \mathfrak{N}$ holds.

One rather trivial sufficient condition for FO-equivalence is the existence of an *isomorphism.* Clearly isomorphic structures satisfy the same FO-formulas. A more interesting sufficient condition for elementary equivalence is that of Duplicator having a winning strategy in all FO Ehrenfeucht-Fraïssé games of finite length. To define this, we first need the following notion:

Definition 2.1.3 (Finite Partial Isomorphism)**.** A *finite partial isomorphism* between structures $\mathfrak{M}$ and $\mathfrak{N}$ is a finite relation $\{(a_1, b_1), \ldots, (a_n, b_n)\}$ between the domains of $\mathfrak{M}$ and $\mathfrak{N}$ such that for all atomic formulas $\varphi(x_1, \ldots, x_n)$, $\mathfrak{M} \models \varphi\ [a_1, \ldots, a_n]$ iff $\mathfrak{N} \models \varphi\ [b_1, \ldots, b_n]$. Since equality statements are atomic formulas, every finite partial isomorphism is (the graph of) an *injective partial function.*

We will also need the following lemma:

Lemma 2.1.4 (Finiteness Lemma)**.** *Fix any set $x_1, \ldots, x_k, X_{k+1}, \ldots, X_m$. In a finite relational vocabulary, up to logical equivalence, with these free variables, there are only finitely many FO-formulas of quantifier depth $\leq n$.*

Proof. This can be shown by induction on k. In a finite relational vocabulary, with finitely many free variables, there are only finitely many atomic formulas. Now, any FO-formula of quantifier depth $k + 1$ is equivalent to a Boolean combination of atoms and formulas of quantifier depth k prefixed by a quantifier. Applying a quantifier to equivalent formulas preserves equivalence and the Boolean closure of a finite set of formulas remains finite, up to logical equivalence. □

We can now introduce the classical FO game:

Definition 2.1.5 (FO Ehrenfeucht-Fraïssé Game)**.** The FO Ehrenfeucht-Fraïssé game of length n on structures $\mathfrak{M}$ and $\mathfrak{N}$ (notation: $EF^n_{FO}(\mathfrak{M}, \mathfrak{N})$) is as follows. There are two players, Spoiler and Duplicator. The game has n rounds, each of which consists of a move of Spoiler followed by a move of Duplicator. Spoiler's moves consist of picking an element from one of the two structures, and Duplicator's responses consist of picking an element in the other structure. In this way, Spoiler and Duplicator build up a finite binary relation between the domains of the two structures: initially, the relation is empty; each round, it is extended with another pair. The winning conditions are as follows: if at some point of the game the constructed binary relation is not a finite partial isomorphism, then Spoiler wins immediately. If after each round the relation is a finite partial isomorphism, then the game is won by Duplicator.

Theorem 2.1.6 (FO Adequacy)**.** *Assume a finite relational first-order language. Duplicator has a winning strategy in the game $EF^n_{FO}(\mathfrak{M},\mathfrak{N})$ iff $\mathfrak{M} \equiv^n_{FO} \mathfrak{N}$. In particular, Duplicator has a winning strategy in all EF-games of finite length between $\mathfrak{M}$ and $\mathfrak{N}$ if and only if $\mathfrak{M} \equiv_{FO} \mathfrak{N}$.*

The proof for the first-order case is classic. We refer the reader to the proof given in [104], to the one given in [58] or to the the more elaborate $\mathsf{FO}(\mathsf{TC}^1)$ adequacy proof that we give in Chapter 3 (the proof for FO can be obtained in a straightforward way by forgetting about the TC move and TC quantifier).

With such a tool at hand, one can for instance show that the transitive closure $\leq$ of the $<_{ch}$ relation is not definable in FO and that neither is the class of finite sibling-ordered trees. As these results are folklore, we only provide a short detailed proof (inspired by [63]) for the first one as an example of proof using the FO-game and we omit the proof of the second one. We refer to [104] for other detailed proofs of simpler related propositions.

Proposition 2.1.7. *On finite linear orders, the predicate $\leq$ is not definable in FO with basic vocabulary $\{<_{ch}\}$.*

Proof. We assume that $\leq$ is definable by a FO-formula $\varphi(x,y)$ of quantifier-depth n, from which we will derive a contradiction. Let $\mathcal{L}$ be a finite linear order in vocabulary $\{<_{ch}\}$ and $\bar{a}$ a finite sequence of parameters in $dom(\mathcal{L})$. By $N^{\mathcal{L}}_r(\bar{a})$ we mean the restriction of $\mathcal{L}$ to all the elements which are at distance at most r along the $<_{ch}$ relation from one of the parameters in $\bar{a}$. This structure, which is not necessarily a linear order can be seen as the "r-neighborhood" (see [104]) of the elements in $\bar{a}$. Now let $r_n = (3^n - 1)/2$ and note that $r_{n+1} = 3r_n + 1$. Consider a linear order $\mathcal{L}$ of length at least r_{n+2} with four distinguished elements $a_{start}, a_{end}, a_*, a^* \in dom(\mathcal{L})$ such that a_{start} is the root, a_{end} the last element and a_*, a^* are points wich distance between them, as well as between the endpoints, is greater than $2r_n + 1$, i.e., greater than 3^n.[4] Let $\mathcal{L}'$ be copy of $\mathcal{L}$, with four parameters $b_{start}, b_{end}, b_*, b^* \in dom(\mathcal{L}')$ such that in $\mathcal{L}'$, b_* is interpreted as a^* and b^* as a_*, while b_{start} and b_{end} are interpreted as a_{start} and a_{end} respectively. Observe that the formula $\varphi(x,y)$ defining $\leq$ is satisfied in $\mathcal{L}, a_*, a^*$ if and only if it is not satisfied in $\mathcal{L}', b_*, b^*$. We will show that $(\mathcal{L}, a_{start}, a_{end}, a_*, a^*) \equiv^n_{\mathsf{FO}} (\mathcal{L}', b_{start}, b_{end}, b_*, b^*)$. This will be enough, as this will contradict our assumption that $\varphi(x,y)$ defines $\leq$.

We consider the game $EF^n_{\mathsf{FO}}((\mathcal{L}, a_{start}, a_{end}, a_*, a^*), (\mathcal{L}', b_{start}, b_{end}, b_*, b^*))$ and we let a_i's and b_i's be the elements played in $\mathcal{L}$ and $\mathcal{L}'$, respectively. We let $\bar{a}^i$ stand for the sequence $a_{start}, a_{end}, a_*, a^*, a_1, \ldots, a_i$ and let $\bar{b}^i$ be defined likewise. We show that Duplicator can play in such a way that $N^{\mathcal{L}}_{r_{n-i}}(\bar{a}^i)$ is isomorphic to $N^{\mathcal{L}'}_{r_{n-i}}(\bar{b}^i)$ after i rounds. It will follow that he has a winning strategy in

[4]Note that we could have picked a smaller model. However, the proof will be shorter with a model of this size, as we will not need to consider any overlapping of neighborhoods.

$EF^n_{\mathsf{FO}}((\mathcal{L}, a_{start}, a_{end}, a_*, a^*), (\mathcal{L}', b_{start}, b_{end}, b_*, b^*))$, since after n rounds a partial isomorphism will be given by the map $\bar{a}^n \to \bar{b}^n$ (as $N_0^{\mathcal{L}}(\bar{a}^n)$ will be isomorphic to $N_0^{\mathcal{L}'}(\bar{b}^n)$). The proof is by induction on the length of the game n. The base case $i = 0$ follows immediately from the assumption on the distance between a_{start} a_{end}, a_* and a^* (and thus b_{start}, b_{end}, b_* and b^*). For the induction case (going from i to $i+1$), let $r = r_{n-(i+1)}$ so that $r_{n-i} = 3r+1$. The hypothesis tells us that there is an isomorphism $h : N^{\mathcal{L}}_{3r+1}(\bar{a}^i) \to N^{\mathcal{L}'}_{3r+1}(\bar{b}^i)$. Assume (as it is symmetric) that Spoiler plays in $\mathcal{L}$. There are two cases.

1. Spoiler plays $a_{i+1} \in N^{\mathcal{L}}_{2r+1}(\bar{a}^i)$. Pick $b_{i+1} = h(a_{i+1})$. Since h is an isomorphism, we have an isomorphism between $N_r^{\mathcal{L}}(\bar{a}^{i+1})$ and $N_r^{\mathcal{L}'}(\bar{b}^{i+1})$.

2. Spoiler plays $a_{i+1} \notin N^{\mathcal{L}}_{2r+1}(\bar{a}^i)$. In particular, $N_r^{\mathcal{L}}(a_{i+1})$ and $N_r^{\mathcal{L}}(\bar{a}^i)$ are disjoint and there are no edges between them. Pick any b_{i+1} so that $N_r^{\mathcal{L}'}(b_{i+1})$ and $N_r^{\mathcal{L}'}(\bar{b}^i)$ are disjoint; it exists because the sizes of $N_r^{\mathcal{L}}(\bar{a}^i)$ and $N_r^{\mathcal{L}'}(\bar{b}^i)$, and the sizes of $\mathcal{L}$ and $\mathcal{L}'$ are the same. Then clearly $N_r^{\mathcal{L}}(\bar{a}^{i+1})$ and $N_r^{\mathcal{L}'}(\bar{b}^{i+1})$ are isomorphic.

□

Proposition 2.1.8. *The class of unranked sibling-ordered finite trees is not definable in* FO *with basic vocabulary* $\{\leq, \preceq\}$.

We omit the proof of Proposition 2.1.8, but we will see in Chapter 3 that in order to define the class of finite trees, one needs an "induction axiom" that is beyond FO's expressive reach. This particular restriction in expressive power is a symptom of a more general feature of FO: it lacks a means to express recursive properties, in other words it is "local" (see [104]). We will now introduce a few extensions of FO which will be of particular interest to us and which are expressive enough to define the class of finite trees (we will explain how in Chapter 3).

Monadic Second-Order Logic

A first way to extend the expressive power of FO is to allow second-order quantifiers over relation symbols. We will focus here on monadic second-order logic MSO, which only allows a restricted form of such quantification. MSO is the extension of first-order logic in which we can quantify over the subsets of the domain. It is a very standard and widely used logic on trees and we will see that it has many nice features.

Definition 2.1.9 (Syntax and semantics of MSO)**.** Let At be a first-order atomic formula, x a first-order variable and X a set variable, we inductively define the set of MSO formulas in the following way:

$$\varphi := At \mid Xx \mid \varphi \vee \psi \mid \neg\varphi \mid \exists x\ \varphi \mid \exists X\ \varphi$$

We use $\forall X\varphi$ as shorthand for $\neg\exists X\neg\varphi$ and $\forall x\varphi$, $\varphi\wedge\psi$, $\varphi\rightarrow\psi$ as shorthand in the usual way. We define the *quantifier depth* of a MSO formula as the maximal number of nested first-order and second-order quantifiers. We interpret MSO formulas in relational structures. Like for FO formulas, the truth of a MSO-formula in a relational structure $\mathfrak{M}$ is defined modulo a valuation g of variables as objects. Additionally, g is also defined over set variables, to which it assigns subsets of the domain. We let $g[a/x]$ be the assignment which differs from g only in assigning a to x (similarly for $g[A/X]$). The truth of atomic formulas is defined by the usual FO clauses plus the following:

$$\mathfrak{M}, g \models Xx \text{ iff } g(x) \in g(X) \text{ for } X \text{ a set variable}$$

The truth of compound formulas is defined by induction, with the same clauses as in FO and an additional one:

$$\mathfrak{M}, g \models \exists X\varphi \text{ iff there is } A \subseteq dom(\mathfrak{M}) \text{ such that } \mathfrak{M}, g[A/X] \models \varphi$$

Observe that in MSO, $\leq$ and $\preceq$ are definable by means of $<_{ch}$ and $\prec_{ns}$:

- $x \leq y := \forall X((Xx \rightarrow \forall u \forall v((Xu \wedge u <_{ch} v) \rightarrow Xv)) \rightarrow Xy)$
- $x \preceq y := \forall X((Xx \rightarrow \forall u \forall v((Xu \wedge u \prec_{ns} v) \rightarrow Xv)) \rightarrow Xy)$

These two formulas say that x and y are, respectively, in the *reflexive transitive closure* of the relations $<_{ch}$ and $\prec_{ns}$.

Monadic Transitive Closure Logic

The second logic we are interested in is a fragment of MSO called monadic transitive closure logic, $\mathsf{FO}(\mathsf{TC}^1)$, which extends FO by closing it under the reflexive transitive closure of binary definable relations.

Definition 2.1.10 (Syntax and semantics of $\mathsf{FO}(\mathsf{TC}^1)$)**.** Let u, v, x, y be first-order variables, $\varphi(x, y)$ a $\mathsf{FO}(\mathsf{TC}^1)$ formula (which, besides x and y, possibly contains other free variables), we inductively define the set of $\mathsf{FO}(\mathsf{TC}^1)$ formulas in the following way:

$$\varphi := At \mid Xx \mid \varphi \vee \psi \mid \neg\varphi \mid \exists x\ \varphi \mid [\mathsf{TC}_{xy}\varphi(x, y)](u, v)$$

We use $\forall x\varphi$, $\varphi \wedge \psi$, $\varphi \rightarrow \psi$ as shorthand in the usual way. We define the *quantifier depth* of a $\mathsf{FO}(\mathsf{TC}^1)$ formula as the maximal number of nested first-order quantifiers and TC operators. We interpret $\mathsf{FO}(\mathsf{TC}^1)$ formulas in relational structures. The notion of assignation and the truth of atomic formulas is defined as in FO. The truth of compound formulas is defined by induction, with the same clauses as in FO and an additional one:

$$\mathfrak{M}, g \models [\mathsf{TC}_{xy}\varphi](u, v)$$
$$\text{iff}$$
$$\text{for all } A \subseteq M, \text{ if } g(u) \in A$$
$$\text{and for all } a, b \in dom(\mathfrak{M}),\ a \in A \text{ and } \mathfrak{M}, g[a/x, b/y] \models \varphi(x, y) \text{ implies } b \in A,$$
$$\text{then } g(v) \in A.$$

Proposition 2.1.11. *On standard structures, the following semantical clause for the* TC *operator is equivalent to the one given above:*

$$\mathfrak{M}, g \models [\mathsf{TC}_{xy}\varphi(x, y)](u, v)$$
$$\textit{iff}$$
$$\textit{there exist } a_1 \ldots a_n \in M \textit{ with } n \geq 0,\ g(u) = a_1,\ g(v) = a_n$$
$$\textit{and } \mathfrak{M}, g \models \varphi(a_i, a_{i+1}) \textit{ for all } 0 < i < n$$

Proof. Indeed, suppose there is a finite sequence of points $a_1 \ldots a_n$ such that $g(u) = a_1$, $g(v) = a_n$, and for each $i < n$, $\mathfrak{M}, g[x/a_i; y/a_{i+1}] \models \varphi(x, y)$. Then for any subset A closed under φ and containing a_1, we can show by induction on the length of the sequence $a_1 \ldots a_n$ that a_n belongs to A. Now, on the other hand, suppose that there is no finite sequence like described above. To show that there is a subset A of the required form, we simply take A to be the set of all points that can be reached from u via φ by a finite sequence. By assumption, v does not belong to this set and the set is closed under φ. □

Intuitively this means that for a formula of the form $[\mathsf{TC}_{xy}\varphi](u, v)$ to hold on a standard structure, there must be a *finite* "φ path" between the points that are named by the variables u and v.

Observe that in $\mathsf{FO}(\mathsf{TC}^1)$, $\leq$ and $\preceq$ are definable in a straightforward way by means of $<_{ch}$ and $\prec_{ns}$:

- $x \leq y := [\mathsf{TC}_{xy} x <_{ch} y](x, y)$
- $x \preceq y := [\mathsf{TC}_{xy} x \prec_{ns} y](x, y)$

Finally, note that we talked about *monadic* transitive closure logic, because $\mathsf{FO}(\mathsf{TC}^1)$ extends FO with the transitive closure of binary relations, but it is also possible to define $\mathsf{FO}(\mathsf{TC}^k)$ by extending FO with the transitive closure of $2k$-ary relations (i.e., binary relations between k-tuples), see [58]. In the remaining, we will focus exclusively on $\mathsf{FO}(\mathsf{TC}^1)$.

Fixed-Points

Finally, we will be interested in yet another way of extending FO with recursive means. It consists in adding explicit *fixed-point constructs* to the FO language.

Before we move on to properly introducing the syntax and semantics of fixed-point logics, let us first recall a few basic points concerning the theory of fixed-point operators. Given a set U, an operator on the powerset of U is a mapping $F : \wp(U) \rightarrow \wp(U)$. Now a set $X \in \wp(U)$ is said to be a *prefixed-point* of F if $F(X) \subseteq X$, a *postfixed-point* of F if $F(X) \supseteq X$ and, finally, a *fixed-point* of F if $F(X) = X$. Sometimes, F does not have any fixed-point at all. For instance, the operator F defined on the powerset of the set $U = \{0\}$ by $F(\{0\}) = \emptyset$ and $F(\emptyset) = \{0\}$, never reaches a fixed-point.[5] On the other hand, some operators satisfying well-known structural properties always have a fixed-point. Some even always have a *least* fixed-point (a set $X \subseteq U$ is a least fixed-point of F if it is a fixed-point, and for every other fixed-point Y of F, $X \subseteq Y$). *Monotone operators* are of this sort. We say that an operator F is *monotone* whenever:

$$X \subseteq Y \text{ implies } F(X) \subseteq F(Y)$$

In the theory of monotone operators, the following key result is both classic and convenient[6]:

Theorem 2.1.12 (Tarski-Knaster). *Every monotone operator $F : \wp(U) \rightarrow \wp(U)$ has a least fixed-point $lfp(F)$ which can be defined as*

$$lfp(F) = \bigcap\{X|F(X) \subseteq X\}$$

.

The least fixed-point can also be constructed explicitly by considering the following sequence of approximants sets X^α, indexed by ordinals:

$$\begin{array}{lcl} X^0 & := & \emptyset \\ X^{\alpha+1} & := & F(X^\alpha) \\ X^\lambda & := & \bigcup_{\xi<\lambda} F(X^\xi) \text{ for limit ordinals } \lambda \end{array}$$

Whenever F is monotone, it is also *inductive*, i.e. the sequence above is *increasing*. Hence, it stabilizes at some step α and by Theorem 2.1.12, X^α is the least fixed-point of F.

In this thesis, we will be mainly interested in fixed-point of monotone operators. Only in Chapter 6, we will be concerned with *inflationary* fixed-points. We say that an operator F is inflationary whenever:

$$X \subseteq F(X) \text{ for all } X \subseteq U$$

[5]We take here a very simple, if not trivial, example, where F is defined on a unique set, but the situation is of course more interesting when more general definitions are given, i.e., when operators are defined for larger classes of sets.

[6]Actually, we present here this theorem in a restricted form, as it is usually stated for operators on *complete lattices*, that is, partially ordered sets $(U, <)$ where every - finite or infinite - subset of U has a greatest lower bound and a least upper bound in the ordering $<$ (see [109]). We will not need this full generality here.

Inflationary operators are inductive, hence they always reach a fixed-point, but contrary to monotone operators, they do not necessarily have a least fixed-point. A very easy way to produce an inflationary operator is to pick an arbitrary operator $F : \wp(U) \to \wp(U)$ and to consider $F' : \wp(U) \to \wp(U)$, given by $F'(X) := X \cup F(X)$. Obviously, F' is inflationary and hence inductive. The *inflationary fixed-point* $ifp(F)$ of F is given by F' and is explicitly constructed by considering the following sequence of approximants sets X^α, indexed by ordinals:

$$\begin{array}{lcl} X^0 & := & \emptyset \\ X^{\alpha+1} & := & X^\alpha \cup F(X^\alpha) \\ X^\lambda & := & \bigcup_{\xi<\lambda} F(X^\xi) \text{ for limit ordinals } \lambda \end{array}$$

As an example, consider again the operator F defined on the powerset of the set $U = \{0\}$ by $F(\{0\}) = \emptyset$ and $F(\emptyset) = \{0\}$. The sequence $F(\emptyset), F(F(\emptyset)), \ldots$ never reaches a fixed-point and keeps oscillating forever in between the value $\{0\}$ and $\emptyset$. On the other hand, the inflationary fixed-point of F is obtained by constructing the sequence $F'(\emptyset), F'(F'(\emptyset)), \ldots$, which stabilizes at the first stage of the fixed-point iteration on the value $\{0\}$.

Fixed-Point Logics

We introduced fixed-points abstractly without any reference to a logical language, but one can also consider relational structures together with operators on the powerset of their domain, or more generally on the powerset of the n-ary Cartesian product of their domain. In this context, various operators are naturally induced by corresponding formulas of matching vocabulary in a given logic. Consider for instance a formula $\varphi(X, \bar{x})$ in vocabulary σ with k free first-order variables and one free second-order variable X of arity k. On any relational σ-structure $\mathfrak{M}$ taken together with a valuation g, $\varphi(X, \bar{x})$ induces an operator $F_\varphi : \wp(dom(\mathfrak{M})^k) \to \wp(dom(\mathfrak{M})^k)$ taking a set $A \subseteq dom(\mathfrak{M})^k$ to the set $\{\bar{a} | \mathfrak{M}, g[A/X, \bar{a}/\bar{x}] \models \varphi\}$. Consider such an operator F_φ. If φ is *positive in* X (a formula is positive in X whenever X only occurs in the scope of an even number of negations), the operator F_φ is monotone and by Theorem 2.1.12, it has a least fixed-point. $\mathsf{FO(LFP)}$ extends FO with second-order variables and an explicit construct defining the least fixed point of φ.

Definition 2.1.13 (Syntax and semantics of $\mathsf{FO(LFP)}$)**.** Let X be a second-order variable of arity k, $\bar{x}$ and $\bar{y}$ two sequences of FO-variables of length k, ψ, ξ $\mathsf{FO(LFP)}$-formulas and $\varphi(\bar{x}, X)$ a $\mathsf{FO(LFP)}$-formula positive in X which, besides $\bar{x}$ and X, possibly contains other free variables, we define the set of $\mathsf{FO(LFP)}$-formulas in the following way:

$$\psi := At \mid X\bar{y} \mid \psi \vee \xi \mid \neg\psi \mid \exists x\ \psi \mid [LFP_{\bar{x},\bar{X}}\varphi(\bar{x}, X)]\bar{y}$$

We use $\forall x\varphi$, $\varphi \wedge \psi$, $\varphi \to \psi$ as shorthand in the usual way. We define the *quantifier depth* of a $\mathsf{FO(LFP)}$-formula as the maximal number of nested first-order quantifiers and fixed-point operators. Again, we interpret FO(LFP)-formulas in relational structures. The notion of assignation and the truth of atomic formulas are defined similarly as in the MSO case (except that now, we have second-order variables of arbitrary arity). The truth of compound formulas is defined by induction, with the same clauses as in FO and an additional one for the least fixed-point operator. Given a formula $\Phi := [LFP_{x,X}\varphi]y$ and a model $\mathfrak{M}$ taken together with a valuation g, we consider the operator $F_\Phi : \wp(dom(\mathfrak{M})^k) \to \wp(dom(\mathfrak{M})^k)$ and define the semantics of Φ in the following way:

$$\mathfrak{M} \models [LFP_{X,\bar{x}}\varphi(X,\bar{x})](\bar{a}) \text{ whenever } \bar{a} \text{ belongs to } lfp(F_\Phi).$$

Note that the LFP operator has a dual operator, denoted GFP (which stands for greatest fixed-point). The greatest fixed-point of a positive formula can be accessed via a similar explicit fixed-point construction as the one we presented earlier for least fixed-points. The only difference is that we now take X^0 to be the set of all possible tuples on the domain of the models, instead of the empty set. This way the value of the computed set keeps shrinking (instead of growing) until it reaches a fixed-point. For more details, we refer to [58].

In Chapter 3, we will be interested in a specific fragment of FO(LFP) called monadic least fixed-point logic (FO(LFP^1)), which extends FO with unary predicate variables and an explicit monadic least fixed-point operator. As this will be useful in that chapter, we give below another, equivalent, formulation of the semantics of FO(LFP^1), which just says that a point belongs to the least fixed-point of a formula whenever it belongs to the intersection of all its prefixed-points (note that a similar semantics could more generally be given for FO(LFP)):

$$\mathfrak{M}, g \models [LFP_{x,X}\varphi]y$$
iff
for all $A \subseteq dom(\mathfrak{M})$, if for all $a \in dom(\mathfrak{M})$, $\mathfrak{M}, g[a/x, A/X] \models \varphi(x, X)$ implies $a \in A$,
then $g(y) \in A$.

Remark 2.1.14. In practice we will often use an equivalent (less intuitive but often more convenient) rephrasing:

$$\mathfrak{M}, g \models [LFP_{x,X}\varphi]y$$
iff
for all $A \subseteq dom(\mathfrak{M})$, if $g(y) \notin A$,
then there exists $a \in dom(\mathfrak{M})$ such that $a \notin A$ and $\mathfrak{M}, g[a/x, A/X] \models \varphi(x, X)$.

⊣

As any positive formula yields a monotone operator, the Tarski-Knaster Theorem can be used in the case of FO(LFP) as a convenient tool insuring that every operator yield by a positive formula has a least fixed-point. But we already noticed that in general, an operator induced by an arbitrary formula is not necessarily monotone and needs not have any fixed point at all. One can still associate other types of fixed-point operators to arbitrary (not necessarily positive) formulas. A common method to do so is to consider the inflationary fixed-point of the formula. We already noticed that such a fixed-point need not be a fixed-point of the formula. The correct way to think about it is rather to view it as the inflationary operator associated to the formula. Unsurprisingly, the syntax and semantics of inflationary fixed-point logic FO(IFP) closely resemble those of FO(LFP) and are as follows:

Definition 2.1.15 (Syntax and semantics of FO(IFP))**.** Let X be a second-order variable of arity k, $\bar{x}$ and $\bar{y}$ two sequences of FO-variables of length k, ψ, ξ FO(IFP)-formulas and $\varphi(\bar{x}, X)$ a FO(IFP)-formula (besides $\bar{x}$ and X, $\varphi(\bar{x}, X)$ possibly contains other free variables), we define the set of FO(IFP) formulas in the following way:

$$\psi := At \mid X\bar{y} \mid \psi \vee \xi \mid \neg\psi \mid \exists x \ \psi \mid [IFP_{x,X}\varphi(\bar{x}, X)]\bar{y}$$

We use $\forall x\varphi$, $\varphi \wedge \psi$, $\varphi \to \psi$ as shorthand in the standard way. Again, we interpret FO(IFP) formulas in relational structures. The notions of *quantifier depth*, assignation and truth of atomic formulas are defined similarly as in the FO(LFP) case. The truth of compound formulas is defined by induction, with the same clauses as in FO and the following additional one:

$$\mathfrak{M} \models [IFP_{X,\bar{x}}\varphi(X, \bar{x})](\bar{a}) \text{ whenever } \bar{a} \text{ belongs to } ifp(F_\varphi).$$

We denote by FO(IFP^1) the monadic fragment of FO(IFP).

Like the LFP operator, the IFP operator has a dual, denoted DFP (which stands for deflationary fixed-point). In Chapter 6, we will be using both the GFP and the DFP operators, as well as special FO(IFP)-formulas that are called simultaneous fixed-point formulas and that we introduce now.

Definition 2.1.16 (Simultaneous fixed-point formulas)**.** Let $X_1, \ldots, X_k$ be relation variables with associated arities r_i and let $\bar{x}_1, \ldots, \bar{x}_k$ be sequence of first-order variables of associated length r_i. Simultaneous formulae are formulae of the form $\Phi(\bar{x}) := [IFPX_i : S](\bar{x})$ in vocabulary σ, where $1 \leq i \leq k$ and

$$S := \begin{cases} X_1(\bar{x}_1) \leftarrow \varphi_1(X_1, \ldots, X_k, \bar{x}_1) \\ \ldots \\ X_k(\bar{x}_k) \leftarrow \varphi_k(X_1, \ldots, X_k, \bar{x}_k) \end{cases}$$

is a system of FO(IFP)-formulas. On a structure $\mathfrak{M}$, each φ_i in S induces an operatordefina

$$F_{\varphi_i} : \wp(dom(\mathfrak{M})^{r_1}) \times \ldots \times \wp(dom(\mathfrak{M})^{r_k}) \to \wp(dom(\mathfrak{M})^{r_i})$$

which to each tuple $(A_1, \ldots, A_k)$ associates

$$\{\bar{a} | \mathfrak{M}, g[A_1/X_1, \ldots, A_k/X_k, \bar{a}/\bar{x}_i] \models \varphi_i(X_1, \ldots, X_k, \bar{x}_i)]\}$$

The stages S^α of an induction on such a system S of formulas are k-tuples of sets $(X_1^\alpha, \ldots, X_k^\alpha)$ defined as:

$$\begin{array}{lll} X_i^0 & := & \emptyset \\ X_i^{\alpha+1} & := & F_{\varphi_i}(X_1^\alpha, \ldots, X_k^\alpha) \\ X_i^\lambda & := & \bigcup_{\xi<\lambda} X^\xi \text{ for limit ordinals } \lambda \end{array}$$

For every σ-structure $\mathfrak{M}$ and any tuple $\bar{a}$ in $dom(\mathfrak{M})$, $\mathfrak{M}, g[\bar{a}/\bar{x}_i] \models \Phi$ if and only if $\bar{a} \in X_i^\infty$, where X_i^∞ denotes the i-th component of the simultaneous fixed-point of S.

The following result is convenient. Its proof is classic and can for instance be found in [104] and [97].

Theorem 2.1.17. *FO(IFP) and its extension with simultaneous fixed-point formulas have the same expressive power.*

Remark 2.1.18 (Partial fixed-point logic)**.** We will also encounter briefly partial fixed-point logic FO(PFP) in Chapter 6. It is in some sense the most general fixed-point extension of FO. Its syntax is defined as for FO(IFP), except that we write PFP for the fixed-point operator. In order to interpret a fixed-point formula $[PFP_{X,\bar{x}}\varphi](\bar{x})$ in a finite model (the semantics of FO(PFP) on infinite models is more complicated, see [97], but we will not need this full generality here), we consider the same sequence of iteration stages as in the case of FO(LFP) formulas. Whenever a fixed-point is reached (which need not be a least fixed-point), then $[PFP_{X,\bar{x}}\varphi](\bar{x})$ holds in a model under some valuation whenever the valuation assigns to $\bar{x}$ a tuple which belongs to this fixed-point. But as there are no restrictions on formulas that can be prefixed by the fixed-point operator, this sequence may not reach a fixed point at all. In this case, the formula is set as equivalent to false. For more details we refer to [58]. ⊣

Now again, like for MSO and FO(TC^1), we can observe that $\leq$ and $\preceq$ are definable by means of $<_{ch}$ and $\prec_{ns}$ in FO(LFP^1):

- $x \leq y := [LFP_{Xz} x <_{ch} z \vee \exists u(Xu \wedge u <_{ch} z)](y)$
- $x \preceq y := [LFP_{Xz} x \prec_{ns} z \vee \exists u(Xu \wedge u <_{ch} z)](y)$

Consider some finite tree and two points x and y such that x is an ancestor of y. In the case of the definition of the $\leq$ operator, the fixed-point iteration on that tree works as follows. At the first stage, the subformula $x <_{ch} z$ plays its role and all the $<_{ch}$-successors of x are put in the set X. Then, comes the turn of the second disjoint $\exists u(X(u) \wedge u <_{ch} z)$, by which at stage $i+1$, every $<_{ch}$-successor of a point which was already in X at stage i joins itself the set. The point y is incorporated in the set at stage n exactly whenever it is separated from x by $n-1$ nodes. The iteration goes on until a fixed-point is reached.

The same definitions can be given in FO(IFP) using similar formulas, obtained by simply replacing LFP operators by IFP operators To see that the procedure is correct, it is enough to look at the least and inflationary explicit fixed-point constructions: in the least fixed-point case, as the operator is monotone, for every stage α, $F(X^\alpha) \subseteq F(X^{\alpha+1})$, which entails $F(X^\alpha) = X^\alpha \cup F(X^\alpha)$. Same definitions can also of course be given in FO(PFP) by replacing LFP operators by PFP operators.

2.1.3 Expressive Power

All the extensions of FO that we presented so far incorporate recursion means, but they do so in different ways. MSO does it by allowing quantification over subsets of the domain, while FO(TC^1) and fixed-point logics are built up by introducing explicit constructs in the language to represent recursive procedures. We only focused on some particularly easy examples of recursion, which were the definition of $<_{ch}$ and $\prec_{ns}$ by means of $\leq$ and $\prec$. A classical example is also very often given: in all the strict extensions of FO that we presented, one can produce formulas which characterize among finite linear orders exactly those which have a domain of even cardinality (while such a counting power is known to be beyond reach of FO). In connection with finite games in extensive form, we will also meet other interesting examples of fixed-point computations in Chapter 6. Now are there some sorts of fixed-point computations which can be expressed in some of these logics and not in others? That is, how do all these different logics compare in expressive power? In this section, we will concentrate on expressive power on finite structures.

We can first immediately extract a few easy inclusions from the semantics. Thus, there is a straightforward recursive procedure transforming any FO(LFP^1) formula φ into a MSO formula φ' such that $\mathfrak{M}, g \models \varphi$ iff $\mathfrak{M}, g \models \varphi'$. The interesting clause is

$$([LFP_{x,X}\varphi(x,X)]y)' = \forall X(\forall x(\varphi(x,X)' \rightarrow Xx) \rightarrow Xy)$$

The other clauses are all of the same type, e.g. $(\varphi \wedge \psi)^* = (\varphi^* \wedge \psi^*)$. This procedure can easily be seen adequate by considering the semantical clause for the LFP operator.

Now there is also a straightforward recursive procedure transforming any FO(TC^1) formula φ into a FO(LFP^1) formula φ'' such that $\mathfrak{M}, g \models \varphi$ iff $\mathfrak{M}, g \models \varphi''$. The interesting clause is

$$([TC_{xy}\varphi](u,v))'' = [LFP_{Xy} y = u \vee \exists x((Xx \wedge \varphi(x,y)''))]v$$

Let us give an argument for this claim. By Proposition 2.1.11 it is enough to show that $[LFP_{Xy} y = u \vee \exists x(Xx \wedge \varphi(x,y)'')]v$ holds if and only if there is a finite φ'' path from u to v. For the right to left direction, suppose there is such a path $a_1 \dots a_n$ with $g(u) = a_1$ and $g(v) = a_n$. Then, for any subset A of the domain, we can show by induction on i that if for all a_i $(1 \leq i \leq n)$, $a_i = u \vee \exists x((Ax \wedge \varphi(x, a_i)'')$ implies $a_i \in A$, then $v \in A$, i.e., $[LFP_{Xy} y = u \vee \exists x((Xx \wedge \varphi(x,y)''))]v$ holds. Now for the left to right direction, suppose there is no such φ'' path. Consider the set A of all points that can be reached from u by a finite φ'' path. By assumption, $\neg Av$ and it holds that $\forall y((y = u \vee \exists x(Ax \wedge \varphi(x,y)'')) \rightarrow Ay)$, i.e., $\neg[LFP_{Xy} y = u \vee \exists x(Xx \wedge \varphi(x,y)'')]v$.

As already pointed out before, FO(LFP) fixed-points are expressible in a straightforward way as FO(IFP) fixed-points. FO(LFP) fixed-points can also be expressed in FO(PFP) simply by substituting every FO(LFP) fixed-point operator by a FO(PFP) fixed-point operator. But one can refine these first easy inclusions. Actually, on arbitrary structures, the following strict inclusions and equalities are known:

FO(PFP)
⊃
FO(IFP) = FO(LFP)
⊃
FO(IFP^1) MSO
⊃ ⊃
FO(LFP^1)
⊃
FO(TC^1)
⊃
FO

Figure 2.2: Extensions of FO, relative expressive power on arbtrary structures

MSO is incomparable to both FO(IFP^1) and FO(IFP). The argument is as follows. Graph 3-Colorability can be expressed in MSO, but not in FO(IFP). For more details see [120] and [48]. On the other hand, by [49], MIC (which is a fragment of $\mathsf{FO}(\mathsf{IFP}^1)$, as explained in the next Section) can express certain context free but non-regular languages which are therefore not MSO definable (as MSO can only define regular languages, see [127]). The strictness of the inclusion $\mathsf{FO} \subset \mathsf{FO}(\mathsf{TC}^1)$ follows from the previous Section, together with Chapter 3 (as $\mathsf{FO}(\mathsf{TC}^1)$ can define the class of finite trees, whereas FO cannot), but also from Proposition 2.1.7. For the strictness of the inclusions $\mathsf{FO}(\mathsf{TC}^1) \subset \mathsf{FO}(\mathsf{LFP}^1)$ and $\mathsf{FO}(\mathsf{LFP}^1) \subset \mathsf{MSO}$, we refer to [58] and [109] respectively, whereas arguments for the strictness of the inclusion $\mathsf{FO}(\mathsf{IFP}^1) \subset \mathsf{FO}(\mathsf{IFP})$ can be found in [77]. The fact that $\mathsf{FO}(\mathsf{LFP})$ and $\mathsf{FO}(\mathsf{IFP})$ have the same expressive power follows from [98] and [85]. Still, the two logics carry different intuitions and formalize computations of a different nature. Also, the equivalence in expressive power does not hold anymore whenever one restricts to the monadic fragments of these logics. And indeed, the translation procedures given in [98] and [85] to go from a $\mathsf{FO}(\mathsf{IFP})$-formula to a $\mathsf{FO}(\mathsf{LFP})$-formula involves a raise in the arity of the second order variables used in the fixed-points. Finally, the fact that the inclusion $\mathsf{FO}(\mathsf{IFP}) \subset \mathsf{FO}(\mathsf{PFP})$ is strict follows from the fact that trace equivalence can be expressed in $\mathsf{FO}(\mathsf{PFP})$ and not in $\mathsf{FO}(\mathsf{IFP})$ (see [97]), also note that on ordered structures $\mathsf{FO}(\mathsf{IFP})$ captures PTIME, whereas $\mathsf{FO}(\mathsf{PFP})$ captures PSPACE, see [104]). The situation is comparable on trees, except that $\mathsf{FO}(\mathsf{LFP}^1)$ and MSO collapse (see [126]) and that to the best of our knowledge, we are not aware of arguments establishing the strictness of the inclusion of $\mathsf{FO}(\mathsf{IFP}^1)$ into $\mathsf{FO}(\mathsf{IFP})$ and of $\mathsf{FO}(\mathsf{IFP})$ into $\mathsf{FO}(\mathsf{PFP})$:

$$\mathsf{FO} \subset_t \mathsf{FO}(\mathsf{TC}^1) \subset_t \mathsf{FO}(\mathsf{LFP}^1) =_t \mathsf{MSO} \subset_t \mathsf{FO}(\mathsf{IFP}^1) \subseteq_t \mathsf{FO}(\mathsf{IFP}) \subseteq_t \mathsf{FO}(\mathsf{PFP})$$

Moreover, on finite linear orders and linear orders of order type ω, $\mathsf{FO}(\mathsf{TC}^1)$ and $\mathsf{FO}(\mathsf{LFP}^1)$ also collapse (see [126]):

$$\mathsf{FO} \subset_l \mathsf{FO}(\mathsf{TC}^1) =_l \mathsf{FO}(\mathsf{LFP}^1) =_l \mathsf{MSO} \subset_l \mathsf{FO}(\mathsf{IFP}^1) \subseteq_l \mathsf{FO}(\mathsf{IFP}) \subseteq_l \mathsf{FO}(\mathsf{PFP})$$

2.2 The Modal Logic Perspective on Trees

2.2.1 Basic Modal Logic

Different applications call for different logics and in some contexts, modal logic provides another interesting perspective on trees. Modal languages talk about relational structures, but instead of using individual variables to quantify directly over the domain of the structure, we will see that they adopt a more local standpoint. Syntactically, they come as extensions of propositional logic by means of modal operators. As in propositional logic, we assume a countable (possibly finite) set of proposition letters $\{p_1, p_2, \ldots\}$, but additionally, we assume a finite set of modal operators. Let us start with the basic modal language, which is built up around a single unary modal operator $\Diamond$. The formulas of the basic modal language are inductively defined as follows:

$$\phi := p_i \mid \bot \mid \neg\phi \mid \phi \vee \psi \mid \Diamond\phi$$

We define a dual operator for $\Diamond$ by using $\Box$ as shorthand for $\neg\Diamond\neg$, while $\varphi \wedge \psi$ and $\varphi \rightarrow \psi$ are introduced as shorthand in the usual way. Now we can also interpret basic modal formulas in relational structures, but the procedure is a bit different from what we saw in the case of extensions of FO. We first need the notion of a *Kripke frame* (or *frame* for short), which is a pair $\mathcal{F} = (W, R)$ where W is a non-empty set, called the domain of $\mathfrak{M}$ and R is a binary relation over W. We call the elements of W *nodes*, *points* or *states*. A frame for the basic modal language is thus simply a relational structure containing one single binary relation. A frame does not incorporate any information about the value of proposition letters and we need to add additional information about it in order to be able to interpret all modal formulas. A *Kripke model* (or simply, *model*) for the basic modal language carries this information. It is a pair $\mathfrak{M} = (\mathcal{F}, V)$ where $\mathcal{F} = (W, R)$ is a frame and V is a *valuation* function, which is a map assigning to each proposition letter p_i a set $V(p_i) \subseteq W$. We say that a model $\mathfrak{M} = (\mathcal{F}, V)$ is *based on the frame* $\mathcal{F}$. Like frames, models can be seen as (richer) relational structures of the form $(W, R, V(p_1), V(p_2), \ldots)$, so that from a model-theoretic point of view, a model *expands*[7] the frame on which it is based with additional unary relations. Now how does one interpret the basic modal language in Kripke models? An important specificity of modal formulas is that they are interpreted locally *at a given state* in a model. Consider a model $\mathfrak{M} = (W, R, V)$ and a point $w \in W$, we call $\mathfrak{M}, w$ a pointed model and we inductively define the truth, or satisfaction of a modal formula in $\mathfrak{M}$ at w in the following way:

$\mathfrak{M}, w \models p$	iff	$w \in V(p)$
$\mathfrak{M}, w \models \neg\varphi$	iff	$\mathfrak{M}, w \not\models \phi$
$\mathfrak{M}, w \models \varphi \vee \psi$	iff	$\mathfrak{M}, w \models \phi$ or $\mathfrak{M} \models \psi$
$\mathfrak{M}, w \models \Diamond\varphi$	iff	there exists $v \in W$ such that $R(w, v)$ and $\mathfrak{M}, v \models \phi$

Besides this local notion of satisfaction, there is also a global notion of satisfaction and we say that a formula is *globally satisfied* or *valid* in a model $\mathfrak{M}$ if it is satisfied at all points in $\mathfrak{M}$.

Note that formulas can also be interpreted in bare frames (see [26]). The perspective leads to an interesting area of modal logic, where the focus is on *frame definability* (see in particular the Goldblat Thomasson Theorem in [26] and the similar results obtained for the μ-calculus - extension of basic modal logic that we will introduce shortly - on trees in [66]). We will nevertheless restrict in this thesis to model satisfaction and focus on definability of classes of models. This

[7]We will formally introduce the notion of expansion later on in the thesis.

being said, we will always assume that the models we are interested in are all based on frames belonging to a specific, given, class of frames. We will be mainly concerned with two classes of trees already mentioned at the beginning of this chapter: finite linear orders and linear orders of order type ω. We will also at some point discuss the case of finite unranked trees.

A natural question again arises: how far does basic modal logic's expressive power goes? An answer can be given via a notion of invariance called *bisimulation*.

Definition 2.2.1 (Bisimulation). Let $\mathfrak{M} = (W, R, V)$ and $\mathfrak{M}' = (W', R', V')$ be two models.

A non-empty binary relation $Z \subseteq W \times W'$ is called a *bisimulation* between $\mathfrak{M}$ and $\mathfrak{M}'$ if the following conditions are satisfied:

(i) If wZw' then w and w' satisfy the same proposition letters.

(ii) If wZw' and $R(w, v)$, then there exists $v' \in W'$ such that zZv' and $R'(w', v')$ (the *forth* condition)

(iii) The converse of (ii): if wZw' and $R'(w', v')$, then there exists $v \in W$ such that zZv' and $R(w, v)$ (the *back* condition)

When Z is a bisimulation linking two states w in $\mathfrak{M}$ and w' in $\mathfrak{M}'$, we say that w and w' are bisimilar. A relation Z is a bisimulation between two pointed models $\mathfrak{M}, w$ and $\mathfrak{M}', w'$ whenever Z is a bisimulation between $\mathfrak{M}$ and $\mathfrak{M}'$ and $(w, w') \in Z$.

We say that a logic is invariant under bisimulation whenever it cannot distinguish between bisimilar models, i.e., whenever every two bisimilar points always satisfy exactly the same formulas. The following is folklore:

Proposition 2.2.2. *Basic modal logic is invariant under bisimulation.*

Let us take one simple example. Consider two modal structures $\mathfrak{M}$ and $\mathfrak{N}$ in vocabulary $\{p\}$ such that the domain of $\mathfrak{M}$ contains one single reflexive node labelled by p, while $\mathfrak{N}$ is a countable linear order isomorphic to $\mathbb{N}$ in which every node is labelled by p: $\mathfrak{M}$ and $\mathfrak{N}$ are bisimilar. This entails that, as FO, basic modal logic is not expressive enough to characterize the class of finite trees. But this also shows that it is far less expressive than FO, in which these two structures could easily be told apart (to do so, it is enough to notice that the FO-formula $\exists x \exists y\ x \neq y$ is satisfied in one structure and not in the other, or, equivalently, that Spoiler has a winning strategy in two rounds in the corresponding game on the two structures). FO is indeed not invariant under bisimulation. But then, how do the expressive power of basic modal logic and FO relate? Are they comparable and can FO be considered as an extension of basic modal logic? The answer to this question is yes and we will now explain how to translate basic modal logic into FO. The procedure is called the *standard translation*. First of all, we need a way to relate the modal language to the FO language. Hence we start by specifying the *correspondence language* in which we will translate our modal formulas. To each proposition letter p_i we will simply associate a unary predicate P_i and to $\Diamond$ we will associate a binary predicate R. We can now define the standard translation.

Definition 2.2.3 (Standard Translation)**.** Let x be a first-order variable. The *standard translation* ST_x taking modal formulas to FO-formulas is defined as follows:

$$\begin{array}{lcl} ST_x(p) & = & P(x) \\ ST_x(\neg\varphi) & = & \neg ST_x(\varphi) \\ ST_x(\varphi \vee \varphi) & = & ST_x(\varphi) \vee ST_x(\psi) \\ ST_x(\Diamond\varphi) & = & \exists y(R(x,y) \wedge ST_x(\varphi)) \end{array}$$

The following is folklore too and can be shown via an easy induction:

Proposition 2.2.4 (Local and global correspondence on models)**.** *Let φ be a modal formula. Then:*

(i) For all $\mathfrak{M}$, all valuation g on $\mathfrak{M}$ and all states w of $\mathfrak{M}$: $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M}, g[w/x] \models ST_x(\varphi)$.

(ii) For all $\mathfrak{M}$: $\mathfrak{M} \models \varphi$ iff $\mathfrak{M} \models \forall x ST_x(\varphi)$.

This yields that modal logic is some fragment of FO, but this does not show precisely to which fragment it corresponds. It actually turns out that this fragment can be elegantly characterized as a *bisimulation invariant fragment*. Let us first give a precise definition to what it means for a FO-formula in one free variable to be bisimulation invariant.

Definition 2.2.5 (Bisimulation invariance for FO)**.** A FO-formula $\varphi(x)$ is *invariant for bisimulation* if for all models $\mathfrak{M}$ and $\mathfrak{N}$, and all states w in $\mathfrak{M}$, v in $\mathfrak{N}$, and all bisimulations Z between $\mathfrak{M}$ and $\mathfrak{N}$ such that wZv, we have $\mathfrak{M}, g[x/w] \models \varphi(x)$ iff $\mathfrak{N}, g[x/v] \models \varphi(x)$.

Now the following result is classic:

Theorem 2.2.6 (Van Benthem Characterization)**.** *Let $\varphi(x)$ be a FO-formula in the correspondence language. Then φ is invariant for bisimulation iff it is equivalent to the standard translation of a modal formula.*

The proof of Theorem 2.2.6 is far from trivial, but we do not need to recall it here and we just refer to [26]. It has many implications, but let us point out one in particular which is of interest to us. In order to establish his characterization result, Johan van Benthem came up with the notion of bisimulation, but it turned out that this very notion was later on independently identified in computer science as an important notion of equivalence for processes, capturing in some sense *behavioral equivalence.* Temporal logics and modal fixed-point logics extend basic modal logic while retaining bisimulation invariance. Temporal logic was originally used in philosophical logic to study reasoning about time, but today it is widely used in verification of programs and processes to study the behavior of systems evolving over time. Modal fixed-point logics are extensions of temporal logics that are also widely used in that perspective. We will now introduce the two frameworks.

2.2.2 Temporal and Fixed-Point Extensions of Basic Modal Logic

With only the basic modal language at hand, node-labelled trees are usually seen as basic Kripke models in the following way. The tree shaped models considered are unranked and the binary relation underlying the modal operator $\Diamond$ plays the role of the relation corresponding to the $<_{ch}$ predicate that we considered in the case of usual relational structures. As the descendant $\leq$ relation is not definable in FO using the $<_{ch}$ relation, here too a modality corresponding to that relation cannot be defined. Hence, whenever the focus is on trees, one also naturally looks for ways to extend the basic modal language. Temporal logic takes this path by allowing a variety of alternative modal operators. Another line is also possible: instead of adding modal operators, one can extend the language by means of fixed-point operators. As regards expressive power, the latter perspective is quite powerful and we will see that the temporal logics we will be concerned with can all be seen as fragments of the modal fixed-point logics we will also be introducing.

Temporal Logic

Considering classes of Kripke models, one often wishes to focus on a specific class of frames. Attention can for instance be restricted to transitive relations, equivalence relations or even to a collection of relations of a certain type (and those can even be of any given arity). For now, we introduced only the basic modal language, but many modal operators can also simultaneously be defined, all based on a different relation defined on the domain of the structure. This is the case in temporal logic, where frames are also supposed to reflect the nature of time. Many alternative options have been considered (see [59]). Time could be bounded (towards the past, towards the future), or not; discrete, or not; dense, or not; linear, or branching... And actually, many other features and refinements have been considered as well. Here we will mainly restrict to one very specific class of frame. We will focus on linear orders of order type ω and we will consider linear time temporal logic LTL, as well as some of its syntactic fragments. Note that this does not necessarily has to imply a determinist vision of time, as it can also be seen as reflecting an a posteriori view on some course of events.

Propositional Linear Temporal Logic In the context of temporal logic, frames are called *flows of time* and actually there is a slight stylistic difference here with the usual technical apparatus of modal logic, because we do not explicitly assume one specific accessibility relation per modality (while we could still equivalently do so). We interpret temporal formulas in structures consisting of a set of worlds (or, time points), a binary relation intuitively representing temporal precedence, and a valuation of proposition letters. A *flow of time* is a structure $\mathcal{T} = (W, <)$, where W is a non-empty set of worlds and $<$ is a binary relation on W. In the context of LTL, one usually focuses on $\mathbf{T}_\omega$, the class of linear orders of order type ω, i.e., frames $(W, <)$ that are isomorphic to $(\mathbb{N}, <)$, where $\mathbb{N}$ is the set of natural numbers with the natural ordering. We freely use $\leq$ to denote the reflexive closure of $<$. We now introduce the syntax and semantics of LTL, following the terminology of [59].

Definition 2.2.7 (LTL)**.** Let σ be a propositional signature. The set of formulas $\mathsf{LTL}[\sigma]$ is defined inductively, as follows:

$$\varphi, \psi := At \mid \top \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \varphi \vee \psi \mid \mathsf{X}\varphi \mid \mathsf{F}\varphi \mid \mathsf{F}^{<}\varphi \mid \varphi\mathsf{U}\psi$$

where $At \in \sigma$. We use G and $\mathsf{G}^{<}$ as shorthand for respectively $\neg\mathsf{F}\neg$ and $\neg\mathsf{F}^{<}\neg$. The relation $\models_{\mathsf{LTL}}$ between LTL-formulas and structures $(\mathcal{T}, V, w)$ is defined as follows (we only list the clauses of the temporal operators, the others are as in the case of classical propositional logic):

- $(\mathcal{T}, V, w) \models_{\mathsf{LTL}} \mathsf{X}\varphi$ iff there exists w' such that $w < w'$, there is no w'' such that $w < w'' < w'$ and $(\mathcal{T}, V, w') \models \varphi$
- $(\mathcal{T}, V, w) \models_{\mathsf{LTL}} \mathsf{F}\varphi$ iff there exists w' such that $w \leq w'$ and $(\mathcal{T}, V, w') \models \varphi$
- $(\mathcal{T}, V, w) \models_{\mathsf{LTL}} \mathsf{F}^{<}\varphi$ iff there exists w' such that $w < w'$ and $(\mathcal{T}, V, w') \models \varphi$
- $(\mathcal{T}, V, w) \models_{\mathsf{LTL}} \varphi\mathsf{U}\psi$ iff there exists w' such that $w \leq w'$, $(\mathcal{T}, V, w') \models \psi$ and for all w'' such that $w \leq w'' < w'$, $(\mathcal{T}, V, w'') \models \varphi$

While the above definition in principle applies to arbitrary pointed structures, the intended semantics will be in terms of structures based on frames in $\mathbf{T}_\omega$.

We define fragments $\mathsf{LTL}(\mathcal{O})$ of LTL by allowing in their syntax only a subset $\mathcal{O} \subseteq \{\mathsf{X}, \mathsf{F}^{<}, \mathsf{F}, \mathsf{U}\}$ of temporal operators. Note that $\mathsf{LTL}(\mathsf{U},\mathsf{X})$ has the same expressive power as LTL, because $\mathsf{F}\varphi$ can be defined as $\top\mathsf{U}\varphi$ and $\mathsf{F}^{<}\varphi$ as $\mathsf{X}(\top\mathsf{U}\varphi)$. The same holds of $\mathsf{LTL}(\mathsf{F}^{<},\mathsf{X})$ and $\mathsf{LTL}(\mathsf{F}^{<},\mathsf{X},\mathsf{F})$, as $\mathsf{F}\varphi$ can be defined as $\varphi \vee \mathsf{F}^{<}\varphi$. Nevertheless, it is known (see [93]), that $\varphi\mathsf{U}\psi$ can be defined neither in $\mathsf{LTL}(\mathsf{F})$ nor in $\mathsf{LTL}(\mathsf{F}^{<},\mathsf{X})$. Also $\mathsf{X}\varphi$ and $\mathsf{F}^{<}\varphi$ can be defined neither in $\mathsf{LTL}(\mathsf{U})$ nor in $\mathsf{LTL}(\mathsf{F})$ (we will see why in Chapter 4, once we introduce the notion of stutter-invariance).

Modal Fixed-Point Logics

In the previous section, we discussed different types of fixed-point operators and introduced some extensions of FO obtained by allowing corresponding fixed-point constructs in the language. Similarly, one can extend the basic modal language by means of least, inflationary or partial fixed-point operators. This gives rise, respectively, to the modal μ-calculus, the modal inflationary calculus MIC and the modal partial iteration calculus MPC. We will be mainly concerned with the μ-calculus interpreted over linear orders of order type ω (in Chapters 4 and 5), but in connection with the ι-calculus that we will introduce in Chapter 6, we will also shortly encounter MIC and MPC on finite trees.

Linear Time μ-calculus A way of increasing the expressive power of LTL is to add a least fixed-point operator to the basic modal language. Basically, this can be seen as the modal counterpart to the extension of FO by $\mathsf{FO(LFP)}$. On arbitrary structures, adding to LTL the least fixed-point operator, referred to as μ, gives the μ-calculus (see for instance [46]). We will here restrict the class of intended structures for the μ-calculus to those based on $\mathbf{T}_w$ and we will call the resulting restricted calculus linear time μ-calculus $\mu\mathsf{TL}$ (see for instance [92]).

Definition 2.2.8 ($\mu\mathsf{TL}$)**.** Let σ be a propositional signature, and $\mathcal{V} = \{x_1, x_2, \ldots\}$ be a disjoint countably infinite stock of *propositional variables*. The set of $\mu\mathsf{TL}$-formulas in vocabulary σ is generated by the following inductive definition:

$$\varphi, \psi, \xi := At \mid \top \mid \neg\varphi \mid \varphi \vee \psi \mid \mathsf{X}\varphi \mid \mathsf{F}\varphi \mid \mathsf{F}^{<}\varphi \mid \varphi \mathsf{U} \psi \mid \mu x_i.\xi$$

where $At \in \sigma \cup \mathcal{V}$ and, in the last clause, x_i occurs only positively in ξ (i.e., within the scope of an even number of negations). We will use $\nu x_i.\varphi(x_i)$ as shorthand for $\neg\mu x_i.\neg\varphi(\neg x_i)$. We will also use $\varphi \wedge \psi$ and $\varphi \rightarrow \psi$ as shorthand in the usual way. The satisfaction relation is defined between $\mu\mathsf{TL}$-formulas and pointed structures $(\mathcal{T}, V, w)$ where $\mathcal{T} \in \mathbf{T}_\omega$. In order to define it inductively, we use an auxiliary assignment to interpret formulas with free variables. The assignment g maps each free variable of φ to a set of worlds. We let $g[x \mapsto A]$ be the assignment which differ from g only by assigning A to x and we only recall:

- $(\mathcal{T}, V, w) \models x_i\ [g]$ iff $w \in g(x_i)$
- $(\mathcal{T}, V, w) \models \mu x.\varphi\ [g]$ iff $\forall A \subseteq W$, if $\{v \mid (\mathcal{T}, V, v) \models \varphi\ [g[x \mapsto A]]\} \subseteq A$, then $w \in A$

To understand this, consider a $\mu\mathsf{TL}$-formula $\varphi(x)$ and a structure $(\mathcal{T}, V, w)$ together with a valuation g. This formula induces an operator F^φ taking a set $A \subseteq W$ to the set $\{v : (\mathcal{T}, V, v) \models_{\mu\mathsf{TL}} \varphi(x)\ g[x \mapsto A]\}$. $\mu\mathsf{TL}$ is concerned with least fixed-points of such operators. If $\varphi(x)$ is positive in x, the operator F^φ is monotone. We already noticed that monotone operators F^φ always have a least fixed-point, defined as the intersection of all their prefixed-points: $\bigcap\{A \subseteq W : \{v : (\mathcal{T}, V, v) \models \varphi(x)\ g[x \mapsto A]\} \subseteq A\}$ (see [5]). The formula $\mu x.\varphi(x)$ denotes this least fixed-point.

As before, we define a fragment $\mu\mathsf{TL}(\mathcal{O})$ for each $\mathcal{O} \subseteq \{\mathsf{X}, \mathsf{F}^<, \mathsf{F}, \mathsf{U}\}$. $\mu\mathsf{TL}(\mathsf{X})$ already as the full expressive power of TL, since $\varphi\mathsf{U}\psi$ can be defined by $\mu y.(\psi \vee (\varphi \wedge \mathsf{X}y))$, $\mathsf{F}^<\varphi$ by $\mu y.(X\varphi \vee \mathsf{X}y)$ and $\mathsf{F}\varphi$ by $\mu y.(\varphi \vee \mathsf{X}y)$. Another fragment of particular interest will be $\mu\mathsf{TL}(\mathsf{U})$. In $\mu\mathsf{TL}(\mathsf{U})$, we can still define $\mathsf{F}\varphi$ in the usual way by $\top\mathsf{U}\varphi$, but we will see in Chapter 4 that $\mathsf{X}\varphi$ and $\mathsf{F}^<\varphi$ are not definable.

Other modal fixed-point logics Inspired by FO(IFP) and FO(PFP), that we already introduced when we dealt with extensions of first-order logic, two other modal fixed-point logics will be of interest to us in Chapter 6. The first one is called the modal iteration calculus MIC and the second one is called the partial iteration calculus MPC. We give here only a very rough presentation of the two frameworks and for more details we refer to [97]. Let us also note that the semantics we give for MPC is restricted to finite models.

Definition 2.2.9 (Modal Inflationary Calculus)**.** MIC extends the basic modal language by the following rule. If $\varphi_1, \ldots, \varphi_k$ are formulae of MIC, and $x_1, \ldots, x_k$ are propositional variables, then

$$S := \begin{cases} x_1 \leftarrow \phi_1(x_1, \ldots, x_k) \\ \ldots \\ x_k \leftarrow \phi_k(x_1, \ldots, x_k) \end{cases}$$

is a system of rules, and for every $1 \leq i \leq k$, $(ifp\ x_i : S)$ is a MIC-formula. The semantics of $(ifp\ x_i : S)$ is defined in a way similar as for the simultaneous fixed-point FO(IFP)-formulas, except that we now require that $(ifp\ x_i : S)$ is true at a state if and only if that state belongs to the fixed-point of x_i which is reached via simultaneous induction.

Definition 2.2.10 (Modal Partial Iteration Calculus)**.** The syntax of MPC is similar to the syntax of MIC, except that we write pfp for the fixed-point operator. We define its semantics only for finite models and we require that $(pfp\ x_i : S)$ is true at a state if and only if that state belongs to the partial fixed-point of x_i which is reached via simultaneous induction.

Let us point out that both MIC and MPC are genuine modal logics, in the sense that they are invariant under bisimulation (c.f. [97]).

2.2.3 Expressive Power

In this section, we will mainly focus on linear time structures (as they will be our main concern in this thesis whenever we will be dealing with modal formalisms).

For the case of branching temporal logics and for the μ-calculus on trees, we refer the reader to [11].

The standard translation We already noticed that, as regards expressive power, there is a regular increase when one goes from fragments of LTL to the linear-time μ-calculus and finally, to MIC and MPC interpreted over linear orders. Additionally, as basic modal logic, these logics also translate to extensions of FO in a natural way. One can consider there suitable "standard translations" in the following way. The main difference is that now, in spite of considering only propositional variables and the $\Diamond$ modality (which corresponds in the case of LTL to X), the correspondence language also sends propositional variables x_i to set variables X_i. Also recall that in the case of MPC and FO(PFP), we restrict attention to finite models.

$$
\begin{array}{lcl}
ST_x(\mathsf{F}\varphi) & = & \exists y(x < y \wedge ST_x(\varphi)) \\
ST_x(\mathsf{F}^{<}\varphi) & = & \exists y(x \leq y \wedge ST_x(\varphi)) \\
ST_x(\varphi \mathsf{U} \psi) & = & \exists y(x \leq y \wedge ST_y(\psi) \wedge \forall z((x \leq z \wedge z < y) \rightarrow ST_z(\varphi))) \\
ST_x(\mathsf{X}\varphi) & = & \exists y(x < y \wedge \neg \exists z x < z < y \wedge ST_y(\varphi)) \\
ST_x(\mu x.\varphi(x_i)) & = & [LFP_{X_i,x} ST_x(\varphi)](x) \\
ST_x((ifp\ x_i : S)) & = & [IFP_{X_i,x} ST_x(S)](x) \\
ST_x((pfp\ x_i : S)) & = & [PFP_{X_i,x} ST_x(S)](x)
\end{array}
$$

Note that we translated μ-formulas in FO(LFP^1), but we could equivalently have translated them in MSO (remember that in the previous section, we remarked that these two logics were equi-expressive on trees). Let us also point out that the translations that we give for fixed-point formulas also make sense on trees for formulas containing as modal operators only the X-operator (which is simply interpreted as the standard $\Diamond$-operator of the basic modal language). We will see now, that in the style of the Van Benthem characterization Theorem, refined characterization results exist.

Characterization results It is quite straightforward to go from the extensions of basic modal logic we presented here to extensions of FO. The converse direction is far less easy. The first result of this type is due to Kamp ([93]) and dates back from 1968. Kamp originally formulated his theorem for a more general class of linear orders, called *Dedekind complete* linear orders and he did not consider satisfaction at the root, but satisfaction in general. Hence he considered an extension of LTL with past tense operators. It is enough in our perspective (especially Chapter 4) to consider satisfaction at the root of the linear order. This perspective is called "initial semantics" and enables the following alternative version of Kamp's Teorem (see [67]):

Theorem 2.2.11 (Kamp's Theorem for initial semantics)**.** LTL *has exactly the same expressive power as* FO *over linear orders of order type ω (with monadic vocabularies and a binary predicate for the order) with respect to initial semantics.*

Kamp's Theorem is a surprising result and it has at first been interpreted as a sign that LTL was not a useful formalism (see [131]). Time has shown this first impression to be wrong and LTL is now ubiquitous in the field of verification. The point is that expressive power is not everything and LTL and FO have different computational properties, as LTL reasoning is in PSPACE, while FO reasoning is non-elementary (see [131]). We will not be dealing with complexity in the thesis, but the point is still worth pointing out.

Whenever one looks a bit closer at Kamp's Theorem, the statement can also be slightly refined: FO can be replaced by FO^3, which is the three variables fragment of first-order logic. In that line, another result of this type has been given in 1998 by Vardi, Etessami and Wilke ([62]):

Theorem 2.2.12 (Vardi, Etessami, Wilke)**.** *Unary* LTL *(i.e.,* LTL *without the Until operator) has exactly the same expressive power as* FO^2 *(i.e.,* FO *with two variables) over linear orders of order type* ω *with respect to initial semantics.*

In the same line again, Janin and Walukiewicz identified in 1996 ([91]) the μ-calculus as the bisimulation invariant fragment of MSO:

Theorem 2.2.13 (Janin, Walukiewicz)**.** *An MSO-formula is invariant under bisimulation if, and only if, it is equivalent to a* L_μ*-formula.*

This holds on arbitrary structures and also on trees. But on the linear orders we are interested in, bisimulations are trivial and the following holds with respect to initial semantics (see [4]):

Theorem 2.2.14. *MSO on linear orders of order type* ω *and the linear-time* μ*-calculus have the same expressive power with respect to initial semantics.*

Let us note that the expressive power of MIC and MPC go far beyond MSO (see [97]). Additionally for these logics there is no known characterization result similar to the ones we just recalled here. What is known is simply that MIC is contained in the bisimulation-invariant fragment of $\mathsf{FO}(\mathsf{IFP}^1)$, even if simultaneous fixed-points are not allowed and that MPC is contained in the bisimulation-invariant fragment of FO(PFP) (see [97]).

It follows from these results that in some sense, we will actually be talking about extensions of FO all the time. Whenever we will be talking about modal logic, as we will mostly restrict to linear orders, one could indeed say that we are in fact talking about extensions of FO *through* modal logic, and not even *modulo bisimulation.* (Except when we will be mentioning modal logic on finite trees, but this will not be the central issue.) In fact, besides the complexity point of view, that we already mentioned in passing, the modal perspective has other advantages. We will see in Chapter 4 and 5 that characterizing the stutter-invariant fragment of MSO and that characterizing the fragments of MSO which satisfy interpolation on linear orders is particularly easy whenever one goes via modal logic.

2.3 Tools and Concepts

We introduced two families of logics, as extensions of respectively, basic modal logic and first-order logic. In order to deal with them, we will be mainly using model-theoretic techniques, but as we focus on trees, the automata-theoretic framework will also lightly come into play. In this thesis, we will be interested in three kinds of things: definability, complete axiomatizations and interpolation. We already gave an account of definability and we reserve the issue of interpolation for Chapter 4, so let us now shortly discuss two closely related notions, decidability and complete axiomatizations.

2.3.1 Decidability

Now that we introduced all these logics, gave their syntax and semantics and discussed their relative expressive power, we can ask one other natural question. Given one of these logics, can we determine an effective procedure which, given a formula φ, allows to decide whether φ is satisfiable? This problem is called the *satisfiability problem* of the logic and - whenever the logic is closed under negation - it can equivalently be formulated as its *validity problem* (can we decide whether φ is valid?). Whenever such a procedure exists, we say that the logic is *decidable*. Historically, this problem became prominent in 1928, when David Hilbert asked a more general and ambitious question. He was interested in whether the validity of any mathematical statement could be shown to be decidable. He was believing that this could in some sense be the case and he had a vast program consisting in reducing mathematics to logic. One of the first step in that program would have been to show that FO (and hence the minimal part of mathematics it can formalize) is decidable (it is known as *the classical decision problem*, see [31]). Hilbert's hopes were soon dashed. In 1936 and 1937, by giving a precise mathematical meaning to the notion of *effective procedure*, both Church and Turing published results implying that FO was not decidable ([43], [42] and [129]). The classical decision problem was then reconsidered and the focus shifted on decidable fragments of FO. In that direction, Löwenheim had for instance already proven in 1915 that the fragment of FO where only unary predicates are allowed is decidable. Another related line of research focuses on decidable *FO-theories*. Gurevich for instance showed in 1964 that the FO theory of ordered abelian groups is decidable (see [82, 83]). He also came later to considering decidable theories formulated in extensions of FO with quantification over some specific classes of subsets of the domain (see [84] for a discussion). In the case where quantification is allowed over the whole powerset of the domain, this amounts to considering decidable MSO-theories. Monadic theories are a good source of theories that are both expressive and manageable. Consequently, a lot of attention has been given to them in the field of theoretical computer science, where there is a special focus on the determination of effective methods. It is interesting to put the development of computer science in perspective with the formulation, discussion and reformulation of the classical decision problem. This logical problem actually contributed to the elaboration of concepts which form now the backbone of computability theory. Also, many "industrial logics" were designed out of logics first investigated in that context (again, see [131]). Indeed, having practical decision procedures for classes of logical formulas is for instance very relevant in the context of formal verification or in the context of database theory. We will now introduce the *Büchi Theorem*, a logical result which is classic in theoretical computer science and which implies that MSO is decidable on linear orders of order type ω. One of the main tools in logics on such linear orders, or more generally on trees, is given by automata-theory. Let us now introduce very shortly some basic notions from this framework which are needed for the formulation of the Büchi Theorem. It is also important to keep them in mind in order to have a good picture of the landscape of logics on trees.

Definition 2.3.1 (ω-word)**.** Let Σ be a finite alphabet. We call *ω-word* any string of letters of length ω over Σ and we represent it by a function $\alpha : \omega \to \Sigma$ assigning to each position a letter. We call ω-word language any set of ω-words over Σ.

Note that we will be using later on a similar notion of *finite word*. Given an alphabet Σ, a finite word is any finite string of letters over Σ and a finite word language is a set of finite words over Σ.

Definition 2.3.2 (Büchi-automata)**.** To specify recognizable ω-word languages, we refer to nondeterministic ω-automata over a finite alphabet Σ, which are presented in the form $\mathcal{A} = (Q, \Sigma, q_0, \Delta, Acc)$ where Q is a finite set of states, q_0 the initial state, $\Delta \subseteq Q \times \Sigma \times Q$ the transition relation, and Acc the acceptance component. A run of $\mathcal{A}$ on a given input ω-word $\alpha = \alpha(0)\alpha(1)\dots$ with $\alpha(i) \in \Sigma$ is a sequence $\rho = \rho(0)\rho(1)\dots \in Q^\omega$ such that $\rho(0) = q_0$ and $(\rho(i), \alpha(i), \rho(i+1)) \in \Delta$ for $i \geq 0$.

Now we write $\exists^\omega$ for the quantifier "there exists infinitely many" and consider the set

$$Inf(\rho) = \{q \in Q|\ \exists^\omega i\ \rho(i) = q\}$$

For a run to be accepting, the Büchi condition requires that some *final state* in a set $F \subseteq Q$ occurs infinitely often in ρ, that is $Inf(\rho) \cap F \neq \emptyset$. An ω-word α is accepted by $\mathcal{A}$ if there is an accepting run of $\mathcal{A}$ on α. The language $L(\mathcal{A})$ recognized by $\mathcal{A}$ is the set of all ω-words over Σ accepted by $\mathcal{A}$.

Note that other acceptance conditions exist for automata on ω-words (for more details see [127]). There are also corresponding notions of automata on finite words and on other classes of trees. In order to keep things simple, we will not develop them here and we refer the reader to [127]. One can go back and forth from ω-words to linear orders of order type ω and here too one can have a notion of "correspondence language" (for more details we refer to Chapter 4). Hence, talking about linear orders of order type ω or talking about ω-words amounts to adopting two different perspectives on one and the same class of structures. Moreover, the Büchi Theorem establishes that characterizing classes of such structures via MSO or via Büchi-automata is actually equivalent:

Theorem 2.3.3 (Büchi [35])**.** *A language of ω-words is recognizable by a Büchi-automaton if and only if it is MSO-definable. Both conversions, from automata to formulas, and vice versa, are effective.*

Theorem 2.3.3 has the following important corollary:

Corollary 2.3.4. *MSO is decidable on linear orders of order type ω.*

Proof. In order to check if a MSO-formula φ is satisfiable on linear orders of order type ω, we first convert it to an equivalent Büchi-automata $\mathcal{A}$. Then, we check whether $L(\mathcal{A})$ is non-empty (this is a decidable problem, see [127]). By Theorem 2.3.3, if $L(\mathcal{A}) \neq \emptyset$, then φ is satisfiable, otherwise it is not satisfiable. □

A version of Theorem 2.3.3 was first shown for finite words by Büchi in 1960 and by Elgot in 1961. Another version was shown for finite binary trees by Thatcher and Wright in 1968, Doner in 1970 and for infinite binary trees by Rabin in 1969 (see [127]). This also implies (via the decidability of the emptiness problem of the automata involved) that MSO is decidable on these classes of structures. The Büchi and the Rabin Theorem (which are also often referred to as establishing the decidability of the monadic theories of one successor $S1S$ and two successors $S2S$, respectively) are key results which brought pioneering methods in the field. In particular, many theories and many modal and temporal logics (among which the μ-calculus) have been shown to be decidable by an interpretation in $S2S$.

Theorem 2.3.5 (Rabin [127])**.** *MSO is decidable on infinite binary trees.*

Moreover, we already noted that unranked trees can be encoded as binary trees, so MSO is also decidable over finite unranked trees and over infinite unranked trees. Additionally, these results imply that all the fragments of MSO we considered are decidable on these classes of structures (see also [134] for an investigation of fragments of temporal logic as characterizing classes of regular languages). On the other hand, MIC is known to be undecidable already on the class of finite words (see for instance [1]), which entails that MPC, FO(IFP^1), FO(IFP) and FO(LFP) are also undecidable there. This clearly sets apart these formalisms from the other logics we will be dealing with. Actually, going beyond the expressive power of MSO while retaining decidability on trees seems to be delicate, while it is known to be possible in some very restricted cases (see for instance [47]). Enriching tree structures has also most of the time the same effect.

For instance, introducing a relation of "simultaneity"[8] into the infinite binary tree of $S2S$ makes MSO undecidable (see [102] and see [24] more generally, for a survey on the topic of decidability on trees).

2.3.2 Complete Axiomatizations

Another property which is related to decidability is *complete axiomatizability*, which entails *semi-decidability*. A theory is semi-decidable if there is an effective method which, given an arbitrary formula, will always tell correctly when the formula is in the theory, but may give no answer at all when the formula is not in the theory. In this thesis, we will provide some complete axomatizations for different theories of classes of trees. An axiomatization is a set of special formulas called *axioms*, which come together with a set of *rules* that can be used in order to derive proofs of other formulas from the axioms. An axiomatization is *sound* whenever only valid formulas can be derived in the system, i.e., whenever every axiom is a valid formula and whenever every application of a rule to valid formulas produces only a valid formula. Soundness is an essential property that one expect an axiomatization to satisfy, but it is usually an easy property to check. Another important property that is expected from a "good" axiomatization is that it is *complete*, i.e., that every valid formula can be proved using the axiomatization. Completeness results are interesting for different reasons. Axioms and rules are usually chosen because they are simple and easy to understand. Hence, in some sense, a logic that is completely axiomatized using intuitive axioms and rules is well-understood. Comparing complete axiom systems is also a good way to understand differences and common points among families of logics, witness the classification of normal modal logics established by reference to their axiom systems. Finally, completeness results are also interesting in the way they establish a link in between semantic validity and syntactic provability. Verifying directly the validity of a formula is indeed generally non-constructive, whereas verifying whether a sequence of formulas is a proof can be done very easily.

Let us now give an example of a complete axiomatization with the simple case of basic modal logic. The proof system for basic ML is called K and it is designed to produce all the ML theorems. A K proof is a finite sequence of formulas, all prefixed by $\vdash$, each of which is an *axiom*, or follows from one or more earlier items in the sequence by applying a *rule of proof*. The axioms and rules of K are given in Figure 2.3.

A formula φ is *K-provable* if $\vdash \varphi$ occurs in some K-proof and it is *K-consistent* if its negation is not K-provable. A set of ML-formulas Γ is K-consistent if there does not exist $\psi_1, \ldots, \psi_k \in \Gamma$ such that $(\psi_1 \wedge \ldots \wedge \psi_k) \rightarrow \bot$ is K-provable. The completeness Theorem for K establishes that in the context of K, the relations $\models$

[8]This relation is also called the "equi-level relation" and is an equivalence relation that holds between two points if they have the same distance from the root.

Taut.	$\vdash \varphi$ where φ is a tautology of sentential calculus
K.	$\vdash \Box(p \to q) \to (\Box p \to \Box q)$
Dual.	$\vdash \Diamond p \leftrightarrow \neg\Box\neg p$
Modus Ponens	if $\vdash \varphi$ and $\vdash \varphi \to \psi$, then $\vdash \psi$
Uniform substitution	if $\vdash \varphi$, then $\vdash \theta$, where θ is obtained from φ by uniformly replacing proposition letters in φ by arbitrary formulas
Generalization	if $\vdash \varphi$, then $\vdash \Box\varphi$

Figure 2.3: Axioms and rules of K

and $\vdash$ are equivalent:

Theorem 2.3.6 (Completeness for K)**.** *A set of ML-formulas is K-consistent if and only if it is satisfiable in a Kripke model.*

The proof is classic (see for instance [26]) and relies on the construction of a *canonical model*. Note that as ML has the finite model property (i.e., every satisfiable ML-formula is satisfiable in a finite model, see [26]) and as its model checking is decidable (i.e., given a formula and a model, it is decidable whether the formula is satisfied in that model), it also follows from Theorem 2.3.6 that ML is decidable on the class of Kripke models. Recursive enumerability of the set of validities of a logic, together with co-recursive enumerability (i.e., recursive enumerability of the set of non-validities of the logic) indeed entails decidability. Recursive enumerability is given here by the complete axiomatization and co-recursive enumerability follows from the fact that attention can be restricted to finite models: if a formula is not valid, then it has a counter-model which can be found in a finite amount of time simply by enumerating all finite models and each time we list one, examining whether the formula is satisfied in it.

Similar complete axiomatizations exist for LTL (see [37]), some of its fragments (for LTL without the X operator, see [110], for LTL with only the X operator, see [121]) and for the modal μ-calculus (see [133] in the general case, [92] on ω-words and [40] on finite trees, as well as on finite words). The arguments are there more involved (especially in the case of [133]). But sometimes, constructing a canonical model and transforming it slightly happens to be enough (this is for instance the case in [40]). Decidability follows from these results in a similar way.

We already noted FO to be undecidable, but it is still completely axiomatizable. Let us recall here this classical result (for the FO axiomatization, we refer to Figure 3.1, which can be found in Chapter 3). A formula φ is *FO-provable* if $\vdash \varphi$ occurs in some FO-proof and it is *FO-consistent* if its negation is not FO-provable. A set of FO-formulas Γ is FO-consistent if there does not exist $\psi_1, \ldots, \psi_k \in \Gamma$ such that $(\psi_1 \wedge \ldots \wedge \psi_k) \to \bot$ is FO-provable.

Theorem 2.3.7 (Completeness of FO)**.** *A set of FO-formulas is FO-consistent if and only if it is satisfiable in a relational structure.*

The standard proof involves the construction of a *Henkin model*. For details we refer to [60] or to the next chapter, where we show completeness theorems for *extensions* of FO on so called *Henkin structures* (the FO-proof can easily be abstracted from these more elaborate proofs).

Historically, the first logical systems considered in the foundations of mathematics were actually far more expressive than FO, but it finally turned out that no complete axiomatizations could be established for such expressive logics ([74]). Hence the completeness theorem contributed to give a very special status to FO, which is still used as the reference logic in model theory. We saw that the picture is a bit different on trees, where the yardstick is the more expressive logic MSO. We will give complete axiom systems for MSO, $\mathsf{FO}(\mathsf{TC}^1)$ and $\mathsf{FO}(\mathsf{LFP}^1)$ on a specific class of finite trees in Chapter 3. We will also show a weaker form of completeness than the one which we just stated for K and FO and we will be concerned with *single* sentences that are consistent and not with consistent sets of sentences. This distinction is called *weak* versus *strong* completeness.

2.4 Summary

In this Chapter, we gave a general overview of temporal and fixed-point logics on trees. We first discussed different classes of trees which are of interest in the field and which we consider in the remainder of the thesis. Then, we introduced the fixed-point extensions of FO that we use in Chapter 3 and Chapter 6 and we discussed their relative expressive power on arbitrary structures, but also on trees and linear orders. We also introduced the notion of Ehrenfeucht-Fraissé game, which is an important model theoretic-tool used in Chapter 3. We then introduced linear-time temporal logic and the linear-time μ-calculus, used in Chapter 4 and 5. We also mentioned some other modal fixed-point logics that we will encounter in Chapter 6. We discussed the expressive power of these modal formalisms and recalled a few classic characterization results, like the van Benthem Characterization Theorem, which says that modal logic is the bisimulation invariant fragment of FO, or the Janin-Walukiewicz Theorem, which extends this result by saying that the μ-calculus is the bisimulation-invariant fragment of MSO. Finally, we recalled classical results like the Büchi and the Rabin Theorems, which entail that MSO and its fragments are decidable on important classes of trees. We also introduced the notion of complete axiomatization, which is central in Chapter 3 and 5.

Chapter 3

Complete Axiomatization of Fragments of MSO on Finite Trees

In this chapter, we develop a uniform method for obtaining complete axiomatizations of fragments of MSO on finite trees. In particular, we obtain a complete axiomatization for MSO, $\mathsf{FO}(\mathsf{TC}^1)$, and $\mathsf{FO}(\mathsf{LFP}^1)$ on finite node labelled sibling-ordered trees. We take inspiration from Kees Doets, who proposed in [57] complete axiomatizations of FO-theories in particular on the class of node-labelled finite trees without sibling-order (see Section 3.2, where we discuss his work in more details). A similar result for FO on node-labelled finite trees with sibling order was shown in [7] in the context of model-theoretic syntax and in [41] in the context of XML query languages. We use the signature of [41] and extend the set of axioms proposed there to match the richer syntax of the logics we consider.

We already pointed out that finite trees are basic and ubiquitous structures which are of interest at least to mathematicians, computer scientists (e.g. tree-structured documents) and linguists (e.g. parse trees). We also explained that the logics we study are known to be very well-behaved on this particular class of structures and to have an interestingly high expressive power. In particular, they all allow to express reachability, but at the same time, they have the advantage of being decidable on trees.

As XML documents are tree-structured data, our results are particularly relevant to XML query languages. Query languages are logical languages used to make queries into database and information systems. In [126] and [75], MSO and $\mathsf{FO}(\mathsf{TC}^1)$ have been proposed as a yardstick of expressivity on trees for such languages. It is known that $\mathsf{FO}(\mathsf{LFP}^1)$ has the same expressive power as MSO on trees, but the translations between the two are non-trivial, and hence it is not clear whether an axiomatization for one language can be obtained from an axiomatization for the other language in any straightforward way.

In applications to computational linguistics, finite trees are used to represent the grammatical structure of natural language sentences. In the context of *model theoretic syntax*, Rogers advocates in [118] the use of MSO in order to characterize derivation trees of context free grammars. Kepser also argues in [95] that MSO should be used in order to query treebanks. A treebank is a text corpus in which each sentence has been annotated with its syntactic structure (represented as a tree structure). In [96] and [128] Kepser and Tiede propose to consider various transitive closure logics, among which $\mathsf{FO}(\mathsf{TC}^1)$, arguing that they constitute very natural formalisms from the logical point of view, allowing concise and intuitive phrasing of parse tree properties.

The remainder of the chapter is organized as follows: in Section 3.1 we start by stating our three axiomatizations. In Section 3.2, we introduce non standard semantics called *Henkin semantics* for which our axiomatizations are easily seen to be complete. We prove in details the $\mathsf{FO}(\mathsf{LFP}^1)$ Henkin completeness proof. Section 3.3 introduces operations on Henkin structures: substructure formation and a general operation of Henkin structures combination. We obtain Feferman-Vaught theorems for this operation by means of Ehrenfeucht-Fraïssé games. This section contains in particular the definitions and adequacy proofs of the Ehrenfeucht-Fraïssé games that we also use there to prove our Feferman-Vaught theorems. In Section 3.4, we prove *real* completeness (that is, on the more restricted class of finite trees). For that purpose, we consider substructures of trees that we call forests and use the general operation discussed in Section 3.3 to combine a set of forests into one new forest. Our Feferman-Vaught theorems apply to such constructions and we use them in our main proof of completeness, showing that no formula of our language can distinguish Henkin models of our axioms from real finite trees. We also point out that every standard model of our axioms actually is a finite tree. Finally, we notice in Section 3.5 that a simplified version of our method can be used to show similar results for the class of node-labelled finite linear orders.

3.1 The Axiomatizations

In this chapter, we are interested in *finite node-labelled sibling-ordered trees*: finite trees in which the children of each node are linearly ordered. Also, the nodes can be labelled by unary predicates. In the remaining of the chapter, we will call these structures *finite trees* for short.

Definition 3.1.1 (Finite tree)**.** Assume a fixed finite set of unary predicate symbols $\{P_1, \dots, P_n\}$. By a finite tree, we mean a finite structure $\mathfrak{M} = (M, <, \prec, P_1, \dots, P_n)$, where $(M, <)$ is a tree (with $<$ the descendant relation) and $\prec$ linearly orders the children of each node.

As many arguments in this chapter equally hold for MSO, $\mathsf{FO}(\mathsf{TC}^1)$ and $\mathsf{FO}(\mathsf{LFP}^1)$, we let $\Lambda \in \{\mathsf{MSO}, \mathsf{FO}(\mathsf{TC}^1), \mathsf{FO}(\mathsf{LFP}^1)\}$ and use Λ as a symbol for any one of them. The axiomatization of Λ on finite trees consists of three parts:

FO1.	Tautologies of sentential calculus
FO2.	$\vdash \forall x\varphi \rightarrow \varphi^x_t$, where t is substitutable for x in φ
FO3.	$\vdash \forall x(\varphi \rightarrow \psi) \rightarrow (\forall x\varphi \rightarrow \forall x\psi)$
FO4.	$\vdash \varphi \rightarrow \forall x\varphi$, where x does not occur free in φ
FO5.	$\vdash x = x$
FO6.	$\vdash x = y \rightarrow (\varphi \rightarrow \psi)$, where φ is atomic and ψ is obtained from φ by replacing x in zero or more (but not necessarily all) places by y.
Modus Ponens	if $\vdash \varphi$ and $\vdash \varphi \rightarrow \psi$, then $\vdash \psi$
FO Generalization	if $\vdash \varphi$, then $\vdash \forall x\varphi$

Figure 3.1: Axioms and rules of FO

COMP.	$\vdash \exists X\forall x(Xx \leftrightarrow \varphi)$, where X does not occur free in φ
MSO1.	$\vdash \forall X\varphi \rightarrow \varphi[X/T]$, where T (which is either a set variable or a monadic predicate) is substitutable in φ for X.
MSO2.	$\vdash \forall X(\varphi \rightarrow \psi) \rightarrow (\forall X\varphi \rightarrow \forall X\psi)$
MSO3.	$\vdash \varphi \rightarrow \forall X\varphi$, where X does not occur free in φ
MSO Generalization	if $\vdash \varphi$, then $\vdash \forall X\varphi$

Figure 3.2: Axioms and inference rule of MSO

FO(TC¹) axiom	$\vdash [TC_{xy}\varphi](u, v) \rightarrow ((\psi(u) \wedge \forall x\forall y(\psi(x) \wedge \varphi(x, y) \rightarrow \psi(y))) \rightarrow \psi(v))$ where ψ is any FO(TC¹) formula
FO(TC¹) Generalization	if $\vdash \xi \rightarrow ((P(u) \wedge \forall x\forall y(P(x) \wedge \varphi(x, y) \rightarrow P(y))) \rightarrow P(v))$, and P does not occur in ξ, then $\vdash \xi \rightarrow [TC_{xy}\varphi](u, v)$

Figure 3.3: Axiom and inference rule of FO(TC¹)

FO(LFP¹) axiom	$\vdash [LFP_{x,X}\varphi]y \rightarrow (\forall x(\varphi(x, \psi) \rightarrow \psi(x)) \rightarrow \psi(y))$ where ψ is any FO(LFP¹) formula and $\varphi(x, \psi)$ is the result of the replacement in $\varphi(x, X)$ of each occurrence of X by ψ (renaming variables when needed)
FO(LFP¹) Generalization	if $\vdash \xi \rightarrow (\forall x(\varphi(x, P) \rightarrow P(x)) \rightarrow P(y))$, and P positive in φ does not occur in ξ, then $\vdash \xi \rightarrow [LFP_{X,x}\varphi](y)$

Figure 3.4: Axiom and inference rule of FO(LFP¹)

T1.	$\forall x \forall y \forall z(x < y \wedge y < z \rightarrow x < z)$	$<$ is transitive
T2.	$\neg \exists x(x < x)$	$<$ is irreflexive
T3.	$\forall x \forall y(x < y \rightarrow \exists z(x <_{ch} z \wedge z \leq y))$	immediate child
T4.	$\exists x \forall y \neg(y < x)$	there is a root
T5.	$\forall x \forall y \forall z(x < z \wedge y < z \rightarrow x \leq y \vee y \leq x)$	linearly ordered branches
T6.	$\forall x \forall y \forall z(x \prec y \wedge y \prec z \rightarrow x \prec z)$	$\prec$ is transitive
T7.	$\neg \exists x(x \prec x)$	$\prec$ is irreflexive
T8.	$\forall x \forall y(x \prec y \rightarrow \exists z(x \prec_{ch} z \wedge z \preceq y))$	immediate next sibling
T9.	$\forall x \exists y(y \preceq x \wedge \neg \exists z(z \prec y))$	there is a least sibling
T10.	$\forall x \forall y((x \prec y \vee y \prec x) \leftrightarrow (\exists z(z <_{ch} x \wedge z <_{ch} y) \wedge x \neq y))$	linearly ordered siblings
T11.	$\forall x \forall y(x = y \vee x < y \vee y < x \vee \exists x' y'(x' < x \wedge y' < y \wedge (x' \prec y' \vee y' \prec x')))$	connectedness
Ind.	$\forall x(\forall y((x < y \vee x \prec y) \rightarrow \varphi(y)) \rightarrow \varphi(x)) \rightarrow \forall x \varphi(x)$	induction scheme

where

$\varphi(x)$ ranges over Λ-formulas in one free variable x

and

$x <_{ch} y$ is shorthand for $x < y \wedge \neg \exists z(z < y \wedge x < z)$,
$x \prec_{ch} y$ is shorthand for $x \prec y \wedge \neg \exists z(x \prec z \wedge z \prec y)$

Figure 3.5: Specific axioms on finite trees

the axioms of first-order logic, the specific axioms of Λ, and the specific axioms on finite trees.

To axiomatize FO, we adopt the infinite set of logical axioms and the two rules of inference given in Figure 3.1 (like in [60], except from the fact that we use a generalization rule). To axiomatize MSO, the axioms and rule of Figure 3.2 are added to the axiomatization of FO and we call the resulting system $\vdash_{\mathsf{MSO}}$. $COMP$ stands for "comprehension" by analogy with the comprehension axiom of set theory. $MSO1$ plays a similar role as $FO2$, $MSO2$ as $FO3$ and $MSO3$ as $FO4$. To axiomatize $\mathsf{FO(TC^1)}$, the axiom and rule of Figure 3.3 are added to the axiomatization of FO and we call the resulting system $\vdash_{\mathsf{FO(TC^1)}}$. To axiomatize $\mathsf{FO(LFP^1)}$, the axiom and rule of Figure 3.4 are added to the axiomatization of FO and we call the resulting system $\vdash_{\mathsf{FO(LFP^1)}}$. We are interested in axiomatizing Λ on the class of finite trees. For that purpose, we restrict the class of considered structures by adding to $\vdash_\Lambda$ the axioms given in Figure 3.5 and we call the resulting system $\vdash_\Lambda^{tree}$. Note that the induction scheme in Figure 3.5 allows to reason by induction on *properties definable in* Λ only. Also, for technical convenience, we adopt the following convention:

Definition 3.1.2. Let Γ be a set of Λ-formulas and φ a Λ-formula. By $\Gamma \vdash_\Lambda \varphi$ we will always mean that there are $\psi_1, \ldots, \psi_n \in \Gamma$ such that $\vdash_\Lambda (\psi_1 \wedge \ldots \wedge \psi_n) \to \varphi$.

Now the main result of this chapter is that on standard structures, the Λ theory of finite trees is completely axiomatized by $\vdash_\Lambda^{tree}$. In the remaining sections we will progressively build a proof of it.

3.2 Henkin Completeness

As it is well known, MSO, $\mathsf{FO(TC^1)}$ and $\mathsf{FO(LFP^1)}$ are highly undecidable on arbitrary standard structures and hence not recursively enumerable (by arbitrary, we mean any sort of structure: infinite trees, arbitrary graphs, partial orders...). So in order to show that our axiomatizations $\vdash_\Lambda^{tree}$ are complete on finite trees, we refine a trick used by Kees Doets in his PhD thesis [57]. We proceed in two steps (the second step being the one inspired by Kees Doets). First, we show completeness theorems, based on a non-standard (so called Henkin) semantics for MSO, $\mathsf{FO(TC^1)}$ and $\mathsf{FO(LFP^1)}$ (on the general topic of Henkin semantics, see [88], the original paper by Henkin and also [108]). Each semantics respectively extends the class of standard structures with non standard (Henkin) MSO, $\mathsf{FO(TC^1)}$ and $\mathsf{FO(LFP^1)}$-structures. By the Henkin completeness theorems, our axiomatic systems $\vdash_\Lambda^{tree}$ naturally turn out to be complete on the wider class of their Henkin-models. But we will see that compactness also follows from these completeness results and some of these Henkin models are infinite. As a second step, we show in Section 3.4 that *no* Λ*-sentence can distinguish between standard and non-standard*

Λ-Henkin-models among models of our axioms. Every finite Henkin model being also a standard model, this entails that our axioms are complete on the class of (standard) finite trees, i.e., each Λ-sentence valid on this class is provable using the system $\vdash_{\Lambda}^{tree}$.

Now let us point out that Kees Doets was interested in complete axiomatizations of monadic "Π^1_1-theories" of various classes of linear orders and trees. Considering such theories in fact amounts to considering *first-order* theories of such structures extended with finitely many unary predicates. Thus, he was relying on the FO completeness theorem and if he was working with non-standard models of particular FO-theories, he was not concerned with non standard Henkin-structures in our sense. In particular, he used Ehrenfeucht-Fraïssé games in order to show that "definably well-founded" node-labelled trees have well-founded n-equivalents for all n. In Section 3.4.2, Lemma 3.4.4 which is the key lemma to our main completeness result, establishes a similar result for definably well-founded Henkin-models of the Λ-theory of finite node-labelled sibling-ordered finite trees. Hence, what makes the originality of the method developed in this chapter is its use of Henkin semantics: we first create a Henkin model and then "massage" it in order to obtain a model which is among our intended ones. Similar methods are commonly used to show completeness results in modal logic, where "canonical models" are often transformed in order to obtain intended models (see [26]). Remarkably, the completeness proof for the μ-calculus on finite trees given in [40], which is directly inspired by the methods used here, proceeds in that way. There are numerous examples of that sort in modal logic (and especially, in temporal logic), but there is also one notable example in classical model theory. In 1970, Keisler provided a complete axiomatization of FO extended with the quantifier "there exist uncountably many" (see [94]). His completeness proof, which is established for standard models, is surprisingly simple, it relies on the construction of an elementary chain of Henkin structures and then uses the omitting types theorem. Hence all in all, these structures seem to provide a particularly convenient tool, not only for simple Henkin completeness proofs, but also for more refined completeness proofs with respect to interesting subclasses of Henkin models like standard models.

Let us now introduce Henkin structures formally. Such structures are particular cases among structures called *frames* (note that such frames are unrelated to "Kripke frames") and it is convenient to define frames before defining Henkin-structures. In our case, a frame is simply a relational structure together with some subset of the powerset of its domain called its *set of admissible subsets*. A Henkin structure is a frame which set of admissible subsets satisfies some natural closure conditions.

Definition 3.2.1 (Frames)**.** Let σ be a purely relational vocabulary. A **σ-frame** $\mathfrak{M}$ consists of a non-empty domain $dom(\mathfrak{M})$, an interpretation in $dom(\mathfrak{M})$ of the predicates in σ and a set of admissible subsets $\mathbb{A}_{\mathfrak{M}} \subseteq \wp(dom(\mathfrak{M}))$.

Whenever $\mathbb{A}_{\mathfrak{M}} = \wp(dom(\mathfrak{M}))$, $\mathfrak{M}$ can be identified to a standard structure. Assignments g into $\mathfrak{M}$ are defined as in standard semantics, except that if X is a set variable, then we require that $g(X) \in \mathbb{A}_{\mathfrak{M}}$.

Definition 3.2.2 (Interpretation of Λ-formulas in frames)**.** Λ-formulas are interpreted in frames as in standard structures, except for the three following clauses. The set quantifier clause of MSO becomes:

$$\mathfrak{M}, g \models \exists X\varphi \text{ iff there is } A \in \mathbb{A}_{\mathfrak{M}_\Gamma} \text{ such that } \mathfrak{M}, g[A/X] \models \varphi$$

The TC clause of $\mathsf{FO}(\mathsf{TC}^1)$ becomes:

$$\begin{gathered} \mathfrak{M}, g \models [TC_{xy}\varphi](u, v) \\ \text{iff} \\ \text{for all } A \in \mathbb{A}_{\mathfrak{M}}, \text{ if } g(u) \in A \\ \text{and for all } a, b \in dom(\mathfrak{M}),\ a \in A \text{ and } \mathfrak{M}, g[x/a, b/y] \models \varphi \text{ imply } b \in A, \\ \text{then } g(v) \in A. \end{gathered}$$

And finally the LFP clause of $\mathsf{FO}(\mathsf{LFP}^1)$ becomes:

$$\begin{gathered} \mathfrak{M}, g \models [LFP_{x,X}\varphi]y \\ \text{iff} \\ \text{for all } A \in \mathbb{A}_{\mathfrak{M}}, \text{ if for all } a \in dom(\mathfrak{M}),\ \mathfrak{M}, g[a/x, A/X] \models \varphi(x, X) \text{ implies } a \in A, \\ \text{then } g(y) \in A. \end{gathered}$$

Definition 3.2.3 (Λ-Henkin-Structures)**.** A Λ-**Henkin-structure** is a frame $\mathfrak{M}$ that is closed under parametric Λ-definability, i.e., for each Λ-formula φ and assignment g into $\mathfrak{M}$:

$$\{a \in M \mid \mathfrak{M}, g[a/x] \models \varphi\} \in \mathbb{A}_{\mathfrak{M}}$$

We call a Λ-Henkin-structure $\mathfrak{M}$ *standard* whenever every subset in $dom(\mathfrak{M})$ belongs to $\mathbb{A}_{\mathfrak{M}}$.

Remark 3.2.4. Note that any finite Λ-Henkin-structure is a standard structure, as every subset of the domain is parametrically definable in a finite structure. Hence, non standard Henkin structures are always infinite. ⊣

Theorem 3.2.5. Λ *is completely axiomatized on* Λ*-Henkin-structures by* $\vdash_\Lambda$*, in fact for every set of* Λ*-formulas* Γ *and* Λ*-formula* φ*,* φ *is true in all* Λ*-Henkin-models of* Γ *if and only if* $\Gamma \vdash_\Lambda \varphi$*.*

We do not detail here the MSO proof, as it is a special case of the proof of completeness for the theory of types given in [108]. We focus only on the $\mathsf{FO}(\mathsf{LFP}^1)$ case, as the $\mathsf{FO}(\mathsf{TC}^1)$ case is very similar, except that there is no need to consider set variables. Up to now we have been working with purely relational vocabularies. Here we will be using individual constants in the standard way, but only for the sake of readability (we could dispense with them and use FO variables instead). Also, whenever this is clear from the context, we will use $\vdash$ as shorthand for $\vdash_{\mathsf{FO(LFP^1)}}$. Let us now begin the Henkin completeness proof for $\mathsf{FO(LFP^1)}$. This will achieve the proof of Theorem 3.2.5.

Lemma 3.2.6 (Generalization Lemma for FO Quantifiers)**.** *If* $\Gamma \vdash \varphi$ *and* x *does not occur free in* Γ*, then* $\Gamma \vdash \forall x\varphi$*.*

Proof. We refer the reader to the proof given by Enderton in [60]. □

Definition 3.2.7. We say that a set of $\mathsf{FO(LFP^1)}$ formulas Δ contains $\mathsf{FO(LFP^1)}$ Henkin witnesses if and only if the two following conditions hold. First, for every formula φ, if $\neg\forall x\varphi \in \Delta$, then $\neg\varphi[x/t] \in \Delta$ for some term t and if $\neg[LFP_{xX}\varphi]y \in \Delta$, then $\neg Py \wedge \neg\exists x(\neg Px \wedge \varphi(P, x)) \in \Delta$ for some monadic predicate P. Second, if $\varphi \in \Delta$ and x is a free variable of φ, then $\forall x(Px \leftrightarrow \varphi(x)) \in \Delta$ for some monadic predicate P.

Definition 3.2.8. We say that a $\mathsf{FO(LFP^1)}$-formula φ is *FO(LFP¹)-provable* if $\vdash_{\mathsf{FO(LFP^1)}} \varphi$ occurs in some $\mathsf{FO(LFP^1)}$-proof and it is *FO(LFP¹)-consistent* if its negation is not $\mathsf{FO(LFP^1)}$-provable.

The originality of the $\mathsf{FO(LFP^1)}$ case essentially lies in the notion of $\mathsf{FO(LFP^1)}$-Henkin witness of Definition 3.2.7. In order to use this notion in the proof of Lemma 3.2.10, we also need the following lemma:

Lemma 3.2.9. *Let* Γ *be a consistent set of FO(LFP¹)-formulas and* θ *a FO(LFP¹)-formula of the form* $\forall x(\varphi \leftrightarrow Px)$ *with* P *a fresh monadic predicate (i.e. not appearing in* Γ*). Then* $\Gamma \cup \{\theta\}$ *is also consistent.*

Proof. Suppose $\Gamma\cup\{\forall x(\varphi \leftrightarrow Px)\}$ is inconsistent, so there is some proof of $\bot$ from formulas in $\Gamma \cup \{\forall x(\varphi \leftrightarrow Px)\}$. We first rename all bound variables in the proof with variables which had no occurrence in the proof or in $\forall x(\varphi \leftrightarrow Px)$ (this is possible since proofs are finite objects and we have a countable stock of variables). Also, whenever in the proof the $\mathsf{FO(LFP^1)}$ generalization rule is applied on some unary predicate P, we make sure that this P is different from the unary predicate that we want to substitute by φ and which does not appear in the proof; this is always possible because we have a countable set of unary predicates. Now, we replace in the proof all occurrences of Px by φ (as we renamed bound variables, there is no accidental binding of variables by wrong quantifiers). Then, every occurrence of $\forall x(\varphi \leftrightarrow Px)$ in the proof becomes an occurrence of $\forall x(\varphi \leftrightarrow \varphi)$, i.e., we have obtained a proof of $\bot$ from $\Gamma \cup \{\forall x(\varphi \leftrightarrow \varphi)\}$, i.e., from Γ ($\forall x(\varphi \leftrightarrow \varphi)$ is an axiom, as it can be obtained by FO generalization from a tautology of sentential calculus). It entails that Γ is already inconsistent, which contradicts the consistency of Γ. Now it remains to show that the replacement procedure of all occurrences of Px by φ is correct, so that we still have a proof of $\bot$ after it.

Every time the replacement occurs in an axiom (or its generalization, which is still an axiom as we defined it), then the result is still an instance of the given axiom schema (even for FO(LFP^1) generalizations, because we took care that P is never used in the proof for a FO(LFP^1) generalization). Also, as replacement is applied uniformly in the proof, every application of modus ponens stays correct: consider $\psi \rightarrow \xi$ and ψ. Obviously the result ψ^* of the substitution will allow to derive the result ξ^* of the substitution from $\psi^* \rightarrow \xi^*$ and ψ^*. Also $\bot^*$ is simply $\bot$, so the procedure gives us a proof of $\bot$. □

Lemma 3.2.10. *(FO(LFP^1) Lindenbaum Lemma) Let $\sigma^* = \sigma \cup \{c_n \mid \ \in \mathbb{N}\} \cup \{P_n \mid n \in \mathbb{N}\}$ with $c_i, P_i \notin \sigma$. If a set Γ of FO(LFP^1)-formulas in vocabulary σ is consistent, then there exists a maximally consistent set Γ^* of σ^* formulas such that $\Gamma \subseteq \Gamma^*$ and Γ^* contains FO(LFP^1)-Henkin witnesses.*

Proof. Let Γ be a consistent set of well formed FO(LFP^1)-formulas in a countable vocabulary. We expand the language by adding countably many new constants and countably many new monadic predicates. Then Γ remains consistent as a set of well formed formulas in the new language. For every pair constituted by one formula and one FO variable of σ^*, we adopt the following fix exhaustive enumeration:

$$< \varphi_1, x_1 >, < \varphi_2, x_2 >, < \varphi_3, x_3 >, < \varphi_4, x_4 >, \ldots$$

(this is possible since the language is countable), where the φ_i are formulas and the x_i are FO variables.

- Let θ_{3n-2} be $\neg\forall x_n\varphi_n \rightarrow \neg\varphi[x_n/c_l]$, where c_l is the first of the new constants neither occurring in φ_n nor in θ_k with $k < 3n-2$.
- Let θ_{3n-1} be $\neg[LFP_{xX}\varphi_n]x_n \rightarrow (\neg P_l x_n \wedge \neg\exists x(\neg P_l x \wedge \varphi(P_l, x)))$, where P_l is the first of the new monadic predicates neither occurring in φ_n nor in θ_k with $k < 3n-1$.
- Let θ_{3n} be $\forall x_n(\varphi_n \leftrightarrow P_l x_n)$, where P_l is the first of the new monadic predicates neither occurring in φ_n nor in θ_k with $k < 3n$.

Call Θ the set of all the θ_i.

Claim 3.2.11. $\Gamma \cup \Theta$ *is consistent*

If not, then because deductions are finite, for some $m \geq 0$, $\Gamma \cup \{\theta_1, \ldots, \theta_m, \theta_{m+1}\}$ is inconsistent. Take the least such m, then by the *reductio ad absurdum* rule (which is, as in FO, admissible in FO(LFP^1)), $\Gamma \cup \{\theta_1, \ldots, \theta_m\} \vdash \neg\theta_{m+1}$. Now there are three cases:

(1) θ_{m+1} is of the form $\neg\forall x\varphi \rightarrow \neg\varphi[x/c]$ i.e. either $\Gamma \cup \{\theta_1, \ldots, \theta_m\} \vdash \neg\forall x\varphi$ and $\Gamma \cup \{\theta_1, \ldots, \theta_m\} \vdash \varphi[x/c]$. Since c does not appear in any formula on the left, by Lemma 3.2.6, $\Gamma \cup \{\theta_1, \ldots, \theta_m\} \vdash \forall x\varphi$, which contradicts the minimality of m (or the consistency of Γ if $m = 0$).

(2) θ_{m+1} is of the form $\neg[LFP_{xX}\varphi]y \to (\neg Py \wedge \neg\exists x(\neg Px \wedge \varphi(P,x)))$. In such a case both $\Gamma \cup \{\theta_1 \dots \theta_m\} \vdash \neg[LFP_{xX}\varphi]y$ and $\Gamma \cup \{\theta_1 \dots \theta_m\} \vdash \neg Py \wedge \neg\exists x(\neg Px \wedge \varphi(P,x))$ hold. Since P does not appear in any formula on the left, by FO(LFP^1) generalization, $\Gamma \cup \{\theta_1 \dots \theta_m\} \vdash [LFP_{xX}\varphi]y$, which contradicts the minimality of m (or the consistency of Γ if $m = 0$).

(3) θ_{m+1} is of the form $\forall x(\varphi \leftrightarrow Px)$. By Lemma 3.2.9, this is not possible.

We then turn $\Gamma \cup \Theta$ into a maximal consistent set Γ^* in the standard way. □

We will now show that if Γ^* is maximally consistent and contains FO(LFP^1)-Henkin witnesses, then Γ^* has a FO(LFP^1)-Henkin model $\mathfrak{M}_{\Gamma^*}$.

Definition 3.2.12. Let $\Gamma^* \subseteq FORM(\sigma)$ be maximally consistent and contain FO(LFP^1)-Henkin witnesses. We define an equivalence relation on the set of FO terms, by letting $t_1 \equiv_{\Gamma^*} t_2$ iff $t_1 = t_2 \in \Gamma^*$. We denote the equivalence class of a term t by $|t|$.

Proposition 3.2.13. $\equiv_{\Gamma^*}$ *is an equivalence relation.*

Proof. By $FO5$ and $FO6$. □

Definition 3.2.14. We define $\mathfrak{M}_{\Gamma^*}$ (together with a valuation g_{Γ^*}) out of Γ^*.

- $M = \{|t| : t \text{ is a FO term }\}$
- $\mathbb{A}_{\mathfrak{M}_{\Gamma^*}} = \{A_T : T \text{ is a set variable or a monadic predicate}\}$ where $A_T = \{|t| : Tt \in \Gamma^*\}$
- $(|t_1|, \dots, |t_n|) \in P^{\mathfrak{M}}_{\Gamma^*}$ iff $Pt_1 \dots t_n \in \Gamma^*$
- $c^{\mathfrak{M}_{\Gamma^*}} = |c|$
- $g_{\Gamma^*}(x) = |x|$
- $g_{\Gamma^*}(X) = A_X$

Proposition 3.2.15. $\mathfrak{M}_{\Gamma^*}$ *is a FO(LFP^1)-Henkin structure.*

Proof. By construction of Γ^* this is immediate (we introduced a monadic predicate for each parametrically definable subset). □

Lemma 3.2.16. *(Truth lemma) For every FO(LFP^1) formula φ, $\mathfrak{M}_{\Gamma^*}, g_{\Gamma^*} \models \varphi$ iff $\varphi \in \Gamma^*$.*

Proof. By induction on φ.

The base case follows from the definition of $\mathfrak{M}_{\Gamma^*}$ together with the maximality of Γ^*. Now consider the inductive step:

- Boolean connectives and FO quantifier: as in FO
- LFP operator: we want to show that

$$\mathfrak{M}_{\Gamma^*}, g_{\Gamma^*} \models [LFP_{xX}\varphi]y \text{ iff } [LFP_{xX}\varphi]y \in \Gamma^*$$

  – We first show that

$$\mathfrak{M}_{\Gamma^*}, g_{\Gamma^*} \models [LFP_{xX}\varphi]y \text{ implies } [LFP_{xX}\varphi]y \in \Gamma^*.$$

  So suppose $\mathfrak{M}_{\Gamma^*}, g_{\Gamma^*} \models [LFP_{xX}\varphi]y$ i.e. for all monadic predicates $P_i \in \sigma^*$, if $g_{\Gamma^*}(y) \notin A_{P_i}$ then there exists $|t_k| \in M$, such that $|t_k| \notin A_{P_i}$ and $\mathfrak{M}_{\Gamma^*}, g_{\Gamma^*}[x/|t_k|, X/A_{P_i}] \models \varphi$ i.e. for all P_i such that $\neg P_i y$ there exists t_k such that $\mathfrak{M}_{\Gamma^*}, g_{\Gamma^*} \models (\neg P_i t_k \wedge \varphi(t_k, P_i))$ and by induction hypothesis $\neg P_i t_k \wedge \varphi(t_k, P_i) \in \Gamma^*$. And so by the same argument as the one used in the FO quantifier step of the present induction, $\neg P_i y \to \exists x(\neg P_i x \wedge \varphi(x, P_i)) \in \Gamma^*$. Now suppose $[LFP_{xX}\varphi]y \notin \Gamma^*$ i.e. $\neg[LFP_{xX}\varphi]y \in \Gamma^*$. Then as Γ^* contains FO(LFP^1) Henkin witnesses, there is a predicate P_m such that $\neg P_m y \wedge \neg\exists x(\neg P_m x \wedge \varphi(P_m, x) \in \Gamma^*$. But that contradicts the maximal consistency of Γ^*. Then $\neg[LFP_{xX}\varphi]y \notin \Gamma^*$ and by maximal consistency of Γ^*, $[LFP_{xX}\varphi]y \in \Gamma^*$.

  – We now show that $[LFP_{xX}\varphi]y \in \Gamma^*$ implies $\mathfrak{M}_{\Gamma^*}, g_{\Gamma^*} \models [LFP_{xX}\varphi]y$. We consider the contraposition

$$\mathfrak{M}_{\Gamma^*}, g_{\Gamma^*} \not\models \text{ implies } [LFP_{xX}\varphi]y \notin \Gamma^*.$$

  So suppose $\mathfrak{M}_{\Gamma^*}, g_{\Gamma^*} \not\models [LFP_{xX}\varphi]y$ i.e. $\mathfrak{M}_{\Gamma^*}, g_{\Gamma^*} \models \neg[LFP_{xX}\varphi]y$ i.e. there exists $A_{P_i} \in \mathbb{A}_{\mathfrak{M}_{\Gamma^*}}$ such that, $g(y) \notin A_{P_i}$ and for all $|t_k| \in M$, $|t_k| \in A_{P_i}$ or $\mathfrak{M}_{\Gamma^*}, g_{\Gamma^*}[x/|t_k|, X/P_i] \models \neg\varphi$ and by induction hypothesis for all for all t_k, $\neg P_i y \wedge (P_i t_k \vee \neg\varphi(P_i, t_k)) \in \Gamma^*$. And so by the same argument as the one used in the FO quantifier step of the present induction, $\neg P_i y \wedge \forall x(P_i x \vee \neg\varphi(P_i, x)) \in \Gamma^*$ i.e. (by maximal consistency) $\neg P_i y \wedge \neg\exists x(\neg P_i x \wedge \varphi(P_i, x)) \in \Gamma^*$. Now suppose $[LFP_{xX}\varphi]y \in \Gamma^*$. Then by the LFP axiom, for every monadic predicate P_m, $\neg P_m y \to \exists x(\neg P_m(x) \wedge \varphi(x, P_m)) \in \Gamma^*$. But that contradicts the maximal consistency of Γ^*. Then $[LFP_{xX}\varphi]y \notin \Gamma^*$ and by maximal consistency of Γ^*, $\neg[LFP_{xX}\varphi]y \in \Gamma^*$.

□

Theorem 3.2.17. *Every consistent set* Γ *of FO(LFP^1)-formulas is satisfiable.*

Proof. First turn Γ into a FO(LFP^1) maximal consistent set Γ^* with FO(LFP^1)-Henkin witnesses in a possibly richer signature (with extra individual constants and monadic predicates) σ^*. Then build a structure $\mathfrak{M}_{\Gamma^*}$ out of this Γ^*. Then the structure $\mathfrak{M}_{\Gamma^*}$ satisfies Γ^* under the valuation g_{Γ^*} and hence it satisfies also Γ (Γ being a subset of Γ^*). □

Compactness follows directly from Definition 3.1.2 and Theorem 3.2.5, i.e., a possibly infinite set of Λ-sentences has a Λ-Henkin model if and only if every finite subset of it has a Λ-Henkin model. It also follows directly from Theorem 3.2.5 that $\vdash_\Lambda^{tree}$ is complete on the class of its Λ-Henkin-models. Nevertheless, by compactness the axioms of $\vdash_\Lambda^{tree}$ are also satisfied on infinite trees. We overcome this problem by defining a slightly larger class of Henkin structures, which we will call *definably well-founded* Λ*-quasi-trees.*[1]

Definition 3.2.18. A Λ-*quasi-tree* is any Λ-Henkin structure

$$(T, <, \prec, P_1, \ldots, P_n, \mathbb{A}_T)$$

(where $\mathbb{A}_T$ is the set of admissible subsets of T) satisfying the axioms and rules of $\vdash_\Lambda$ and the axioms T1–T11 of Figure 3.5. A Λ-quasi-tree is *definably well founded* if, in addition, it satisfies all instances of the induction scheme Ind of Figure 3.5.

Corollary 3.2.19. *A* Λ*-Henkin-structure satisfies the axioms of* $\vdash_\Lambda^{tree}$ *if and only if it is a definably well-founded* Λ*-quasi-tree.*

3.3 Operations on Henkin-Structures

Let $\Lambda \in \{\mathsf{MSO}, \mathsf{FO(TC^1)}, \mathsf{FO(LFP^1)}\}$. As noted in Remark 3.2.4, every finite Λ-Henkin structure is also a standard structure. Hence, when working in finite model theory, it is enough to rely on the usual FO constructions to define operations on structures. On the other hand, even though our main completeness result concerns finite trees, inside the proof we need to consider infinite (Λ-Henkin) structures and operations on them. In this context, methods for forming new structures out of existing ones have to be redefined carefully. We first propose a notion of substructure of a Λ-Henkin-structure generated by one of its parametrically definable admissible subsets:

Definition 3.3.1 (Λ-substructure)**.** Let $\mathfrak{M} = (dom(\mathfrak{M}), Pred, \mathbb{A}_\mathfrak{M})$ be a Λ-Henkin-structure (where $Pred$ is the interpretation of the predicates). We call $\mathfrak{M}_{\mathsf{FO}} = (dom(\mathfrak{M}), Pred)$ the relational structure underlying $\mathfrak{M}$. Given a parametrically definable set $A \in \mathbb{A}_\mathfrak{M}$, the Λ-substructure of $\mathfrak{M}$ generated by A is the structure $\mathfrak{M} \upharpoonright A = (\langle A \rangle_{\mathfrak{M}_{\mathsf{FO}}}, \mathbb{A}_{\mathfrak{M}\upharpoonright A})$, where $\langle A \rangle_{\mathfrak{M}_{\mathsf{FO}}}$ is the relational substructure of $\mathfrak{M}_{\mathsf{FO}}$ generated by A (note that A forms the domain of $\langle A \rangle_{\mathfrak{M}_{\mathsf{FO}}}$, as the vocabulary is purely relational) and $\mathbb{A}_{\mathfrak{M}\upharpoonright A} = \{X \cap A | X \in \mathbb{A}_\mathfrak{M}\}$.

Note that in the case of MSO and $\mathsf{FO(LFP^1)}$, we could also have defined $\mathbb{A}_{\mathfrak{M}\upharpoonright A}$ in an alternative way:

[1]For a nice picture of a quasi-tree which is *not* definably well-founded, see [7].

Proposition 3.3.2. *Take $\mathfrak{M}$ and A as previously and consider the structure $(\mathfrak{M} \upharpoonright A)' = (\langle A \rangle_{\mathfrak{M}_{FO}}, \mathbb{A}_{(\mathfrak{M}\upharpoonright A)'})$, where $\mathbb{A}_{(\mathfrak{M}\upharpoonright A)'} = \{X \subseteq A | X \in \mathbb{A}_{\mathfrak{M}}\}$. Whenever $\mathfrak{M}$ is a MSO-Henkin structure or a FO(LFP^1)-Henkin structure, $\mathfrak{M} \upharpoonright A$ and $(\mathfrak{M} \upharpoonright A)'$ are one and the same structure.*

Proof. Indeed, take $B \in \mathbb{A}_{\mathfrak{M}\upharpoonright A}$. So there exists $B' \in \mathbb{A}_{\mathfrak{M}}$ such that $B = B' \cap A$. We want to show that also $B' \cap A \in \mathbb{A}_{(\mathfrak{M}\upharpoonright A)'}$ i.e. $B' \cap A \subseteq A$ (which obviously holds) and $B' \cap A \in \mathbb{A}_{\mathfrak{M}}$. The second condition holds because both B' and A are parametrically definable in $\mathfrak{M}$, so their intersection also is ($B' \cap A = \{x \mid \mathfrak{M} \models Ax \wedge B'x\}$). Conversely, consider $B \in \mathbb{A}_{(\mathfrak{M}\upharpoonright A)'}$. As $B \subseteq A$ and $B \in \mathbb{A}_{\mathfrak{M}}$ it follows that $B \in \mathbb{A}_{\mathfrak{M}\upharpoonright A}$ (we can take $B = B \cap A$). □

Now, in order to show that Λ-substructures are Λ-Henkin-structures, we introduce a notion of *relativization* and a corresponding *relativization lemma.* This lemma establishes that for every Λ-Henkin-structure $\mathfrak{M}$ and Λ-substructure $\mathfrak{M} \upharpoonright A$ of $\mathfrak{M}$ (with A a set parametrically definable in $\mathfrak{M}$), if a set is parametrically definable in $\mathfrak{M} \upharpoonright A$ then it is also parametrically definable in $\mathfrak{M}$. This result will be useful again in Section 3.4.2.

Definition 3.3.3 (Relativization mapping)**.** Given two Λ-formulas φ, ψ having no variables in common and given a FO variable x occurring free in ψ, we define $REL(\varphi, \psi, x)$ by induction on the complexity of φ and call it the *relativization of φ to ψ*:

- If φ is an atom, $REL(\varphi, \psi, x) = \varphi$,
- If $\varphi :\approx \varphi_1 \wedge \varphi_2$, $REL(\varphi, \psi, x) = REL(\varphi_1, \psi, x) \wedge REL(\varphi_2, \psi, x)$ (similar for $\vee, \rightarrow, \neg$),
- If $\varphi :\approx \exists y \chi$, $REL(\varphi, \psi, x) = \exists y(\psi[y/x] \wedge REL(\chi, \psi, x))$,
- If $\varphi :\approx \exists Y \chi$, $REL(\varphi, \psi, x) = \exists Y((Yx \rightarrow \psi) \wedge REL(\chi, \psi, x))$,
- If $\varphi :\approx [TC_{yz}\chi](u, v)$,
  $REL(\varphi, \psi, x) = [TC_{yz}REL(\chi, \psi, x) \wedge \psi[y/x] \wedge \psi[z/x]](u, v)$,
- If $\varphi :\approx [LFP_{Xy}\chi]z$, $REL(\varphi, \psi, x) = [LFP_{Xy}\chi \wedge \psi[y/x]]z$.

where $\psi[y/x]$ is the formula obtained by replacing in ψ every occurrence of x by y and similarly for $\psi[z/x]$.

Hence for instance, $REL(\exists y P(y), Q(x), x) = \exists y(P(y) \wedge Q(y))$, which is satisfied in any model $\mathfrak{M}$ of which the submodel induced by Q contains an element satisfying P.

Lemma 3.3.4 (Relativization lemma)**.** *Let $\mathfrak{M}$ be a Λ-Henkin-structure, g a valuation on $\mathfrak{M}$, φ, ψ Λ-formulas having no variable in common and $A = \{x \mid \mathfrak{M}, g \models \psi\}$. If $g(y) \in A$ for every variable y occurring free in φ and $g(Y) \in \mathbb{A}_{\mathfrak{M}\upharpoonright A}$ for every set variable Y occurring free in φ, then $\mathfrak{M}, g \models REL(\varphi, \psi, x) \Leftrightarrow \mathfrak{M} \upharpoonright A, g \models \varphi$.*

Proof. By induction on the complexity of φ. Let g be an assignment satisfying the required conditions. Base case: φ is an atom and $REL(\varphi, \psi, x) = \varphi$. So $\mathfrak{M}, g \models \varphi \Leftrightarrow \mathfrak{M} \upharpoonright A, g \models \varphi$ (by hypothesis, g is a suitable assignment for both models). Inductive hypothesis: the property holds for every φ of complexity at most n. Now consider φ of complexity $n + 1$.

- $\varphi :\approx \varphi_1 \wedge \varphi_2$ and $REL(\varphi_1 \wedge \varphi_2, \psi, x) :\approx REL(\varphi_1, \psi, x) \wedge REL(\varphi_2, \psi, x)$. By induction hypothesis, the property holds for φ_1 and for φ_2. By the semantics of $\wedge$, it also holds for $\varphi_1 \wedge \varphi_2$. (Similar for $\vee, \rightarrow, \neg$.)

- $\varphi :\approx \exists y \chi$ and $REL(\exists y \chi) :\approx \exists y(\psi[y/x] \wedge REL(\chi, \psi, x))$. By inductive hypothesis, for every node $a \in A$, $\mathfrak{M}, g[a/y] \models REL(\chi, \psi, x) \Leftrightarrow \mathfrak{M} \upharpoonright A, g[a/y] \models \chi$. Hence, by the semantics of $\exists$ and by definition of A, $\mathfrak{M}, g \models \exists y(\psi[y/x] \wedge REL(\chi, \psi, x)) \Leftrightarrow \mathfrak{M} \upharpoonright A, g \models \exists y \chi$.

- $\varphi :\approx \exists Y \chi$ and $REL(\exists Y \chi, \psi, x) = \exists Y((Yx \rightarrow \psi) \wedge REL(\chi, \psi, x))$. As every admissible subset of $\mathfrak{M} \upharpoonright A$ is also admissible in $\mathfrak{M}$ (by Proposition 3.3.2) it follows by inductive hypothesis that for every $B \in \mathfrak{M} \upharpoonright \mathbb{A}$, $\mathfrak{M}, g[B/Y] \models REL(\chi, \psi, x) \Leftrightarrow \mathfrak{M} \upharpoonright A, g[B/Y] \models \chi$. Hence, by the semantics of $\exists$ and by definition of A, $\mathfrak{M}, g \models \exists Y((Yx \rightarrow \psi) \wedge REL(\chi, \psi, x)) \Leftrightarrow \mathfrak{M} \upharpoonright A, g \models \exists Y \chi$.

- $\varphi :\approx [TC_{yz}\chi](u, v)$ and $REL([TC_{yz}\chi](u, v), \psi, x) = [TC_{yz}REL(\chi, \psi, x) \wedge \psi[y/x] \wedge \psi[z/x]](u, v)$. By definition of TC, the following are equivalent:

  1. $\mathfrak{M} \upharpoonright A, g \models [TC_{yz}\chi](u, v)$,
  2. for all $B \in \mathbb{A}_{\mathfrak{M}\upharpoonright A}$, if $g(u) \in B$ and for all $a, b \in A$, $a \in B$ and $\mathfrak{M} \upharpoonright A, g[a/y, b/z] \models \chi$ implies $b \in B$, then $g(v) \in B$.

  By inductive hypothesis, for all $a, b \in A$,
  $\mathfrak{M}, g[a/y, b/z] \models REL(\chi, \psi, x) \Leftrightarrow \mathfrak{M} \upharpoonright A, g[a/y, b/z] \models \chi$. Hence 2. $\Leftrightarrow$ 3.:

  3. for all $B \in \mathbb{A}_{\mathfrak{M}\upharpoonright A}$, if $g(u) \in B$ and for all $a, b \in A$, $a \in B$ and $\mathfrak{M}, g[a/y, b/z] \models REL(\chi, \psi, x)$ implies $b \in B$, then $g(v) \in B$,

  By definition of A, 3. $\Leftrightarrow$ 4.:

  4. for all $B \in \mathbb{A}_{\mathfrak{M}\upharpoonright A}$, if $g(u) \in B$ and for all $a, b \in dom(\mathfrak{M})$, $a \in B$ and $\mathfrak{M}, g[a/y, b/z] \models REL(\chi, \psi, x) \wedge \psi[y/x] \wedge \psi[z/x]$ implies $b \in B$, then $g(v) \in B$,

  We claim that 4. $\Leftrightarrow$ 5.:

5. for all $C \in \mathbb{A}_{\mathfrak{M}}$, if $g(u) \in C$ and for all $a, b \in dom(\mathfrak{M})$, $a \in C$ and $\mathfrak{M}, g[a/y, b/z] \models REL(\chi, \psi, x) \wedge \psi[y/x] \wedge \psi[z/x]$ implies $b \in C$, then $g(v) \in C$,

which, by the semantics of TC, is equivalent to:

6. $\mathfrak{M}, g \models [TC_{yz}REL(\chi, \psi, x) \wedge \psi[y/x] \wedge \psi[z/x]](u, v)$.

It is clear that 5. $\Rightarrow$ 4.. For the 4. $\Rightarrow$ 5. direction, assume 4.. Take any set $C \in \mathbb{A}_{\mathfrak{M}}$ such that $g(u) \in C$ and for all $a, b \in dom(\mathfrak{M})$, $a \in C$ and $\mathfrak{M}, g[a/y, b/z] \models REL(\chi, \psi, x) \wedge \psi[y/x] \wedge \psi[z/x]$ implies $b \in C$. Let $B = A \cap C$. By Definition 3.3.1, $B \in \mathbb{A}_{\mathfrak{M}\restriction A}$. Now by our assumptions on g and by definition of A, $g[a/y, b/z]$ only assigns points in A. So as $B = A \cap C$, $g(u) \in B$ and for all $a, b \in dom(\mathfrak{M})$, $a \in B$ and $\mathfrak{M}, g[a/y, b/z] \models REL(\chi, \psi, x) \wedge \psi[y/x] \wedge \psi[z/x]$ implies $b \in B$. So by 4., $g(v) \in B$. As $B \subseteq C$, it follows that $g(v) \in C$.

- $\varphi :\approx [LFP_{Xy}\chi]z$ and $REL([LFP_{Xy}\chi]z, \psi, x) :\approx [LFP_{Xy}\chi \wedge \psi[y/x]]z$. By definition of LFP, the following are equivalent:

1. $\mathfrak{M} \restriction A, g \models [LFP_{Xy}\chi]z$,
2. for all $B \in \mathbb{A}_{\mathfrak{M}\restriction A}$, if for all $a \in A$, $\mathfrak{M} \restriction A, g[a/y, B/X] \models \chi$ implies $a \in B$, then $g(z) \in B$.

By inductive hypothesis, for all $a \in A$, $B \in \mathfrak{M} \restriction \mathbb{A}$, $\mathfrak{M}, g[a/y, B/X] \models REL(\chi, \psi, x) \Leftrightarrow \mathfrak{M} \restriction A, g[a/y, B/X] \models \chi$. Hence 2. is equivalent to 3.:

3. for all $B \in \mathbb{A}_{\mathfrak{M}\restriction A}$, if for all $a \in A$, $\mathfrak{M}, g[a/y, B/X] \models REL(\chi, \psi, x)$ implies $a \in B$, then $g(z) \in B$,

By definition of A, 3. $\Leftrightarrow$ 4.:

4. for all $B \in \mathbb{A}_{\mathfrak{M}\restriction A}$, if for all $a \in dom(\mathfrak{M})$,
$\mathfrak{M}, g[a/y, B/X] \models REL(\chi, \psi, x) \wedge \psi[y/x]$ implies $a \in B$,
then $g(z) \in B$,

We claim that 4. $\Leftrightarrow$ 5.:

5. for all $C \in \mathbb{A}_{\mathfrak{M}}$, if for all $a \in dom(\mathfrak{M})$,
$\mathfrak{M}, g[a/y, C/X] \models REL(\chi, \psi, x) \wedge \psi[y/x]$ implies $a \in C$,
then $g(z) \in C$,

which, by the semantics of LFP, is equivalent to:

6. $\mathfrak{M}, g \models [LFP_{Xy}REL(\chi, \psi, x) \wedge \psi[y/x]]z$.

It is clear that 5. $\Rightarrow$ 4.. For the 4. $\Rightarrow$ 5. direction, assume 4.. Take any set $C \in \mathbb{A}_{\mathfrak{M}}$ such that for all $a \in dom(\mathfrak{M})$, $\mathfrak{M}, g[a/y, C/X] \models REL(\chi, \psi, x) \wedge \psi[y/x]$ implies $a \in C$. Let $B = A \cap C$. By Definition 3.3.1, $B \in \mathbb{A}_{\mathfrak{M}\restriction A}$. Consider $a \in dom(\mathfrak{M})$ such that $\mathfrak{M}, g[a/y, B/X] \models REL(\chi, \psi, x) \wedge \psi[y/x]$. As $REL(\chi, \psi, x)$ is positive in X and X does not occur in ψ, $\mathfrak{M}, g[a/y, C/X] \models REL(\chi, \psi, x) \wedge \psi[y/x]$. Also by hypothesis $a \in C$. Now as $\mathfrak{M}, g[a/y] \models \psi[y/x]$, by definition of A, $a \in A$. So $a \in A \cap C$, i.e, $a \in B$ and since we proved it for arbitrary $a \in dom(\mathfrak{M})$, by 4., $g(z) \in B$. As $B \subseteq C$, it follows that $g(z) \in C$.

□

Theorem 3.3.5. $\mathfrak{M} \restriction A$ *is a* Λ-Henkin-*structure.*

Proof. Take B parametrically definable in $\mathfrak{M} \restriction A$, i.e., there is a Λ-formula $\varphi(y)$ and an assignment g such that $B = \{a \in dom(\mathfrak{M} \restriction A) \mid \mathfrak{M} \restriction A, g[a/y] \models \varphi(y)\}$. Now we know that A is also parametrically definable in $\mathfrak{M}$, i.e., there is a Λ-formula $\psi(x)$ and an assignment g' such that $A = \{a \in dom(\mathfrak{M}) \mid \mathfrak{M}, g'[a/x] \models \psi(x)\}$. Assume without loss of generality that φ and ψ have no variables in common. We define an assignment g^* by letting $g^*(z) = g'(z)$ for every variable z occurring in ψ and $g^*(z) = g(z)$ otherwise. The situation with set variables is symmetric. Now by Lemma 3.3.4, $B = \{a \in dom(\mathfrak{M}) \mid \mathfrak{M}, g^*[a/x] \models REL(\varphi, \psi, x)\}$ and hence $B \in \mathbb{A}_{\mathfrak{M}}$. By definition 3.3.1 it follows that $B \in \mathbb{A}_{\mathfrak{M}\restriction A}$ (because $B = B \cap A$). □

There is in model theory a whole range of methods to form new structures out of existing ones. A standard reference on the matter is [64], written in a very general algebraic setting. Familiar constructions like disjoint unions of relational structures are redefined as particular cases of a new notion of *generalized product* of FO-structures and abstract properties of such products are studied. In particular, an important theorem now called the Feferman-Vaught theorem for FO is proven in [64]. We are particularly interested in one of its corollaries, which establishes that generalized products of relational structures preserve elementary equivalence. We show an analogue of this result for a particular case of generalized product of Λ-Henkin-structures that we call *fusion*, this notion being itself a generalization of a notion of disjoint union of Λ-Henkin-structures defined below.

Definition 3.3.6 (Disjoint union of Λ-Henkin-structures)**.** Let σ be a purely relational vocabulary and $\sigma^* = \sigma \cup \{Q_1, \ldots, Q_k\}$, with $\{Q_1, \ldots, Q_k\}$ a set of new monadic predicates. For any Λ-Henkin-structures $\mathfrak{M}_1, \ldots, \mathfrak{M}_k$ in vocabulary σ with disjoint domains, define their *disjoint union* $\biguplus_{1\leq i\leq k} \mathfrak{M}_i$ (or, *direct sum*) to be the σ^*-frame that has as its domain the union of the domains of the structures $\mathfrak{M}_i$ and likewise for the relations, except for the predicates Q_i, whose interpretations are respectively defined as the domain of the structures $\mathfrak{M}_i$ (we will use Q_i to label the elements of M_i). The set of admissible subsets $\mathbb{A}_{\biguplus_{1\leq i\leq k} \mathfrak{M}_i}$ is the closure under finite union of the union of the sets of admissible subsets of the $\mathfrak{M}_i$. That is:

- $dom(\biguplus_{1\leq i\leq k}\mathfrak{M}_i) = \bigcup_{1\leq i\leq k} dom(\mathfrak{M}_i)$
- $P^{\biguplus_{1\leq i\leq k}\mathfrak{M}_i} = \bigcup_{1\leq i\leq k} P^{\mathfrak{M}_i}$ (with $P \in \sigma$) and $Q_i^{\biguplus_{1\leq i\leq k}\mathfrak{M}_i} = dom(\mathfrak{M}_i)$
- $A \in \mathbb{A}_{\biguplus_{1\leq i\leq k}\mathfrak{M}_i}$ iff $A = \bigcup_{1\leq i\leq k} A_i$ for some $A_i \in \mathbb{A}_{\mathfrak{M}_i}$

Definition 3.3.7 (f-fusion of Λ-Henkin-structures)**.** Let σ be a purely relational vocabulary and $\sigma^* = \sigma \cup \{Q_1, \ldots, Q_k\}$, with $\{Q_1, \ldots, Q_k\}$ a set of new monadic predicates. Let f be a function mapping each n-ary predicate $P \in \sigma$ to a quantifier-free first-order formula over σ^* in variables $x_1, \ldots, x_n$. For any Λ-Henkin-structures $\mathfrak{M}_1, \ldots, \mathfrak{M}_k$ in vocabulary σ with disjoint domains, define their *f-fusion* to be the σ-frame $\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i$ that has the same domain and set of admissible subsets as $\biguplus_{1\leq i\leq k}\mathfrak{M}_i$. For every $P \in \sigma$, the interpretation of P in $\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i$ is the set of n-tuples satisfying $f(P)$ in $\biguplus_{1\leq i\leq k}\mathfrak{M}_i$.

An easy example of f-fusion on standard structures (it is simpler to give an example on standard structures, as we do not have to say anything about admissible sets) is the ordered sum of two linear orders $(M_1, <_1), (M_2, <_2)$, where all the elements of M_1 are before the elements of M_2. In this case, σ consists of a single binary relation $<$, the elements of M_1 are indexed with Q_1, those of M_2 with Q_2 and f maps $<$ to $x_1 < x_2 \vee (Q_1x_1 \wedge Q_2x_2)$. Another notable example of f-fusion is the $\sigma \cup \{Q_1, \ldots, Q_k\}$-structure $\biguplus_{1\leq i\leq k}\mathfrak{M}_i = \bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i^+$, where f is the identity function and for each $1 \leq i \leq k$, $\mathfrak{M}_i^+$ is the expansion of the σ-structure $\mathfrak{M}_i$ in which $Q_i^{\mathfrak{M}_i^+} = dom(\mathfrak{M}_i)$ and $Q_j^{\mathfrak{M}_i^+} = \emptyset$ for every $i \neq j$. In this sense, disjoint union as we defined it above can be seen as a special case of fusion.

We show preservation results involving f-fusions of Λ-Henkin-structures. Hence we deal with analogues of elementary equivalence for these logics and we refer to Λ-*equivalence*. Let us recall that by quantifier depth of a Λ-formula, we mean the maximal number of nested quantifiers in the formula (by "quantifier", we mean FO and MSO-quantifiers, as well as TC or LFP-operators).

Definition 3.3.8. Given two Λ-Henkin-structures $\mathfrak{M}$ and $\mathfrak{N}$, we write $\mathfrak{M} \equiv_\Lambda \mathfrak{N}$ and say that $\mathfrak{M}$ and $\mathfrak{N}$ are Λ-*equivalent* if they satisfy the same Λ-sentences. Also, for any natural number n, we write $\mathfrak{M} \equiv^n_\Lambda \mathfrak{N}$ and say that $\mathfrak{M}$ and $\mathfrak{N}$ are n-Λ-*equivalent* if $\mathfrak{M}$ and $\mathfrak{N}$ satisfy the same Λ-sentences of quantifier depth at most n. In particular, $\mathfrak{M} \equiv_\Lambda \mathfrak{N}$ holds iff, for all n, $\mathfrak{M} \equiv^n_\Lambda \mathfrak{N}$ holds.

Now we are ready to introduce the "Feferman-Vaught theorems" that we will show in Section 3.3.2 and which establish that f-fusions of Λ-Henkin-structures preserve Λ-equivalence, that is:

Theorem 3.3.9. *Let $\mathfrak{M}_1, \ldots, \mathfrak{M}_k$, $\mathfrak{N}_1, \ldots, \mathfrak{N}_k$ be Λ-Henkin structures. Whenever $\mathfrak{M}_i \equiv^n_\Lambda \mathfrak{N}_i$ for all $1 \leq i \leq k$, then also $\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i \equiv^n_\Lambda \bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i$.*

We will also show in this section that every f-fusion of Λ-Henkin-structures is a Λ-Henkin-structure. Comparable work had already been done by Makowski in [106] for extensions of FO, but an important difference is that he only considered standard structures, whereas we need to deal with Λ-Henkin-structures. Our proofs make use of Ehrenfeucht-Fraïssé games for each of the logics Λ.

3.3.1 Ehrenfeucht-Fraïssé Games on Henkin-Structures

Let $\Lambda \in \{\mathsf{MSO}, \mathsf{FO(TC^1)}, \mathsf{FO(LFP^1)}\}$. We survey Ehrenfeucht-Fraïssé games for FO, MSO, $\mathsf{FO(TC^1)}$, and $\mathsf{FO(LFP^1)}$ which are suitable to use on Henkin structures. We also provide an adequacy proof for the $\mathsf{FO(TC^1)}$ game. The MSO game is a rather straightforward extension of the FO case and has already been used by other authors (see for instance [100]). The $\mathsf{FO(LFP^1)}$ game is borrowed from Uwe Bosse [32]. It also applies to Henkin structures, as careful inspection shows. The $\mathsf{FO(TC^1)}$ game has already been mentioned in passing by Erich Grädel in [76] as an alternative to the game he used and we show that it is adequate for Henkin semantics. It looks also similar to a system of partial isomorphisms given in [38]. However it is important to note that this game is very different from the $\mathsf{FO(TC^1)}$ game which is actually used in [76]. The two games are equivalent when played on standard structures, but not when played on $\mathsf{FO(TC^1)}$-Henkin structures. This is so because the game used in [38] relies on the alternative semantics for the TC operator given in Proposition 2.1.11, so that only finite sets of points can be chosen by players ; whereas the game we use involves choices of not necessarily finite admissible subsets. These are not equivalent approaches. Indeed, on $\mathsf{FO(TC^1)}$-Henkin structures a simple compactness argument shows that the semantical clause of Proposition 2.1.11 (defined in terms of existence of a *finite* path) is not adequate.

Let us first introduce basic notions connected to these games. One rather trivial sufficient condition for Λ-equivalence is the existence of an *isomorphism*. Clearly isomorphic structures satisfy the same Λ-formulas. A more interesting sufficient condition for Λ-equivalence is that of Duplicator having a winning strategy in all Λ Ehrenfeucht-Fraïssé games of finite length. To define this, we first need this notion:

Definition 3.3.10 (Finite Partial Isomorphism)**.** A *finite partial isomorphism* between structures $\mathfrak{M}$ and $\mathfrak{N}$ is a finite relation $\{(a_1, b_1), \ldots, (a_n, b_n)\}$ between the domains of $\mathfrak{M}$ and $\mathfrak{N}$ such that for all atomic formulas $\varphi(x_1, \ldots, x_n)$, $\mathfrak{M} \models \varphi\ [a_1, \ldots, a_n]$ iff $\mathfrak{N} \models \varphi\ [b_1, \ldots, b_n]$. Since equality statements are atomic formulas, every finite partial isomorphism is (the graph of) a *injective partial function.*

We will also need the following lemma:

Lemma 3.3.11 (Finiteness Lemma). *Fix any set $x_1, \ldots, x_k, X_{k+1}, \ldots, X_m$. In a finite relational vocabulary, up to logical equivalence, with these free variables, there are only finitely many Λ-formulas of quantifier depth $\leq n$.*

Proof. This can be shown by induction on k. In a finite relational vocabulary, with finitely many free variables, there are only finitely many atomic formulas. Now, any Λ-formula of quantifier depth $k+1$ is equivalent to a Boolean combination of atoms and formulas of quantifier depth k prefixed by a quantifier. Applying a quantifier to equivalent formulas preserves equivalence and the Boolean closure of a finite set of formulas remains finite, up to logical equivalence. □

Now, as we are concerned with extensions of FO, every Λ-game will be defined as an extension of the classical FO game, that we recall here:

Definition 3.3.12 (FO Ehrenfeucht-Fraïssé Game). The FO Ehrenfeucht-Fraïssé game of length n on standard structures $\mathfrak{M}$ and $\mathfrak{N}$ (notation: $EF^n_{FO}(\mathfrak{M}, \mathfrak{N})$) is as follows. There are two players, Spoiler and Duplicator. The game has n rounds, each of which consists of a move of Spoiler followed by a move of Duplicator. Spoiler's moves consist of picking an element from one of the two structures, and Duplicator's responses consist of picking an element in the other structure. In this way, Spoiler and Duplicator build up a finite binary relation between the domains of the two structures: initially, the relation is empty; each round, it is extended with another pair. The winning conditions are as follows: if at some point of the game the constructed binary relation is not a finite partial isomorphism, then Spoiler wins immediately. If after each round the relation is a finite partial isomorphism, then the game is won by Duplicator.

Theorem 3.3.13 (FO Adequacy). *Assume a finite relational first-order language. Duplicator has a winning strategy in the game $EF^n_{FO}(\mathfrak{M}, \mathfrak{N})$ iff $\mathfrak{M} \equiv^n_{FO} \mathfrak{N}$. In particular, Duplicator has a winning strategy in all EF-games of finite length between $\mathfrak{M}$ and $\mathfrak{N}$ if and only if $\mathfrak{M} \equiv_{FO} \mathfrak{N}$.*

The proof for the first order case is classic. We refer the reader to the proof given in [58] or to the one in [104].

For technical convenience in the course of inductive proofs, we extend the notion of FO parameter by considering set parameters, i.e., instead of interpreting a set variable as a name of the admissible set A, we can add a new monadic predicate A to the signature. The new predicates and the sets they name are called set parameters. (This is similar to the FO notion which can be found in [89].) We will work with *parametrized* (or *expanded*) Henkin-structures, that is, structures considered together with partial valuations. This means that the assignment is possibly non empty at the beginning of the game, which can start with some "handicap" for Duplicator, i.e., some preliminary set of already "distinguished objects and sets".

We first define a necessary and sufficient condition for MSO equivalence by extending Ehrenfeucht-Fraïssé games from FO to MSO. This game has already been defined in the literature, see for instance [100].

Definition 3.3.14 (MSO Ehrenfeucht-Fraïssé Game)**.** Consider two MSO-Henkin structures $\mathfrak{M}$ together with $\bar{A} \in \mathbb{A}^r_{\mathfrak{M}}$, $\bar{a} \in dom(\mathfrak{M})^s$ and $\mathfrak{N}$ together with $\bar{B} \in \mathbb{A}^r_{\mathfrak{N}}$, $\bar{b} \in dom(\mathfrak{N})^s$ and $r \geq 0$, $s \geq 0$, $n \geq 0$. The MSO Ehrenfeucht-Fraïssé game $EF^n_{\mathsf{MSO}}((\mathfrak{M}, \bar{A}, \bar{a}), (\mathfrak{N}, \bar{B}, \bar{b}))$ of length n on expanded structures $(\mathfrak{M}, \bar{A}, \bar{a})$ and $(\mathfrak{N}, \bar{B}, \bar{b})$ is defined as for the first-order case, except that each time she chooses a structure, Spoiler can choose either an element or an admissible subset of its domain. For a given $A_{r+1} \in \mathbb{A}_{\mathfrak{M}}$ chosen by Spoiler, $(\mathfrak{M}, \bar{A}, \bar{a})$ is expanded to $(\mathfrak{M}, \bar{A}, A_{r+1}, \bar{a})$. Duplicator then responds by choosing $B_{r+1} \in \mathbb{A}_{\mathfrak{N}}$ and $(\mathfrak{N}, \bar{B}, \bar{b})$ is expanded to $(\mathfrak{N}, \bar{B}, B_{r+1}, \bar{b})$. The game goes on with the so expanded structures. The winning conditions are as follows: if at some point of the game $\bar{a} \mapsto \bar{b}$ is not a finite partial isomorphism from $(\mathfrak{M}, \bar{A}, A_{r+1})$ to $(\mathfrak{N}, \bar{B}, B_{r+1})$, then Spoiler wins immediately. If after each round the relation is a finite partial isomorphism, then the game is won by Duplicator.

Theorem 3.3.15 (MSO Adequacy)**.** *Assume a finite relational MSO language. Given $\mathfrak{M}$ and $\mathfrak{N}$, $\bar{A} \in \mathbb{A}^r_{\mathfrak{M}}$, $\bar{B} \in \mathbb{A}^r_{\mathfrak{N}}$, $\bar{a} \in dom(\mathfrak{M})^s$, $\bar{b} \in dom(\mathfrak{N})^s$ and $r \geq 0$, $s \geq 0$, $n \geq 0$, Duplicator has a winning strategy in the game $EF^n_{\mathsf{MSO}}((\mathfrak{M}, \bar{A}, \bar{a}), (\mathfrak{N}, \bar{B}, \bar{b}))$ iff $(\mathfrak{M}, \bar{A}, \bar{a})$ and $(\mathfrak{N}, \bar{B}, \bar{b})$ satisfy the same MSO formulas of quantifier depth n. In particular, Duplicator has a winning strategy in all EF_{MSO}-games of finite length between $(\mathfrak{M}, \bar{A}, \bar{a})$ and $(\mathfrak{N}, \bar{B}, \bar{b})$ if and only if $(\mathfrak{M}, \bar{A}, \bar{a})$ and $(\mathfrak{N}, \bar{B}, \bar{b})$ satisfy the same MSO formulas.*

We omit the proof, because it parallels the FO case. The proof works regardless whether MSO is interpreted in the standard or in the Henkin way. What matters here is that the game-theoretic meaning of a "quantification" over a given "domain", lies in the choice of an element from that domain (including one consisting of "higher-order elements", e.g., sets).

Corollary 3.3.16. *For MSO-Henkin-structures $\mathfrak{M}$, $\mathfrak{N}$ and $n \geq 0$, Duplicator has a winning strategy in $EF^n_{\mathsf{MSO}}(\mathfrak{M}, \mathfrak{N})$ if and only if $\mathfrak{M} \equiv^n_{\mathsf{MSO}} \mathfrak{N}$. In particular, Duplicator has a winning strategy in all EF_{MSO}-games of finite length between $\mathfrak{M}$ and $\mathfrak{N}$ if and only if $\mathfrak{M} \equiv_{\mathsf{MSO}} \mathfrak{N}$.*

The $\mathsf{FO}(\mathsf{TC}^1)$ game that we will be introducing now had been already mentioned in passing by Erich Grädel in [76] as an alternative to the game he used. We will show that it is adequate on Henkin-structures.

Definition 3.3.17 ($\mathsf{FO}(\mathsf{TC}^1)$ Ehrenfeucht-Fraïssé Game)**.** Consider two $\mathsf{FO}(\mathsf{TC}^1)$-Henkin structures $\mathfrak{M}$ and $\mathfrak{N}$ together with $\bar{a} \in dom(\mathfrak{M})^s$, $\bar{b} \in dom(\mathfrak{N})^s$ and $s \geq 0$, $n \geq 0$. The $\mathsf{FO}(\mathsf{TC}^1)$-game $EF^n_{\mathsf{FO}(\mathsf{TC}^1)}((\mathfrak{M}, \bar{a}), (\mathfrak{N}, \bar{b}))$ of length n on expanded structures $(\mathfrak{M}, \bar{a})$ and $(\mathfrak{N}, \bar{b})$ is defined as for the first-order case, except that each time she chooses a structure, Spoiler can either choose only one element or an admissible subset together with two elements of its domain. In the first case we say that she plays an $\exists$ (or point) move and in the second case, a TC-move (which we will define more precisely below). Each point move results in an extension of the assignment $\{\bar{a} \mapsto \bar{b}\}$ with elements $a_{s+1} \in dom(\mathfrak{M}), b_{s+1} \in dom(\mathfrak{N})$. Each TC-move results in an extension of the assignment $\{\bar{a} \mapsto \bar{b}\}$ with elements $a_{s+1}, a_{s+2} \in dom(\mathfrak{M}), b_{s+1}, b_{s+2} \in dom(\mathfrak{N})$. At each round, Spoiler chooses the kind of move to be played.

The $\exists$ move is defined as in the FO case. The TC-move is as follows:

Spoiler considers two pebbles (a_i, b_i) and (a_j, b_j) on the board (i.e., corresponding couples of parameters taken in each structure) and depending on the structure that he chooses to consider, he plays:

- either a set $A \in \mathbb{A}_{\mathfrak{M}}$ with $a_i \in A$ and $a_j \notin A$. Duplicator then answers with a set $B \in \mathbb{A}_{\mathfrak{N}}$ such that $b_i \in B$ and $b_j \notin B$. Spoiler now picks $b_{s+1} \in B, b_{s+2} \notin B$ and Duplicator answers with $a_{s+1} \in A, a_{s+2} \notin A$.

- or a set $B \in \mathbb{A}_{\mathfrak{N}}$ with $b_i \in B$ and $b_j \notin B$. Duplicator then answers with a set $A \in \mathbb{A}_{\mathfrak{M}}$ such that $a_i \in A$ and $a_j \notin A$. Spoiler now picks $a_{s+1} \in A, a_{s+2} \notin A$ and Duplicator answers with $b_{s+1} \in B, b_{s+2} \notin B$.

In each TC-move, the assignment is extended with $a_{s+1} \mapsto b_{s+1}, a_{s+2} \mapsto b_{s+2}$. After n moves, Duplicator has won if the constructed assignment $\bar{a} \mapsto \bar{b}$ is a partial isomorphism (i.e. the game continues with the two new pebbles in each structure, but the sets A and B are forgotten).

Theorem 3.3.18 (FO(TC^1) Adequacy)**.** *Assume a finite relational FO(TC^1) language. Given two FO(TC^1)-Henkin structures $\mathfrak{M}$ and $\mathfrak{N}$, $\bar{a} \in dom(\mathfrak{M})^s$, $\bar{b} \in dom(\mathfrak{N})^s$ and $r \geq 0$, $s \geq 0$, $n \geq 0$, Spoiler has a winning strategy in the game $EF^n_{FO(TC^1)}((\mathfrak{M}, \bar{a}), (\mathfrak{N}, \bar{b}))$ iff there is a FO(TC^1) formula of quantifier depth n distinguishing $(\mathfrak{M}, \bar{a})$ and $(\mathfrak{N}, \bar{b})$.*

Proof.

- $\Rightarrow$ From the existence of a winning strategy for Spoiler in the FO(TC^1)-game of length n in between $(\mathfrak{M}, \bar{a})$ and $(\mathfrak{N}, \bar{b})$, we will infer the existence of a FO(TC^1)-formula of quantifier depth n distinguishing $(\mathfrak{M}, \bar{a})$ and $(\mathfrak{N}, \bar{b})$.

  By induction on n.

  Base step: With 0 round the initial match between distinguished objects must have failed to be a partial isomorphism for Spoiler to win. This implies that $(\mathfrak{M}, \bar{a})$ and $(\mathfrak{N}, \bar{b})$ disagree on some atomic formula.

  Inductive step: The inductive hypothesis says that for every two structures, if Spoiler can win their comparison game over n rounds, then the structures disagree on some FO(TC^1)-formula of quantifier depth n. Now assume that for some structures $(\mathfrak{M}, \bar{a}), (\mathfrak{N}, \bar{b})$, Spoiler has a winning strategy for the game over $n+1$ rounds. Let us reason on Spoiler's first move in the game. It can either be a TC or an $\exists$ move.

If it is an $\exists$ move, then it means that Spoiler picks an element a in one of the two structures, so that no matter what element b Duplicator picks in the other, Spoiler has an n-round winning strategy. But then we can use the induction hypothesis, and find for each such b a formula $\varphi_b(x)$ that distinguishes $(\mathfrak{M}, \bar{a}, a)$ from $(\mathfrak{N}, \bar{b}, b)$. In fact we can assume that in each case the respective formula is true of $(\mathfrak{M}, \bar{a}, a)$ and false of $(\mathfrak{N}, \bar{b}, b)$ (by negating the formula if needed). Now take the big conjunction $\varphi(x)$ of all these formulas (which is equivalent to a finite formula according to Lemma 3.3.11) and prefix it with an existential quantifier. Then the resulting formula is true in $(\mathfrak{M}, \bar{a})$ but false in $(\mathfrak{N}, \bar{b})$. It is true in $(\mathfrak{M}, \bar{a})$ if we pick a for the existentially quantified variable. And no matter which element we pick in $(\mathfrak{N}, \bar{b})$, it will always falsify one of the conjuncts in the formula, by construction. So, the new formula is false in $(\mathfrak{N}, \bar{b})$. I.e., $\exists x \varphi(x)$ of quantifier depth $n+1$ distinguishes $(\mathfrak{M}, \bar{a})$ and $(\mathfrak{N}, \bar{b})$.

If Spoiler's first move is a TC-move, then it means that Spoiler picks a subset in one structure, let say $A \in \mathbb{A}_{\mathfrak{M}}$ (with $a_i \in A$ and $a_j \notin A$), so that no matter which $B \in \mathbb{A}_{\mathfrak{N}}$ (with $b_i \in B$ and $b_j \notin B$) Duplicator picks in the other structure, Spoiler can pick $b_k \in B$, $b_{k+1} \notin B$ such that no matter which $a_k \in A$, $a_{k+1} \notin A$ Duplicator picks, Spoiler has an n-round winning strategy. For each B that might be chosen by Duplicator, Spoiler's given strategy gives a fixed couple b_k, b_{k+1}. For each response a_k, a_{k+1} of Duplicator, we thus obtain by inductive hypothesis a discriminating formula $\varphi_{B,a_k,a_{k+1}}(x, y)$ that we can assume to be true in $(\mathfrak{N}, \bar{b})$ for b_k, b_{k+1} and false in $(\mathfrak{M}, \bar{a})$ for a_k, a_{k+1}. Now for each B, let us take the big conjunction $\Phi_B(x, y)$ of all these formulas (which is finite, by Lemma 3.3.11). We can then construct the big disjunction $\Phi(x, y)$ (again finite, by the same lemma) of all the formulas $\Phi_B(x, y)$.

Considering the first round in the game together with the inductive hypothesis, note that the MSO formula $\exists X(a_i \in X \wedge a_j \notin X \wedge \forall xy((x \in X \wedge y \notin X) \rightarrow \neg\Phi(x, y)))$ holds in $(\mathfrak{M}, \bar{a})$. Indeed, by induction hypothesis, any couple $a_k \in A, a_{k+1} \notin A$ that Duplicator might choose in $dom(\mathfrak{M})$ will always falsify at least one of the conjuncts of each $\Phi_B(x, y)$. Finally, the formula $\Phi(x, y)$ being constructed as the disjunction of all the formulas $\Phi_B(x, y)$, any such couple a_k, a_{k+1} will also falsify $\Phi(x, y)$. Now the MSO formula $\exists X(a_i \in X \wedge a_j \notin X \wedge \forall xy((x \in X \wedge y \notin X) \rightarrow \neg\Phi(x, y)))$ is equivalent to $\exists X(a_i \in X \wedge a_j \notin X \wedge \neg\exists xy(x \in X \wedge \Phi(x, y) \wedge y \notin X))$, which means that $(\mathfrak{M}, \bar{a}) \not\models [TC_{xy}\Phi(x, y)](a_i, a_j)$.

On the other hand for the same reasons, note that it holds in $(\mathfrak{N}, \bar{b})$ that $\forall X((b_i \in X \wedge b_j \notin X) \rightarrow \exists xy(x \in X \wedge y \notin X \wedge \Phi(x,y)))$. Indeed, by induction hypothesis, for each B that Duplicator might choose in $\mathbb{A}_{\mathfrak{N}}$ Spoiler will always be able to find a couple $b_k \in B, b_{k+1} \notin B$ satisfying all the conjuncts of the corresponding formulas $\Phi_B(x,y)$. Finally, the formula $\Phi(x,y)$ being constructed as the disjunction of all the formulas $\Phi_B(x,y)$, such a couple a_k, a_{k+1} will also satisfy $\Phi(x,y)$. Now $\forall X((b_i \in X \wedge b_j \notin X) \rightarrow \exists xy(x \in X \wedge y \notin X \wedge \Phi(x,y)))$ is equivalent to $\forall X(b_i \notin X \vee b_j \in X \vee \exists xy(x \in X \wedge y \notin X \wedge \Phi(x,y)))$, which means that $(\mathfrak{N}, \bar{b}) \models [TC_{xy}\Phi(x,y)](b_i, b_j)$.

Let u be a name for the parameters a_i, b_i and v for b_i, b_j. $[TC_{xy}\Phi(x,y)](u,v)$ of quantifier depth $n+1$ distinguishes $(\mathfrak{N}, \bar{a})$ and $(\mathfrak{M}, \bar{b})$.

$\Leftarrow$ From the existence of a $\mathsf{FO(TC^1)}$ formula of quantifier depth n distinguishing $(\mathfrak{M}, \bar{a})$ and $(\mathfrak{N}, \bar{b})$ we will infer the existence of a winning strategy for Spoiler in $EF^n_{FO+TC}((\mathfrak{M}, \bar{a}), (\mathfrak{N}, \bar{b}))$.

By induction on n.

Base step: Doing nothing is a strategy for Spoiler.

Inductive step: The inductive hypothesis says that, for every two structures, if they disagree on some $\mathsf{FO(TC^1)}$ formula of quantifier depth n, then Duplicator has a winning strategy in the n-round game. Now, assume that some expanded structures $(\mathfrak{M}, \bar{a})$, $(\mathfrak{N}, \bar{b})$ disagree on some $\mathsf{FO(TC^1)}$ formula χ of quantifier depth $n+1$. Any such formula must be equivalent to a Boolean combination of formulas of the form $\exists x\psi(x)$ and $[TC_{xy}\varphi(x,y)](u,v)$ with ψ, φ of quantifier depth at most n. If χ distinguishes the two structures, then there is at least one component of this Boolean combination which suffices distinguishing them.

Let us first suppose that it is of the form $\exists x\psi(x)$. We may assume without loss of generality that $(\mathfrak{M}, \bar{a}) \models \exists x\psi(x)$ whereas $(\mathfrak{N}, \bar{b}) \not\models \exists x\psi(x)$. Then it means that there exists an object $a \in dom(\mathfrak{M})$ such that $(\mathfrak{M}, \bar{a}) \models \psi(a)$ whereas for every object $b \in dom(\mathfrak{N})$, $(\mathfrak{N}, \bar{b}) \not\models \psi(b)$. But then we can use our induction hypothesis and find for each such b a winning strategy for Spoiler in $EF^n_{\mathsf{FO(TC^1)}}((\mathfrak{M}, \bar{a}, a), (\mathfrak{N}, \bar{b}, b))$. We can infer that Spoiler has a winning strategy in $EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M}, \bar{a}), (\mathfrak{N}, \bar{b}))$. His first move consists in picking the object a in $dom(\mathfrak{M})$ and for each response b in $dom(\mathfrak{N})$ of Duplicator, the remaining of his winning strategy is the same as in $EF^n_{\mathsf{FO(TC^1)}}((\mathfrak{M}, \bar{a}, a), (\mathfrak{N}, \bar{b}, b))$.

Let us now suppose that $[TC_{xy}\varphi(x,y)](u,v)$ of quantifier depth $n+1$ distinguishes the two structures. We may assume without loss of generality that $(\mathfrak{M}, \bar{a}) \models [TC_{xy}\varphi(x,y)](u,v)$ i.e. it holds in $(\mathfrak{M}, \bar{a})$ that $\forall X((a_i \in X \wedge a_j \notin X) \rightarrow \exists xy(x \in X \wedge y \notin X \wedge \varphi(x,y)))$, whereas $(\mathfrak{N}, \bar{b}) \not\models [TC_{xy}\varphi(x,y)](u,v)$ i.e. it holds in $(\mathfrak{N}, \bar{b})$ that $\exists X(b_i \in X \wedge b_j \notin X \wedge$

$\neg\exists xy(x \in X \wedge \varphi(x,y) \wedge y \notin X))$. We want to show that Spoiler has a winning strategy in $EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M},\bar{a}),(\mathfrak{N},\bar{b}))$. Let us describe her first move. She first chooses $(\mathfrak{N},\bar{b})$ and $B \in \mathbb{A}_{\mathfrak{N}}$ such that $b_i \in B \wedge b_j \notin B \wedge \neg\exists xy(x \in B \wedge \varphi(x,y) \wedge y \notin B)$. By definition of TC, such a set exists. Duplicator has to respond by picking a set A in $\mathbb{A}_{\mathfrak{M}}$ containing a_i and not a_j. Spoiler then picks $a_k \in A$ and $a_{k+1} \notin A$ such that $(\mathfrak{M},\bar{a}) \models \varphi(a_k,a_{k+1})$. This is possible because by definition of TC, for any possible choice A of Duplicator (i.e., any set A containing a_i and not a_j) we have $\exists xy(x \in A \wedge y \notin A \wedge \varphi(x,y))$. But that means that Duplicator is now stuck and has to pick $b_k \in B$ and $b_{k+1} \notin B$ such that $(\mathfrak{N},\bar{b}) \not\models \varphi(b_k,b_{k+1})$. Consequently, we have $(\mathfrak{N},\bar{b},b_k,b_{k+1}) \not\models \varphi(x,y)$, whereas $(\mathfrak{M},\bar{a},a_k,a_{k+1}) \models \varphi(x,y)$. As $\varphi(x,y)$ is of quantifier depth n, by induction hypothesis, Spoiler has a winning strategy in $EF^{n}_{\mathsf{FO(TC^1)}}((\mathfrak{M},\bar{a},a_k,a_{k+1}),(\mathfrak{N},\bar{b},b_k,b_{k+1}))$. The remaining of Spoiler's winning strategy in $EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M},\bar{a}),(\mathfrak{N},\bar{b}))$ (i.e. after her first move, that we already accounted for) is consequently as in $EF^{n}_{\mathsf{FO(TC^1)}}((\mathfrak{M},\bar{a},a_k,a_{k+1}),(\mathfrak{N},\bar{b},b_k,b_{k+1}))$.

□

Corollary 3.3.19. *For structures $\mathfrak{M}$, $\mathfrak{N}$ and $n \geq 0$, Duplicator has a winning strategy in $EF^{n}_{\mathsf{FO(TC^1)}}(\mathfrak{M},\mathfrak{N})$ if and only if $\mathfrak{M} \equiv^{n}_{\mathsf{FO(TC^1)}} \mathfrak{N}$. In particular, Duplicator has a winning strategy in all $EF_{\mathsf{FO(TC^1)}}$-games of finite length between $\mathfrak{M}$ and $\mathfrak{N}$ if and only if $\mathfrak{M} \equiv_{\mathsf{FO(TC^1)}} \mathfrak{N}$.*

Let us finally consider the $\mathsf{FO(LFP^1)}$ case. There are two classical equivalent syntactic ways of defining the syntax of $\mathsf{FO(LFP^1)}$: the one we used in Section 2.1.2 and another one, dispensing with restrictions to positive formulas, but allowing negations only in front of atomic formulas and introducing a greatest fixed-point operator as the dual of the least fixed-point operator (also $\forall$ cannot be defined using $\exists$ and has to be introduced separately, similarly for the Boolean connectives). This second way of defining $\mathsf{FO(LFP^1)}$ turns out to be more convenient to define an adequate Ehrenfeucht-Fraïssé game. The game is suitable to use on Henkin structures because the semantics on which it relies is merely a syntactical variant of the one given in Section 3.2. Now the $\mathsf{FO(LFP^1)}$-formulas $[LFP_{x,X}\varphi(x,X)]y$ and $[GFP_{x,X}\varphi(x,X)]y$, stating that a point belongs to the least fixed-point, or respectively, to the greatest fixed-point induced by the formula φ satisfy the following equations:

$$[LFP_{x,X}\varphi(x,X)]y \leftrightarrow \forall X(\neg Xy \to \exists x(\neg Xx \wedge \varphi(x,X)))$$
$$[GFP_{x,X}\varphi(x,X)]y \leftrightarrow \exists X(Xy \wedge \forall x(Xx \to \varphi(x,X)))$$

Note that this holds no matter whether we be concerned with $\mathsf{FO(LFP^1)}$ and MSO on standard structures or on Henkin structures. The consideration of these equations is the key idea behind an Ehrenfeucht-Fraïssé game defined by Uwe

Bosse in [32] for least fixed-point logic FO(LFP) (i.e. where fixed-points are not only considered for monadic operators, but for any n-ary operator). FO(LFP^1) being simply the monadic fragment of FO(LFP), the game for FO(LFP) can be adapted to FO(LFP^1) in a straightforward way:

Definition 3.3.20 (FO(LFP^1) Ehrenfeucht-Fraïssé game)**.** Consider FO(LFP^1)-Henkin structures $\mathfrak{M}$ and $\mathfrak{N}$ together with $\bar{a} \in dom(\mathfrak{M})^s$, $\bar{b} \in dom(\mathfrak{N})^s$, $\bar{A} \in \mathbb{A}^r_{\mathfrak{M}}$, $\bar{b} \in \mathbb{A}^r_{\mathfrak{N}}$, $r \geq 0$, $s \geq 0$, $n \geq 0$. In the game $EF^n_{\mathsf{FO(LFP^1)}}((\mathfrak{M}, \bar{A}, \bar{a}), (\mathfrak{N}, \bar{B}, \bar{b}))$ of length n, there are two types of moves, point and fixed-point moves. Each move results in an extension of the assignment $\bar{a} \mapsto \bar{b}, \bar{A} \mapsto \bar{B}$ with elements $a_{s+1} \in dom(\mathfrak{M}), b_{s+1} \in dom(\mathfrak{N})$, and possibly (in the case of fixed-point moves) with sets $A_{r+1} \in \mathbb{A}_{\mathfrak{M}}, B_{r+1} \in \mathbb{A}_{\mathfrak{N}}$. Spoiler chooses the kind of move to be played. Now the following moves are possible:

- $\exists$ move: Spoiler chooses $a_{s+1} \in dom(\mathfrak{M})$ and Duplicator $b_{s+1} \in dom(\mathfrak{N})$.
- $\forall$ move: Spoiler chooses $b_{s+1} \in dom(\mathfrak{N})$ and Duplicator $a_{s+1} \in dom(\mathfrak{M})$.

In each point move, the assignment is extended by $a_{s+1} \mapsto b_{s+1}$.

- LFP move: Spoiler chooses $B_{r+1} \in \mathbb{A}_{\mathfrak{N}} \setminus \{dom(\mathfrak{N})\}$ with some pebble $b_i \notin B_{r+1}$ and Duplicator responds with $A_{r+1} \in \mathbb{A}_{\mathfrak{M}} \setminus \{dom(\mathfrak{M})\}$.

  Now Spoiler chooses in $dom(\mathfrak{M})$ a new element $a_{s+1} \notin A_{r+1}$ and Duplicator answers in $dom(\mathfrak{N})$ with $b_{s+1} \notin B_{r+1}$.

- GFP move: Spoiler chooses $A_{r+1} \in \mathbb{A}_{\mathfrak{M}} \setminus \{dom(\mathfrak{M})\}$ with some pebble $a_i \in A_{r+1}$ and Duplicator responds with $B_{r+1} \in \mathbb{A}_{\mathfrak{N}} \setminus \{dom(\mathfrak{N})\}$ such that $B_{r+1} \neq \emptyset$.

  Now Spoiler chooses in $dom(\mathfrak{N})$ a new element $b_{s+1} \in B_{r+1}$ and Duplicator answers in $dom(\mathfrak{M})$ with $a_{s+1} \in A_{r+1}$.

In each fixed-point move the assignment is extended by $A_{r+1} \mapsto B_{r+1}, a_{s+1} \mapsto b_{s+1}$.

After n moves, Duplicator has won if the constructed element assignment $\bar{a} \mapsto \bar{b}$ is a partial isomorphism and for the subset assignment $\bar{A} \mapsto \bar{B}$, for any $1 \leq j \leq r$ and $i \leq s$:

$$a_i \in A_j \text{ implies } b_i \in B_j$$

We call an assignment with these properties a *posimorphism*.

Theorem 3.3.21 (FO(LFP^1) Adequacy)**.** *Assume a finite relational FO(LFP^1) language. Given two FO(LFP^1)-Henkin structures $\mathfrak{M}$ and $\mathfrak{N}$, $\bar{A} \in \mathbb{A}^r_{\mathfrak{M}}$, $\bar{B} \in \mathbb{B}^r_{\mathfrak{N}}$, $\bar{a} \in dom(\mathfrak{M})^s$, $\bar{b} \in dom(\mathfrak{N})^s$ and $r \geq 0$, $s \geq 0$, $n \geq 0$, Duplicator has a winning strategy in the game $EF^n_{\mathsf{FO(LFP^1)}}((\mathfrak{M}, \bar{A}, \bar{a}), (\mathfrak{N}, \bar{B}, \bar{b}))$ iff $(\mathfrak{M}, \bar{A}, \bar{a})$ and $(\mathfrak{N}, \bar{B}, \bar{b})$ satisfy the same FO(LFP^1)-formulas of quantifier depth n.*

For a proof in the case of standard structures, we refer the reader to Uwe Bosse [32]. As pointed out earlier, the same argument works as well in the case of Henkin structures.

3.3.2 Fusion Theorems on Henkin-Structures

Let $\Lambda \in \{\mathsf{MSO}, \mathsf{FO}(\mathsf{TC}^1), \mathsf{FO}(\mathsf{LFP}^1)\}$. We show our analogues of Feferman-Vaught theorem for fusions of Λ-Henkin-structures. We will refer to them as Λ-*fusion Theorems*, even though they will sometimes be formally first stated as corollaries. What we show is, more precisely, that fusion of Λ-Henkin-structures preserve Λ-equivalence for all fixed quantifier-depths.

In order to give inductive proofs for MSO and $\mathsf{FO}(\mathsf{LFP}^1)$, it will be more convenient to consider parametrized Λ-Henkin-structures where the set of set parameters is closed under union, this notion being defined below. This is safe because whenever two parametrized structures $(\mathfrak{M}, \bar{A}, \bar{a})$ and $(\mathfrak{N}, \bar{B}, \bar{b})$ are n-Λ-equivalent, it follows trivially that $\mathfrak{M}$ and $\mathfrak{N}$ considered together with a subset of this set of parameters are also n-Λ-equivalent.

Definition 3.3.22. Let $A_1, \ldots, A_k$ be a finite sequence of set parameters. We define $(A_1, \ldots, A_k)^{\cup}$ as the finite sequence of set parameters obtained by closing the set $\{A_1, \ldots, A_k\}$ under union in such a way that $(A_1, \ldots, A_k)^{\cup} = \{\bigcup_{i \in I} A_i | I \subseteq \{1, \ldots, k\}\}$. (We additionally assume that this set is ordered in a fixed canonical way, depending on the index sets I.)

Theorem 3.3.23 (Fusion Theorem for MSO)**.** *Let $\bar{a}_i$, $\bar{b}_i$ be sequences of first-order parameters of the form $a_{i_1}, \ldots, a_{i_m}$, $b_{i_1}, \ldots, b_{i_m}$, with $m \in \mathbb{N}$ and $\bar{A}_i$, $\bar{B}_i$ sequences of set parameters of the form $A_{i_1}, \ldots, A_{i_{m'}}$, $B_{i_1}, \ldots, B_{i_{m'}}$ with $m' \in \mathbb{N}$. Whenever*

$$(\mathfrak{M}_i, \bar{A}_i, \bar{a}_i) \equiv^n_{MSO} (\mathfrak{N}_i, \bar{B}_i, \bar{b}_i) \text{ for all } 1 \leq i \leq k,$$

then also

$$\bigoplus^f_{1 \leq i \leq k} \mathfrak{M}_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, \bar{a}_1, \ldots, \bar{a}_k \equiv^n_{MSO} \bigoplus^f_{1 \leq i \leq k} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, \bar{b}_1, \ldots, \bar{b}_k.$$

.

Proof. We define a winning strategy for Duplicator in the game

$$EF^n_{\mathsf{MSO}}((\bigoplus^f_{1 \leq i \leq k} \mathfrak{M}_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, \bar{a}_1, \ldots, \bar{a}_k), (\bigoplus^f_{1 \leq i \leq k} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, \bar{b}_1, \ldots, \bar{b}_k))$$

out of her winning strategies in the games $EF^n_{\mathsf{MSO}}((\mathfrak{M}_i, \bar{A}_i, \bar{a}_i), (\mathfrak{N}_i, \bar{B}_i, \bar{b}_i))$ by induction on n.

Base step: $n = 0$, doing nothing is a strategy for Duplicator. We need to show that

$$(\bigoplus_{1\leq i\leq k}^{f} \mathfrak{M}_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, \bar{a_1}, \ldots, \bar{a_k})$$

and

$$(\bigoplus_{1\leq i\leq k}^{f} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, \bar{b_1}, \ldots, \bar{b_k})$$

agree on all atomic formulas. Now in the fusion structures, each atomic formula is defined by f in terms of a σ^*-quantifier free formula that is evaluated in the corresponding disjoint union structure. So it is enough to show that the disjoint union structures agree on all atomic σ^*-formulas and on their Boolean combinations. The initial match between the distinguished objects in $(\mathfrak{M}_i, \bar{A}_i, \bar{a_i})$ and $(\mathfrak{N}_i, \bar{B}_i, \bar{b_i})$ is a partial isomorphism for every $1 \leq i \leq k$, so it is also one for $\biguplus_{1\leq i\leq k} \mathfrak{M}_i, \bar{a_1}, \ldots, \bar{a_k}$ and $\biguplus_{1\leq i\leq k} \mathfrak{N}_i, \bar{b_1}, \ldots, \bar{b_k}$ i.e. the two disjoint union structures extended with FO parameters agree on all σ^*-atomic formulas. We still need to show that it is also one for $\biguplus_{1\leq i\leq k} \mathfrak{M}_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, \bar{a_1}, \ldots, \bar{a_k}$ and $\biguplus_{1\leq i\leq k} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, \bar{b_1}, \ldots, \bar{b_k}$ i.e. the two disjoint union structures extended with FO parameters and the closure under union of set parameters agree on all σ^*-atomic formulas. It is enough to point that for every parameter a_{i_j}, for every $I \subseteq \{i_1, \ldots, i_{m'}, \ldots, k_1, k_{m'}\}$ by construction of $\bigcup_{i\in I} A_i$ in $(\bar{A}_1, \ldots, \bar{A}_k)^{\cup}$, the following are equivalent:

- $\biguplus_{1\leq i\leq k} \mathfrak{M}_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, \bar{a_1}, \ldots, \bar{a_k} \models \bigcup_{i\in I} A_i a_{i_j}$,
- $\biguplus_{1\leq i\leq k} \mathfrak{M}_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, A_{i_l}, \bar{a_1}, \ldots, \bar{a_k} \models A_{i_l} a_{i_j}$ for some i_l in I.

Similarly for every parameter b_{i_j}, by construction of $\bigcup_{i\in I} B_i$ in $(\bar{B}_1, \ldots, \bar{B}_k)^{\cup}$, the following are equivalent:

- $\biguplus_{1\leq i\leq k} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, \bar{b_1}, \ldots, \bar{b_k} \models \bigcup_{i\in I} B_i b_{i_j}$,
- $\biguplus_{1\leq i\leq k} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, B_{i_l}, \bar{b_1}, \ldots, \bar{b_k} \models B_{i_l} b_{i_j}$ for some i_l in I.

But by Duplicator's winning strategy in the small structure games, we know that the following are equivalent:

- $\biguplus_{1\leq i\leq k} \mathfrak{M}_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, A_{i_l}, \bar{a_1}, \ldots, \bar{a_k} \models A_{i_l} a_{i_j}$ for some i_l in I.
- $\biguplus_{1\leq i\leq k} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, B_{i_l}, \bar{b_1}, \ldots, \bar{b_k} \models B_{i_l} b_{i_j}$ for some i_l in I.

So the following are also equivalent:

- $\biguplus_{1\leq i\leq k} \mathfrak{M}_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, \bar{a_1}, \ldots, \bar{a_k} \models \bigcup_{i\in I} A_i a_{i_j}$,
- $\biguplus_{1\leq i\leq k} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, \bar{b_1}, \ldots, \bar{b_k} \models \bigcup_{i\in I} B_i b_{i_j}$,

So the two extended disjoint union structures agree on all σ^*-atomic formulas. Now relying on the semantics of Boolean connectives, it can be shown by induction on the complexity of quantifier free sentences that they also agree on all Boolean combinations of atomic σ^*-sentences.

Inductive step: the inductive hypothesis says that whenever Duplicator has a winning strategy in $EF^n_{\mathsf{MSO}}((\mathfrak{M}_i, \bar{A}_i, \bar{a}_i), (\mathfrak{N}_i, \bar{B}_i, \bar{b}_i))$ for all $1 \leq i \leq k$, he also has one in

$$EF^n_{\mathsf{MSO}}((\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, \bar{a}_1, \ldots, \bar{a}_k), (\bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, \bar{b}_1, \ldots, \bar{b}_k)).$$

We want to show that this also holds when the length of the games is $n + 1$. Suppose Duplicator has a winning strategy in $EF^{n+1}_{\mathsf{MSO}}((\mathfrak{M}_i, \bar{A}_i, \bar{a}_i), (\mathfrak{N}_i, \bar{B}_i, \bar{b}_i))$ for all $1 \leq i \leq k$. We describe Duplicator's answer to Spoiler's first move in $EF^{n+1}_{\mathsf{MSO}}((\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i, \bar{A}_1, \ldots, \bar{A}_k, \bar{a}_1, \ldots, \bar{a}_k), (\bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i, \bar{B}_1, \ldots, \bar{B}_k, \bar{b}_1, \ldots, \bar{b}_k))$. It will then follow by induction hypothesis, that he has a winning strategy in the remaining n-length game.

- Spoiler's first move is a point move. Suppose Spoiler picks a in $\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i$. Then $a \in dom(\mathfrak{M}_i)$ for some $1 \leq i \leq k$. So Duplicator uses his winning strategy in $EF^{n+1}_{\mathsf{MSO}}((\mathfrak{M}_i, \bar{A}_i, \bar{a}_i), (\mathfrak{N}_i, \bar{B}_i, \bar{b}_i))$ to pick $b \in dom(\mathfrak{N}_i)$, so that he still has a winning strategy in $EF^n_{\mathsf{MSO}}((\mathfrak{M}_i, \bar{A}_i, \bar{a}_i, a), (\mathfrak{N}_i, \bar{B}_i, \bar{b}_i, b))$. By induction hypothesis he also has one in the remaining n-length MSO game between the following two structures:

  $$(\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, \bar{a}_1, \ldots, \bar{a}_k, a)$$

  and

  $$(\bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, \bar{b}_1, \ldots, \bar{b}_k, b)$$

- Spoiler's first move is a set move. Suppose Spoiler chooses a set A in the set of admissible subsets of $\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i$. Then A is necessarily of the form $A_1 \cup \ldots \cup A_k$, with A_i an admissible subset of $\mathfrak{M}_i$. We now define locally his response $B = B_1 \cup \ldots \cup B_k$, using his winning strategies in the small structures, so that he still has a winning strategy in $EF^n_{\mathsf{MSO}}((\mathfrak{M}_i, \bar{A}_i, A_i, \bar{a}_i), (\mathfrak{N}_i, \bar{B}_i, B_i, \bar{b}_i))$ for all $1 \leq i \leq k$. By induction hypothesis, he also has one in the remaining n-length MSO game between the following two structures:

  $$(\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i, (\bar{A}_1, A_1, \ldots, \bar{A}_k, A_k)^{\cup}, \bar{a}_1, \ldots, \bar{a}_k)$$

and

$$(\bigoplus_{1\le i\le k}^{f} \mathfrak{N}_i, (\bar{B}_1, B_1, \ldots, \bar{B}_k, B_k)^{\cup}, \bar{b_1}, \ldots, \bar{b_k}).$$

(Note that this is enough, because $A \in (\bar{A}_1, A_1, \ldots, \bar{A}_k, A_k)^{\cup}$.)

□

Now an analogue of this result for disjoint unions can easily be derived as a corollary of Theorem 3.3.23. For the convenience of the reader, we provide here the detailed argument:

Corollary 3.3.24. *Whenever* $(\mathfrak{M}_i, \bar{A}_i, \bar{a}_i) \equiv^n_{\mathsf{MSO}} (\mathfrak{N}_i, \bar{B}_i, \bar{b_i})$ *for all* $1 \le i \le k$ *(with* $\bar{a}_i$ *a sequence of first-order parameters of the form* $a_{i_1}, \ldots, a_{i_m}$ *with* $m \in \mathbb{N}$ *and* $\bar{A}_i$ *a sequence of set parameters of the form* $A_{i_1}, \ldots, A_{i_{m'}}$ *with* $m' \in \mathbb{N}$, *similarly for the* $\bar{b_i}$ *and* $\bar{B}_i$*), then also* $\biguplus_{1\le i\le k} \mathfrak{M}_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, \bar{a}_1, \ldots, \bar{a}_k \equiv^n_{\mathsf{MSO}} \biguplus_{1\le i\le k} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, \bar{b_1}, \ldots, \bar{b_k}$.

Proof. Let $(\mathfrak{M}_i, \bar{A}_i, \bar{a}_i) \equiv^n_{MSO} (\mathfrak{N}_i, \bar{B}_i, \bar{b_i})$ for all $1 \le i \le k$ (with $\bar{a}_i$ a sequence of first-order parameters of the form $a_{i_1}, \ldots, a_{i_m}$ with $m \in \mathbb{N}$ and $\bar{A}_i$ a sequence of set parameters of the form $A_{i_1}, \ldots, A_{i_{m'}}$ with $m' \in \mathbb{N}$, similarly for the $\bar{b_i}$ and $\bar{B}_i$).

Now consider the following expansions $\mathfrak{M}'_i$ and $\mathfrak{N}'_i$ of the σ structures $\mathfrak{M}_i$ and $\mathfrak{N}_i$ to $\sigma^* = \sigma \cup \{Q_1, \ldots, Q_k\}$: the interpretation of Q_j is empty in $\mathfrak{M}'_i$ (respectively $\mathfrak{N}'_i$) whenever $i \ne j$ and it is the domain of $\mathfrak{M}'_i$ (respectively $\mathfrak{N}'_i$) whenever $i = j$.

Clearly $(\mathfrak{M}'_i, \bar{A}_i, \bar{a}_i) \equiv^n_{\mathsf{MSO}} (\mathfrak{N}'_i, \bar{B}_i, \bar{b_i})$ for all $1 \le i \le k$.

Now consider a mapping f such that for every n-ary predicate $P \in \sigma^*$, $f(P) = Px_1 \ldots x_n$. By Theorem 3.3.23 we have that

$$\bigoplus_{1\le i\le k}^{f} \mathfrak{M}'_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, \bar{a}_1, \ldots, \bar{a}_k \equiv^n_{\mathsf{MSO}} \bigoplus_{1\le i\le k}^{f} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, \bar{b_1}, \ldots, \bar{b_k}.$$

Corollary 3.3.24 follows, because

$$\bigoplus_{1\le i\le k}^{f} \mathfrak{M}'_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, \bar{a}_1, \ldots, \bar{a}_k \text{ and } \bigoplus_{1\le i\le k}^{f} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, \bar{b_1}, \ldots, \bar{b_k}$$

are isomorphic to

$$\biguplus_{1\le i\le k} \mathfrak{M}_i, (\bar{A}_1, \ldots, \bar{A}_k)^{\cup}, \bar{a}_1, \ldots, \bar{a}_k \text{ and } \biguplus_{1\le i\le k} \mathfrak{N}_i, (\bar{B}_1, \ldots, \bar{B}_k)^{\cup}, \bar{b_1}, \ldots, \bar{b_k}$$

respectively. □

Another important corollary of Theorem 3.3.23 is the fact that fusions of MSO-Henkin structures are also MSO-Henkin structures. Let us the stress the importance of this fact, which is needed for the correctness of our main completeness argument.

Corollary 3.3.25. $\mathbb{A}_{\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i}$ *is closed under MSO parametric definability and so* $\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i$ *is a MSO-Henkin structure.*

Proof. First note that the following are equivalent:

- A is MSO parametrically definable in $\mathfrak{M}$,
- for some n, there is a finite sequence of parameters $\bar{a}, \bar{A}$ such that A is defined by a MSO formula φ of quantifier depth n using $\bar{a}, \bar{A}$,
- for some n, for every two points a and a' in $dom(\mathfrak{M})$, if they are MSO n-indistinguishable using $\bar{a}, \bar{A}$, then $a \in A$ iff $a' \in A$.

Now suppose for the sake of contradiction that there is $A \subseteq dom(\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i)$ MSO parametrically definable in $\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i$ using $\bar{a'}, \bar{A}'$, but $A \notin \mathbb{A}_{\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i}$. So it means that for some $1 \leq i \leq k$, $A_i = A \cap dom(\mathfrak{M}_i)$ is not MSO parametrically definable in $\mathfrak{M}_i$ i.e. there are two MSO parametrically indistinguishable points $a \in A$, $a' \notin A$. So for all n, for all sequence of parameters $\bar{a}, \bar{A}$ in $\mathfrak{M}_i$,

$$(\mathfrak{M}_i, \bar{a}, \bar{A}, a) \equiv^n_{\mathsf{FO(TC^1)}} (\mathfrak{M}_i, \bar{a}, \bar{A}, a')$$

and by the fusion theorem,[2]

$$\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i, \bar{a}, \bar{A}, \bar{a'}, \bar{A}', a \equiv^n_{\mathsf{FO(TC^1)}} \bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i, \bar{a}, \bar{A}, \bar{a'}, \bar{A}', a'$$

But this entails that A is not MSO parametrically definable in $\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i$ using $\bar{a'}, \bar{A}'$, which is a contradiction. □

Corollary 3.3.26. $\mathbb{A}_{\biguplus_{1\leq i\leq k}\mathfrak{M}_i}$ *is closed under MSO parametric definability and so* $\biguplus_{1\leq i\leq k}\mathfrak{M}_i$ *is a MSO-Henkin structure.*

Proof. Analogous to the proof of Corollary 3.3.25 (as $\mathbb{A}_{\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i} = \mathbb{A}_{\biguplus_{1\leq i\leq k}\mathfrak{M}_i}$). □

[2]There is no need to consider the case where $\bar{a'}, \bar{A}'$ is empty, because if a set is parametrically definable using no parameter, it is also definable using parameters.

Let us now consider the $\mathsf{FO(TC^1)}$ case. As TC moves can only be played when there are already two pebbles on the board, it is more convenient to show first a version of our $\mathsf{FO(TC^1)}$ fusion theorem in which each small structure comes with at least two parameters. This allows us to define Duplicator's answer to a TC move played in a big structure, by means of his winning strategies in the corresponding small structures. We then derive as a corollary the fusion theorem for non-parametrized structures.

Theorem 3.3.27 (Fusion Theorem for $\mathsf{FO(TC^1)}$). *Let $\bar{a_i}$, $\bar{b_i}$ be sequences of first-order parameters of the form $a_{i_1}, \ldots, a_{i_m}$, $b_{i_1}, \ldots, b_{i_m}$, with $m \in \mathbb{N}$ and and $m \geq 2$. Whenever*

$$(\mathfrak{M}_i, \bar{a_i}) \equiv^n_{\mathsf{FO(TC^1)}} (\mathfrak{N}_i, \bar{b_i}) \text{ for all } 1 \leq i \leq k,$$

then also

$$\bigoplus_{1\leq i\leq k}^{f} \mathfrak{M}_i, \bar{a_1}, \ldots, \bar{a_k} \equiv^n_{\mathsf{FO(TC^1)}} \bigoplus_{1\leq i\leq k}^{f} \mathfrak{N}_i, \bar{b_1}, \ldots, \bar{b_k}.$$

As a special case, whenever dealing with single point structures (structures which domain contains only one point), we allow the parameters to be non distinct objects.

Proof. We define a winning strategy for Duplicator in the game

$$EF^n_{\mathsf{FO(TC^1)}}((\bigoplus_{1\leq i\leq k}^{f} \mathfrak{M}_i, \bar{a_1}, \ldots, \bar{a_k}), (\bigoplus_{1\leq i\leq k}^{f} \mathfrak{N}_i, \bar{b_1}, \ldots, \bar{b_k}))$$

out of her winning strategies in the games $EF^n_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}), (\mathfrak{N}_i, \bar{b_i}))$ by induction on n.

Base step: $n = 0$, doing nothing is a strategy for Duplicator. We need to show that the $\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i, \bar{a_1}, \ldots, \bar{a_k}$ and $\bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i, \bar{b_1}, \ldots, \bar{b_k}$ agree on all atomic formulas. Now in the fusion structures, each atomic formula is defined by f in terms of a σ^*-quantifier free formula that is evaluated in the corresponding disjoint union structure. So it is enough to show that the disjoint union structures agree on all atomic σ^*-formulas and on their Boolean combinations. The initial match between the distinguished objects in $(\mathfrak{M}_i, \bar{a_i})$ and $(\mathfrak{N}_i, \bar{b_i})$ is a partial isomorphism for every $1 \leq i \leq k$, so it is also one for $\biguplus_{1\leq i\leq k} \mathfrak{M}_i, \bar{a_1}, \ldots, \bar{a_k}$ and $\biguplus_{1\leq i\leq k} \mathfrak{N}_i, \bar{b_1}, \ldots, \bar{b_k}$ i.e. the two disjoint union structures agree on all σ^*-atomic formulas. Now relying on the semantics of Boolean connectives, it can be shown by induction on the complexity of quantifier free sentences that they also agree on all Boolean combinations of atomic σ^*-sentences.

Inductive step: the inductive hypothesis says that whenever Duplicator has a winning strategy in $EF^n_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}), (\mathfrak{N}_i, \bar{b_i}))$ for some $(\mathfrak{M}_i, \bar{a_i}), (\mathfrak{N}_i, \bar{b_i})$ satisfying the required conditions on parameters and $1 \leq i \leq k$, he also has one in $EF^n_{\mathsf{FO(TC^1)}}((\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i, \bar{a_1}, \ldots, \bar{a_k}), (\bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i, \bar{b_1}, \ldots, \bar{b_k}))$.

We want to show that this also holds whenever the length of the game is $n+1$. Suppose Duplicator has a winning strategy in $EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i,\bar{a_i}),(\mathfrak{N}_i,\bar{b_i}))$ for all $1 \leq i \leq k$. We describe Duplicator's answer to Spoiler's first move in $EF^{n+1}_{\mathsf{FO(TC^1)}}((\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i,\bar{a_1},\dots,\bar{a_k}),(\bigoplus^f_{1\leq i\leq k}\mathfrak{N}_i,\bar{b_1},\dots,\bar{b_k}))$. It will then follow by induction hypothesis, that he has a winning strategy in the remaining n-length game.

- Spoiler's first move is an $\exists$ move. Suppose Spoiler chooses a point $a \in dom(\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i)$, then $a \in dom(\mathfrak{M}_i)$ for some $1 \leq i \leq k$. So Duplicator can use his winning strategy in $EF^{n}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i,\bar{a_i}),(\mathfrak{N}_i,\bar{b_i}))$ and pick a corresponding point b in the other structure. Now he still has a winning strategy in $EF^{n}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i,\bar{a_i},a),(\mathfrak{N}_i,\bar{b_i},b))$. So by induction hypothesis he also has one in the remaining n length game
$$EF^{n}_{\mathsf{FO(TC^1)}}((\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i,\bar{a_1},\dots,\bar{a_k},a),(\bigoplus^f_{1\leq i\leq k}\mathfrak{N}_i,\bar{b_1},\dots,\bar{b_k},b)).$$
- Spoiler's first move is a TC move. Suppose Spoiler chooses a set A in the set of admissible subsets of $\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i$. Then A is necessarily of the form $A_1 \cup \ldots \cup A_k$, with A_i an admissible subset (possibly empty) of $\mathfrak{M}_i$. Her response $B = B_1 \cup \ldots \cup B_k$ can now be defined locally for each B_i using her winning strategies in the small structures. So let Spoiler choose $A = A_1 \cup \ldots \cup A_k$. Keeping in mind that each non single point small structure comes with at least two distinct parameters, there are four cases:

  a) in $dom(\mathfrak{M}_i)$, there is a distinguished object inside, but also outside A_i, so Duplicator considers A_i together with these two parameters and constructs B_i by using his winning strategy in $EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i,\bar{a_i}),(\mathfrak{N}_i,\bar{b_i}))$.

  b) in $dom(\mathfrak{M}_i)$, there are only distinguished objects inside A_i[3], so Duplicator considers any one of these distinguished objects, let say a_j and looks at $A_i\backslash\{a_j\}$ together with some parameter inside A_i, so that he can use his winning strategy in $EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i,\bar{a_i}),(\mathfrak{N}_i,\bar{b_i}))$ to construct an answer that we call B'_i. Now $B_i = B'_i \cup \{b_j\}$;

  c) in $dom(\mathfrak{M}_i)$, there are only distinguished objects outside A_i,[4] so Duplicator similarly considers some distinguished object a_j and looks at $A_i \cup \{a_j\}$ together with some other parameter outside A_i, so that he can use his winning strategy in $EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i,\bar{a_i}),(\mathfrak{N}_i,\bar{b_i}))$ to construct an answer that we call B'_i. Now $B_i = B'_i\backslash\{b_j\}$;

  d) $\mathfrak{M}_i$ is a single point structure, then $B_i = \emptyset$ if $A_i = \emptyset$ and $B_i = dom(\mathfrak{M}_i)$ if $A_i = dom(\mathfrak{N}_i)$.

[3] Note that as a special case we may have $A_i = dom(\mathfrak{M}_i)$.

[4] Note that as a special case we may have $A_i = \emptyset$.

Once $B = B_1 \cup \ldots \cup B_k$ has been constructed, Spoiler picks two points $b \in B$ and $b' \notin B$. There are two cases:

1. b and b' belong to the domain of one and the same small structure $\mathfrak{N}_i$; now $dom(\mathfrak{M}_i)$ is as previously described in $a), b), c)$ (but not d), because two distinct points cannot belong to one and the same single point structure) and in each case Duplicator does the following:

    a) Duplicator answers with a, a' according to his winning strategy in $EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}), (\mathfrak{N}_i, \bar{b_i}))$, so that he still has a winning strategy in $EF^{n}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}, a, a'), (\mathfrak{N}_i, \bar{b_i}, b, b'))$. By induction hypothesis he also has one in the remaining n length game

    $$EF^{n}_{\mathsf{FO(TC^1)}}((\bigoplus_{1 \leq i \leq k}^{f} \mathfrak{M}_i, \bar{a_1}, \ldots, \bar{a_k}, a, a'), (\bigoplus_{1 \leq i \leq k}^{f} \mathfrak{N}_i, \bar{b_1}, \ldots, \bar{b_k}, b, b'));$$

    b) suppose first that $b' \neq b_j$, so Duplicator considers $A_i \backslash \{a_j\}$ together with a_j and with some other parameter inside this set and uses his winning strategy in $EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}), (\mathfrak{N}_i, \bar{b_i}))$ to pick corresponding a, a' in $\mathfrak{M}_i$, so that he still has a winning strategy in $EF^{n}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}, a, a'), (\mathfrak{N}_i, \bar{b_i}, b, b'))$. By induction hypothesis he also has one in the remaining n length game

    $$EF^{n}_{\mathsf{FO(TC^1)}}((\bigoplus_{1 \leq i \leq k}^{f} \mathfrak{M}_i, \bar{a_1}, \ldots, \bar{a_k}, a, a'), (\bigoplus_{1 \leq i \leq k}^{f} \mathfrak{N}_i, \bar{b_1}, \ldots, \bar{b_k}, b, b'));$$

    Next, suppose $b = b_j$. Then we choose $a = a_j$. The parameter a_j already matches b i.e. Duplicator has a winning strategy in

    $$EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}, a), (\mathfrak{N}_i, \bar{b_i}, b)),$$

    so Duplicator uses it to pick a', answering as if it was a point move (i.e a' has to be n-equivalent to b'), so that he still has a winning strategy in $EF^{n}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}, a, a'), (\mathfrak{N}_i, \bar{b_i}, b, b'))$. By induction hypothesis he also has one in the remaining n length game

    $$EF^{n}_{\mathsf{FO(TC^1)}}((\bigoplus_{1 \leq i \leq k}^{f} \mathfrak{M}_i, \bar{a_1}, \ldots, \bar{a_k}, a, a'), (\bigoplus_{1 \leq i \leq k}^{f} \mathfrak{N}_i, \bar{b_1}, \ldots, \bar{b_k}, b, b')).$$

    This works, except that there is the additional condition $a' \notin A_i$ that Duplicator must also maintain in order to respect the rules of the game. A slightly more refined argument shows, however that there has to be an n-equivalent point to b' which is outside

A_i. Indeed, instead of b, Spoiler could have picked any other point $b^* \in B_i$ together with $b' \notin B_i$ and Duplicator's winning strategy would have provided a correct answer $a^* \in A_i$, $a' \notin A_i$, which means that Duplicator would have found some a' point which is at least n-equivalent to b' and outside A_i (because if Duplicator has a winning strategy in $EF^n_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}, a^*, a'), (\mathfrak{N}_i, \bar{b_i}, b^*, b'))$ then he also has one in $EF^n_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}, a'), (\mathfrak{N}_i, \bar{b_i}, b'))$ and hence in $EF^n_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}, a, a'), (\mathfrak{N}_i, \bar{b_i}, b, b')))$.

c) suppose first that $b \neq b_j$, so Duplicator considers $A_i \cup \{a_j\}$ together with a_j and with some other parameter outside this set and uses his winning strategy in $EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}), (\mathfrak{N}_i, \bar{b_i}))$, so that he still has a winning strategy in $EF^n_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}, a, a'), (\mathfrak{N}_i, \bar{b_i}, b, b'))$. By induction hypothesis he also has one in the remaining n length game

$$EF^n_{\mathsf{FO(TC^1)}}((\bigoplus^f_{1 \leq i \leq k} \mathfrak{M}_i, \bar{a_1}, \ldots, \bar{a_k}, a, a'), (\bigoplus^f_{1 \leq i \leq k} \mathfrak{N}_i, \bar{b_1}, \ldots, \bar{b_k}, b, b'));$$

otherwise $b' = b_j$, then $a' = a_j$ because the parameter a_j already matches b' i.e. Duplicator has a winning strategy in

$$EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}, a'), (\mathfrak{N}_i, \bar{b_i}, b')),$$

so we can show by a similar argument as the one used in the above item, that he can use it to pick $a \in A_i$, so that he still has a winning strategy in $EF^n_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}, a, a'), (\mathfrak{N}_i, \bar{b_i}, b, b'))$. By induction hypothesis he also has one in the remaining n length game

$$EF^n_{\mathsf{FO(TC^1)}}((\bigoplus^f_{1 \leq i \leq k} \mathfrak{M}_i, \bar{a_1}, \ldots, \bar{a_k}, a, a'), (\bigoplus^f_{1 \leq i \leq k} \mathfrak{N}_i, \bar{b_1}, \ldots, \bar{b_k}, b, b')).$$

2. otherwise $b \in dom(\mathfrak{N}_i, \bar{b_i})$ and $b' \in dom(\mathfrak{N}_j, \bar{b_j})$ with $i \neq j$; we can again use a similar argument to show that Duplicator can use his winning strategy in

$$EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}), (\mathfrak{N}_i, \bar{b_i})) \text{ and } EF^{n+1}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_j, \bar{a_j}), (\mathfrak{N}_j, \bar{b_j}))$$

to pick a, a' in the right part of the structure (that is, inside or outside A_i), so that he still has a winning strategy in the games $EF^n_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, \bar{a_i}, a), (\mathfrak{N}_i, \bar{b_i}, b))$ and $EF^n_{\mathsf{FO(TC^1)}}((\mathfrak{M}_j, \bar{a_j}, a'), (\mathfrak{N}_j, \bar{b_j}, b'))$ (in the special case where for instance, $\mathfrak{M}_j$ is a single point structure, Duplicator picks the only available point in the other structure). By induction hypothesis he also has one in the remaining n length game

$$EF^n_{\mathsf{FO(TC^1)}}((\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i, \bar{a_1}, \ldots, \bar{a_k}, a, a'), (\bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i, \bar{b_1}, \ldots, \bar{b_k}, b, b')).$$

□

We now show a corollary of the preceding lemma, in which the small structures do not come with any distinguished objects:

Corollary 3.3.28. *Whenever* $\mathfrak{M}_i \equiv^n_{\mathsf{FO(TC^1)}} \mathfrak{N}_i$ *for all* $1 \leq i \leq k$*, then also* $\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i \equiv^n_{\mathsf{FO(TC^1)}} \bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i$*.*

Proof. We know that Spoiler's first two moves in the $\mathsf{FO(TC^1)}$-game of length $n+1$ bewteen $\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i$ and $\bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i$ must be quantifier moves, because the TC move can only be played once there are two pebbles on the board. Let us look at the first move. Suppose Spoiler plays a point $a \in dom(\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i)$. So $a \in dom(\mathfrak{M}_i)$ for some $1 \leq i \leq k$. By Duplicator's winning strategy in $EF^n_{\mathsf{FO(TC^1)}}(\mathfrak{M}_i, \mathfrak{N}_i)$, he has an answer $b \in dom(\mathfrak{N}_i)$ such that $(\mathfrak{M}_i, a) \equiv^n_{\mathsf{FO(TC^1)}} (\mathfrak{N}_i, b)$. Let us rename a with a_{i_1} and b with b_{i_1}. Similarly, for every $j \neq i$ such that $1 \leq j \leq k$, fix some random point a_{j_1} coming from the domain of $\mathfrak{M}_j$, Spoiler could have played this point and so Duplicator would have had an adequate answer b_{j_1} such that $(\mathfrak{M}_j, a_{j_1}) \equiv^n_{\mathsf{FO(TC^1)}} (\mathfrak{N}_j, b_{j_1})$. Now for the second round in the game, some point $a' = a_{l_2}$ or $b' = b_{l_2}$ coming from the domain of respectively $\mathfrak{M}_l$ or $\mathfrak{N}_l$ will be played by Spoiler and Duplicator will be able to answer so that $(\mathfrak{M}_l, a_{l_1}, a_{l_2}) \equiv^{n-2}_{\mathsf{FO(TC^1)}} (\mathfrak{N}_l, b_{l_1}, b_{l_2})$. Similarly, for each $\mathfrak{M}_j$ such that $j \neq l$, we can find points such that $(\mathfrak{M}_j, a_{j_1}, a_{j_2}) \equiv^{n-2}_{\mathsf{FO(TC^1)}} (\mathfrak{N}_i, b_{j_1}, b_{j_2})$. Now as for all $1 \leq i \leq k$, Duplicator has a winning strategy in $EF^{n-2}_{\mathsf{FO(TC^1)}}((\mathfrak{M}_i, a_{i_1}, a_{i_2}), (\mathfrak{N}_i, b_{i_1}, b_{i_2}))$, by the previous lemma, he has one in

$$EF^{n-2}_{\mathsf{FO(TC^1)}}(\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i, a_{1_1}, a_{1_2}, \ldots, a_{k_1}, a_{k_2}), (\bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i, b_{1_1}, b_{1_2}, \ldots, b_{k_1}, b_{k_2})),$$

so he also has one in $EF^{n-2}_{\mathsf{FO(TC^1)}}(\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i, a, a'), (\bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i, b, b'))$. □

Corollary 3.3.29. *Whenever* $\mathfrak{M}_i \equiv^n_{\mathsf{FO(TC^1)}} \mathfrak{N}_i$ *for all* $1 \leq i \leq k$*, then also* $\biguplus_{1\leq i\leq k} \mathfrak{M}_i \equiv^n_{\mathsf{FO(TC^1)}} \biguplus_{1\leq i\leq k} \mathfrak{N}_i$*.*

Proof. Analogous to the proof of Corollary 3.3.24. □

Corollary 3.3.30. $\mathbb{A}_{\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i}$ *is closed under* $\mathsf{FO(TC^1)}$ *parametric definability and so* $\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i$ *is a* $\mathsf{FO(TC^1)}$*-Henkin structure.*

Proof. Analogous to the proof of Corollary 3.3.25. □

Corollary 3.3.31. $\mathbb{A}_{\biguplus_{1\leq i\leq k}\mathfrak{M}_i}$ *is closed under FO(TC^1) parametric definability and so* $\biguplus_{1\leq i\leq k}\mathfrak{M}_i$ *is a FO(TC^1)-Henkin structure.*

Proof. Analogous to the proof of Corollary 3.3.26. □

In the FO(LFP^1) case, the situation parallels the FO(TC^1) case. As LFP moves can only be played when there is already one pebble on the board, it is more convenient to show first a version of our FO(LFP^1) fusion theorem in which each small structure comes with at least one FO parameter. This allows us to define Duplicator's answer to a LFP move played in the big structure, by means of his winning strategies in the small structures. We then derive as a corollary the fusion theorem for non-parametrized structures.

Theorem 3.3.32 (Fusion Theorem for FO(LFP^1))**.** *Let* $\bar{a_i}$, $\bar{b_i}$ *be* non empty *sequences of first-order parameters of the form* $a_{i_1},\dots,a_{i_m}$, $b_{i_1},\dots,b_{i_m}$, *with* $m\in\mathbb{N}$ *and* $\bar{A}_i$, $\bar{B}_i$ *sequences of set parameters of the form* $A_{i_1},\dots,A_{i_{m'}}$, $B_{i_1},\dots,B_{i_{m'}}$ *with* $m'\in\mathbb{N}$. *Whenever*

$$(\mathfrak{M}_i,\bar{A}_i,\bar{a_i})\equiv^n_{\mathsf{FO(LFP^1)}}(\mathfrak{N}_i,\bar{B}_i,\bar{b_i})\ \textit{for all}\ 1\leq i\leq k,$$

then also

$$\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i,(\bar{A}_1,\dots,\bar{A}_k)^{\cup},\bar{a_1},\dots,\bar{a_k}\equiv^n_{\mathsf{FO(LFP^1)}}\bigoplus^f_{1\leq i\leq k}\mathfrak{N}_i,(\bar{B}_1,\dots,\bar{B}_k)^{\cup},\bar{b_1},\dots,\bar{b_k}.$$

Proof. We define a winning strategy for Duplicator in the FO(LFP^1)-game of length n in between the structures $(\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i,(\bar{A}_1,\dots,\bar{A}_k)^{\cup},\bar{a_1},\dots,\bar{a_k})$ and $(\bigoplus^f_{1\leq i\leq k}\mathfrak{N}_i,(\bar{B}_1,\dots,\bar{B}_k)^{\cup},\bar{b_1},\dots,\bar{b_k})$ out of her winning strategies in the games $EF^n_{\mathsf{FO(LFP^1)}}((\mathfrak{M}_i,\bar{A}_i,\bar{a_i}),(\mathfrak{N}_i,\bar{B}_i,\bar{b_i}))$ by induction on n.

Base step: $n=0$, doing nothing is a strategy for Duplicator (this can be justified by a similar argument as in the MSO case).

Inductive step: the inductive hypothesis says that whenever Duplicator has a winning strategy in $EF^n_{\mathsf{FO(LFP^1)}}((\mathfrak{M}_i,\bar{A}_i,\bar{a_i}),(\mathfrak{N}_i,\bar{B}_i,\bar{b_i}))$ for some pairs of structures $(\mathfrak{M}_i,\bar{A}_i,\bar{a_i}),(\mathfrak{N}_i,\bar{B}_i,\bar{b_i})$ satisfying the required conditions on parameters and $1\leq i\leq k$, he also has one in the FO(LFP^1)-game of length n in between $(\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i,(\bar{A}_1,\dots,\bar{A}_k)^{\cup},\bar{a_1},\dots,\bar{a_k})$ and $(\bigoplus^f_{1\leq i\leq k}\mathfrak{N}_i,(\bar{B}_1,\dots,\bar{B}_k)^{\cup},\bar{b_1},\dots,\bar{b_k})$.

We want to show that this also holds when the length of the games is $n+1$. Suppose Duplicator has a winning strategy in $EF^{n+1}_{\mathsf{FO(LFP^1)}}((\mathfrak{M}_i,\bar{A}_i,\bar{a_i}),(\mathfrak{N}_i,\bar{B}_i,\bar{b_i}))$ for all $1\leq i\leq k$. We describe Duplicator's answer to Spoiler's first move in the FO(LFP^1)-game of length $n+1$ in between $(\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i,(\bar{A}_1,\dots,\bar{A}_k)^{\cup},\bar{a_1},\dots,\bar{a_k})$ and $(\bigoplus^f_{1\leq i\leq k}\mathfrak{N}_i,(\bar{B}_1,\dots,\bar{B}_k)^{\cup},\bar{b_1},\dots,\bar{b_k})$. It then follows by induction hypothesis, that he has a winning strategy in the remaining n-length game.

- Spoiler's first move is an $\exists$ move.

  Same argument as for MSO and $\mathsf{FO(TC^1)}$.

- Spoiler's first move is a $\forall$ move.

  Symmetric.

- Spoiler's first move is a GFP move.

  Suppose Spoiler chooses a set A in the set of admissible subsets of $\bigoplus^f_{1\leq i\leq k}\mathfrak{M}_i$ with some pebble $a_{i_j}\in A$. Then A is necessarily of the form $A_1\cup\ldots\cup A_k$, with A_i an admissible subset of $\mathfrak{M}_i$. Her response $B=B_1\cup\ldots\cup B_k$ can now be defined locally for each B_i using her winning strategies in the small structures. So let Spoiler choose $A=A_1\cup\ldots\cup A_k$. Keeping in mind that each small structure comes with at least one parameter, there are four cases:

  1) in $dom(\mathfrak{M}_i)$, there is a distinguished object inside A_i and $A_i\neq dom(\mathfrak{M}_i)$, so Duplicator considers A_i together with this parameter and constructs B_i by using his winning strategy in $EF^{n+1}_{\mathsf{FO(LFP^1)}}((\mathfrak{M}_i,\bar{A}_i,\bar{a}_i),(\mathfrak{N}_i,\bar{B}_i,\bar{b}_i))$.
  2) in $dom(\mathfrak{M}_i)$, there are only distinguished objects outside A_i and $A_i\neq\emptyset$, so Duplicator considers any one of these distinguished objects, let say a_j and looks at $A_i\cup\{a_j\}$, so that he can use his winning strategy in $EF^{n+1}_{\mathsf{FO(LFP^1)}}((\mathfrak{M}_i,\bar{A}_i,\bar{a}_i),(\mathfrak{N}_i,\bar{B}_i,\bar{b}_i))$ to construct an answer that we call B'_i. Now $B_i=B'_i\backslash\{b_j\}$. This is a correct answer, because the (posimorphism) condition to be maintained is that for every pebble a_l on the board at the end of the game, $a_l\in A_i\Rightarrow b_l\in B_i$. But by Duplicator's winning strategy in $EF^{n+1}_{\mathsf{FO(LFP^1)}}((\mathfrak{M}_i,\bar{A}_i,A_i\cup\{a_j\},\bar{a}_i),(\mathfrak{N}_i,\bar{B}_i,B'_i,\bar{b}_i))$, we know already that for every such pebble, $a_l\in A_i\cup\{a_j\}\Rightarrow b_l\in B'_i$, so also $a_l\in A_i\Rightarrow b_l\in B'_i\backslash\{b_j\}$, since the winning conditions will assure that $a_l=a_j$ if and only if $b_l=b_j$.
  3) $B_i=dom(\mathfrak{M}_i)$. So $A_i=dom(\mathfrak{N}_i)$. As pebbles are only chosen using Duplicator's winning strategies in the small structures, the posimorphism condition will be maintained.
  4) $B_i=\emptyset$. So $A_i=\emptyset$. As no pebble can belong to this set, the posimorphism condition will be maintained.

  Now that $B=B_1\cup\ldots\cup B_k$ has been constructed, Spoiler picks a new element $b\in B$ which belongs to the domain of one particular small structure $\mathfrak{N}_i$ (so $b\in B_i$) and $dom(\mathfrak{M}_i)$ is as previously described either in 1), 2) or 3) (but not 4), because b cannot belong to the empty set) and in each case Duplicator does the following:

  1) Duplicator answers with a according to his winning strategy in

$$EF^{n+1}_{\mathsf{FO(LFP^1)}}((\mathfrak{M}_i,\bar{A}_i,\bar{a}_i),(\mathfrak{N}_i,\bar{B}_i,\bar{b}_i));$$

2) Duplicator again considers $A_i \cup \{a_j\}$ and answers according to his winning strategy in $EF^{n+1}_{\mathsf{FO(LFP^1)}}((\mathfrak{M}_i, \bar{A}_i, A_i \cup \{a_j\}, \bar{a}_i), (\mathfrak{N}_i, \bar{B}_i, B_i', \bar{b}_i))$. This is safe, because the pebble to be chosen may be assumed to be fresh, so it won't be a_j;
3) Duplicator picks some random pebble a_j in $dom(\mathfrak{M}_i)$ and considers $dom(\mathfrak{M}_i)\backslash\{a_j\}$. His winning strategy provides him with a correct answer.

So in any case (either 1), 2) or 3)), Duplicator has a winning strategy in $EF^{n}_{\mathsf{FO(LFP^1)}}((\mathfrak{M}_i, \bar{A}_i, A_i, \bar{a}_i, a), (\mathfrak{N}_i, \bar{B}_i, B_i, \bar{b}_i, b))$. Now for all $j \neq i$, $1 \leq j \leq k$, he also has one in $EF^{n}_{\mathsf{FO(LFP^1)}}((\mathfrak{M}_j, \bar{A}_j, A_j, \bar{a}_j), (\mathfrak{N}_j, \bar{B}_j, B_j, \bar{b}_j))$. So by induction hypothesis, he has one in the remaining n-length $\mathsf{FO(LFP^1)}$ game between the following two structures:

$$(\bigoplus_{1\leq i\leq k}^{f} \mathfrak{M}_i, (\bar{A}_1, A_1, \ldots, \bar{A}_k, A_k)^{\cup}, \bar{a}_1, \ldots, \bar{a}_k, a)$$

and

$$(\bigoplus_{1\leq i\leq k}^{f} \mathfrak{N}_i, (\bar{B}_1, B_1, \ldots, \bar{B}_k, B_k)^{\cup}, \bar{b}_1, \ldots, \bar{b}_k, b)$$

- Spoiler's first move is a LFP move.

  Symmetric.

□

Corollary 3.3.33. *Whenever $\mathfrak{M}_i \equiv^n_{\mathsf{FO(LFP^1)}} \mathfrak{N}_i$ for all $1 \leq i \leq k$, then also $\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i \equiv^n_{\mathsf{FO(LFP^1)}} \bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i$.*

Proof. We know that Spoiler's first move in $EF^{n+1}_{\mathsf{FO(LFP^1)}}(\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i, \bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i)$ must be a FO quantifier move, because the LFP move can only be played once there is a pebble on the board. Let us look at the first move. Suppose Spoiler plays a point $a \in dom(\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i)$. So $a \in dom(\mathfrak{M}_i)$ for some $1 \leq i \leq k$. By Duplicator's winning strategy in $EF^{n}_{\mathsf{FO(LFP^1)}}(\mathfrak{M}_i, \mathfrak{N}_i)$, he has an answer $b \in dom(\mathfrak{N}_i)$ such that $(\mathfrak{M}_i, a) \equiv^n_{\mathsf{FO(LFP^1)}} (\mathfrak{N}_i, b)$. Let us rename a with a_i and b with b_i. Similarly, for every $j \neq i$ such that $1 \leq j \leq k$, fix some random point a_j coming from the domain of $\mathfrak{M}_j$, Spoiler could have played this point and so Duplicator would have had an adequate answer b_j such that $(\mathfrak{M}_j, a_j) \equiv^n_{\mathsf{FO(LFP^1)}} (\mathfrak{N}_j, b_j)$. Now as for all $1 \leq i \leq k$, Duplicator has a winning strategy in $EF^{n-1}_{\mathsf{FO(LFP^1)}}((\mathfrak{M}_i, a_i), (\mathfrak{N}_i, b_i))$, by the previous lemma, he has one in $EF^{n-1}_{\mathsf{FO(LFP^1)}}(\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i, a_1, \ldots, a_k), (\bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i, b_1, \ldots, b_k))$, so he also has one in $EF^{n-1}_{\mathsf{FO(LFP^1)}}(\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i, a), (\bigoplus^f_{1\leq i\leq k} \mathfrak{N}_i, b))$. □

Corollary 3.3.34. *Whenever* $\mathfrak{M}_i \equiv^n_{\mathsf{FO(LFP^1)}} \mathfrak{N}_i$ *for all* $1 \leq i \leq k$*, then also* $\biguplus_{1\leq i\leq k} \mathfrak{M}_i \equiv^n_{\mathsf{FO(LFP^1)}} \biguplus_{1\leq i\leq k} \mathfrak{N}_i$*.*

Proof. Analogous to the proof of Corollary 3.3.24. □

Corollary 3.3.35. $\mathbb{A}_{\bigoplus^f_{1\leq i\leq k}}$ *is closed under* FO(LFP^1) *parametric definability and so* $\bigoplus^f_{1\leq i\leq k} \mathfrak{M}_i$ *is a* FO(LFP^1)*-Henkin structure.*

Proof. Analogous to the proof of Corollary 3.3.25. □

Corollary 3.3.36. $\mathbb{A}_{\biguplus_{1\leq i\leq k} \mathfrak{M}_i}$ *is closed under* FO(LFP^1) *parametric definability and so* $\biguplus_{1\leq i\leq k} \mathfrak{M}_i$ *is a* FO(LFP^1)*-Henkin structure.*

Proof. Analogous to the proof of Corollary 3.3.26. □

3.4 Putting it Together: Completeness on Finite Trees

3.4.1 Forests and Operations on Forests

In Section 3.4.2, we will prove that no Λ-sentence can distinguish Λ-Henkin-models of $\vdash^{tree}_{\Lambda}$ from standard models of $\vdash^{tree}_{\Lambda}$. More precisely, we will show that for each n, every definably well-founded Λ-quasi-tree is n-Λ-equivalent to a finite tree. In order to give an inductive proof, it will be more convenient to consider a stronger version of this result concerning a class of finite and infinite Henkin structures that we call *quasi-forests*. In this section, we give the definition of quasi-forest and we show how they can be combined into bigger quasi-forests using the notion of fusion from Section 3.3. Whenever quasi forests are finite, we simply call them *finite forests*. As a simple example, consider a finite tree and remove the root node, then it is no longer a finite tree. Instead it is a finite sequence of trees, whose roots stand in a linear (sibling) order.[5] It does not have a unique root, but it does have a unique *left-most root*. For technical reasons it will be convenient in the definition of quasi forests to add an extra monadic predicate R labeling the roots.

Definition 3.4.1 (Λ-quasi-forest)**.** Let $T = (dom(T), <, \prec, P_1, \ldots, \ldots P_n, \mathbb{A}_T)$ be a Λ-quasi-tree. Given a node a in T, consider the Λ-substructure of T generated by the set $\{x \mid \exists z(a \preceq z \wedge z \leq x)\}$, which is the set formed by a together with all its siblings to the right and their descendants. The Λ-quasi-forest T_a is obtained by labeling each root in this substructure with R ($Rx \Leftrightarrow_{def} \neg\exists y\; y < x$). Whenever T is a tree, we simply call T_a a forest.

[5]Note that, as far as roots are concerned, two nodes can be siblings without sharing any parent. This would not happen in a quasi tree.

We will show in our main proof of completeness that for each n and for each node a in a definably well-founded Λ-quasi-tree, the Λ-quasi-forest T_a is n-Λ-equivalent to a finite forest. Our proof will use a notion of composition of Λ-quasi-forests which is a special case of fusion. Given a single node forest F_1 and two Λ-quasi-forests F_2 and F_3, we construct a new Λ-quasi-forest $\bigoplus^{f^\triangle}(F_1, F_2, F_3)$ by letting the unique element in F_1 be the left-most root, the roots of F_2 become the children of this node and the roots of F_3 become its siblings to the right. We then derive a corollary of the Λ-fusion theorem for compositions of Λ-quasi-forests and use it in Section 3.4.2.

Definition 3.4.2. Let $\sigma = \{<, \prec, R, P_1, \ldots, P_n\}$, be a relational vocabulary with only monadic predicates except $<$ and $\prec$. Given three additional monadic predicates Q_1, Q_2, Q_3, we define a mapping $f^\triangle$ from σ to quantifier-free formulas over $\sigma \cup \{Q_1, Q_2, Q_3\}$ by letting

- $f^\triangle(P_i) = P_i(x_1)$
- $f^\triangle(<) = x_1 < x_2 \vee (Q_1(x_1) \wedge Q_2(x_2))$
- $f^\triangle(\prec) = x_1 \prec x_2 \vee (Q_1(x_1) \wedge Q_3(x_2) \wedge R(x_2))$
- $f^\triangle(R) = (Q_3(x_1) \wedge R(x_1)) \vee Q_1(x_1)$

Corollary 3.4.3. *Let F_1 be a single node forest and F_2, F_3 Λ-quasi forests. If $F_2 \equiv^n_\Lambda F'_2$ and $F_3 \equiv^n_\Lambda F'_3$ then $\bigoplus^{f^\triangle}(F_1, F_2, F_3) \equiv^n_\Lambda \bigoplus^{f^\triangle}(F_1, F'_2, F'_3)$.*

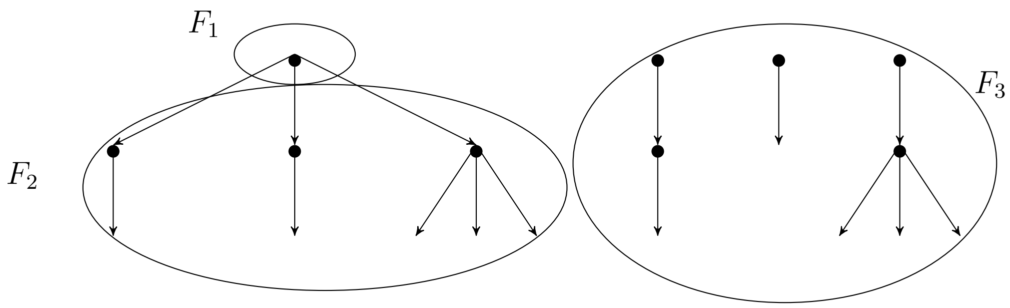


Figure 3.6: A composition of forests using the mapping $f^\triangle$

Figure 3.6 represents a composition of three forests F_1, F_2, F_3 which uses the mapping $f^\triangle$. Only new $<_{ch}$-arrows are represented, linking the unique node in F_1 to the root nodes in F_2. But new $\preceq$-links have also been added and the roots in F_3 have became the siblings to the right of the root in F_1. This is implicitly indicated by the left to right organization of the picture.

3.4.2 Main Proof of Completeness

Lemma 3.4.4. *For all $n \in \mathbb{N}$, every definably well-founded Λ-quasi-tree of finite signature is n-Λ-equivalent to a finite tree. In particular, a Λ-sentence is valid on definably well-founded Λ-quasi-trees iff it is valid on finite trees.*

Proof. Let T be a Λ-quasi-tree, without loss of generality assume that a monadic predicate R labels its root (and only that node in the tree). During this proof, it will be convenient to work with Λ-quasi-forests. Note that finite Λ-quasi-forests are simply finite forests and finite Λ-quasi-trees are simply finite trees. Let X_n be the set of all nodes a of T for which it holds that T_a is n-Λ-equivalent to a finite forest. We first show that 'belonging to X_n" is a property definable in T (Claim 1). We then use the induction scheme to show that every node of a definably well-founded Λ-quasi-tree (so in particular, the root) has this property (Claim 2).

Claim 1: X_n is invariant for $n+1$-Λ-equivalence (i.e., $(T,a) \equiv^{\Lambda}_{n+1} (T,b)$ implies that $a \in X_n$ iff $b \in X_n$), and hence is defined by a Λ-formula of quantifier depth $n+1$.

Proof of claim. Suppose that $(T,a) \equiv^{\Lambda}_{n+1} (T,b)$. We will show that $T_a \equiv^{\Lambda}_{n} T_b$, and hence, by the definition of X_n, $a \in X_n$ iff $b \in X_n$. By the definition of Λ-quasi-forests, $dom(T_a) = \{x \mid \exists z(a \preceq z \wedge z \leq x)\}$. Let φ be any Λ-sentence of quantifier depth n. We can assume without loss of generality that φ does not contain the variables z and x (otherwise we can rename in φ these two variables). By lemma 3.3.4, $(T,a) \models REL(\varphi, \exists z(a \preceq z \wedge z \leq x), x)$ iff $T_a \models \varphi$. Notice that $REL(\varphi, \exists z(a \preceq z \wedge z \leq x), x)$ expresses precisely that φ holds in (T,a) within the subforest T_a. Moreover, the quantifier depth of $REL(\varphi, \exists z(a \preceq z \wedge z \leq x)$ is at most $n+1$. It follows that $(T,a) \models REL(\varphi, \exists z(a \preceq z \wedge z \leq x), x)$ iff $(T,b) \models REL(\varphi, \exists z(b \preceq z \wedge z \leq x), x)$, and hence $T_a \models \varphi$ iff $T_b \models \varphi$.

For the second part of the claim, note that by Lemma 3.3.11, up to logical equivalence, there are only finitely many Λ-formulas of any given quantifier depth, as the vocabulary is finite. $\dashv$

Claim 2: If all descendants and siblings to the right of a belong to X_n, then a itself belongs to X_n.

Proof of claim. Let us consider the case where a has both a descendant and a following sibling (all other cases are simpler). Then, by axioms T3, T5, T8, T9 and T10, a has a first child b, and an immediate next sibling c. Moreover, we know that both b and c are in X_n. In other words, T_b and T_c are n-Λ-equivalent to finite forests T'_b and T'_c. Now, we construct a finite Λ-quasi-forest T'_a by taking a $f^{\triangle}$-fusion of T'_b, T'_c and of the Λ-substructure of T generated by $\{a\}$, which unique element becomes a common parent of all roots of T'_b and a left sibling of all roots of T'_c. So we get $T'_a = \bigoplus^{f^{\triangle}}(T \restriction \{a\}, T'_b, T'_c))$. It is not hard to see that T'_a is again a finite forest. Moreover, by the fusion theorem, $\bigoplus^{f^\triangle}(T \restriction \{a\}, T_b, T_c)) \equiv_n^\Lambda T'_a$. Now to show that $\bigoplus^{f^\triangle}(T \restriction \{a\}, T_b, T_c))$ is isomorphic to T_a (which entails $T_a \equiv_n^\Lambda T'_a$ i.e. T_a is n-Λ-equivalent to a finite forest), it is enough to show $\mathbb{A}_{T_a} = \mathbb{A}_{\bigoplus^{f^\triangle}(T\restriction\{a\},T_b,T_c)}$. It holds that $\mathbb{A}_{\bigoplus^{f^\triangle}(T\restriction\{a\},T_b,T_c)} \subseteq \mathbb{A}_{T_a}$ because we can define in T_a each such union of sets by means of a disjunction. Now to show $\mathbb{A}_{T_a} \subseteq \mathbb{A}_{\bigoplus^{f^\triangle}(T\restriction\{a\},T_b,T_c)}$, take $A \in \mathbb{A}_{T_a}$, so $A = A_1 \cup A_2 \cup A_3$ with $A_1 \in \mathbb{A}_{T\restriction\{a\}}$, $A_2 \in \mathbb{A}_{T_b}$, $A_3 \in \mathbb{A}_{T_c}$. The domain of each of these three structures is definable in T_a. Let φ_1 define $dom(T \restriction \{a\})$, φ_2 define $dom(T_b)$ and φ_3 define $dom(T_c)$. So each A_i component is definable in T_a (just take the conjunction $\varphi_i(x) \wedge Ax$). But then A_i was already definable in $\bigoplus^{f^\triangle}(T \restriction \{a\}, T_b, T_c)$ (by construction of this structure). ⊣

It follows from these two claims, by the induction scheme for definable properties, that X_n contains all nodes of the Λ-quasi-tree, including the root, and hence T is n-Λ-equivalent to a finite tree (to a finite forest actually, but the root of the Λ-quasi-tree being labelled by R, it can also be viewed as a Λ-quasi-forest). For the second statement of the lemma, it suffices to note that every Λ-sentence has a finite vocabulary and a finite quantifier depth. □

Theorem 3.4.5. *Let* $\Lambda \in \{\mathsf{MSO}, \mathsf{FO(TC^1)}, \mathsf{FO(LFP^1)}\}$. *The* Λ*-theory of finite trees is completely axiomatized by* $\vdash_\Lambda^{tree}$.

Proof. Theorem 3.4.5 follows directly from Lemma 3.4.4 and Corollary 3.2.19. □

3.4.3 Definability of the Class of Finite Trees

Proposition 3.4.6 below shows together with Theorem 3.4.5 that on standard structures, the set of $\vdash_\Lambda^{tree}$ consequences actually defines the (not FO-definable) class of finite trees. That is, $\vdash_\Lambda^{tree}$ has *no infinite standard model* at all.

Proposition 3.4.6. *Let* $\Lambda \in \{\mathsf{FO(TC^1)}, \mathsf{FO(LFP^1)}, \mathsf{MSO}\}$. *On standard structures, there is a* Λ*-formula which defines the class of finite trees.*

Sketch of the proof. It is enough to show it for $\Lambda = \mathsf{FO(TC^1)}$. It follows by Section 2.1.3 that it also holds for MSO and $\mathsf{FO(LFP^1)}$.

We merely give a sketch of the proof. For additional details we refer the reader to [96]. It can be shown that on standard structures, the finite conjunction of the axioms T1–T11 in Figure 3.5 "almost" defines the class of finite trees, i.e., any finite structure satisfying this conjunction is a finite tree. Now we will explain how to construct another sentence, which together with this one, actually defines on arbitrary standard structures the class of finite trees. Let L be a shorthand for the formula labeling the leaves in the tree ($Lx \Leftrightarrow_{def} \neg\exists y x < y$) and R a shorthand for the formula labelling the root ($Rx \Leftrightarrow_{def} \neg\exists y y < x$). Consider the depth-first left-to-right ordering of nodes in a tree and the $\mathsf{FO(TC^1)}$ formula $\varphi(x, y)$ saying "the node that comes after x in this ordering is y":

$$\varphi(x,y) :\approx (\neg Lx \wedge x <_{ch} y \wedge \neg\exists zz \prec y) \vee (Lx \wedge x \prec_{ch} y) \vee (Lx \wedge \neg\exists zx \prec z \wedge \exists z(z < x \wedge z \prec_{ch} y \wedge \neg\exists ww < x \wedge z < w \wedge \exists uw \prec_{ch} u))$$

There is also a $\mathsf{FO}(\mathsf{TC}^1)$ formula which says that "x is the very last node in this ordering". $\varphi(x,y)$ can be combined with this formula into an $\mathsf{FO}(\mathsf{TC}^1)$ formula χ expressing that the tree is finite by saying that (we rely here for the interpretation of χ on the alternative semantics for the TC operator given in Proposition 2.1.11) "there is a finite sequence of nodes $x_1 \dots x_n$ such that x_1 is the root, x_{i+1} the node that comes after x_i in the above ordering, for all i, and x_n is the very last node of the tree in the above ordering".

$$\chi :\approx \exists u \exists z(Rz \wedge [TC_{xy}\varphi](z,u) \wedge \neg\exists u'(u \neq u' \wedge [TC_{xy}\varphi](u,u')))$$

□

Theorem 3.4.7. *The set of* $\vdash^{tree}_{\Lambda}$ *consequences* defines *the class of finite trees.*

Proof. By Proposition 3.4.6 we can express in Λ by means of some formula χ that a structure is a finite tree. So χ is necessarily a consequence of $\vdash^{tree}_{\Lambda}$ (as it is a Λ-formula valid on the class of finite trees). □

3.5 Finite Linear Orders

Let us note that a simplified version of this method can be used in order to show the completeness of MSO, $\mathsf{FO}(\mathsf{TC}^1)$ and $\mathsf{FO}(\mathsf{LFP}^1)$ on finite node-labelled linear orders. The relevant simpler axioms are the ones listed in Figures 3.1, 3.7 and respectively, Figures 3.2, 3.3 and 3.4.

L1.	$\forall xyz(x < y \wedge y < z \rightarrow x < z)$	$<$ is transitive
L2.	$\neg\exists x(x < x)$	$<$ is irreflexive
L3.	$\forall xy(x < y \rightarrow \exists z(x <_{ch} z \wedge z \leq y))$	immediate children
L4.	$\exists x \forall y \neg(y < x)$	there is a root
L5.	$\forall xy(x = y \vee x < y \vee y < x)$	$<$ is total
Ind.	$\forall x(\forall y((x < y \rightarrow \varphi(y)) \rightarrow \varphi(x)) \rightarrow \forall x\varphi(x)$	

where

$\varphi(x)$ ranges over Λ-formulas in one free variable x

and

$x <_{ch} y$ is shorthand for $x < y \wedge \neg\exists z(z < y \wedge x < z)$

Figure 3.7: Specific axioms on finite linear orders

3.6 Conclusion

In this chapter, taking inspiration from Kees Doets [57] we developed a uniform method for obtaining complete axiomatizations of fragments of MSO on finite trees. For that purpose, we had to adapt classical tools and notions from finite model theory to the specificities of Henkin semantics. The presence of admissible subsets called for some refinements in model theoretic constructions such as formation of substructure or disjoint union. Also, we noticed that not every Ehrenfeucht-Fraïssé game that has been used for $\mathsf{FO}(\mathsf{TC}^1)$ was suitable to use on Henkin-structures. We focused on a game which does not seem to have been used previously in the literature. We also established analogues of the FO Feferman-Vaught theorem for MSO, $\mathsf{FO}(\mathsf{TC}^1)$ and $\mathsf{FO}(\mathsf{LFP}^1)$ on Henkin-structures (let us recall that related work for the case of standard structures can be found in [106]). We considered fusions, a particular case of the Feferman-Vaught notion of generalized product and obtained results for Henkin-structures which might be interesting to generalize and use in other contexts.

We applied our method to MSO, $\mathsf{FO}(\mathsf{TC}^1)$ and $\mathsf{FO}(\mathsf{LFP}^1)$, but it would be worth also examining other fragments of MSO or logics such as monadic deterministic transitive closure logic ($\mathsf{FO}(\mathsf{DTC}^1)$, which was advocated in [128] as particularly relevant in the context of applications to model-theoretic syntax) or monadic alternating transitive closure logic ($\mathsf{FO}(\mathsf{ATC}^1)$), see also [38].

An important feature of our main completeness argument (the idea of which was borrowed from Kees Doets) is the way we used the inductive scheme of Figure 3.5. Hence, extending our approach to another class of finite structures would involve finding a comparable scheme. We also know that we should focus on a logic which is decidable on this class, as on finite structures recursive enumerability is equivalent to decidability (as long as the model-checking is decidable). This suggests that other natural candidates would be fragments of MSO on classes of finite structures with bounded treewidth.

Finally, let us recall that we noted in Chapter 2 that MSO is also known to be decidable over infinite trees and over linear orders of order type ω. It would be interesting to look for a model-theoretic argument which would work on a Henkin model and produce an intended model of one of these theories in a way comparable to what we did here or to what Keisler did in [94]. Note that related complete axiomatizations of monadic theories of classes of infinite structures can be found in [36], [124] and [136], but that instead of relying on Henkin-semantics, the completeness proofs there are based on automata-theoretic techniques.

Chapter 4

Interpolation for Linear Temporal Languages

4.1 Introduction

Craig's interpolation theorem in classical model theory dates back from the late fifties [44]. It states that if a first-order formula φ (semantically) entails another first-order formula ψ, then there is an *interpolant* first-order formula θ, such that every non-logical symbol in θ occurs both in φ and ψ, φ entails θ and θ entails ψ. The key idea of the Craig interpolation theorem is to relate different logical theories via their common non-logical vocabulary. In his original paper, Craig presents his work as a generalization of Beth's definability theorem, according to which implicit (semantic) definability is equivalent to explicit (syntactic) definability. Indeed, Beth's definability theorem follows from Craig's interpolation theorem, but the latter is more general.

From the point of view of applications in computer science, interpolation is often a desirable property of a logic. For instance, in fields such as automatic reasoning and software development, interpolation is related to modularization [2, 103], a property which allows systems or specifications to be developed efficiently by first building component subsystems (or modules). Interpolation for temporal logics is also an increasingly important topic. Temporal logics in general are widely used in systems and software verification, and interpolation has proven to be useful for building efficient model-checkers [45]. This is particularly true of a strong form of Craig interpolation known as uniform interpolation, which is quite rare in modal logic, but that the modal μ-calculus satisfies (see [46]), whereas most temporal logics lack even Craig interpolation (see [107]).

We study Craig interpolation for fragments and extensions of propositional linear temporal logic (LTL). We use the framework of [12] and work with a general notion of *abstract temporal language* which allows us to consider a general notion of extension of such languages. We consider different sets of temporal connectives and, for each, identify the smallest extension of the fragment of LTL with these

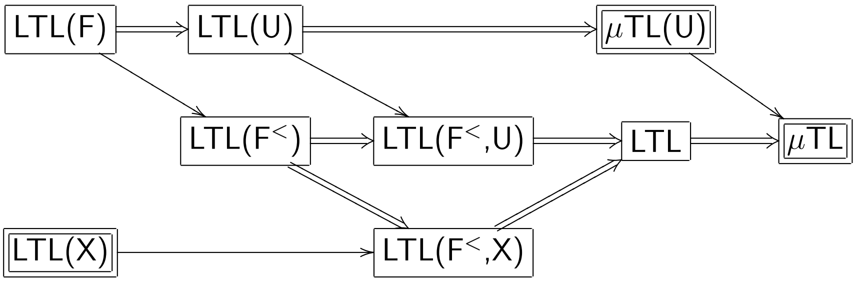


Figure 4.1: Hierarchy of temporal languages

temporal connectives that has Craig interpolation. Depending on the set of temporal connectives, the resulting logic turns out to be either the fragment of LTL with only the Next operator, or the extension of LTL with a fixed-point operator μ (known as linear time μ-calculus), or the fixed-point extension of the fragment of LTL with only the Until operator (which we will show to be the stutter-invariant fragment of the linear time μ-calculus). The diagram in Figure 4.1 summarizes our results. A simple arrow linking two languages means that the first one is an extension of the second one and a double arrow means that, furthermore, every extension of the first one having Craig interpolation is an extension of the second one. Temporal languages with Craig interpolation (in fact, uniform interpolation) are represented in a double frame. Thus we have for instance that μTL(U) is the least expressive extension of LTL(F) with Craig interpolation.

Outline of the chapter: In Section 4.2, we introduce a general notion of *abstract temporal language*. We then introduce LTL, some of its natural fragments and its fixed-point extension known as linear time μ-calculus (μTL) as samples of abstract temporal languages.

Section 4.3 contains some technical results that are used in subsequent sections. One of these relates projective definability in LTL to definability in the fixed-point extension μTL. Another result relates in a similar way LTL(U) and μTL(U). Along the way, we show that μTL(U) is the stutter invariant fragment of μTL. Stutter-invariance is a property that is argued by some authors [101] to be natural and desirable for a temporal logic. Roughly, a temporal logic is stutter-invariant if it cannot detect the addition of identical copies of a state.

In Section 4.4, we give three positive interpolation results. Among the fragments of LTL obtained by restricting the set of temporal operators, we show that only one (the "Next-only" fragment) has Craig interpolation. In fact, this fragment satisfies a stronger form of interpolation, called uniform interpolation. The logics μTL and μTL(U) also have uniform interpolation.

Section 4.5 completes the picture by showing that μTL and μTL(U) are the least extensions of LTL(F) and LTL($F^{<}$), respectively, with Craig interpolation.

4.2 Preliminaries

4.2.1 Abstract Temporal Languages

We will be dealing with a variety of temporal languages. They are all interpreted in structures consisting of a set of worlds (or, time points), a binary relation intuitively representing temporal precedence, and a valuation of proposition letters. In this section, we give an *abstract model-theoretic* definition of temporal languages (on the general topic of abstract model theory, we refer to [12]).

Let us recall that a *flow of time*, or *frame*, is a structure $\mathcal{T} = (W, <)$, where W is a non-empty set of worlds and $<$ is a binary relation on W. We will focus here on $\mathbf{T}_\omega$, the class of linear orders of order type ω, i.e., frames $(D, <)$ that are isomorphic to $(\mathbb{N}, <)$, where $\mathbb{N}$ is the set of natural numbers with the natural ordering. We will also freely use $\leq$ to denote the reflexive closure of $<$.

By a *propositional signature* we mean a finite non-empty set of propositional letters $\sigma = \{p_i \mid i \in I\}$. A *pointed σ-structure* is a structure $\mathfrak{M} = (\mathcal{T}, V, w)$ where $\mathcal{T} = (W, R)$ is a frame, $V : \sigma \to \wp(W)$ a valuation and $w \in W$ a world. The class of all pointed σ-structures is denoted by $Str[\sigma]$ and we call them σ-structures for short. Furthermore, for any class of frames $\mathbf{T}$, $Str_{\mathbf{T}}[\sigma]$ will denote the class of σ-structures of which the underlying frame belongs to $\mathbf{T}$. Let $\sigma \subseteq \tau$ be propositional signatures. Given a τ-structure $\mathfrak{M} = (\mathcal{T}, V, w)$, we define its *$\sigma$-reduct* $\mathfrak{M} \upharpoonright \sigma$ as the σ-structure $(\mathcal{T}, V \upharpoonright \sigma, w)$ where $V \upharpoonright \sigma$ is the restriction of the valuation to the propositional letters in σ. We call $\mathfrak{M}$ a *τ-expansion* of $\mathfrak{M} \upharpoonright \sigma$. We also write $K \upharpoonright \sigma$ for $\{\mathfrak{M} \upharpoonright \sigma \mid \mathfrak{M} \in K\}$. Let $(\mathcal{T}, V, w)$ be a σ-structure and $A \subseteq W$ a subset of its domain. By $V[A/p]$, we will refer to the valuation V extended with $V(p) = A$ (p being a fresh proposition letter). We will refer to the corresponding $\sigma \cup \{p\}$-expansion of $(\mathcal{T}, V, w)$ by $(\mathcal{T}, V[A/p], w)$.

Definition 4.2.1 (Abstract temporal language)**.** An abstract temporal language (*temporal language* for short) is a pair $\mathcal{L} = (\mathcal{L}, \models_{\mathcal{L}})$, where $\mathcal{L} : \sigma \mapsto \mathcal{L}[\sigma]$ is a map from propositional signatures to sets of objects that we call formulas and $\models_{\mathcal{L}}$ is a relation between formulas and pointed structures satisfying the following conditions, for all propositional signatures σ, τ:

1. **Expansion property.** If $\sigma \subseteq \tau$ then $\mathcal{L}[\sigma] \subseteq \mathcal{L}[\tau]$. Furthermore, for all $\varphi \in \mathcal{L}[\sigma]$ and $\mathfrak{M} \in Str[\tau]$, $\mathfrak{M} \models_{\mathcal{L}} \varphi$ iff $\mathfrak{M} \upharpoonright \sigma \models_{\mathcal{L}} \varphi$. If $\mathfrak{M} \in Str[\sigma]$ and $\mathfrak{M} \models_{\mathcal{L}} \varphi$, then $\varphi \in \mathcal{L}[\sigma]$.

2. **Closure under uniform substitution.** For all $\psi \in \mathcal{L}[\sigma]$, $p \notin \sigma$ and $\varphi \in \mathcal{L}[\sigma \cup \{p\}]$, there is a formula of $\mathcal{L}[\sigma]$, which we will denote by $\varphi[p/\psi]$, such that for every $(\mathcal{T}, V, w) \in Str[\sigma]$ the following holds:

$$(\mathcal{T}, V, w) \models_{\mathcal{L}} \varphi[p/\psi] \text{ iff } (\mathcal{T}, V', w) \models_{\mathcal{L}} \varphi$$

   where $V' = V[\{w \mid (\mathcal{T}, V, w) \models_{\mathcal{L}} \psi\}/p]$.

3. **Negation property.** For each $\varphi \in \mathcal{L}[\sigma]$ there is a formula of $\mathcal{L}[\sigma]$, which we will denote by $\neg\varphi$, s.t. for all $\mathfrak{M} \in Str[\sigma]$, $\mathfrak{M} \models_{\mathcal{L}} \neg\varphi$ iff $\mathfrak{M} \not\models_{\mathcal{L}} \varphi$.

For any class of frames $\mathbf{T}$, $\models_{\mathcal{L},\mathbf{T}}$ will denote the restriction of $\models_{\mathcal{L}}$ to pointed structures based on $\mathbf{T}$. For $\varphi \in \mathcal{L}[\sigma]$, we will use $Mod^{\sigma}(\varphi)$ as shorthand for $\{\mathfrak{M} \in Str[\sigma] \mid \mathfrak{M} \models_{\mathcal{L},\mathbf{T}} \varphi\}$ and $Mod^{\sigma}_{\mathbf{T}}(\varphi)$ when restricting to a frame class $\mathbf{T}$. Whenever this is clear from the context, we will be omitting superscript and subscripts in $Mod^{\sigma}_{\mathbf{T}}(\varphi)$ and $\models_{\mathcal{L},\mathbf{T}}$. We say that a class of pointed structures $\mathbf{K} \subseteq Str_{\mathbf{T}}[\sigma]$ is *definable* in an abstract temporal language $\mathcal{L}$ (relative to the frame class $\mathbf{T}$) if there is a $\mathcal{L}$-formula φ such that for every $(\mathcal{T}, V, w) \in Str_{\mathbf{T}}[\sigma]$, $(\mathcal{T}, V, w) \models \varphi$ iff $(\mathcal{T}, V, w) \in \mathbf{K}$.

Definition 4.2.2 (Extension of a temporal language)**.** Let $\mathcal{L}_1 = (\mathcal{L}_1, \models_{\mathcal{L}_1})$, $\mathcal{L}_2 = (\mathcal{L}_2, \models_{\mathcal{L}_2})$ be temporal languages. $\mathcal{L}_2$ extends $\mathcal{L}_1$ (notation: $\mathcal{L}_1 \subseteq \mathcal{L}_2$) if for all σ, for all $\varphi \in \mathcal{L}_1[\sigma]$, there exists $\varphi^* \in \mathcal{L}_2[\sigma]$ such that $Mod_{\sigma}(\varphi) = Mod_{\sigma}(\varphi^*)$. Also, whenever $\mathcal{L}_1 \subseteq \mathcal{L}_2$, we say that $\mathcal{L}_1$ is a *fragment* of $\mathcal{L}_2$. Whenever restricting attention to a frame class $\mathbf{T}$ we write $\mathcal{L}_1 \subseteq_{\mathbf{T}} \mathcal{L}_2$.

The following notion is related to existential second-order quantification over propositional letters. Allowing such a form of quantification in a given temporal language indeed amounts to considering its projective classes. It is a classical notion in abstract modal theory and it will be useful in the context of Δ-interpolation (see Definition 4.5.2).

Definition 4.2.3 (Projective class)**.** Let σ be a propositional signature, $\mathbf{T}$ a frame class and let $K \subseteq Str_{\mathbf{T}}[\sigma]$. Then K is a projective class of a temporal language $\mathcal{L}$ relative to $\mathbf{T}$ if there is a $\varphi \in \mathcal{L}[\tau]$ with $\tau \supseteq \sigma$ a propositional signature, such that $K = Mod(\varphi) \restriction \sigma$.

Lemma 4.2.4. *Let* $\mathbf{T}$ *be a frame class. If* $\mathcal{L}_1 \subseteq_{\mathbf{T}} \mathcal{L}_2$*, then every projective class of* $\mathcal{L}_1$ *relative to* $\mathbf{T}$ *is also a projective class of* $\mathcal{L}_2$ *relative to* $\mathbf{T}$.

Proof. Let K be a projective class of $\mathcal{L}_1$ relative to a frame class $\mathbf{T}$. So there is $\varphi \in \mathcal{L}_1[\tau]$ with $\tau \subseteq \sigma$ a propositional signature, such that $K = Mod^{\tau}_{\mathcal{L},\mathbf{T}}(\varphi) \restriction \sigma$. As $\mathcal{L}_1 \subseteq \mathcal{L}_2$, there is also $\varphi^* \in \mathcal{L}_2[\tau]$ such that $Mod^{\tau}_{\mathcal{L}_1}(\varphi) = Mod^{\tau}_{\mathcal{L}_2}(\varphi^*)$. It follows that $K = Mod^{\tau}_{\mathcal{L}_2}(\varphi^*) \restriction \sigma$. □

Definition 4.2.5 (Entailment)**.** Let $\mathcal{L}$ be a temporal language, σ a propositional signature, $\mathbf{T}$ a frame class and $\varphi, \psi \in \mathcal{L}[\sigma]$. We say that φ entails ψ in $\mathcal{L}$ over $\mathbf{T}$ and write $\varphi \models_{\mathcal{L},\mathbf{T}} \psi$ if for any $(\mathcal{T}, V, w) \in Str_{\mathbf{T}}[\sigma]$, whenever $(\mathcal{T}, V, w) \models_{\mathcal{L},\mathbf{T}} \varphi$, then also $(\mathcal{T}, V, w) \models_{\mathcal{L},\mathbf{T}} \psi$.

4.2.2 Propositional Linear Temporal Logic

Recall that $\mathbf{T}_{\omega}$ denotes the linear orders of order type ω. We now recall the syntax and semantics of LTL, following the terminology of [59].

Definition 4.2.6 (LTL)**.** Let σ be a propositional signature. The set of formulas $\mathsf{LTL}[\sigma]$ is defined inductively, as follows:

$$\varphi, \psi := At \mid \top \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \varphi \vee \psi \mid \mathsf{X}\varphi \mid \mathsf{F}\varphi \mid \mathsf{F}^{<}\varphi \mid \varphi \mathsf{U} \psi$$

where $At \in \sigma$. We use G and $\mathsf{G}^{<}$ as shorthand for respectively $\neg\mathsf{F}\neg$ and $\neg\mathsf{F}^{<}\neg$. The relation $\models_{\mathsf{LTL}}$ between LTL-formulas and structures $(\mathcal{T}, V, w)$ is defined as follows (we only list the clauses of the temporal operators, the others are as in the case of classical propositional logic):

- $(\mathcal{T}, V, w) \models_{\mathsf{LTL}} \mathsf{X}\varphi$ iff there exists w' such that $w < w'$, there is no w'' such that $w < w'' < w'$ and $(\mathcal{T}, V, w') \models \varphi$
- $(\mathcal{T}, V, w) \models_{\mathsf{LTL}} \mathsf{F}\varphi$ iff there exists w' such that $w \leq w'$ and $(\mathcal{T}, V, w') \models \varphi$
- $(\mathcal{T}, V, w) \models_{\mathsf{LTL}} \mathsf{F}^{<}\varphi$ iff there exists w' such that $w < w'$ and $(\mathcal{T}, V, w') \models \varphi$
- $(\mathcal{T}, V, w) \models_{\mathsf{LTL}} \varphi\mathsf{U}\psi$ iff there exists w' such that $w \leq w'$, $(\mathcal{T}, V, w') \models \psi$ and for all w'' such that $w \leq w'' < w'$, $(\mathcal{T}, V, w'') \models \varphi$

While the above definition in principle applies to arbitrary pointed structures, the intended semantics will be, of course, in terms of structures based on frames in $\mathbf{T}_\omega$, and in what follows we will always restrict attention to such frames.

We define fragments $\mathsf{LTL}(\mathcal{O})$ of LTL by allowing in their syntax only a subset $\mathcal{O} \subseteq \{\mathsf{X}, \mathsf{F}^{<}, \mathsf{F}, \mathsf{U}\}$ of temporal operators. Note that LTL(U,X) has the same expressive power as LTL, because $\mathsf{F}\varphi$ can be defined as $\top\mathsf{U}\varphi$ and $\mathsf{F}^{<}\varphi$ as $\mathsf{X}(\top\mathsf{U}\varphi)$. The same holds of $\mathsf{LTL}(\mathsf{F}^{<},\mathsf{X})$ and $\mathsf{LTL}(\mathsf{F}^{<},\mathsf{X},\mathsf{F})$, as $\mathsf{F}\varphi$ can be defined as $\varphi \vee \mathsf{F}^{<}\varphi$. Nevertheless, it is known (see [93]), that $\varphi\mathsf{U}\psi$ can be defined neither in LTL(F) nor in $\mathsf{LTL}(\mathsf{F}^{<},\mathsf{X})$. Also $\mathsf{X}\varphi$ and $\mathsf{F}^{<}\varphi$ can be defined neither in LTL(U) nor in LTL(F) (we will see why later on in this chapter, once we introduce the notion of stutter-invariance).

4.2.3 Linear Time μ-Calculus

A way of increasing the expressive power of temporal languages is to add a fixed-point operator. On arbitrary structures, adding to LTL the least fixed-point operator μ gives the μ-calculus (see for instance [46]). Here, the class of intended structures for μ-calculus is restricted to those based on $\mathbf{T}_w$ and the resulting restricted temporal language is called $\mu\mathsf{TL}$ (see for instance [92]). We also recall here its syntax and semantics.

Definition 4.2.7 (μTL)**.** Let σ be a propositional signature and $\mathcal{V} = \{x_1, x_2, \ldots\}$ a disjoint countably infinite stock of *propositional variables*. We define $\mu\mathsf{TL}[\sigma]$ as the set of all formulas *without free variables* that are generated by the following inductive definition:

$$\varphi, \psi, \xi := At \mid \top \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \mathsf{X}\varphi \mid \mathsf{F}\varphi \mid \mathsf{F}^{<}\varphi \mid \varphi\mathsf{U}\psi \mid \mu x_i.\xi$$

where $At \in \sigma \cup \mathcal{V}$ and, in the last clause, x_i occurs only positively in ξ (i.e., within the scope of an even number of negations). We will use $\varphi \to \psi$ as shorthand in the usual way and $\nu x_i.\varphi(x_i)$ as shorthand for $\neg\mu x_i.\neg\varphi(\neg x_i)$. The relation $\models_{\mu\mathsf{TL}}$ is defined between $\mu\mathsf{TL}$-formulas and pointed structures $(\mathcal{T}, V, w)$ where $\mathcal{T} \in \mathbf{T}_\omega$. In order to define it inductively, we use an auxiliary assignment to interpret formulas with free variables. The assignment g maps each free variable of φ to a set of worlds. We let $g[x \mapsto A]$ be the assignment which differ from g only by assigning A to x and we only recall:

- $(\mathcal{T}, V, w) \models_{\mu\mathsf{TL}} x_i\ [g]$ iff $w \in g(x_i)$
- $(\mathcal{T}, V, w) \models_{\mu\mathsf{TL}} \mu x.\varphi\ [g]$ iff $\forall A \subseteq W$, if $\{v \mid (\mathcal{T}, V, v) \models_{\mu\mathsf{TL}} \varphi\ [g[x \mapsto A]]\} \subseteq A$, then $w \in A$

It is easy to see that, for formulas without free variables, the assignment is irrelevant, and therefore $\models_{\mu\mathsf{TL}}$ defines a binary relation between (the set of sentences of) $\mu\mathsf{TL}$ and pointed structures. In this way, $\mu\mathsf{TL}$ is an abstract modal language in the sense of Definition 4.2.1.

As before, we define a fragment $\mu\mathsf{TL}(\mathcal{O})$ for each $\mathcal{O} \subseteq \{\mathsf{X}, \mathsf{F}^<, \mathsf{F}, \mathsf{U}\}$. $\mu\mathsf{TL}(\mathsf{X})$ already as the full expressive power of TL, since $\varphi\mathsf{U}\psi$ can be defined by $\mu y.(\psi \vee (\varphi \wedge \mathsf{X}y))$, $\mathsf{F}^<\varphi$ by $\mu y.(X\varphi \vee \mathsf{X}y)$ and $\mathsf{F}\varphi$ by $\mu y.(\varphi \vee \mathsf{X}y)$. Another fragment of particular interest will be $\mu\mathsf{TL}(\mathsf{U})$. In $\mu\mathsf{TL}(\mathsf{U})$, we can still define $\mathsf{F}\varphi$ in the usual way by $\top\mathsf{U}\varphi$, but we will see that $\mathsf{X}\varphi$ and $\mathsf{F}^<\varphi$ are not definable.

4.3 Projective Definability versus Definability with Fixed-Points

In this section, we discuss two results that relate projective definability in languages without fixed-point operators to explicit definability in the corresponding language with fixed-point operators. Along the way, we also show that $\mu\mathsf{TL}(\mathsf{U})$ is the stutter-invariant fragment of $\mu\mathsf{TL}$. These results will be put to use in Section 4.4 and 4.5.

We first state a general result relating projective definability in LTL and definability in $\mu\mathsf{TL}$. It will be convenient to consider also definability in MSO and definability by a Büchi automaton (for background on Büchi-automata and on MSO, we refer to Chapter 2). In order to be fully precise, we first provide the following definition:

Definition 4.3.1. Let $\sigma = \{p_1, \ldots, p_n\}$ be a propositional signature. We define $\Sigma = \wp(\sigma)$ as the *corresponding alphabet over ω-words* and $\sigma_{\mathsf{FO}} = \{<, P_1, \ldots, P_n\}$ as the *corresponding FO signature over* $\mathbf{T}_\omega$. Now let $\mathcal{T} = (D, <) \in \mathbf{T}_{fin}$ with $D = \{w_0, w_1, \ldots\}$ and $w_i < w_{i+1}$ for all $i \geq 0$. Given a σ-structure $(\mathcal{T}, V, w_j)$, we define the *corresponding ω-word* $(\mathcal{T}, V)^{w_j}$ in signature Σ and the *corresponding relational structure* $(\mathcal{T}, V)^{w_j}_{\mathsf{FO}}$ in signature σ_{FO} in the following way:

- let $w_i^V = \{p \in \sigma \mid w_i \in V(p)\}$, we define $(\mathcal{T}, V)^{w_j}$ as the word $w_j^V w_{j+1}^V \ldots$ (i.e., w_j^V is the first letter of the word and for every $i \geq j$, w_{i+1}^V is the letter immediately following w_i^V)
- $(\mathcal{T}, V)_{\mathsf{FO}}^{w_j}$ is the relational structure $(D^{w_j}, <^{w_j}, P_1^{w_j}, \ldots, P_n^{w_j})$ in signature σ_{FO} with
  - a domain $D^{w_j} = \{w_i \in D \mid i \geq j\}$
  - a binary relation $<^{w_j} = < \upharpoonright \{w_i \in D \mid i \geq j\}$ (i.e., $<^{w_j}$ is the restriction of the relation $<$ to the points in D that are greater or equal in $<$ to w_j)
  - for every $l \geq 1$, a unary relation $P_l^{w_j} = \{w_i \mid w_i \in V(p_l) \text{ and } i \geq j\}$

Now we can state the general result we are interested in.

Theorem 4.3.2. *Let σ be a propositional signature. For any $K \subseteq Str_{\mathbf{T}_\omega}[\sigma]$, the following are equivalent:*

1. *there is an* MSO *sentence φ in signature σ_{FO} such that*
$$K = \{(\mathcal{T}, V, w) \mid (\mathcal{T}, V)_{\mathsf{FO}}^{w} \models_{\mathsf{MSO}} \varphi\}$$
2. *there is a Büchi automata $\mathcal{A}$ over the alphabet $\wp(\sigma)$ such that*
$$K = \{(\mathcal{T}, V, w) \mid (\mathcal{T}, V)^{w} \text{ is accepted by } \mathcal{A}\}$$
3. *K is a projective class of* $\mathsf{LTL}(\mathsf{F}^<, \mathsf{X})$ *relative to $\mathbf{T}_\omega$*
4. *there is a* $\mu\mathsf{TL}$ *sentence φ such that*
$$K = \{(\mathcal{T}, V, w) \mid (\mathcal{T}, V, w) \models_{\mu\mathsf{TL}} \varphi\}$$

Proof.

1 ⇒ 2 This is a known result (see [127]).

2 ⇒ 3 Let $\mathcal{A} = (Q, \Sigma = \wp(P_1, \ldots, P_m), \Delta, q_0, Acc)$ be a Büchi automaton. Assume $Q = \{q_0, \ldots, q_k\}$ and let $r_1, \ldots, r_k$ be pairwise distinct propositional letters not in σ. We will construct a $\mathsf{LTL}(\mathsf{F}^<, \mathsf{X})$-formula which holds in a $\sigma \cup \{r_0, \ldots, r_k\}$-structure $(\mathcal{T}, V, w)$ (with $\mathcal{T} \in \mathbf{T}_\omega$) if and only if $(\mathcal{T}, V \upharpoonright \sigma)^w$ is accepted by $\mathcal{A}$. Given an ω-word $(\mathcal{T}, V \upharpoonright \sigma)^w \in L(\mathcal{A})$ of the form $\alpha(0)\alpha(1)\ldots$, the sentence will state the existence of a successful run $\rho(0), \rho(1), \ldots$ of $\mathcal{A}$, i.e., with $\rho(0) = q_0$ $(\rho(i), \alpha(i), \rho(i+1)) \in \Delta$ for $i \geq 0$, and $Inf(\rho) \cap F = \emptyset$. We introduced new propositional letters because we can code such a state sequence by a tuple of propositional letters $(r_0, \ldots, r_k)$

of pairwise disjoint subsets of $\{0, 1, \ldots\}$ such that r_i contains those positions of $\alpha(0)\alpha(1)\ldots$ where state q_i is assumed. The automaton should be able to reach a final state infinitely often. For every $\alpha \in \wp(P_1, \ldots, P_m)$ let α^* be $\bigwedge_{p_i \in \alpha} p_i \wedge \bigwedge_{p_i \notin \alpha} \neg p_i$. Thus, $\mathcal{A}$ accepts the nonempty word $(\mathcal{T}, V \restriction \sigma)^{\geq w}$ iff

$$(\mathcal{T}, V, w) \models (r_0 \wedge \bigwedge_{i>0} \neg r_i)$$

(r_0 contains the first position in w, i.e., r_0 is true at the first node of w)

$$\wedge(\bigwedge_{i \neq j} \neg \mathsf{F}^{<}(r_i \wedge r_j))$$

(all other r_i positions are pairwise different, i.e., if r_i contains a position in w, then if $i \neq j$, r_j does not contain this position)

$$\wedge(\mathsf{G}^{<} \bigvee_{(q_i, \alpha, q_j) \in \Delta} (r_i \wedge \alpha^* \wedge \mathsf{X} r_j))$$

(the next position is consistent with Δ)

$$\wedge \bigvee_{q_j \in F} G^{<}\mathsf{F}^{<} r_j$$

(some state in F occurs infinitely often)

$3 \Rightarrow 1$ Let K be projectively definable relative to $\mathbf{T}_\omega$ by a $\mathsf{LTL}(\mathsf{F}^{<}, \mathsf{X})$-formula φ in an extension σ' of σ. Construct the *standard translation* of φ (this is a FO formula in signature σ', see Chapter 2) and call it φ^*. Now consider $p_1, \ldots, p_n \in \sigma' \backslash \sigma$ and replace uniformly in φ^* the corresponding FO predicates $P_1, \ldots, P_n \in \sigma'_{\mathsf{FO}} \backslash \sigma_{\mathsf{FO}}$ by set variables $X_1, \ldots, X_n$. We obtain the formula $\varphi^*[X_1/P_1, \ldots, X_n/P_n]$ that we can now prefix with existential set quantifiers over $X_1, \ldots, X_n$. The obtained formula

$$\exists X_1 \ldots \exists X_n \varphi^*[X_1/P_1, \ldots, X_n/P_n]$$

is in signature σ_{FO} and has the desired property.

$4 \Leftrightarrow 1$ This is a known result (see [111] and [5]).

□

Below, we will show a similar theorem linking projective definability in $\mathsf{LTL}(\mathsf{U})$ (which was shown in [116, 61] to be the stutter-invariant fragment of LTL) to definability in $\mu\mathsf{TL}(\mathsf{U})$, which we show here to be the stutter-invariant fragment of linear time μ-calculus. Before stating this second result, we first define stuttering.

Intuitively, a stuttering of a linearly ordered structure $\mathfrak{M}$ is a structure obtained from $\mathfrak{M}$ by replacing each world by a non-empty finite sequence of worlds, all satisfying the same proposition letters.

Definition 4.3.3 (Stuttering)**.** Let σ be a propositional signature and $\mathfrak{M} = ((W,<),V,w)$, $\mathfrak{M}' = ((W',<),V',w')$ be in $Str_{\mathbf{T}_\omega}[\sigma]$. We say that $\mathfrak{M}'$ is a stuttering of $\mathfrak{M}$ if and only if there is a surjective function $s : W' \to W$ such that

1. $s(w') = w$
2. for every $w_i, w_j \in W'$, $w_i < w_j$ implies $s(w_i) \leq s(w_j)$
3. for every $w_i \in W'$ and $p \in \sigma, w_i \in V'(p)$ iff $s(w_i) \in V(p)$

Some notation will be useful later on. For any $w \in W$, we let $s^{-1}(w) = \{w' \in W' \mid s(w') = w\}$. We also extend s and s^{-1} to subsets of W' in the following way: for any $A' \subseteq W', A \in W$, we let $s(A') = \{s(v') \mid v' \in A'\}$ and $s^{-1}(A) = \bigcup_{v\in A} s^{-1}(v)$.

Lemma 4.3.4. *Let $\mathfrak{M} = ((W,<),V,w)$, $\mathfrak{M}' = ((W',<),V',w')$ be in $Str_{\mathbf{T}_\omega}[\sigma]$ and $\mathfrak{M}'$ be a stuttering of $\mathfrak{M}$, then the following hold:*

1. *$\forall v' \in W', \forall A' \subseteq W'$ such that $v' \in A'$ implies $s^{-1}(s(v')) \subseteq A'$:*

$$((W',<),V'[A'/p],v') \text{ is a stuttering of } ((W,<),V[s(A')/p],s(v'))$$

2. *$\forall v \in W, \forall A \subseteq W, \forall v' \in s^{-1}(v)$:*

$$((W',<),V'[s^{-1}(A)/p],v') \text{ is a stuttering of } ((W,<),V[A/p],v)$$

Definition 4.3.5 (Stutter-Invariant Class of Pointed Structures)**.** Let σ be a propositional signature and $\mathbf{K} \subseteq Str_{\mathbf{T}_\omega}[\sigma]$. Then $\mathbf{K}$ is a stutter-invariant class relative to $\mathbf{T}_\omega$ iff for every $\mathfrak{M} \subseteq Str_{\mathbf{T}_\omega}[\sigma]$ and for every stuttering $\mathfrak{M}'$ of $\mathfrak{M}$, $\mathfrak{M} \in \mathbf{K} \Leftrightarrow \mathfrak{M}' \in \mathbf{K}$.

Definition 4.3.6 (Stutter-free Pointed Structure)**.** We say that a pointed structure $\mathfrak{M}$ is *stutter-free* whenever for all $\mathfrak{M}'$ such that $\mathfrak{M}$ is a stuttering of $\mathfrak{M}'$, $\mathfrak{M}'$ is isomorphic to $\mathfrak{M}$.

Only stutter-invariant classes of structures in $Str_{\mathbf{T}_\omega}[\sigma]$ are definable in $\mathsf{LTL}(\mathsf{U})$ and $\mu\mathsf{TL}(\mathsf{U})$. This is known for $\mathsf{LTL}(\mathsf{U})$ (see [61, 116]), but it also holds for $\mu\mathsf{TL}(\mathsf{U})$.

Proposition 4.3.7. *Let σ be a propositional signature. For every $\mu\mathsf{TL}(\mathsf{U})$-sentence φ in signature σ, $Mod(\varphi)$ is stutter-invariant.*

Proof. By induction on the sentence complexity. For the sake of the induction, we can use expanded σ-structures as in classical model theory. Hence we consider two base cases, one for propositional letters and one for propositional variables. The propositional letter case is clear. We handle the propositional variable case x_i similarly, except that we use σ-models expanded with the value of x_i (i.e., models considered together with a partial auxiliary valuation, so that x_i can be seen as a sentence). The induction hypothesis says that for any propositional signature σ and $\mu\mathsf{TL}(\mathsf{U})$-sentence φ of complexity n in signature σ, $Mod(\varphi)$ is a stutter-invariant invariant class. Now consider the case were φ is of complexity $n+1$. We handle the Boolean connectives and the U operator as in the $\mathsf{LTL}(\mathsf{U})$ case. For the U case, suppose $\varphi :\approx \psi U\xi$. We want to show that for every $\mathfrak{M} = ((<, W), V, w) \subseteq Str_{\mathbf{T}}[\sigma]$ and for every stuttering $\mathfrak{M}' = ((<, W'), V', w')$ of $\mathfrak{M}$:

$$\mathfrak{M} \in Mod^{\sigma}_{\mu\mathsf{LTL}(\mathsf{U}),\mathbf{T}}(\psi U\xi) \Leftrightarrow \mathfrak{M}' \in Mod^{\sigma}_{\mu\mathsf{LTL}(\mathsf{U}),\mathbf{T}}(\psi U\xi)$$

- $\Rightarrow$ Suppose $((W, <), V, w) \models \psi U\xi$, i.e., there exists w_i such that $w = w_i$ or $w < w_i$, $((W, <), V, w_i) \models \xi$ and for all w_j such that $w < w_j < w_i$, $(\mathcal{T}, V, w_j) \models \psi$. Let w_i be the first point such that $(\mathcal{T}, V, w_i) \models \xi$, then all points w_j before it are such that $(\mathcal{T}, V, w_j) \models \psi$. It follows from the definition of stuttering that the minimal point $s \in s^{-1}(w_j)$ is the first point such that $(\mathcal{T}, V, s) \models \xi$ and all points $s' \in s^{-1}(w_j)$ (with w_j before w_i) are such that $(\mathcal{T}, V, s') \models \psi$, i.e., $((W', <), V', w') \models \psi U\xi$.

- $\Leftarrow$ The reasoning is similar.

Now for the fixed-point case, suppose $\varphi :\approx \mu x.\psi(x)$. We want to show that for every $\mathfrak{M} \subseteq Str_{\mathbf{T}}[\sigma]$ and for every stuttering $\mathfrak{M}'$ of $\mathfrak{M}$:

$$\mathfrak{M} = ((<, W), V, w) \in Mod(\mu x.\psi(x)) \Leftrightarrow \mathfrak{M}' = ((<, W'), V', w') \in Mod(\mu x.\psi(x))$$

For the left to right direction, suppose $((W, <), V, w) \models \mu x.\psi(x)$, i.e., $\forall A \subseteq W$, if $\{v \mid ((W, <), V[A/p], v) \models \psi(p)\} \subseteq A$, then $w \in A$. Consider $A' \subseteq W'$ such that $\{v \mid ((W', <), V'[A'/p], v) \models \psi(p)\} \subseteq A'$. We want to show that $w' \in A'$. Let us first show that $v' \in A'$ implies $s^{-1}(s(v')) \subseteq A'$. For every $v' \in A'$, we have that $((W', <), V'[A'/p], v') \models \psi(p)$. Now by induction hypothesis for any $v \in s^{-1}(s(v'))$, $((W', <), V'[A'/p], v) \models \psi(p)$ and by hypothesis on A', $v \in A'$. It follows from this property of A' that $\mathfrak{M}'$ being a stuttering of $\mathfrak{M}$, by Lemma 4.3.4 for any $v' \in W'$, $((<, W'), V'[A'/p], v')$ is also a stuttering of $((<, W), V[s(A')/p], s(v'))$ and by induction hypothesis:

$$((W', <), V'[A'/p], v') \models \psi(p) \text{ iff } ((<, W), V[s(A')/p], s(v')) \models \psi(p)$$

Hence $\{v \mid ((W, <), V[s(A')/p], v) \models \psi(p)\} \subseteq s(A')$. But $\mathfrak{M} \models \mu x.\psi(x)$. It follows that $w \in S(A')$, so $s(w) \in A'$, i.e., $w' \in A'$.

Now for the right to left direction, suppose $((W', <), V', w') \models \mu x.\psi(x)$, i.e., $\forall A' \subseteq W'$, if $\{v \mid (W', <), V'[A'/p], v \models \psi(p)\} \subseteq A'$, then $w' \in A'$. Consider $A \subseteq W$ such that $\{v \mid (W, <), V[A/p], v \models \psi(p)\} \subseteq A$. We want to show that $w \in A$. $\mathfrak{M}'$ being a stuttering of $\mathfrak{M}$, by Lemma 4.3.4, for any $v \in W$, $v' \in s^{-1}(v)$, $((<, W'), V'[s^{-1}(A)/p], v')$ is also a stuttering of $((<, W), V[A/p], v)$ and by induction hypothesis, for any $v \in W, v' \in s^{-1}(v)$:

$$((W', <), V'[s^{-1}(A)/p], v') \models \psi(p) \text{ iff } ((W, <), V[A/p], v) \models \psi(p)$$

Hence $\{v \mid ((W', <), V'[s^{-1}(A)/p], v) \models \psi(p)\} \subseteq s^{-1}(A)$. But $\mathfrak{M}' \models \mu x.\psi(x)$. It follows that $w' \in s^{-1}(A)$, so $s^{-1}(w') \subseteq A$, i.e., $w \in A$. □

Corollary 4.3.8. *Let $K \subseteq Str_{\mathbf{T}_\omega}[\sigma]$ be stutter-invariant and let $\varphi \in \mu\mathsf{TL}(\mathsf{U})[\sigma]$ be a sentence such that for each stutter-free $\mathfrak{M} \in Str_{\mathbf{T}_\omega}[\sigma]$, $\mathfrak{M} \models \varphi$ if and only if $\mathfrak{M} \in K$. Then φ defines K.*

We now show that (over $\mathbf{T}_\omega$) $\mu\mathsf{TL}(\mathsf{U})$ is the stutter-invariant fragment of $\mu\mathsf{TL}$. The proof is a variant of [116], where Peled and Wilke show that stutter-invariant LTL properties are expressible without X. We give it in detail, as the construction procedure below will be useful again later on in the chapter.

Lemma 4.3.9. *Let σ be a modal vocabulary. For every $\mu\mathsf{TL}$ sentence φ in vocabulary σ, there exists a $\mu\mathsf{TL}(\mathsf{U})$ sentence φ^* in vocabulary σ that agrees with φ on all stutter-free σ-structures over $\mathbf{T}_\omega$:*

$$\mathfrak{M} \models \varphi \leftrightarrow \varphi^* \textit{ for all stutter free pointed structures } \mathfrak{M} \in Str_{\mathbf{T}_\omega}[\sigma]$$

Proof. Assume $\sigma = \{p_0, \ldots, p_{n-1}\}$. The proof goes by induction on the structure of φ. For convenience, we use expanded structures. The base case is clear: $p^* = p$ for any propositional variable or letter p. Now as regards the induction step, we can set $(\neg\psi)^* = \neg\psi^*$, $(\psi \wedge \xi)^* = \psi^* \wedge \tau^*$, $(\psi \mathsf{U} \xi)^* = \psi^* \mathsf{U} \xi^*$, $(\mu x.\psi)^* = \mu x.\psi^*$. If φ is of the form $\mathsf{X}\psi$, we let B be the set of all possible valuations $\sigma \to \{\bot, \top\}$, and for each $g \in B$, we let β_g be the formula $\alpha_0 \wedge \ldots \wedge \alpha_{n-1}$ where $\alpha_j = p_j$ if $g(p_j) = \top$ and $\alpha_j = \neg p_j$ if $g(p_j) = \bot$. Now observe that if $g, g' \in B$ are such that $g \neq g'$, then

$$\mathfrak{M}, w \models \beta_g \wedge \mathsf{X}\beta_{g'} \leftrightarrow \beta_g \mathsf{U} \beta_{g'} \text{ for } \mathfrak{M} \in Str_{\mathbf{T}}[\sigma] \text{ stutter-free}$$

We have $\mathfrak{M}, w \models \mathsf{X}\psi$ if and only if every point in it satisfies the same set of proposition letters and $\mathfrak{M}, w \models \psi$, or the valuation function does not send the same set of proposition letters to w and to its immediate successor w' and $\mathfrak{M}, w' \models \varphi$. Thus we can set:

$$(\mathsf{X}\psi)^* = \bigvee_{g \in G} ((\mathsf{G}\beta_g \wedge \psi^*) \vee \bigvee_{g \neq g'} (\beta_g \wedge \beta_g \mathsf{U}(\beta_{g'} \wedge \psi^*)))$$

□

Theorem 4.3.10. *Let $\varphi \in \mu\mathsf{TL}[\sigma]$ be a sentence such that $Mod_\sigma(\varphi)$ is stutter-invariant. Then there exists $\varphi^* \in \mu\mathsf{TL}(\mathsf{U})[\sigma]$ such that $Mod_\sigma(\varphi) = Mod_\sigma(\varphi^*)$.*

Proof. Follows from Lemma 4.3.9 and Corollary 4.3.8. □

Following [61], we now introduce a variant of the notion of projective class, that we call *harmonious projective class*, which preserves stutter-invariance. Before we define it, we first introduce the notion of a *harmonious expansion*. For any propositional signature σ and worlds w, w', we write $w \equiv_\sigma w'$ if w and w' satisfy the same propositions in σ.

Definition 4.3.11 (Harmonious expansion)**.** Let $\sigma \subseteq \tau$ be propositional signatures and $\mathfrak{M} \in Str_{\mathbf{T}_\omega}[\tau]$. We say that $\mathfrak{M}$ is a harmonious expansion of $\mathfrak{M} \upharpoonright \sigma$ whenever $\forall w, w' \in W$ such that w' is a direct successor of w, $w \equiv_\sigma w'$ implies $w \equiv_\tau w'$.

Definition 4.3.12 (Harmonious projective class)**.** Let σ be a propositional signature and $K \subseteq Str_{\mathbf{T}_\omega}[\sigma]$. Then K is a *harmonious projective class* of a temporal language $\mathcal{L}$ relative to $\mathbf{T}_\omega$ whenever there is $\varphi \in \mathcal{L}[\tau]$ with $\tau \supseteq \sigma$ such that for all $\mathfrak{M} \in Str_{\mathbf{T}_\omega}[\sigma]$: $\mathfrak{M} \in K$ iff there is a harmonious τ-expansion $\mathfrak{M}^+$ of $\mathfrak{M}$ such that $\mathfrak{M}^+ \models \varphi$.

We will be using the following proposition in order to show Theorem 4.3.14. It refers to the notion of ω-regular language (cf. [127], an ω-regular language is a language of ω-words which is definable in MSO or, equivalently, which is recognizable by a Büchi automata). The proof of the proposition in [61] uses a notion of stutter-invariant ω-automata.

Proposition 4.3.13 ([61])**.** *On $\mathbf{T}_\omega$, harmonious projective classes of* LTL(U) *define exactly the stutter-invariant ω-regular languages.*

Now we are able to show the following theorem:

Theorem 4.3.14. *Let σ be a propositional signature. For any $K \subseteq Str_{\mathbf{T}}[\sigma]$, the following are equivalent:*

1. *K is a harmonious projective class of* LTL(U) *relative to $\mathbf{T}_\omega$*
2. *K is definable by a μ*TL(U)*-sentence φ relative to $\mathbf{T}_\omega$*

Proof. Follows from Theorem 4.3.2 and Proposition 4.3.13, because by [61, 116], LTL(U) is the stutter-invariant fragment of LTL and by Theorem 5.2.9, μTL(U) is the stutter-invariant fragment of μTL. □

4.4 Temporal Languages with Craig Interpolation

In this section, we show that three of the temporal languages previously discussed have Craig interpolation.

Definition 4.4.1 (Craig interpolation property)**.** Let $\mathcal{L}$ be a temporal language and $\mathbf{T}$ a frame class. Then $\mathcal{L}$ has the Craig interpolation property over $\mathbf{T}$ whenever the following holds. Let $\varphi \in \mathcal{L}[\sigma]$, $\psi \in \mathcal{L}[\sigma']$. Whenever $\varphi \models_{\mathcal{L},\mathbf{T}} \psi$, then there exists $\theta \in \mathcal{L}[\sigma \cap \sigma']$ such that $\varphi \models_{\mathcal{L},\mathbf{T}} \theta$ and $\theta \models_{\mathcal{L},\mathbf{T}} \psi$.

They even satisfy a stronger form of interpolation called uniform interpolation. Intuitively if a temporal language has uniform interpolation, it means that the interpolant can be constructed so that it depends only on the signature of the antecedent and its intersection with the signature of the consequent.

Definition 4.4.2 (Uniform Interpolation)**.** Let $\mathcal{L}$ be a temporal language and $\mathbf{T}$ a frame class. $\mathcal{L}$ has the *uniform interpolation* property over $\mathbf{T}$ if, for all signatures $\sigma \subseteq \tau$ and for each formula $\varphi \in \mathcal{L}[\tau]$ there is a formula $\theta \in \mathcal{L}[\sigma]$ such that $\varphi \models_{\mathcal{L}} \theta$ and for each formula $\psi \in \mathcal{L}[\tau']$ with $\tau \cap \tau' \subseteq \sigma$, if $\varphi \models_{\mathcal{L}} \psi$ then $\theta \models_{\mathcal{L}} \psi$.

Theorem 4.4.3. $\mu\mathsf{TL}$ *has uniform interpolation over* $\mathbf{T}_\omega$*.*

Proof. MSO has uniform interpolation (for monadic predicates) on any class of structures (so in particular on $\mathbf{T}_\omega$) because it has set quantifiers (see [45]). By [111, 5], $\mu\mathsf{TL}$ is expressively complete for MSO. Hence $\mu\mathsf{TL}$ uniform interpolants can always be obtained via translation into MSO and back. □

Theorem 4.4.4. $\mu\mathsf{TL}(\mathsf{U})$ *has uniform interpolation over* $\mathbf{T}_\omega$*.*

Proof. Let $\sigma \subseteq \tau$ be modal signatures and let $\varphi \in \mu\mathsf{TL}(\mathsf{U})[\tau]$. By Theorem 4.4.3, there exists $\theta \in \mu\mathsf{TL}[\sigma]$ such that $\varphi \models \theta$ and for each formula $\psi \in \mu\mathsf{TL}[\tau']$ with $\tau \cap \tau' \subseteq \sigma$, if $\varphi \models \psi$, then $\theta \models \psi$. Now let $\theta^* \in \mu\mathsf{TL}(\mathsf{U})$ be the formula that agrees with θ on all stutter-free structures based on $\mathbf{T}_\omega$ (by Lemma 4.3.9, such a formula exists). We want to show that $\varphi \models \theta^*$ and that for each formula $\psi \in \mu\mathsf{TL}(\mathsf{U})[\tau']$ with $\tau \cap \tau' \subseteq \sigma$, if $\varphi \models \psi$, then $\theta^* \models \psi$. Let $SMod(\varphi)$ denote the set of stutter free structures in $Mod(\varphi)$. As $Mod(\varphi) \subseteq Mod(\theta)$, $SMod(\varphi) \subseteq SMod(\theta)$. Now by construction of θ^* also $SMod(\varphi) \subseteq SMod(\theta^*)$. $Mod(\varphi)$ and $Mod(\theta^*)$ are both stutter-invariant classes. It follows from Corollary 4.3.8 that the closure under stuttering of $SMod(\varphi)$ is included in the closure under stuttering of $SMod(\theta^*)$, i.e., $Mod(\varphi) \subseteq Mod(\theta^*)$, i.e., $\varphi \models \theta^*$. The argument for $\theta^* \models \psi$ is similar. □

Theorem 4.4.5. $\mathsf{LTL}(\mathsf{X})$ *has uniform interpolation over* $\mathbf{T}_\omega$*.*

Proof. We will show something much stronger, namely that every projective class of $\mathsf{LTL}(\mathsf{X})$ is definable by a $\mathsf{LTL}(\mathsf{X})$-formula.
Let $\varphi \in \mathsf{LTL}(\mathsf{X})[\sigma \cup \tau]$ with $\tau = \{p_1, \ldots, p_l\}$. We will show how to construct a formula $\psi \in \mathsf{LTL}(\mathsf{X})[\sigma]$ that defines the class of σ-reducts of models of φ.
We first show that for every $\sigma \cup \tau$-pointed structure $\mathfrak{M}, w$, there exists $\varphi^S \in \mathsf{LTL}(\mathsf{X})[\sigma]$ such that $\mathfrak{M}, w \models \varphi$ if and only if $\mathfrak{M} \restriction \sigma \models \varphi^S$ and for every σ-pointed structure $\mathfrak{N}, v$, $\mathfrak{N}, v \models \varphi^S$ implies that there exists a $\sigma \cup \tau$-expansion $\mathfrak{N}^+$ of $\mathfrak{N}$ such that $\mathfrak{N}^+, v \models \varphi$. Let $md(\varphi) = n$ be the modal depth of φ, i.e., the maximal nesting depth of X-operators in φ. Intuitively, φ can only talk about the first n worlds in the pointed structure (starting from the designated world w). For each p_i, we can represent the valuation of p_i in $\mathfrak{M}$ at these n first worlds by a set $S_i \subseteq \{0, \ldots, n\}$, where $k \in S_i$ represents that p_i is true at the k-th world starting from w. We denote by $S = (S_1, \ldots, S_l)$ the ordered sequence of all the S_i. Now we define φ^S as follows: we replace each occurrence of p_i in φ that is in the scope of $k \leq n$ X-operators by $\top$ if $k \in S_i$ and $\bot$ otherwise. We can now show by induction on $md(\varphi)$ that for every $\sigma \cup \tau$-pointed structure $\mathfrak{M}, w$, $\mathfrak{M}, w \models \varphi$ iff $\mathfrak{M} \restriction \sigma, w \models \varphi^S$ and for every σ-pointed structure $\mathfrak{N}, v$, $\mathfrak{N}, v \models \varphi^S$ implies that there exists a $\sigma \cup \tau$-expansion $\mathfrak{N}^+$ of $\mathfrak{N}$ such that $\mathfrak{N}^+, v \models \varphi$. Whenever $md(\varphi) = 0$, then we are just in the propositional case and the property immediately follows. Now assume the property holds for all formulas ψ with $md(\psi) = n$ and consider φ with $md(\varphi) = n + 1$. φ is equivalent to a Boolean combination of formulas which are either of modal depth $\leq n$ (and to which the inductive hypothesis applies directly), or which are of the form $\mathsf{X}\xi$ with $md(\xi) = n$. Let w' be the first successor of w. For every such ξ, by induction hypothesis $\mathfrak{M}, w' \models \xi$ iff $\mathfrak{M} \restriction \sigma, w' \models \xi^{S'}$ and for every σ-pointed structure $\mathfrak{N}, v$, $\mathfrak{N}, v \models \xi^{S'}$ implies that there exists a $\sigma \cup \tau$-expansion $\mathfrak{N}^+$ of $\mathfrak{N}$ such that $\mathfrak{N}^+, v \models \xi$, where S' encodes the valuation of the proposition letters in τ at each of the n first states starting from w'. By the semantics of the X-operator, it follows that $\mathfrak{M}, w \models \mathsf{X}\xi$ iff $\mathfrak{M} \restriction \sigma, w \models \mathsf{X}(\xi^{S'})$. Also, assuming there is a state v' in $\mathfrak{N}$ which is the immediate predecessor of v, $\mathfrak{N}, v' \models \mathsf{X}\xi^{S'}$ implies that there exists a $\sigma \cup \tau$-expansion $\mathfrak{N}^+$ of $\mathfrak{N}$ such that $\mathfrak{N}^+, v' \models \mathsf{X}\xi$. Now it is enough to remark that $\mathsf{X}(\xi^{S'})$ and $(\mathsf{X}\xi)^S$ denote one and the same formula. Hence $\mathfrak{M}, w \models \mathsf{X}\xi$ iff $\mathfrak{M} \restriction \sigma, w \models \mathsf{X}\xi^S$ and $\mathfrak{N}, v' \models \mathsf{X}\xi^{S'}$ iff $\mathfrak{N}, v' \models \mathsf{X}\xi^S$. So the property also follows for φ.
Finally, the number of proposition variables in τ being finite, we can quantify over the finite number of all such possible valuations S and we let $\psi = \bigvee_S \varphi^S$. Assume ψ holds in a pointed σ-structure $\mathfrak{M}, w$. Then for some S there is φ^S such that $\mathfrak{M}, w \models \varphi^S$, i.e., $\mathfrak{M}^+, w \models \varphi$ where $\mathfrak{M}^+$ is a $\sigma \cup \tau$-expansion of $\mathfrak{M}, w$ in which the valuation of the p_i's is as described by S. Now assume $\mathfrak{M}, w$ has a $\sigma \cup \tau$-expansion satisfying φ. Then the valuation of the p_i's in the first n worlds after w can be represented by some S and $\mathfrak{M}, w \models \varphi^S$, which yields $\mathfrak{M}, w \models \psi$. This means that ψ holds in a pointed σ-structure $\mathfrak{M}, w$ iff $\mathfrak{M}, w$ has a $\sigma \cup \tau$-expansion satisfying φ , i.e., ψ defines the class of σ-reducts of models of φ. □

4.5 Interpolation Closure Results for Temporal Languages

In this section, we look at the fragments of LTL that do not have Craig interpolation, and we address the question how much expressive power must be added in order to regain interpolation. We will phrase our main results in terms of the notion of interpolation closure, which we define by taking inspiration from abstract model theory (see [12]):

Definition 4.5.1 (Interpolation Closure)**.** Let $\mathbf{T}$ be a frame class. $\mathcal{L}_2$ is the interpolation closure of $\mathcal{L}_1$ over $\mathbf{T}$ if $\mathcal{L}_1 \subseteq_{\mathbf{T}} \mathcal{L}_2$, $\mathcal{L}_2$ has interpolation over $\mathbf{T}$, and for every abstract temporal language $\mathcal{L}_3$, if $\mathcal{L}_1 \subseteq \mathcal{L}_3$ and $\mathcal{L}_3$ has Craig interpolation on $\mathbf{T}$, then $\mathcal{L}_2 \subseteq_{\mathbf{T}} \mathcal{L}_3$.

4.5.1 The Interpolation Closure of $\mathsf{LTL}(\mathsf{F}^<)$

A useful tool (see [12]) for proving interpolation closure results is the following lemma:

Definition 4.5.2 (Δ-interpolation property)**.** Let $\mathcal{L}$ be a temporal language and $\mathbf{T}$ a frame class. Then $\mathcal{L}$ has the Δ-interpolation property over $\mathbf{T}$ whenever the following holds: let σ be a propositional signature and $K \subseteq Str_{\mathbf{T}}[\sigma]$, if both K and $\bar{K}$ are projective classes of $\mathcal{L}$ relative to $\mathbf{T}$, there is a $\mathcal{L}$-formula φ such that $K = Mod^{\sigma}_{\mathbf{T}}(\varphi)$.

Lemma 4.5.3. *Let $\mathcal{L}$ be a temporal language with Craig interpolation on $\mathbf{T}_\omega$. Then $\mathcal{L}$ has Δ-interpolation over $\mathbf{T}_\omega$.*

Lemma 4.5.4 (Δ-interpolation follows from Craig interpolation)**.** *Let $\mathcal{L}$ be a temporal language with Craig interpolation on some frame class $\mathbf{T}$. Then $\mathcal{L}$ has Δ-interpolation over $\mathbf{T}$.*

Proof. Let $\mathbf{K} \subseteq Str_{\mathbf{T}}[\sigma]$ such that both $\mathbf{K}$ and $Str_{\mathbf{T}}[\sigma]\backslash\mathbf{K}$ are projective classes of $\mathcal{L}$ relative to $\mathbf{T}$. We want to show that there is a $\xi \in \mathcal{L}[\sigma]$ such that $\mathbf{K} = Mod_{\mathcal{L},\mathbf{T}}(\xi)$.

Since $\mathbf{K}$ and $Str_{\mathbf{T}}[\sigma]\backslash\mathbf{K}$ are projective classes, there are formulas $\varphi \in \mathcal{L}[\sigma \cup \tau]$ such that $\mathbf{K} = Mod_{\mathcal{L},\mathbf{T}}(\varphi) \upharpoonright \sigma$ and $\psi \in \mathcal{L}[\sigma \cup \tau']$ such that $Str_{\mathbf{F}}[\sigma]\backslash\mathbf{K} = Mod_{\mathcal{L},\mathbf{T}}(\psi) \upharpoonright \sigma$. It follows that $\varphi \models_{\mathcal{L},\mathbf{T}} \neg\psi$. Without loss of generality, we can assume that τ and τ' are disjoint. Indeed, suppose $\tau \cap \tau' = p$ (we consider only the case where $\tau \cap \tau'$ contains one single propositional letter, as the other cases only generalize this simpler one). Now, let q be a fresh propositional letter. By closure under uniform substitution of $\mathcal{L}$, for every $\mathcal{T} \in \mathbf{T}$ and $(\mathcal{T}, V, w) \in Str_{\mathbf{T}}[\sigma \cup \tau]$ the following holds:

$$(\mathcal{T}, V, w) \models \varphi[q/p] \text{ iff } (\mathcal{T}, V', w) \models \varphi$$

where V' extends V with $V(q) = V(p)$. Hence $\mathbf{K} = Mod_{\mathcal{L},\mathbf{T}}(\varphi) \upharpoonright \sigma$ and so $\mathbf{K} = Mod_{\mathcal{L},\mathbf{T}}(\varphi[q/p]) \upharpoonright \sigma$ and the intersection of the signatures of $\varphi[q/p]$ and ψ does not contain any propositional letter not in σ.

Since $\mathcal{L}$ has interpolation, there must be a $\theta \in \mathcal{L}[\sigma]$ such that $\varphi \models_{\mathcal{L},\mathbf{T}} \theta$ and $\theta \models_{\mathcal{L},\mathbf{T}} \neg\psi$. As a last step, we will show that $Mod_{\mathcal{L},\mathbf{T}}(\theta) = \mathbf{K}$.

Suppose $\mathfrak{M} \in \mathbf{K}$. Then $\mathfrak{M} = \mathfrak{N} \upharpoonright \sigma$ for some $\mathfrak{N} \in Mod_{\mathcal{L},\mathbf{T}}(\varphi)$. Since $\varphi \models_{\mathcal{L},\mathbf{T}} \theta$, it follows that $\mathfrak{N} \models \theta$. By the expansion property, $\mathfrak{M} \models \theta$. Conversely, suppose $\mathfrak{M} \notin K$. Then $\mathfrak{M} = \mathfrak{N} \upharpoonright \sigma$ for some $\mathfrak{N} \in Mod_{\mathcal{L},\mathbf{T}}(\psi)$. Since $\theta \models_{\mathcal{L},\mathbf{T}} \neg\psi$, it follows that $\mathfrak{N} \not\models \theta$. By the expansion property, $\mathfrak{M} \not\models \theta$. □

The proof of Lemma 4.5.3 given below is similar to the one given in [39] (we only need to remark that the substitution property assumed here of abstract temporal languages is stronger than the *renaming* property assumed in [39] of abstract modal languages).

Now we will show that $\mathsf{LTL}(\mathsf{F}^<,\mathsf{X})$ is contained in the interpolation closure of $\mathsf{LTL}(\mathsf{F}^<)$ over $\mathbf{T}_\omega$. As an intermediate step, we show that in every extension of $\mathsf{LTL}(\mathsf{F}^<)$ having Craig interpolation, the property $\mathsf{X}p$ is "definable". By this, we mean the following:

Lemma 4.5.5. *Let $\mathcal{L}$ be an extension of $\mathsf{LTL}(\mathsf{F}^<)$ with Craig-interpolation over $\mathbf{T}_\omega$. Then there is $\xi \in \mathcal{L}[\{p\}]$ such that $Mod(\xi) = Mod(\mathsf{X}p)$.*

Proof. Let q, r be new distinct propositional letters. Consider the two following projective classes of $\mathsf{LTL}(\mathsf{F}^<)$: $Mod(\mathsf{F}^<(p \wedge q) \wedge \neg \mathsf{F}^<\mathsf{F}^<q) \upharpoonright \{p\}$ and $Mod((\mathsf{F}^<(\neg p \wedge r) \wedge \neg \mathsf{F}^<\mathsf{F}^<r) \vee \mathsf{G}^<\bot) \upharpoonright \{p\}$. As $\mathsf{LTL}(\mathsf{F}^<) \subseteq \mathcal{L}$, these two classes are also projective classes of $\mathcal{L}$ (by Lemma 4.2.4). They also complement each other, as a $\{p\}$-structure belongs to the first class exactly when the first node of this structure has a successor node where p holds and it belongs to the second class in all other cases. By Δ-interpolation for $\mathcal{L}$ on $\mathbf{T}$, it follows that the first class is definable in $\mathcal{L}$ by means of some formula ξ in signature $\{p\}$, i.e., there is $\xi \in \mathcal{L}[\{p\}]$ such that $Mod(\mathsf{X}p) = Mod(\xi)$. □

Theorem 4.5.6. *Every extension of $\mathsf{LTL}(\mathsf{F}^<)$ with Craig interpolation over $\mathbf{T}_\omega$ is an extension of $\mathsf{LTL}(\mathsf{F}^<,\mathsf{X})$ over $\mathbf{T}_\omega$.*

Proof. Let $\mathcal{L}$ be an extension of $\mathsf{LTL}(\mathsf{F}^<)$ with Craig interpolation over $\mathbf{T}_\omega$ and σ a propositional signature. We show by induction on the complexity of φ (number of Boolean and temporal operators in φ) that for all $\varphi \in \mathsf{LTL}(\mathsf{F}^<,\mathsf{X})[\sigma]$, there exists $\varphi' \in \mathcal{L}[\sigma]$ such that $Mod(\varphi) = Mod(\varphi')$. The base case is clear. The induction hypothesis says that for all σ, for all $\varphi \in \mathsf{LTL}(\mathsf{F}^<,\mathsf{X})[\sigma]$ of complexity at most n, there exists $\varphi' \in \mathcal{L}[\sigma]$ such that $Mod(\varphi) = Mod(\varphi')$. Now let φ be of complexity $n+1$. If $\varphi := \mathsf{X}\psi$, by induction hypothesis there exists $\psi' \in \mathcal{L}[\sigma]$ such that $Mod(\psi) = Mod(\psi')$. Pick any $p \notin \sigma$. By Lemma 4.5.5 and the expansion property we know:

1. There is $\xi \in \mathcal{L}[\sigma \cup \{p\}]$ such that $Mod(\mathsf{X}p) = Mod(\xi)$.

We will define φ' as $\xi[p/\psi'] \in \mathcal{L}[\sigma]$ (by closure under uniform substitution of $\mathcal{L}$, such a formula exists). We need to show that $Mod(\mathsf{X}\psi) = Mod(\xi[p/\psi'])$. From 1 we can derive as a particular case:

2. For any $(\mathcal{T}, V, w) \in Str_{\mathbf{T}}[\sigma \cup \{p\}]$ where $V(p) = \{w_i \mid (F, V, w_i) \models \psi'\}$, $(\mathcal{T}, V, w) \models \xi$ iff there exists $w' \in D$ such that $w < w'$, there is no w'' such that $w < w'' < w'$ and $(\mathcal{T}, V, w') \models p$.

Now by closure under uniform substitution of $\mathcal{L}$, 2 is equivalent to the following:

3. For any $(\mathcal{T}, V, w) \in Str_{\mathbf{T}}[\sigma]$, $(F, V, w) \models \xi[p/\psi']$ iff there exists $w' \in D$ such that $w < w'$, there is no w'' such that $w < w'' < w'$ and $(F, V, w') \models p[p/\psi']$.

Finally, ψ' and $p[p/\psi']$ holding exactly in the same models, we can replace $p[p/\psi']$ by ψ' in the second member of the equivalence in 3. It follows that $Mod(\mathsf{X}\psi) = Mod(\xi[p/\psi'])$. We can use similar arguments for the operator $\mathsf{F}^<$ and for Boolean connectives. □

By putting Lemma 4.5.3 to use, we now improve Theorem 4.5.6 and identify the interpolation closure of $\mathsf{LTL}(\mathsf{F}^<)$.

Theorem 4.5.7. *$\mu\mathsf{TL}$ is the interpolation closure of $\mathsf{LTL}(\mathsf{F}^<,\mathsf{X})$ over $\mathbf{T}_\omega$.*

Proof. Let σ be a propositional signature. Now let $K \subseteq Str_{\mathbf{T}_\omega}[\sigma]$ be definable by a $\mu\mathsf{TL}$-sentence φ in signature σ. As $\mu\mathsf{TL}$ is closed under negation, there is a $\mu\mathsf{TL}$-sentence $\neg\varphi$ in signature σ, which defines the complement of K over $Str_{\mathbf{T}_\omega}[\sigma]$. It follows by Theorem 4.3.2 that both K and its complement are projective classes of $\mathsf{LTL}(\mathsf{F}^<,\mathsf{X})$. Now consider a temporal language $\mathcal{L} \supseteq \mathsf{LTL}(\mathsf{F}^<,\mathsf{X})$ with Craig interpolation over $\mathbf{T}_\omega$. By Lemma 4.2.4, K and its complement are also projective classes of $\mathcal{L}$ and by Lemma 4.5.3, it follows that K is definable in $\mathcal{L}$. □

4.5.2 The Interpolation Closure of $\mathsf{LTL}(\mathsf{F})$

For the case of the stutter-invariant languages $\mathsf{LTL}(\mathsf{F})$ and $\mathsf{LTL}(\mathsf{U})$, we need to refine the notion of Δ-interpolation, by considering harmonious projective classes.

Definition 4.5.8 (Harmonious Δ-interpolation property)**.** Let $\mathcal{L}$ be a temporal language. Then $\mathcal{L}$ has the harmonious Δ-interpolation property over $\mathbf{T}_\omega$ whenever the following holds. Let K be a class of $\mathcal{L}$-structures based on $\mathbf{T}_\omega$. If both K and $\bar{K}$ are harmonious projective classes of $\mathcal{L}$ relative to $\mathbf{T}_\omega$, there is a $\mathcal{L}$-formula φ such that $K = Mod_{\mathbf{T}_\omega}(\varphi)$.

Lemma 4.5.9. *If $\mathcal{L}_1 \subseteq \mathcal{L}_2$, then every harmonious projective class of $\mathcal{L}_1$ is also a harmonious projective class of $\mathcal{L}_2$.*

Definition 4.5.10 (Harmonious temporal language)**.** A temporal language $\mathcal{L}$ is *harmonious* for $\mathbf{T}_\omega$ if the following holds. For every $\sigma \subseteq \tau$ propositional signatures, there is a formula $\varphi \in \mathcal{L}[\tau]$ such that for every $\mathfrak{M} \in Str_{\mathbf{T}_\omega}[\tau]$, $\mathfrak{M} \models \varphi$ if and only if $\mathfrak{M}$ is an harmonious expansion of $\mathfrak{M} \restriction \sigma$.

Proposition 4.5.11. $\mathsf{LTL}(\mathsf{U})$ *and its extensions are harmonious for* $\mathbf{T}_\omega$*.*

Proof. Fix $\sigma \subseteq \tau$ with $|\sigma| = n$, $|\tau \backslash \sigma| = m$. We can represent any valuation over σ by a finite conjunction of atoms and negations of atoms. Let $\{\sigma_i \mid i \in 2^n\}$ be the set of all such conjunctions. Also, for each σ_i, we define the corresponding set $\{\tau^i_j \mid j \in 2^m\}$ as the set of conjunctions representing all possible ways of extending to τ the valuation represented by σ_i. Now for every $\mathfrak{M} \in Str_{\mathbf{T}}[\tau]$,

$$\mathfrak{M} \models \bigwedge_{i,j \in 2^n} (\sigma_i \mathsf{U} \sigma_j \rightarrow \bigvee_{k,l \in 2^m} \tau^i_k \mathsf{U} \tau^j_l)$$

if and and only if $\mathfrak{M}$ is an harmonious expansion of $\mathfrak{M} \restriction \sigma$, i.e., $\mathsf{LTL}(\mathsf{U})$ is harmonious. It is immediate from definition 4.2.2 that every extension of a temporal language which is harmonious for $\mathbf{T}_\omega$ is also harmonious for $\mathbf{T}_\omega$. □

Lemma 4.5.12. *Let* $\mathcal{L}$ *be a temporal language which has Craig interpolation and is harmonious for* $\mathbf{T}_\omega$*. Then* $\mathcal{L}$ *has harmonious* Δ*-interpolation over* $\mathbf{T}_\omega$*.*

Proof. $\mathcal{L}$ being harmonious, we can use the formula φ in Definition 4.5.10 and appeal for the proof of Lemma 4.5.12 to the same classical argument as for Lemma 4.5.3. Let $\mathbf{K} \subseteq Str[\sigma]$ such that both $\mathbf{K}$ and $Str_{\mathbf{T}}[\sigma] \backslash \mathbf{K}$ are harmonious projective classes of $\mathcal{L}$ relative to $\mathbf{T}$. Then there is $\varphi \in Fml_{\mathcal{L}}[\tau]$ with $\tau \supseteq \sigma$ such that for all $\mathfrak{M} \in Str_{\mathbf{T}}[\tau]$, $\mathfrak{M} \restriction \sigma \in K$ iff $\mathfrak{M} \models \varphi$ and $\mathfrak{M}$ is an harmonious expansion of $\mathfrak{M} \restriction \sigma$. Also there is $\psi \in Fml_{\mathcal{L}}[\tau']$ with $\tau' \supseteq \sigma$ such that for all $\mathfrak{M} \in Str_{\mathbf{T}}[\tau']$, $\mathfrak{M} \restriction \sigma \in K$ iff $\mathfrak{M} \models \psi$ and $\mathfrak{M}$ is an harmonious expansion of $\mathfrak{M} \restriction \sigma$. As $\mathcal{L}$ is harmonious for $\mathbf{T}$, it follows that there is ξ such that $\varphi \wedge \xi \models_{\mathcal{L},\mathbf{T}} \neg(\psi \wedge \xi)$. The remaining of the proof is as in Theorem 4.5.3. □

Theorem 4.5.13. *Every extension of* $\mathsf{LTL}(\mathsf{F})$ *with Craig interpolation over* $\mathbf{T}_\omega$ *is an extension of* $\mathsf{LTL}(\mathsf{U})$ *over* $\mathbf{T}_\omega$*.*

Proof. The reasoning is similar as in the case of Lemma 4.5.6 and Theorem 4.5.6, but we consider $Mod(p\mathsf{U}q) = Mod(\mathsf{G}(\mathsf{F}r \rightarrow r) \wedge \mathsf{F}(q \wedge r) \wedge \mathsf{G}((r \wedge \neg q) \rightarrow p)) \restriction \{p, q\}$ and $Mod(\neg p\mathsf{U}q) = Mod(\mathsf{F}q \rightarrow (\mathsf{F}(\neg p \wedge r) \wedge \mathsf{G}(\mathsf{F}r \rightarrow \neg q))) \restriction \{p, q\}$. □

Theorem 4.5.14. $\mu\mathsf{TL}(\mathsf{U})$ *is the interpolation closure of* $\mathsf{LTL}(\mathsf{U})$ *over* $\mathbf{T}_\omega$*.*

Proof. Let σ be a modal signature. Now let $K \subseteq Str_{\mathbf{T}_\omega}[\sigma]$ be definable by a $\mu\mathsf{TL}(\mathsf{U})$-sentence φ in signature σ. As $\mu\mathsf{TL}(\mathsf{U})$ is closed under negation, there is a $\mu\mathsf{TL}(\mathsf{U})$-sentence $\neg\varphi$ in signature σ, which defines the complement $\bar{K} \subseteq Str_{\mathbf{T}_\omega}[\sigma]$ of K over $Str_{\mathbf{T}_\omega}[\sigma]$. By Theorem 4.3.14, both K and $\bar{K}$ are harmonious projective classes of $\mathsf{LTL}(\mathsf{U})$. Now consider a temporal language $\mathcal{L} \supseteq \mathsf{LTL}(\mathsf{U})$ with Craig interpolation over $\mathbf{T}$. By Lemma 4.5.9, K and $\bar{K}$ are also harmonious projective classes of $\mathcal{L}$. By Proposition 4.5.11, $\mathcal{L}$ is harmonious and by Lemma 4.5.12, it follows that K is definable in $\mathcal{L}$, i.e., $\mathcal{L} \supseteq \mu\mathsf{TL}(\mathsf{U})$. □

4.6 Finite Linear Orders

We restricted our attention to the frame class $\mathbf{T}_\omega$, but our results easily extend to finite linear orders. Let $\mathbf{T}_{\mathit{fin}}$ be the class of frames $(D, <)$ where D is a finite set and $<$ is a strict linear order on D. All the definitions and results that we gave relative to $\mathbf{T}_\omega$ also apply to $\mathbf{T}_{\mathit{fin}}$. An analogous of Theorem 4.3.2 for $\mathbf{T}_{\mathit{fin}}$ can be obtained by considering automata on finite words. The proof of Proposition 4.3.13 can similarly be adapted by considering stutter-invariant automata on finite words. In the proof of Lemma 4.3.9, we can define $(\mathsf{X}\psi)^*$ as $\bigvee_{g \neq g'}(\beta_g \mathsf{U}(\beta_{g'} \wedge \psi^*))$ (i.e., we keep only the second disjoint, as no finite stutter free linear order exhibits two successor points satisfying the same set of proposition letters). The remaining of our arguments do not need any further adjustment.

4.7 Conclusion

In this chapter, we studied the temporal fragments of linear time μ-calculus satisfying Craig interpolation, showing essentially that there are only three distinct such fragments: $\mu\mathsf{TL}$ itself, $\mu\mathsf{TL}(\mathsf{U})$, and $\mathsf{LTL}(\mathsf{X})$. These results reconfirm the robustness of (linear time) μ-calculus as compared to less expressive temporal logics. They also allow to identify $\mu\mathsf{TL}(\mathsf{U})$ as a particularly well-behaved linear-time logic which does not seem to have been studied before. In particular, complete axiomatizations were already known for $\mu\mathsf{TL}$ and $\mathsf{LTL}(\mathsf{X})$ (see Chapter 2), but this was not the case for $\mu\mathsf{TL}(\mathsf{U})$. In the next Chapter, we will study this logic further by providing such a complete axiomatization.

We are currently working on extending our interpolation results to other flows of time such as finite trees, infinite trees, and infinite linear orders other than the natural numbers (as in [34]). There are some important differences in these settings. For example, it is known (see [3]) that the branching time temporal logic with only Since and Until has Craig interpolation, while linear time fails to have this property. Also there is still no definitive consensus on the appropriate notion of stuttering for infinite branching time (see [81]). Finally, let us note that whether Propositional Dynamic Logic PDL (see [26]), which can be defined as a semantic fragment of the μ-calculus, satisfies some form of interpolation is still an open problem. It would be worth trying to obtain at least partial results for PDL on finite trees by using our methods.

Chapter 5

Complete Axiomatization of the Stutter-Invariant Fragment of the Linear Time μ-Calculus

5.1 Introduction

In the previous chapter, we encountered the phenomenon of stutter-invariance, which is is a property that is argued by some authors (see [101]) to be natural and desirable for a temporal logic, especially in the context of concurrent systems. Let us recall that, roughly, a temporal logic is stutter-invariant if it cannot detect the addition of identical copies of a state. The stutter-invariant fragment of linear-time temporal logic LTL is known to be its "Until"-only fragment $\mathsf{LTL(U)}$ and is obtained by disallowing the use of the "Next" operator (see [116]). It has been extensively studied and it is widely used as a specification language. Nevertheless, it has been pointed out (see in particular [61]) that $\mathsf{LTL(U)}$ fails to characterize the class of stutter-invariant ω-regular languages. In order to extend the expressive power of this framework, while retaining stutter-invariance, some ways of extending it have been proposed. In [61], Kousha Etessami proposed for instance the logic $\mathsf{SI\text{-}QLTL}$, which extends $\mathsf{LTL(U)}$ by means of a certain restricted type of quantification over proposition letters. He showed that $\mathsf{SI\text{-}QLTL}$ characterizes exactly stutter-invariant ω-regular languages.

In this chapter, we will focus on $\mu\mathsf{TL(U)}$, which we defined as the fixpoint extension of the "Until"-only fragment of linear-time temporal logic. In the previous chapter we showed that $\mu\mathsf{TL(U)}$ has exactly the same expressive power as $\mathsf{SI\text{-}QLTL}$, which implies that it also characterizes exactly stutter-invariant ω-regular languages. We also showed that it satisfies uniform interpolation, which is a sign that $\mu\mathsf{TL(U)}$ is a well-behaved logic. Additionally, it is known that $\mathsf{LTL(U)}$ is PSPACE complete both for model checking and for satisfiability (c.f. [53]). It is also know that $\mu\mathsf{TL}$ is PSPACE complete both for model checking and for satisfiability (c.f. [130]). So PSPACE completeness follows for $\mu\mathsf{TL(U)}$ in both cases. This is another argument in favor of $\mu\mathsf{TL}(\mathsf{U})$: while much more expressive than $\mathsf{LTL}(\mathsf{U})$, it has the same complexity. Here we further contribute to the study of the logical properties of $\mu\mathsf{TL}(\mathsf{U})$ by completely axiomatizing it over the class of ω-words and over the class of finite words. We introduce for this end another logic, which we call $\mu\mathsf{TL}(\Diamond_\Gamma)$, and which is a variation of $\mu\mathsf{TL}$ where the Next time operator is replaced by the family of its stutter-invariant counterparts. We use this logic as a technical tool to show completeness results for $\mu\mathsf{TL}(\mathsf{U})$.

Outline of the chapter: In Section 5.2, we recall basic facts and notions about linear-time μ-calculus $\mu\mathsf{TL}$. We also give a precise definition of the notion of stutter-invariance and introduce $\mu\mathsf{TL}(\mathsf{U})$, the stutter-invariant fragment of $\mu\mathsf{TL}$. In Section 5.3, we introduce the logic $\mu\mathsf{TL}(\Diamond_\Gamma)$ and show that $\mu\mathsf{TL}(\mathsf{U})$ and $\mu\mathsf{TL}(\Diamond_\Gamma)$ have exactly the same expressive power on finite and ω-words. In section 5.4, we give axiomatizations of $\mu\mathsf{TL}(\Diamond_\Gamma)$ that we respectively show to be complete on these two classes of structures. Finally, these results are put to use in Section 5.5, where we show similar completeness results for $\mu\mathsf{TL}(\mathsf{U})$.

5.2 Preliminaries

In this section, we recall the syntax and semantics of linear time μ-calculus $\mu\mathsf{TL}$. We also recall its axiomatization on some interesting classes of linear orders, as well as the notion of stutter-invariance.

5.2.1 Linear Time μ-Calculus

By a *propositional vocabulary* we mean a countable (possibly finite) non-empty set of propositional letters $\sigma = \{p_i \mid i \in I\}$.

Definition 5.2.1 (Syntax of $\mu\mathsf{TL}$)**.** Let σ be a propositional vocabulary and $\mathcal{V} = \{x_1, x_2, \ldots\}$ a disjoint countably infinite set of *propositional variables.* We inductively define the set of $\mu\mathsf{TL}$-formulas in vocabulary σ as follows:

$$\varphi, \psi, \xi := At \mid \top \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \Diamond\varphi \mid \mu x_i.\xi$$

where $At \in \sigma \cup \mathcal{V}$ and, in the last clause, x_i occurs only positively in ξ (i.e., within the scope of an even number of negations). We will use $\varphi \to \psi$, $\nu x_i.\xi$, $\Box\varphi$, $\varphi\mathsf{U}\psi$, $\mathsf{F}\varphi$ as shorthand for, respectively, $\neg(\varphi \wedge \neg\psi)$, $\neg\mu x_i.\neg\xi(\neg x_i)$, $\neg\Diamond\neg\varphi$, $\mu y.(\psi \vee (\varphi \wedge \Diamond y))$ and $\mu y.(\varphi \vee \Diamond y)$. We will also use $\mathsf{G}\varphi$ as shorthand for $\neg\mathsf{F}\neg\varphi$.

A *linear flow of time* is a structure $\mathcal{L} = (W, <)$, where W is a non-empty set of points and $<$ is a linear order on W. A *linear time σ-structure* is a structure $\mathfrak{M} = (\mathcal{L}, V)$ where $\mathcal{L} = (W, R)$ is a linear flow of time and $V : \sigma \to \wp(W)$ a valuation. Whenever $w \in W$ is a point, we call $\mathfrak{M}, w$ a *pointed σ-structure.* Linear time μ-calculus is usually considered over restricted classes of linear orders. In this paper, we will only consider it over the following classes:

- $\mathbf{L}_\omega$, the class of linear orders of order type ω, i.e., flows of time $(W, <)$ that are isomorphic to $(\mathbb{N}, <)$, where $\mathbb{N}$ is the set of natural numbers with the natural ordering,
- $\mathbf{L}_{\mathit{fin}}$, the class of finite linear orders,
- the union $\mathbf{L}_\omega \cup \mathbf{L}_{\mathit{fin}}$ of these two classes

We will often refer to structures based on $\mathbf{L}_\omega$ as *ω-words* or $\mathbf{L}_\omega$-structures, to structures based on $\mathbf{L}_{\mathit{fin}}$ as *finite words* or $\mathbf{L}_{\mathit{fin}}$-structures and more generally, to structures based on $\mathbf{L}$ as $\mathbf{L}$-structures.

Definition 5.2.2 (Semantics of $\mu\mathsf{TL}$)**.** Given a $\mu\mathsf{TL}$-formula φ, a structure $\mathfrak{M} = ((W, <), V)$ and an assignment $g : \mathcal{V} \to \wp(W)$, we define a subset $[\![\varphi]\!]_{\mathfrak{M},g}$ of $\mathfrak{M}$ that is interpreted as the set of points at which φ is true. This subset is defined by induction in the usual way. Let $\mathrm{ImSuc}(w)$, be the set of direct successors of the point w with respect to $<$, we only recall:

$$[\![\Diamond\varphi]\!]_{\mathfrak{M},g} = \{w \in W : [\![\varphi]\!]_{\mathfrak{M},g} \cap \mathrm{ImSuc}(w) \neq \emptyset\}$$

$$[\![\mu x.\varphi]\!]_{\mathfrak{M},g} = \bigcap\{A \subseteq W : [\![\varphi]\!]_{\mathfrak{M},g[x/A]} \subseteq A\}$$

where $g[x/A]$ is the assignment defined by $g[x/A](x) = A$ and $g[x/A](y) = g(y)$ for all $y \neq x$. If $w \in [\![\varphi]\!]_{\mathfrak{M},g}$, we write $\mathfrak{M}, w \models_g \varphi$ and we say that φ is true at $w \in \mathfrak{M}$ under the assignment g. If φ is a sentence, or if $\mathfrak{M}, w \models_g \varphi$ holds for every valuation g, we simply write $\mathfrak{M}, w \models \varphi$.

Note that the $\Diamond$ operator is interpreted as the "Next" operator of temporal logic and that the temporal operators U and F that we defined as shorthand have their usual meaning that we recall here:

- $(\mathcal{L}, V, w) \models_g \mathsf{F}\varphi$ iff there exists w' such that $w \leq w'$ and $(\mathcal{L}, V, w') \models_g \varphi$
- $(\mathcal{L}, V, w) \models_g \varphi\mathsf{U}\psi$ iff there exists w' such that $w \leq w'$, $(\mathcal{L}, V, w') \models_g \psi$ and for all w'' such that $w \leq w'' < w'$, $(\mathcal{L}, V, w'') \models_g \varphi$

Before we give the complete axiomatization of $\mu\mathsf{TL}$ on $\mathbf{L}_\omega$, $\mathbf{L}_{\mathit{fin}}$ and $\mathbf{L}_\omega \cup \mathbf{L}_{\mathit{fin}}$, let us first recall the axiomatization of the *μ-calculus*. In the μ-calculus, instead of considering a linear order $<$, we consider an arbitrary binary relation R on W. In this more general context, (W, R) can be an arbitrary graph and we call it a *frame*.[1] The corresponding structures are called *Kripke structures*. Let $\mathrm{RSuc}(w) = \{w' : (w, w') \in R\}$, the semantics of $\Diamond$ is now as follows:

$$[\![\Diamond\varphi]\!]_{\mathfrak{M},g} = \{w \in W : [\![\varphi]\!]_{\mathfrak{M},g} \cap \mathrm{RSuc}(w) \neq \emptyset\}$$

[1] Note that on arbitrary graphs, we do not introduce $\mathsf{F}\varphi$, $\mathsf{G}\varphi$ and $\varphi\mathsf{U}\psi$ as shorthands for $\mu\mathsf{TL}$-formulas anymore: as we consider frames instead of linear flows of time, this would not really map the usual meaning of these temporal operators.

Definition 5.2.3. Let σ be a finite propositional vocabulary and $\varphi, \psi \in \mu$TL arbitrary formulas. We call BV(φ) and FV(φ) respectively, the set of bound variables in φ and the set of free variables in φ. The Kozen system K_μ consists of the Modus Ponens, the Substitution rule, the Necessitation rule and the following axioms and rules:

$A1$ propositional tautologies,

$A2$ $\vdash \Box\varphi \leftrightarrow \neg\Diamond\neg\varphi$ (dual),

$A3$ $\vdash \Box(\varphi \to \psi) \to (\Box\varphi \to \Box\psi)$ (K),

$A4$ $\vdash \varphi[x/\mu x.\varphi] \to \mu x.\varphi$ (fixpoint axiom),

FR If $\vdash \varphi[x/\psi] \to \psi$, then $\vdash \mu x.\varphi \to \psi$ (fixpoint rule)

where x does not belong to BV(φ) and FV(ψ) $\cap$ BV(φ) $= \emptyset$.

Theorem 5.2.4. *If φ is a μTL-formula, let $K_\mu + \varphi$ be the smallest set which contains both K_μ and φ and is closed for the Modus Ponens, Substitution, fixpoint and Necessitation rules. The following holds:*

1. *K_μ is complete with respect to the class of Kripke structures.*
2. *$K_\mu + \Diamond\varphi \leftrightarrow \Box\varphi$ is complete with respect to the class of ω-words.*
3. *$K_\mu + \Diamond\varphi \to \Box\varphi + \mu x.\Box x$ is complete with respect to the class of finite words.*
4. *$K_\mu + \Diamond\varphi \to \Box\varphi$ is complete with respect to the class of finite and ω-words.*

Proof. (i) was shown in [133] and the three other completeness results might actually be derivable from it. But direct (and simpler) proofs for (ii) and (iii) can be found respectively in [92] and [40]. In order to establish (iv), we will rely on (i), (ii) and (iii), using an argument from Johan van Benthem and Balder ten Cate (private communication). We first show the following claim:

- *Claim: Let K_x be a system extending K_μ with a finite set of axioms and closed under Substitution, Modus Ponens and the fixpoint and Necessitation rules. Let θ be a closed formula with $K_x \vdash \theta \to \Box\theta$. For all formulas ξ, if $K_x + \theta \vdash \xi$, then $K_x \vdash \theta \to \xi$.*
  The proof goes by induction on the length of K_x-derivations. The only difficult case is whenever the last line in the proof is obtained via the application of the fixpoint rule. So assume the property holds for all derivations of length n and $K_x + \theta \vdash \mu y.\varphi \to \psi$ is the last line of a derivation of length $n + 1$. We want to show that $K_x \vdash \theta \to (\mu y.\varphi \to \psi)$.

By induction hypothesis, $K_x \vdash \theta \to (\varphi[x/\psi] \to \psi)$. So by propositional tautologies also $K_x \vdash (\theta \wedge \varphi[\psi/x]) \to \psi$. By the fixpoint rule, $K_x \vdash \mu x.(\theta \wedge \varphi) \to \psi$. Now from $K_x \vdash \theta \to \Box\theta$ it follows by propositional tautologies that $K_x \vdash (\Diamond\neg\theta \vee \neg\theta) \to \neg\theta$ and by the fixpoint rule $K_x \vdash \mu x.(\neg\theta \vee \Diamond x) \to \neg\theta$, so $K_x \vdash \theta \to \neg\mu x.(\neg\theta \vee \Diamond x)$. Now it is valid in the μ-calculus that $\neg\mu x.(\neg\theta \vee \Diamond x) \to (\mu x.(\theta \wedge \varphi) \leftrightarrow \mu x.\varphi)$, so it is also derivable in K_x. It follows that $K_x \vdash \theta \to (\mu x.\varphi \to \psi)$.

Now assume that ξ is valid on finite and ω-words. As it is valid on finite words, by (iii), $K_\mu \ + \ \Diamond\varphi \to \Box\varphi \ + \ \mu x.\Box x \vdash \xi$. As $\mu x.\Box x$ satisfies the condition of the claim we get:

$$\text{(a) } K_\mu \ + \ \Diamond\varphi \to \Box\varphi \vdash \mu x.\Box x \to \xi$$

ξ is also valid on ω-words, and hence by (ii), $K_\mu \ + \ \Diamond\varphi \to \Box\varphi \ + \ \Box\varphi \to \Diamond\varphi \vdash \xi$. Note that $\Diamond\top$ can equivalently be substituted for $\Box\varphi \to \Diamond\varphi$ there. As $K_\mu \ + \ \Diamond\varphi \to \Box\varphi \vdash \neg\mu x.\Box x \to \Diamond\top$, we can also take θ to be $\neg\mu x.\Box x$, which also satisfies the condition of the claim. Indeed by the $\Diamond\varphi \to \Box\varphi$ axiom, it is enough to prove $\neg\mu x.\Box x \to \Diamond\neg\mu x.\Box x$. But this is equivalent to $\Box\mu x.\Box x \to \mu x.\Box x$, which is derivable in K_μ (since $\mu x.\Box x \leftrightarrow \Box(\mu x.\Box x)$). It follows that:

$$\text{(b) } K_\mu \ + \ \Diamond\varphi \to \Box\varphi \vdash \neg(\mu x.\Box x) \to \xi$$

$K_\mu \ + \ \Diamond\varphi \to \Box\varphi \vdash \xi$ follows from (a) and (b), which proves (iv).

□

5.2.2 Stutter-Invariance

We will now recall the syntax and semantics of $\mu\mathsf{TL}(\mathsf{U})$.We also recall our definition of stutter-invariance and recall that, in terms of expressive power, $\mu\mathsf{TL}(\mathsf{U})$ is exactly the stutter-invariant fragment of $\mu\mathsf{TL}$.

Definition 5.2.5 (Syntax of $\mu\mathsf{TL}(\mathsf{U})$)**.** Let σ be a propositional vocabulary, and let $\mathcal{V} = \{x_1, x_2, \ldots\}$ be a disjoint countably infinite set of *propositional variables.* We inductively define the set of $\mu\mathsf{TL}(\mathsf{U})$-formulas in vocabulary σ as follows:

$$\varphi, \psi, \xi := At \mid \top \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \varphi \mathsf{U} \psi \mid \mu x_i.\xi$$

where $At \in \sigma \cup \mathcal{V}$ and, in the last clause, x_i occurs only positively in ξ (i.e., within the scope of an even number of negations). We will use $\varphi \to \psi$, $\nu x_i.\xi$, $\mathsf{F}\varphi$ and $\mathsf{G}\varphi$ as shorthand for, respectively, $\neg(\varphi \wedge \neg\psi$, $\neg\mu x_i.\neg\xi(\neg x_i)$, $\top\mathsf{U}\varphi$ and $\neg(\top\mathsf{U}\neg\varphi)$.

Note that the temporal operators F and G defined as shorthand have their usual meaning. We interpret $\mu\mathsf{TL}(\mathsf{U})$-formulas in the same type of structures as $\mu\mathsf{TL}$-formulas, i.e., structures of the form $\mathfrak{M} = (\mathcal{L}, V)$ where $\mathcal{L} \in \mathbf{L}_{\mathit{fin}} \cup \mathbf{L}_\omega$.

Definition 5.2.6 (Semantics of $\mu\mathsf{TL}(\mathsf{U})$)**.** Given a $\mu\mathsf{TL}(\mathsf{U})$-formula φ, a structure $\mathfrak{M} = ((W,<),V)$ and an assignment $g : \mathcal{V} \to \wp(W)$, we define a subset $[\![\varphi]\!]_{\mathfrak{M},g}$ of $\mathfrak{M}$ that is interpreted as the set of points at which φ is true. This subset is defined by induction in the usual way. We only recall:

$$[\![\varphi \mathsf{U} \psi]\!]_{\mathfrak{M},g} = \{w \in W : \exists w' \geq w, w' \in [\![\psi]\!]_{\mathfrak{M},g} \text{ and } \forall w \leq w'' < w', w'' \in [\![\varphi]\!]_{\mathfrak{M},g}\}$$

$$[\![\mu x.\varphi]\!]_{\mathfrak{M},g} = \bigcap\{A \subseteq W : [\![\varphi]\!]_{\mathfrak{M},g[x/A]} \subseteq A\}$$

where $g[x/A]$ is the assignment defined by $g[x/A](x) = A$ and $g[x/A](y) = g(y)$ for all $y \neq x$.

In the remaining, we always assume $\mathbf{L} \in \{\mathbf{L}_\omega, \mathbf{L}_{\mathit{fin}}, \mathbf{L}_{\mathit{fin}} \cup \mathbf{L}_\omega\}$.

Definition 5.2.7 (Stuttering)**.** Let σ be a propositional signature, and $\mathfrak{M} = ((W,<),V,w)$, $\mathfrak{M}' = ((W',<),V',w')$ pointed $\mathbf{L}$-structures in vocabulary σ. We say that $\mathfrak{M}'$ is a stuttering of $\mathfrak{M}$ if and only if there is a surjective function $s : W' \to W$ such that

1. $s(w') = w$
2. for every $w_i, w_j \in W'$, $w_i < w_j$ implies $s(w_i) \leq s(w_j)$
3. for every $w_i \in W'$ and $p \in \sigma, w_i \in V'(p)$ iff $s(w_i) \in V(p)$

We say that an $\mathbf{L}$-structure $\mathfrak{M}$ is *stutter-free relative to* $\mathbf{L}$ whenever for all $\mathfrak{M}'$ such that $\mathfrak{M}$ is a stuttering of $\mathfrak{M}'$, $\mathfrak{M}'$ is isomorphic to $\mathfrak{M}$.

Let for instance $\mathfrak{M}, w$ be an ω-word in vocabulary $\{p\}$ with $V(p) = W$. $\mathfrak{M}, w$ is stutter-free relative to $\mathbf{L}_\omega$, but it is not stutter-free relative to $\mathbf{L}_{\mathit{fin}} \cup \mathbf{L}_\omega$. Indeed, let $\mathfrak{M}', w'$ be a finite word in vocabulary $\{p\}$ containing one single point w'. Assume $V'(p) = \{w'\}$, then $\mathfrak{M}, w$ is a stuttering of $\mathfrak{M}', w'$ and relative to $\mathbf{L}_{\mathit{fin}} \cup \mathbf{L}_\omega$, $\mathfrak{M}', w'$ is stutter-free, while $\mathfrak{M}, w$ is not.

Definition 5.2.8 (Stutter-Invariant Class of Structures)**.** Let σ be a propositional signature and $\mathbf{K}$ a class of $\mathbf{L}$-structures in vocabulary σ. Then $\mathbf{K}$ is a stutter-invariant class iff for every $\mathbf{L}$-structure $\mathfrak{M}$ in vocabulary σ and for every $\mathbf{L}$-stuttering $\mathfrak{M}'$ of $\mathfrak{M}$, $\mathfrak{M} \in \mathbf{K} \Leftrightarrow \mathfrak{M}' \in \mathbf{K}$.

We say that a sentence φ is stutter-invariant relative to $\mathbf{L}$ whenever the class of $\mathbf{L}$-structures in which φ is satisfied is stutter-invariant. Every $\mu\mathsf{TL}(\mathsf{U})$-sentence is stutter-invariant relative to $\mathbf{L}$ (see Chapter 4). To see that it is not possible in $\mu\mathsf{TL}(\mathsf{U})$ to define $\Diamond\varphi$, it is hence enough to observe that the sentence $\Diamond p$ is not stutter-invariant. Also, considering a $\mathbf{L}$-structure $\mathfrak{M}, w$, there is always a unique (up to isomorphism) $\mathfrak{M}', w'$ which is stutter-free relative to $\mathbf{L}$ and such that $\mathfrak{M}, w$ is a stuttering of $\mathfrak{M}', w'$. Observe that it follows that if a $\mu\mathsf{TL}(\mathsf{U})$-formula is satisfiable in some $\mathbf{L}$-structure, it is also satisfiable in a $\mathbf{L}$-structure which is stutter-free relative to $\mathbf{L}$. Additionally, on $\mathbf{L}$, we can show that $\mu\mathsf{TL}(\mathsf{U})$ is exactly the stutter-invariant fragment of $\mu\mathsf{TL}$:

Theorem 5.2.9. *Let φ be a $\mu\mathsf{TL}$-sentence which is stutter-invariant relative to $\mathbf{L}$. Then, there exists a $\mu\mathsf{TL}(\mathsf{U})$-sentence φ^* which is equivalent to φ on $\mathbf{L}$-structures.*

Proof. The proof can be found in Chapter 4. □

5.3 The Logic $\mu\mathsf{TL}(\Diamond_\Gamma)$

In this Section, we introduce the logic $\mu\mathsf{TL}(\Diamond_\Gamma)$ and we show that, as far as expressivity is concerned, it is a fragment of $\mu\mathsf{TL}$. More precisely, we show that $\mu\mathsf{TL}(\Diamond_\Gamma)$ has exactly the same expressive power as $\mu\mathsf{TL}(\mathsf{U})$. In the last Sections, we will see that $\mu\mathsf{TL}(\Diamond_\Gamma)$ can be used as a very convenient tool to show completeness results for $\mu\mathsf{TL}(\mathsf{U})$.

$\mu\mathsf{TL}(\Diamond_\Gamma)$ is a variation of $\mu\mathsf{TL}$ where instead of the regular $\Diamond$ modality, we consider the family of its stutter-invariant counterparts. For each finite set Γ of $\mu\mathsf{TL}(\Diamond_\Gamma)$-sentences, we consider a $\Diamond_\Gamma$ operator which intuitively means "at the next distinct point with respect to Γ" (i.e., distinct with respect to the values it assigns to the formulas in Γ). To design this operator, we took inspiration from [61], where a "next distinct" operator was mentioned in passing. This operator was interpreted in σ-structures as our $\Diamond_\sigma$ operator. In order to obtain a well-behaved operator, we relativize it here to any finite set Γ of sentences. This gives rise to a better-behaved logic, where we can define a natural notion of substitution and where the truth of σ-formulas in σ-structures is preserved in σ^+-expansions of these structures (with $\sigma^+ \supseteq \sigma$).

We interpret $\mu\mathsf{TL}(\Diamond_\Gamma)$-formulas in the same type of structures as $\mu\mathsf{TL}$-formulas, i.e., structures of the form $\mathfrak{M} = (\mathcal{L}, V)$ where $\mathcal{L} \in \mathbf{L}_{\mathit{fin}} \cup \mathbf{L}_\omega$. For any finite set of $\mu\mathsf{TL}(\Diamond_\Gamma)$-formulas and for any points w, w', we write $w \equiv_\Gamma w'$ if w and w' satisfy the same formulas in Γ.

Definition 5.3.1. Let σ be a finite propositional signature and $\mathcal{V} = \{x_1, x_2, \ldots\}$ a disjoint countably infinite stock of *propositional variables.* We inductively define the set of $\mu\mathsf{TL}(\Diamond_\Gamma)$-formulas as follows:

$$\varphi, \psi, \xi := At \mid \top \mid \neg\varphi \mid \varphi \wedge \psi \mid \varphi \vee \psi \mid \Diamond_\Gamma\varphi \mid \mu x_i.\xi$$

where $At \in \sigma \cup \mathcal{V}$, Γ is a finite set of $\mu\mathsf{TL}(\Diamond_\Gamma)$-formulas and, in the last clause, x_i occurs only positively in ξ (i.e., within the scope of an even number of negations). We use $\Box_\Gamma\varphi$, $\varphi \to \psi$ and $\nu x_i.\xi(x_i)$ as shorthand for $\neg\Diamond_\Gamma\neg\varphi$, $\neg(\varphi \wedge \neg\psi$ and $\neg x_i\mu\neg\xi(\neg x_i)$, respectively. We interpret $\mu\mathsf{TL}(\Diamond_\Gamma)$-formulas as $\mu\mathsf{TL}$-formulas, except that:

$$(\mathcal{L}, V, w) \models \Diamond_\Gamma\varphi \text{ if } \exists w' > w \text{ such that } w \not\equiv_\Gamma w',\ \forall w'' \text{ with } w < w'' < w', w'' \equiv_\Gamma w \text{ and } (\mathcal{L}, V, w') \models \varphi$$

The resulting logic is stuttering invariant. We write Voc(φ) for the vocabulary of φ and Voc(Γ) for $\bigcup_{\varphi\in\Gamma}$ Voc(φ). Note that we include in the vocabulary of a formula *all* the proposition letters occurring in it, including those which occur in the formulas contained in the sets Γ indexing its modalities. This remark particularly matters for the notion of substitution, as whenever a formula is to be uniformly substituted for a proposition letter, the operation has to be done everywhere, including in the formulas contained in the sets indexing the modalities. Otherwise, validity would not be preserved by uniform substitution. Consider for instance $(p \wedge \Diamond_{\{p\}}\top) \to \Diamond_{\{p\}}\neg p$. It is clear that this formula is valid and that for any $\mu\mathsf{TL}(\Diamond_\Gamma)$-formula φ, $\models (\varphi \wedge \Diamond_{\{\varphi\}}\top) \to \Diamond_{\{\varphi\}}\neg\varphi$ also holds. But it is also very clear that $\not\models (\varphi \wedge \Diamond_{\{p\}}\top) \to \Diamond_{\{p\}}\neg\varphi$.

We will now provide a way to compare $\mu\mathsf{TL}(\Diamond_\Gamma)$ and $\mu\mathsf{TL}(\mathsf{U})$, by defining two recursive procedures transforming each formula from one language into an equivalent formula from the other language.

Definition 5.3.2. Let $\Gamma = \{\varphi_0, \ldots, \varphi_{n-1}\}$ be a finite set of $\mu\mathsf{TL}(\Diamond_\Gamma)$-formulas. Whenever $\Gamma \neq \emptyset$, we define B_Γ as the set of all possible mappings $\Gamma \to \{\bot, \top\}$, and for each $g \in B_\Gamma$, we let β_g be the formula $\alpha_0 \wedge \ldots \wedge \alpha_{n-1}$ where $\alpha_j = \varphi_j$ if $g(\varphi_j) = \top$ and $\alpha_j = \neg\varphi_j$ if $g(\varphi_j) = \bot$. By convention, we set $B_\emptyset = \{\bot, \top\}$.[2]

Definition 5.3.3 ($\mu\mathsf{TL}(\mathsf{U})$-translation of a $\mu\mathsf{TL}(\Diamond_\Gamma)$-formula)**.** Let φ be a $\mu\mathsf{TL}(\Diamond_\Gamma)$-formula, we recursively define its $\mu\mathsf{TL}(\mathsf{U})$-translation $\varphi_{\mu\mathsf{TL}(\mathsf{U})}$ via the following procedure. $At_{\mu\mathsf{TL}(\mathsf{U})} = At$, $(\neg\varphi)_{\mu\mathsf{TL}(\mathsf{U})} = \neg\varphi_{\mu\mathsf{TL}(\mathsf{U})}$, $(\varphi \wedge \psi)_{\mu\mathsf{TL}(\mathsf{U})} = \varphi_{\mu\mathsf{TL}(\mathsf{U})} \wedge \psi_{\mu\mathsf{TL}(\mathsf{U})}$, $(\mu x.\varphi)_{\mu\mathsf{TL}(\mathsf{U})} = \mu x.\varphi_{\mu\mathsf{TL}(\mathsf{U})}$, and $(\Diamond_\Gamma\varphi)_{\mu\mathsf{TL}(\mathsf{U})} = \bigvee_{g\in B_\Gamma}(\beta_g \wedge \beta_g\mathsf{U}(\neg\beta_g \wedge \varphi_{\mu\mathsf{TL}(\mathsf{U})}))$.

Proposition 5.3.4. *Let* $\mathbf{L} \in \{\mathbf{L}_\omega, \mathbf{L}_{fin}, \mathbf{L}_\omega \cup \mathbf{L}_{fin}\}$ *and* φ *be a* $\mu\mathsf{TL}(\Diamond_\Gamma)$*-formula,* φ *and* $\varphi_{\mu\mathsf{TL}(\mathsf{U})}$ *are equivalent on* $\mathbf{L}$*-structures.*

Proof. We show that a class of σ-structures based on $\mathbf{L}_{fin} \cup \mathbf{L}_\omega$, is definable by a $\mu\mathsf{TL}(\Diamond_\Gamma)$-formula if and only if it is definable by its $\mu\mathsf{TL}(\mathsf{U})$-translation. Let $((W,<),V,w)$ be a σ-structure (by induction hypothesis, we assume the property holds for φ, $\varphi_{\mu\mathsf{TL}(\mathsf{U})}$).

Assume $((W,<),V,w) \models \Diamond_\Gamma\varphi$, i.e., there exists $w' > w$ such that $w \not\equiv_\Gamma w'$ and $\forall w''$ with $w < w'' < w', w'' \equiv_\Gamma w$ and $((W,<),V,w') \models \varphi$. So there are $g \neq g' \in B_\Gamma$ such that $\mathfrak{M}, w \models \beta_g$ and there exists $w' > w$ with $((W,<),V,w') \models \beta_{g'} \wedge \varphi_{\mu\mathsf{TL}(\mathsf{U})}$ and for all w'' such that $w \leq w'' < w'$, $((W,<),V,w'') \models \beta_g$. By induction hypothesis, $((W,<),V,w) \models \bigvee_{g\in B_\Gamma}(\beta_g \wedge \beta_g\mathsf{U}(\neg\beta_g \wedge \varphi_{\mu\mathsf{TL}(\mathsf{U})}))$.

Assume $((W,<),V,w) \models \bigvee_{g\in B_\Gamma}(\beta_g \wedge \beta_g\mathsf{U}(\neg\beta_g \wedge \varphi_{\mu\mathsf{TL}(\mathsf{U})}))$. So there are $g \neq g'$ such that $\beta_g\mathsf{U}(\beta_{g'} \wedge \varphi_{\mu\mathsf{TL}(\mathsf{U})})$, i.e., there exists w' such that $w \leq w'$, $((W,<),V,w') \models \beta_{g'} \wedge \varphi_{\mu\mathsf{TL}(\mathsf{U})}$ and for all w'' such that $w \leq w'' < w'$, $((W,<),V,w'') \models \beta_g$. As $g \neq g'$, also $w \not\equiv_\Gamma w'$. By induction hypothesis, $((W,<),V,w) \models \Diamond_\Gamma\varphi$. □

[2] We adopt this convention because we allowed Γ to be empty (see the instantiation of Axiom $A6'$ where $\Gamma = \emptyset$, our convention will guaranty that $\Diamond_\emptyset\varphi$, which is not satisfiable, is also inconsistent), but we could also have required that $\Gamma \neq \emptyset$.

Definition 5.3.5 ($\mu\mathsf{TL}(\Diamond_\Gamma)$-translation of a $\mu\mathsf{TL}(\mathsf{U})$-formula)**.** Let φ be $\mu\mathsf{TL}(\mathsf{U})$-formula in vocabulary σ, we recursively define its $\mu\mathsf{TL}(\Diamond_\Gamma)$-translation $\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ via the following procedure. $At_{\mu\mathsf{TL}(\Diamond_\Gamma)} = At$, $(\neg\varphi)_{\mu\mathsf{TL}(\Diamond_\Gamma)} = \neg\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$, $(\varphi \wedge \psi)_{\mu\mathsf{TL}(\Diamond_\Gamma)} = \varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)} \wedge \psi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$, $(\mu x.\varphi)_{\mu\mathsf{TL}(\Diamond_\Gamma)} = \mu x.\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$, and $(\varphi \mathsf{U} \psi)_{\mu\mathsf{TL}(\Diamond_\Gamma)} = \mu x.(\psi_{\mu\mathsf{TL}(\Diamond_\Gamma)} \vee (\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)} \wedge \Diamond_\sigma x))$.

Proposition 5.3.6. *Let* $\mathbf{L} \in \{\mathbf{L}_\omega, \mathbf{L}_{\mathit{fin}}, \mathbf{L}_\omega \cup \mathbf{L}_{\mathit{fin}}\}$ *and* φ *be a* $\mu\mathsf{TL}(\mathsf{U})$*-formula. Then* φ *and* $\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ *are equivalent on* $\mathbf{L}$*-structures.*

Proof. We show that a class of σ-structures based on $\mathbf{L}_{\mathit{fin}} \cup \mathbf{L}_\omega$, is definable by a $\mu\mathsf{TL}(\mathsf{U})$-formula if and only if it is definable by its $\mu\mathsf{TL}(\Diamond_\Gamma)$-translation. Let $((W,<),V,w)$ be a σ-structure (by induction hypothesis, we assume the property holds for φ, $\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ and ψ, $\psi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ respectively).

Assume $((W,<),V,w) \models \varphi \mathsf{U} \psi$. This means either that w satisfies ψ, or w satisfies φ and it is separated from some subsequent w' satisfying ψ by a finite sequence of points which all satisfy φ. So, by induction hypothesis, $((W,<),V,w) \models \mu x.(\psi_{\mu\mathsf{TL}(\Diamond_\Gamma)} \vee (\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)} \wedge \Diamond_\sigma x))$, because $\mu x.(\psi_{\mu\mathsf{TL}(\Diamond_\Gamma)} \vee (\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)} \wedge \Diamond_\sigma x))$ states that the current state belongs to the least fixpoint which contains all the points satisfying $\psi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$, together with all the points that satisfy $\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ and which are immediate predecessors of a point which is already in the fixpoint.

Assume $\mu x.(\psi_{\mu\mathsf{TL}(\Diamond_\Gamma)} \vee (\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)} \wedge \Diamond_\sigma x))$, i.e., w belongs to the least fixpoint which contains all the points satisfying $\psi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$, together with all the points that satisfy $\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ and which are immediate predecessors of a point which is already in the fixpoint. This means that either w satisfies $\psi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$, or it satisfies $\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ and it is separated from some subsequent w' satisfying $\psi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ by a finite sequence of successor points which all satisfy $\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ and by induction hypothesis, $((W,<),V,w) \models \varphi U \psi$. □

Corollary 5.3.7. $\mu\mathsf{TL}(\mathsf{U})$ *and* $\mu\mathsf{TL}(\Diamond_\Gamma)$ *have the same expressive power on the class of finite and* ω*-words.*

Proof. Follows from Propositions 5.3.4 and 5.3.6. □

Remark 5.3.8. It follows that $\Diamond_\Gamma$ can be used as shorthand either in $\mu\mathsf{TL}$ or in $\mu\mathsf{TL}(\mathsf{U})$, that U can be used as shorthand in $\mu\mathsf{TL}(\Diamond_\Gamma)$ and that $\mu\mathsf{TL}(\Diamond_\Gamma)$ is definable as a semantic fragment of $\mu\mathsf{TL}$. In the remainder of the chapter, this will be assumed. ⊣

Now in order to see that both $\mu\mathsf{TL}(\Diamond_\Gamma)$ and $\mu\mathsf{TL}(\mathsf{U})$ strictly extend $\mathsf{LTL}(\mathsf{U})$, let us give an example of a class of finite words which is known to be not definable in $\mathsf{LTL}(\mathsf{U})$, while it is definable in $\mu\mathsf{TL}(\Diamond_\Gamma)$ and $\mu\mathsf{TL}(\mathsf{U})$. The following $\mu\mathsf{TL}(\Diamond_\Gamma)$-formula is satisfied at the root of a finite word in any vocabulary σ expanding $\{p\}$ exactly whenever this word contains an even number of sequences (of arbitrary length) of states satisfying p:

$$\Phi := \mu x.((\neg p \wedge \Box_p x) \vee (p \wedge \Diamond_p \mu y.((\neg p \wedge \Diamond_p y) \vee (p \wedge \Box_p x))))$$

By Proposition 5.3.4, $\Phi_{\mu\mathsf{TL}(\mathsf{U})} \in \mu\mathsf{TL}(\mathsf{U})$ is equivalent to Φ. Note also that by removing the subscripts in the modal operators in Φ, we obtain the following $\mu\mathsf{TL}$-formula:

$$\Phi' := \mu x.((\neg p \wedge \Box x) \vee (p \wedge \Diamond \mu y.((\neg p \wedge \Diamond y) \vee (p \wedge \Box x))))$$

which is satisfied at the root of a finite word in vocabulary σ exactly whenever this word contains an even number of p (i.e., whenever p is satisfied at an even number of states). Note also that the property defined via Φ is the closure under stuttering of the one defined via Φ'. This suggests a natural procedure - via indexing of the modalities in the formula - to characterize in $\mu\mathsf{TL}(\Diamond_\Gamma)$ the closure under stuttering of $\mu\mathsf{TL}$-properties, which illustrates a close connection between the syntax of $\mu\mathsf{TL}$ and $\mu\mathsf{TL}(\Diamond_\Gamma)$. It is admittedly difficult to write specifications in $\mu\mathsf{TL}$ (c.f. [131]), but the difficulty does not seem to be higher in the case of $\mu\mathsf{TL}(\Diamond_\Gamma)$.

5.4 Complete Axiomatization of $\mu\mathsf{TL}(\Diamond_\Gamma)$

In this Section, we show some completeness results for the logic $\mu\mathsf{TL}(\Diamond_\Gamma)$. We will use them in the next Section as a tool to obtain similar results for the logic $\mu\mathsf{TL}(\mathsf{U})$.

Proposition 5.4.1. *Let φ be a $\mu\mathsf{TL}(\Diamond_\Gamma)$-formula in vocabulary σ containing no free occurrence of the variable x. On the class of finite and ω-words, the following formulas are equivalent:*

- $\bigvee_{g\in B_\sigma}(\beta_g \wedge \mu x.((\neg\beta_g \wedge \varphi) \vee (\beta_g \wedge \Diamond_\sigma x))$
- $\bigvee_{g\in B_\sigma}(\beta_g \wedge \mu x.((\neg\beta_g \wedge \varphi) \vee (\beta_g \wedge \Diamond x))$
- $\Diamond_\sigma \varphi$

Proof. Recall that U can be defined as shorthand in $\mu\mathsf{TL}(\Diamond_\Gamma)$. We already noted in Section 5.2 and in Proposition 5.3.6 that on linear orders, the formulas $\varphi\mathsf{U}\psi$, $\mu x.(\psi \vee (\varphi \wedge \Diamond x))$ and $\mu x.(\psi \vee (\varphi \wedge \Diamond_\sigma x))$ are equivalent. We also noted in Proposition 5.3.4 that in this context, the formulas $\Diamond_\sigma\varphi$ and $\bigvee_{g\in B_\sigma}(\beta_g \wedge \beta_g \mathsf{U}(\neg\beta_g \wedge \varphi))$ are equivalent. The Proposition follows. □

Definition 5.4.2. $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ consists of the Modus Ponens, the Substitution rule, for each Γ, the corresponding Necessitation rule (i.e., if $\vdash \varphi$, then $\vdash \Box_\Gamma\varphi$) and the following axioms and rules:

$A1'$ propositional tautologies,

$A2'$ $\vdash \Box_\Gamma\varphi \leftrightarrow \neg\Diamond_\Gamma\neg\varphi$ (dual),

$A3'$ $\vdash \Diamond_\Gamma\varphi \rightarrow \Box_\Gamma\varphi$ (linearity),

$A4'$ $\vdash \Box_\Gamma(\varphi \rightarrow \psi) \rightarrow (\Box_\Gamma\varphi \rightarrow \Box_\Gamma\psi)$ (K),

$A5'$ $\vdash \varphi[x/\mu x.\varphi] \rightarrow \mu x.\varphi$ (fixpoint axiom),

FR$'$ If $\vdash \varphi[x/\psi] \rightarrow \psi$, then $\vdash \mu x.\varphi \rightarrow \psi$(fixpoint rule),

$A6'$ $\vdash \Diamond_\Gamma\varphi \leftrightarrow \bigvee_{g\in B_\Gamma}(\beta_g \wedge \mu x.((\neg\beta_g \wedge \varphi) \vee (\beta_g \wedge \Diamond_\sigma x)))$, where $\mathrm{Voc}(\Diamond_\Gamma\varphi) \subseteq \sigma$ (inductive meaning of $\Diamond_\Gamma$),

for each finite set $\Gamma = \{\varphi_0, \ldots, \varphi_{n-1}\}$ of $\mu\mathsf{TL}(\Diamond_\Gamma)$-sentences and where in the three last Axioms, x does not belong to $\mathrm{BV}(\varphi)$ and $\mathrm{FV}(\psi) \cap \mathrm{BV}(\varphi) = \emptyset$.

Lemma 5.4.3. *Axiom* $A6'$ *is sound on the class of finite and* ω*-words.*

Proof. Let σ be a finite vocabulary, Γ a finite set of $\mu\mathsf{TL}(\Diamond_\Gamma)$-formulas and φ a $\mu\mathsf{TL}(\Diamond_\Gamma)$-formula with $\mathrm{Voc}(\Diamond_\Gamma\varphi) \subseteq \sigma$ and $x \notin \mathrm{FV}(\varphi)$. As $\mu\mathsf{TL}(\Diamond_\Gamma)$ define only stutter-invariant classes of structures, we can consider a stutter-free σ-model $\mathfrak{M}$ with $w \in \mathfrak{M}$ and it is enough to show that the following are equivalent:

1. $\mathfrak{M}, w \models \Diamond_\Gamma\varphi$
2. $\mathfrak{M}, w \models \bigvee_{g\in B_\Gamma}(\beta_g \wedge \mu x.((\neg\beta_g \wedge \varphi) \vee (\beta_g \wedge \Diamond_\sigma x)))$

As for Proposition 5.4.1, this follows from what was observed in Section 5.2 and 5.3.

□

Theorem 5.4.4. $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ *is complete for* $\mu\mathsf{TL}(\Diamond_\Gamma)$ *with respect to the class of* ω*-words and with respect to the class of finite and* ω*-words.*

Proof. Let φ be a $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$-consistent formula in vocabulary σ. By Axiom $A6'$, we can restrict our attention to σ-formulas containing only $\Diamond_\sigma$ modalities. Again by Axiom $A6'$, we can define a recursive procedure transforming φ into a $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$-equivalent formula φ'. We set $At' = At$, $(\neg\varphi)' = \neg\varphi'$, $(\varphi\wedge\psi)' = \varphi'\wedge\psi'$, $(\mu x.\varphi)' = \mu x.\varphi'$, and $(\Diamond_\sigma\varphi)' = \bigvee_{g\in B_\sigma}(\beta_g \wedge \mu y.((\neg\beta_g \wedge \varphi') \vee (\beta_g \wedge \Diamond_\sigma y)))$. Consider now the $\mu\mathsf{TL}$-formula φ'', which we define as the result of removing in φ' all the subscripts of the modalities. Notice that by Proposition 5.4.1, φ' and φ'' are equivalent. We claim that φ'' is $K_\mu + \Diamond\varphi \rightarrow \Box\varphi$-consistent. For suppose not. Then, there exists a proof of $\neg\varphi''$ using the axioms and rules of $K_\mu + \Diamond\varphi \rightarrow \Box\varphi$. Now, replace every occurrence of the operator $\Diamond$ by $\Diamond_\sigma$ in each axiom and rule used in the proof. The result is a correct $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$-proof, where only correct axioms and rules of $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ are used (because the $K_\mu + \Diamond\varphi \rightarrow \Box\varphi$ axioms and rules can be obtained from the

$K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ ones simply by removing the indexes of the modalities). Additionally, this is a proof of the formula $\neg\varphi'$ (as the original φ' can also be obtained from φ'' by adding the subscript σ to every $\Diamond$ in φ''). But this contradicts the fact that φ' was $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$-consistent. So φ'' is $K_\mu + \Diamond\varphi \to \Box\varphi$-consistent. By Theorem 5.2.4, there is an ω-word or a finite word $\mathfrak{M}$ such that $\mathfrak{M}, w \models \varphi''$ and it follows from Proposition 5.4.1 (by which φ' and φ'' are equivalent) that $\mathfrak{M}, w \models \varphi'$, i.e. (by Axiom $A6'$), $\mathfrak{M}, w \models \varphi$. Completeness with respect to the class of ω-words follows too, because every finite word has an ω-word stuttering. □

Theorem 5.4.5. *$K_{\mu\mathsf{TL}(\Diamond_\Gamma)} + \mu x.\Box_\Gamma x$ is complete for $\mu\mathsf{TL}(\Diamond_\Gamma)$ with respect to the class of finite words.*

Proof. We can apply the same reasoning as for the proof of Theorem 5.4.5, using completeness of $K_\mu + \Diamond\varphi \to \Box\varphi + \mu x.\Box x$ on finite words, instead of completeness of $K_\mu + \Diamond\varphi \to \Box\varphi$ on finite and ω-words. □

Let $\mathfrak{M}$ be an ω-word. We say that $\mathfrak{M}$ is a *pseudo-finite* word whenever there exists a finite word $\mathfrak{M}'$ such that $\mathfrak{M}$ is a stuttering of $\mathfrak{M}'$. Note that $K_{\mu\mathsf{TL}(\Diamond_\Gamma)} + \mu x.\Box_\Gamma x$ is also complete for $\mu\mathsf{TL}(\Diamond_\Gamma)$ with respect to the class of finite and pseudo-finite words, as every pseudo-finite word is the stuttering of a finite word.

Remark 5.4.6. Axiom $A6'$ *is not derivable* from the other axioms and rules. Otherwise, every $\Box_\Gamma$ would simply be interpreted as the regular $\Box$ operator of μTL. Now, more precisely, let $K^{-A6'}_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ be the smallest set of $\mu(\Diamond_\Gamma)$-formulas which is closed under all axioms and rules in $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$, except Axiom $A6'$. Suppose Axiom $A6'$ is derivable in $K^{-A6'}_{\mu\mathsf{TL}(\Diamond_\Gamma)}$. Then, $K^{-A6'}_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ would be complete with respect to the class of ω-words. Therefore, as on ω-words $\models (p \wedge \Diamond_{\{p\}}\top) \to \Diamond_p\neg p$, also in $K^{-A6'}_{\mu\mathsf{TL}(\Diamond_\Gamma)}$, $\vdash (p \wedge \Diamond_{\{p\}}\top) \to \Diamond_p\neg p$ and there would exists a $K^{-A6'}_{\mu\mathsf{TL}(\Diamond_\Gamma)}$-proof of this formula. But now we could replace in that proof, every modal operator by the regular $\Diamond$ operator. This would be a correct $K_\mu + \Diamond\varphi \to \Box\varphi$-proof of $(p \wedge \Diamond\top) \to \Diamond\neg p$. But as on $\omega - words$, $\not\models (p \wedge \Diamond\top) \to \Diamond\neg p$, this contradicts the soundness of $K_{\mu\mathsf{TL}} + \Diamond\varphi \to \Box\varphi$. It follows that Axiom $A6'$ is not derivable in $K^{-A6'}_{\mu\mathsf{TL}(\Diamond_\Gamma)}$. ⊣

5.5 Complete Axiomatization of μTL(U)

Recall that LTL(U) is the fragment of μTL(U) where the μ-operator is disallowed. In [110], the authors propose an axiomatization of LTL(U) which is complete on the class of ω-words and finite words. In order to axiomatize μTL(U), we extend here the Axioms and rules in [110] with the usual fixed-point rule and Axiom, together with an additional axiom accounting for the way the Until operator and the μ-operator can interact together. Using the completeness result in [110] with the completeness of $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$, this allows us to derive a similar completeness Theorem for $\mu\mathsf{TL}(\mathsf{U})$. Recall that, in $\mu\mathsf{TL}(\mathsf{U})$, we use $\mathsf{G}\varphi$ as shorthand for $\neg(\top\mathsf{U}\neg\varphi)$ and $\Diamond_\tau\varphi$ as shorthand for $\bigvee_{g\in B_\tau}\beta_g \wedge (\beta_g\mathsf{U}(\neg\beta_g \wedge \varphi))$.

Definition 5.5.1. The $K_{\mu\mathsf{TL}(\mathsf{U})}$ system consists of the Modus Ponens, the G Necessitation rule (i.e., if $\vdash \varphi$, then $\vdash \mathsf{G}\varphi$) the Substitution rule and the following axioms and rules (these rules, as well as Axioms $A1''$ to $A9''$, are borrowed from [110]):

$A1''$ propositional tautologies,

$A2''$ The Until operator is non strict:
$\vdash \varphi \rightarrow \bot\mathsf{U}\varphi$,

$A3''$ For any consistent formula there exists a model that is a discrete linear order:

- $\vdash \mathsf{F}\varphi \rightarrow \neg\varphi\mathsf{U}\varphi$,
- $\vdash \varphi \wedge \mathsf{F}\psi \rightarrow \neg\psi\mathsf{U}(\varphi \wedge \varphi\mathsf{U}(\neg\varphi\mathsf{U}\psi))$,

$A4''$ Properties that hold throughout a computation hold at the initial state:
$\vdash G\varphi \rightarrow \varphi$,

$A5''$ Conventional logical deduction holds within individual states (K axiom):

- $\vdash (\mathsf{G}(\varphi \rightarrow \psi) \rightarrow (\varphi\mathsf{U}\xi \rightarrow \psi\mathsf{U}\xi))$
- $\vdash (\mathsf{G}(\varphi \rightarrow \psi) \rightarrow (\xi\mathsf{U}\varphi \rightarrow \xi\mathsf{U}\psi))$

$A6''$ Persistence of an Until formula until its second argument is satisfied:
$\vdash \varphi\mathsf{U}\psi \rightarrow (\varphi\mathsf{U}\psi)\mathsf{U}\psi$

$A7''$ Immediacy of satisfaction of an Until formula at the current state:
$\vdash \varphi\mathsf{U}(\varphi\mathsf{U}\psi) \rightarrow \varphi\mathsf{U}\psi$

$A8''$ States of the time line are not skipped over in evaluating an Until formula:
$\vdash \varphi\mathsf{U}\psi \wedge \neg(\xi\mathsf{U}\psi) \rightarrow \varphi\mathsf{U}(\varphi \wedge \neg\xi)$

$A9''$ Models are linearly ordered:
$\vdash \varphi U\psi \wedge \xi U\theta \rightarrow ((\varphi \wedge \xi)\mathsf{U}(\psi \wedge \theta) \vee (\varphi \wedge \xi)\mathsf{U}(\psi \wedge \xi) \vee (\varphi \wedge \xi)\mathsf{U}(\varphi \wedge \theta))$

$A10''$ $\vdash \varphi[x/\mu x.\varphi] \rightarrow \mu x.\varphi$, (fixpoint axiom),

FR$''$ If $\vdash \varphi[x/\psi] \rightarrow \psi$, then $\vdash \mu x.\varphi \rightarrow \psi$ (fixpoint rule),

$A11''$ $\vdash \mu x.(\psi \vee (\varphi \wedge \Diamond_\sigma x)) \leftrightarrow \varphi\mathsf{U}\psi$, where $\mathrm{Voc}(\varphi) \cup \mathrm{Voc}(\psi) \subseteq \sigma$ (inductive meaning of U), where in the three last Axioms, x does not belong to $\mathrm{BV}(\varphi) \cup \mathrm{BV}(\psi)$ and $\mathrm{FV}(\psi) \cap \mathrm{BV}(\varphi) = \emptyset$.

Lemma 5.5.2. *Let* $\varphi \in \mu\mathsf{TL}(\mathsf{U})$. *Then* $\phi \leftrightarrow (\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)})_{\mu\mathsf{TL}(\mathsf{U})}$ *is derivable in* $K_{\mu\mathsf{TL}(\mathsf{U})}$.

Proof. By induction on the complexity of φ (number of Boolean, modal and fixed-point operators in φ). The base case is immediate. Assume the property holds for all formulas of complexity n. Let $\varphi \in \mu\mathsf{TL}(\mathsf{U})$ of complexity $n+1$ be of the form $\xi \mathsf{U} \psi$ for some $\xi, \psi \in \mu\mathsf{TL}(\mathsf{U})$ (otherwise, by induction hypothesis, the property follows immediately). We have:

$$(\xi \mathsf{U} \psi)_{\mu\mathsf{TL}(\Diamond_\Gamma)} := \mu x.(\psi_{\mu\mathsf{TL}(\Diamond_\Gamma)} \vee (\xi_{\mu\mathsf{TL}(\Diamond_\Gamma)} \wedge \Diamond_\sigma x))$$

and

$$((\xi \mathsf{U} \psi)_{\mu\mathsf{TL}(\Diamond_\Gamma)})_{\mu\mathsf{TL}(\mathsf{U})} := \mu x.((\psi_{\mu\mathsf{TL}(\Diamond_\Gamma)})_{\mu\mathsf{TL}(\mathsf{U})} \vee ((\xi_{\mu\mathsf{TL}(\Diamond_\Gamma)})_{\mu\mathsf{TL}(\mathsf{U})} \wedge \bigvee_{g \in B_\sigma} (\beta_g \wedge \beta_g \mathsf{U}(\neg \beta_g \wedge x))$$

By induction hypothesis, $((\xi \mathsf{U} \psi)_{\mu\mathsf{TL}(\Diamond_\Gamma)})_{\mu\mathsf{TL}(\mathsf{U})}$ is provably equivalent in $K_{\mu\mathsf{TL}(\mathsf{U})}$ to:

$$\mu x.(\psi \vee ((\xi \wedge \bigvee_{g \in B_\sigma} (\beta_g \wedge \beta_g \mathsf{U}(\neg \beta_g \wedge x))$$

By Axiom $A11''$ the following is derivable in $K_{\mu\mathsf{TL}(\mathsf{U})}$:

$$\mu x.(\psi \vee ((\xi \wedge \bigvee_{g \in B_\sigma} (\beta_g \wedge \beta_g \mathsf{U}(\neg \beta_g \wedge x)) \leftrightarrow \xi \mathsf{U} \psi$$

The property follows. □

Lemma 5.5.3. *The* $\mu\mathsf{TL}(\mathsf{U})$*-translations of the axioms and rules of* $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ *are derivable in* $K_{\mu\mathsf{TL}(\mathsf{U})}$.

Proof. Except for the $\mu\mathsf{TL}(\mathsf{U})$-translation of the fixed-point Axiom and of the fixed-point rule (which are both trivially derivable from $K_{\mu\mathsf{TL}(\mathsf{U})}$, as they also belong to it), as well as Axiom $A6'$, there is no explicit occurrence of the μ-operator in the $\mu\mathsf{TL}(\mathsf{U})$-translation of the Axioms and rules of $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$. As they are sound on the class of ω-words and finite words, by the completeness Theorem in [110], together with Proposition 5.3.4, they are derivable in $\mathsf{LTL}(\mathsf{U})$. It follows that they are also derivable in $K_{\mu\mathsf{TL}(\mathsf{U})}$, because the Axioms and rules of $K_{\mu\mathsf{TL}(\mathsf{U})}$ simply extend those of $\mathsf{LTL}(\mathsf{U})$.

Now consider the $\mu\mathsf{TL}(\mathsf{U})$-translation of Axiom $A6'$:

$$\bigvee_{g \in B_\Gamma} (\beta_g \wedge \beta_g \mathsf{U}(\neg \beta_g \wedge \varphi))$$

$$\leftrightarrow$$

$$\bigvee_{g\in B_\Gamma} (\beta_g \wedge \mu y.((\neg\beta_g \wedge \varphi) \vee (\beta_g \wedge \bigvee_{g'\in B_\sigma} (\beta_{g'} \wedge \beta_{g'}\mathsf{U}(\neg\beta_{g'} \wedge y)))))$$

This formula is derivable from propositional tautologies, together with the substitution rule and Axiom $A11''$ of $K_{\mu\mathsf{TL(U)}} \vdash \mu y.(\psi \vee (\varphi \wedge \Diamond_\sigma x)) \leftrightarrow \varphi\mathsf{U}\psi$ (which is actually shorthand for $\vdash \mu y.(\psi \vee (\varphi \wedge \bigvee_{g\in B_\sigma}(\beta_g \wedge \beta_g\mathsf{U}(\neg\beta_g \wedge y)))) \leftrightarrow \varphi\mathsf{U}\psi$). Finally, let us point out that the restriction of our axioms and rules to LTL(U)-formulas is actually slightly stronger than the axiomatization proposed in [110]. The authors chose to prefix all their modal axioms and rules by G and to allow the generalization rule only on propositional tautologies (our generalization rule is a derived rule in their framework). But our axioms and rule being sound, it is safe to use the completeness of their system as we do here. □

Proposition 5.5.4. *Let $\varphi \in \mu$TL(U) be $K_{\mu\mathsf{TL(U)}}$-consistent, then its μTL($\Diamond_\Gamma$)-translation $\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ is $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$-consistent.*

Proof. Let $\varphi \in \mu$TL(U) be $K_{\mu\mathsf{TL(U)}}$-consistent. Now suppose $\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ is not $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$-consistent. So there is a $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$-proof of $\neg\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$. By Lemma 5.5.2 and 5.5.3, this entails that there is a $K_{\mu\mathsf{TL(U)}}$-proof of $\neg\varphi$, which contradicts the $K_{\mu\mathsf{TL(U)}}$-consistency of φ. □

Corollary 5.5.5. *$K_{\mu\mathsf{TL(U)}}$ is complete for μTL(U) with respect to the class of ω-words.*

Proof. Let φ be a $K_{\mu\mathsf{TL(U)}}$-consistent formula. Now let φ' be the μTL($\Diamond_\Gamma$)-translation of φ. By Proposition 5.5.4, φ' is $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$-consistent and so, by Theorem 5.4.4, φ' is satisfied in some ω-word $\mathfrak{M}, w$. By Proposition 5.3.6, φ and φ' are equivalent on ω-words. Hence also $\mathfrak{M}, w \models \varphi$. □

Proposition 5.5.6. *Let $\varphi \in \mu$TL(U) be $K_{\mu\mathsf{TL(U)}}$ + $\mu y.\Box_\Gamma y$-consistent, then its μTL($\Diamond_\Gamma$)-translation $\varphi_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ is $K_{\mu\mathsf{TL}(\Diamond_\Gamma)}$ + $\mu y.\Box_\Gamma y$-consistent.*

Proof. The proof is similar to the proof of Proposition 5.5.4. □

Corollary 5.5.7. *$K_{\mu\mathsf{TL(U)}}$ + $\mu y.\Box_\Gamma y$ is complete for μTL(U) with respect to the class of finite words.*

Proof. Similarly follows from Proposition 5.3.6, Theorem 5.4.5 and Proposition 5.5.6. □

Remark 5.5.8. Let $K^{-A11''}_{\mu\mathsf{TL(U)}}$ be the smallest set of μTL(U)-formulas which is closed under all axioms and rules in $K_{\mu\mathsf{TL(U)}}$ except Axiom $A11''$. Axiom $A11''$ *is not derivable* in $K^{-A11''}_{\mu\mathsf{TL(U)}}$. Observe that the μTL-translation of every axiom and rule of $K^{-A11''}_{\mu\mathsf{TL(U)}}6$ is sound when instantiated by μTL-formulas and that, by completeness of μTL, their μTL-translations are also derivable in μTL. So if

Axiom $A11''$ was derivable in $K^{-A11''}_{\mu\mathsf{TL}(\mathsf{U})}$, its $\mu\mathsf{TL}$-translation would be also derivable (and hence, valid) in $K_\mu + \Diamond\varphi \to \Box\varphi$. But let $\mathfrak{M}$ be a finite word in vocabulary $\{p\}$ with $W = \{w_0, w_1, w_2\}$, $w_i < w_{i+1}$ and $V(p) = w_2$. Obviously $\mathfrak{M}, w_0 \models \mu x.(p \vee (\Diamond\Diamond p \wedge \Diamond_{\{p\}} x))$, but $\mathfrak{M}, w_0 \not\models (\Diamond\Diamond p)\mathsf{U}p$, i.e., $\mathfrak{M}, w_0 \not\models \mu x.(p \vee (\Diamond\Diamond p \wedge \Diamond_{\{p\}} x)) \leftrightarrow (\Diamond\Diamond p)\mathsf{U}p$. ⊣

5.6 Conclusion

In this chapter, we studied the logic $\mu\mathsf{TL}(\mathsf{U})$. We introduced for that purpose the logic $\mu\mathsf{TL}(\Diamond_\Gamma)$ as a technical tool in order to easily obtain completeness results for $\mu\mathsf{TL}(\mathsf{U})$. In Chapter 4, we used a similar trick to show that $\mu\mathsf{TL}(\mathsf{U})$ satisfies uniform interpolation. A number of other interesting logical properties of $\mu\mathsf{TL}(\mathsf{U})$ remain to be investigated. In particular, we could examine counterparts of the Łoś Tarski Theorem and of the Lyndon Theorem, which the μ-calculus was shown in [46] to satisfy. More generally, the logic $\mu\mathsf{TL}(\Diamond_\Gamma)$ could also be used as a tool in order to easily transfer results from $\mu\mathsf{TL}$ to languages capturing exactly its stutter-invariant fragment (see for instance the frameworks in [61], [117], or [50]).

The method that we used here in order to show completeness results could also be reused in other contexts. It may for instance be applicable to the extension of $\mu\mathsf{TL}(\mathsf{U})$ with past tense operators or to the stutter-invariant fragment of the μ-calculus on trees (either finite or infinite). For a discussion of stuttering on trees, see [33] and [81], or [72], [73] and [80] in the setting of process algebra. It should be noted, though, that on (especially infinite) trees, there is still no general consensus on the appropriate notion of stuttering and that it is questionable whether the "Until only" fragment and the stutter-invariant fragment of the μ-calculus actually coincide. A further generalization would be to consider finite game trees (as studied in the next chapter), which actually carry a bit more structure than plain finite trees. In the context of game equivalence, the notion of stuttering could indeed constitute an interesting alternative to the notion of bisimulation (for a discussion see [13]).

Chapter 6

Fixed-point Logics on Finite Extensive Games

6.1 Game Solution as Rational Procedure

In this chapter, we will focus on finite games, which can be represented in so-called *extensive form* as special enriched tree structures. Logic and games form a natural combination. On the one hand, there are "logic games" that analyze basic notions such as truth, proof, or model comparison, while on the other hand, standard logical systems have proved applicable to many basic issues in the foundations of game theory (cf. [14], [114]). This chapter will concentrate on the second aspect.

Logics that describe games In recent years, many logical analyses have been given of both strategic and extensive games, through introducing formal languages that describe game structure while raising logical questions of definability and axiomatization ([29], [13], [51], [87]). A benchmark for logics in this tradition has been the definition of *Backward Induction* ("BI" for short), the most common method for solving finite extensive games of perfect information ([122], [123]). In this same arena, basic foundational results have been obtained in epistemic game theory, endowing bare games with epistemic assumptions about players. A pilot result was the characterization of the BI outcome in terms of assuming common knowledge, or common true belief, in rationality, meaning that players choose those actions that they believe to be best for themselves ([6]).

Analyzing solution procedures Recently, [18] has suggested that the main focus here should be shifted: away from a static assumption of known or believed rationality to the underlying "procedural rationality" of *plausible procedures* that players engage in when analyzing and playing a game, and the way these result in stable limit models where rationality becomes common knowledge.[1] Thus, [13] shows how game-theoretic equilibrium fits with the computational perspective of fixed-point logics, and [20] gives several dynamic procedures that analyze BI. This chapter will analyze these proposals further, and find their common mathematical background. This will then be our starting point for suggesting a more general line of investigation.

Basics of extensive games We assume some basic game theory, and we will work with *finite extensive games* of perfect information, i.e., finite trees with labelled nodes, where each node is either an end node, or an intermediate node that represents the turn of a unique player.[2] We will mostly think of 2-player games, though much of what we say generalizes to more players. While game trees with moves are simple computational structures, the essence of rational action arises with the way players evaluate outcomes. Thus, there is also a further *preference relation* for each player between end nodes (encoding complete histories) that we will take to be a total order in this chapter, though this requirement could be generalized. Equivalently, such total evaluation orders may be represented in the form of numerical utility values for players at end nodes.

Backward induction We now define our basic procedure in a bit more detail:

Definition 6.1.1 (BI procedure for "generic" extensive games)**.** We call a game *generic* when, for each player, distinct end nodes have different utility values. On such games *Backward Induction* is this inductive algorithm:

> "At end nodes, players already have their values marked. At further nodes, once all daughters are marked, the player to move gets her maximal value that occurs on a daughter, while the other, non-active player gets his value on that maximal node."

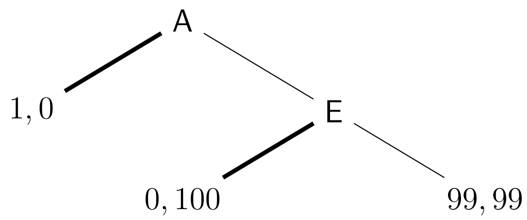
A *strategy* for a player is a map that selects one move at each turn for that player. It is easy to see that BI generates a strategy for each player at her turns: go to the successor node that has your highest value. The resulting set of strategies is the "BI outcome", that leads to a unique play of the game. We will call the set-theoretic union of all these strategies (still a function on nodes) bi. The BI procedure seems obvious, telling us players best course of action. And yet, it is packed with assumptions about how players behave that are worth highlighting. For now, just note that the algorithm subtly changes its interpretation of values on the way. At leaves, these values encode plain utilities or preferences, but at nodes higher up in the game tree, the BI values clearly mix in additional considerations of plausibility, incorporating beliefs about what others will do.

[1] Note that even the common word "solution" has an ambiguity between a procedure ("Solution is not easy") and a static product of such a procedure ("Show me your solution").

[2] Only towards the end, we will briefly consider games with imperfect information.

Delicate cases BI can produce debatable outcomes, as in the next illustration:

Example 6.1.2 (A simple BI outcome)**.** In the following game, players' preferences are encoded in utility values, as pairs (value for A, value for E). Backward Induction tells player E to turn left at her turn, which gives A a belief that this will happen, and so, based on this belief about his counter-player, A should turn left at the start, making both worse off than they might have been:[3]



[4]

The fact that the BI prediction or recommendation is not always intuitive has motivated much logical analysis of the procedure and the reasoning underpinning it. We will not enter this debate here. We neither endorse nor reject Backward Induction, but we merely take it as our point of entry into the logic of game solution procedures. Our starting points are three different proposals for explaining what makes BI tick, that we will explain in due course. But before getting there, let us first make a generalization of what we mean by Backward Induction.

6.2 From Functional to Relational Strategies

Strategies as subrelations of the *move* relation A game-theoretic strategy is usually taken to be a function on nodes in a game tree, yielding a unique recommendation for play there. But in many settings, it makes more sense to think of strategies as nondeterministic *binary subrelations* of the total relation *move* (the union of all labelled actions in the game) that merely constrain further moves by selecting one or more as admissible. This is in line with the colloquial use of the term "strategy", it also reflects a common view of plans for action, and technically, it facilitates logical definitions of strategies in propositional dynamic logic [19].

[3]People defend this outcome by saying that the game is "competitive", but that amounts to giving information about the players that is not explicit in the game tree. If such extra information is relevant to solution, we may need a richer notion of game from the start.

[4]Frankly, we have dramatized things a bit here to catch the reader's attention. Since the numbers just encode ordinal preferences, the same point might have been made with values 0, 1, 2 and 3. But the undesirable point remains that the computed outcome is not Pareto-optimal. An outcome of a game is Pareto optimal if there is no other outcome that makes every player at least as well off and at least one player strictly better off.

Relational BI, first version Indeed, one common numerical formulation of BI already has this relational flavor. We now (as for the remainder of the chapter) drop the assumption that games are generic:

Definition 6.2.1 (Relational Backward Induction, first version)**.** Starting from the leaves, one now assigns values for players at nodes using the rule:

> Suppose that E is to move at a node, and all values for daughters are known. The E-value is the maximum of all the E-values on the daughters, while the A-value is the minimum of the A-values at all E-best daughters.[5]

The relation bi arising from this algorithm connects nodes to all daughters with maximal values for the active player, of which there may be more than one. This method focuses on minimal values that can be guaranteed when doing the best within one's power.

Solution algorithms make assumptions about players But while this looks like an obvious numerical rule, it does embody special assumptions about players. In particular, taking the minimum value is a worst-case assumption that my counter-player does not care about my interests after her own are satisfied. But we might also assume that she does, choosing among her maximum nodes one that is best for me. In that case, the second numerical value in the algorithm would be a maximum rather than a minimum. And other options are possible.[6] This variety of relational versions of game solution is not a problem. It rather highlights an important feature of game theory: mathematical "solution methods" are not neutral, they encode significant assumptions about players. But the variety does suggest that we start by finding a general base version of BI that is not too specific:

A minimal notion of rationality: avoid stupid moves Here is one logical analysis of the variety for relational versions of BI. Let us first view matters from a somewhat higher standpoint. Suppose that I need to compare different moves of mine, each of which, given the relational nature of the procedure, still allows for many leaves (end nodes) that can be reached via further bi-play.[7] A minimal notion of *Rationality* would then say that

I do not play a move when I have another move whose outcomes I prefer.

[5]The dual calculation for values at A's turns is completely analogous.

[6]Of course, one might view such alternatives as calling for a change in players' utilities. We will not get into this perennial issue of game preference transformations here.

[7]In this perspective with total outcomes of the game, we make a shift from the original version of the BI algorithm, which looked at daughters of the current node only.

A source of variety: different set preferences This seems plausible, but what notion of preference is involved here? It is easy to see that, in the above first version of the BI algorithm, the following choice is made. Player i preferred a set Y of leaves reachable by further bi-play to another set X if the *minimum* of its values for i is higher. This means that we have the following $\forall\exists$ pattern for set preference:[8]

$$\forall y \in Y \; \exists x \in X : x <_i y$$

But clearly, staying with the same over-all notion of Rationality, there are several alternatives for comparisons between reachable sets of outcomes. One common notion of preference for Y over X in the logical literature ([135], [105]) is the $\forall\forall$ stipulation that

$$\forall y \in Y \; \forall x \in X : x <_i y$$

Relational backward induction, second version Clearly, avoiding moves that should not be taken under this stronger notion of preference is a weaker constraint on behavior of players. Still, it fits with a minimal game-theoretic solution procedure for strategic games called eliminating strictly dominated strategies ([112]). We will take this second relational version of Backward Induction as our running example:

Definition 6.2.2 (Relational Backward Induction, second version)**.** First, mark all moves as "active". Call a move to a node x *dominated* if x has a sibling from which all reachable endpoints via active moves are preferred by the current player to all reachable endpoints via active moves from x itself. The second version of the BI algorithm works in stages:

> At each stage, it marks dominated moves in the $\forall\forall$ sense of set preference as "passive", leaving all others active. In this preference comparison between sets of outcomes, the "reachable endpoints" by an active move are all those that can be reached via a sequence of moves that are still active at this stage.

In another well-known terminology, players play a "best response".

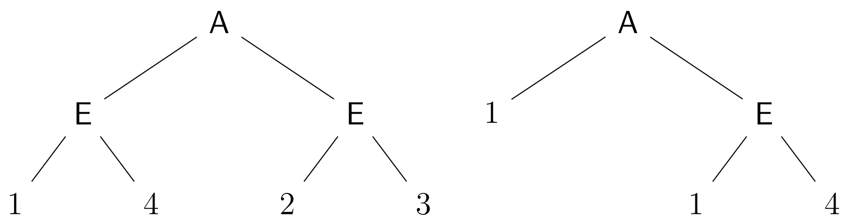
Henceforth, we will use BI to refer to this algorithm, and the subrelation of the total *move* relation produced by it at the end. It is a cautious notion of game solution making fewer assumptions about the behavior of other agents than the earlier version. Of course, the two versions agree on generic games, for which the subset of the move relation obtained as output is always a function.

[8]Given that we have finite total orders, we could also replace this by

$$\exists x \in X \; \forall y \in Y : x <_i y$$

.

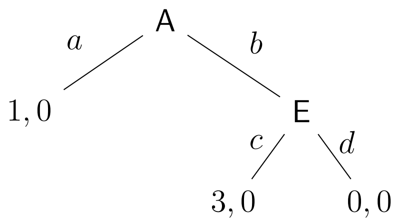
Example 6.2.3 (Some comparisons)**.** Consider the following two games, where the values indicated are utilities for player A. For simplicity, we assume that player E has no preference between her moves:



In the game to the left, our first version of Backward Induction makes A go right, since the minimum 2 is greater than the 1 on the left. But our cautious BI will accept both moves for A, as no move strictly dominates the other.

Moreover, both versions will accept all moves in the game to the right. This may seem strange, since most players would probably go right at the start: they have nothing to lose, and a lot to gain. But analyzing all variants for preference comparisons between sets of outcomes is not our focus here. We will return to the issue of further possible solution concepts in later sections.

Important remark Our style of analysis chooses one particular line toward generalizing Backward Induction to non-generic games. But others make sense, too, as pointed out by Cédric Dégremont. For instance, if one thinks of strategy profiles in Nash equilibrium, the following game would have two:



Both profiles (a, d) and (b, c) are in equilibrium. But our algorithm will leave both options for E, and tell A to go left. This chapter will not address the alternative analysis of the BI-output in terms of sets of strategy profiles, leaving this as a challenge to fixed-point logics over richer models.

6.3 Defining BI as a Unique Static Relation

Many definitions for the BI relation on generic games have been published by logicians and game-theorists (cf. the survey in [55]). Our point of departure here is a version involving a modal language of a-labelled moves, i.e., binary transition relations a on nodes with matching modalities $\langle a \rangle$, plus a modal preference operator interpreted as follows at nodes of a game tree:

$\langle pref_i \rangle \varphi$: player i prefers some node where φ holds to the current one

The original result Here is a result from [23]:

Theorem 6.3.1. *On generic games, the BI strategy is the unique function σ which is total on non-terminal nodes and satisfies the following modal axiom for all propositions p - viewed as sets of nodes - for all players i:*

$$(turn_i \wedge \langle \sigma^* \rangle (end \wedge p)) \rightarrow [move]\langle \sigma^* \rangle (end \wedge \langle pref_i \rangle p)$$

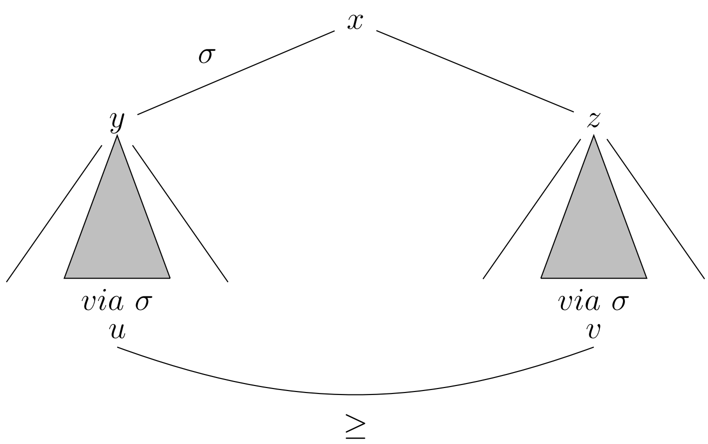
For a proof (a laborious but straightforward induction on finite tree depth), we refer to the cited paper. Here we just concentrate on the meaning of the crucial axiom, that may be brought out by a standard modal frame correspondence, where frame truth quantifies universally over all sets of objects for proposition letters ([26]). The frames here are games extended with one more binary relation σ. What we find is a notion of *Rationality* like before:

Fact 6.3.2. An extended game makes $(turn_i \wedge \langle \sigma^* \rangle (end \wedge p)) \rightarrow [move]\langle \sigma^* \rangle (end \wedge \langle pref_i \rangle p)$ true for all i at all nodes iff it has this property for all i:

> RAT-1: No other available move for the current player i yields a set of outcomes by further play using σ that has a higher minimal value for i than the outcomes of playing σ all the way down the tree from the current node.

Proof. This is a standard modal correspondence argument that we omit. The correspondence language uses the reflexive-transitive closure of the relation σ, but this is a simple extension of known techniques ([16]). □

The typical picture to keep in mind here, and also later on, is this:



RAT-1 is equivalent to this *confluence property* for action and preference:

$$\mathsf{CF1}: \bigwedge_i \forall x(turn_i(x) \to \forall y(\sigma(x,y) \to (move(x,y) \wedge \forall u((end(u) \wedge \sigma^*(y,u)) \to \forall z(move(x,z) \to \exists v(end(v) \wedge \sigma^*(z,v) \wedge v \leq_i u))))))$$[9]

This $\forall\forall\forall\exists$ form is a comparison between sets of outcomes that negates an earlier notion of preference: the *minimum value* on the reachable endpoints after z is not larger than that after y. It is easy to show that any relation σ which assigns a successor to each non terminal node and which satisfies this property matches the BI solution level by level on generic games.

Capturing BI in logical terms But now let us look at our favored relational generalization of BI. First, we reformulate the stated non-dominance property:

> RAT-2. No alternative move for the current player i guarantees outcomes via further play using σ that are all strictly better for i than all outcomes resulting from starting at the current move and then playing σ all the way down the tree.

A logical formula defining this has the following $\forall\forall\exists\exists$ form:

$$\mathsf{CF2}: \bigwedge_i \forall x \forall y((turn_i(x) \wedge \sigma(x,y)) \to (move(x,y) \wedge \forall z(move(x,z) \to \exists u \exists v(end(u) \wedge end(v) \wedge \sigma^*(y,v) \wedge \sigma^*(z,u) \wedge u \leq_i v))))$$

Theorem 6.3.3. *BI is the largest subrelation of the* move *relation in a finite game tree satisfying the two properties that* (*a*) *the relation has a successor at each non-terminal node, and* (*b*) *CF2 holds.*[10]

Proof. First, the given algorithm clearly leaves at least one active move at each node, by the definition of preference. Moreover, at the final state, when no more deactivations occur, CF2 must hold: there are no more dominated moves, and that is what it says.

That the relation defined in this way is maximal may be seen as follows. If we reactivate anywhere a move that is inactive, that move had disappeared at some stage because it was dominated there by another move. But then it would still be dominated in the whole tree by the same move. For, all that can have happened in the further stages of the algorithm is that fewer endpoints have become reachable through active paths from the two moves, and their $\forall\forall$-dominance relationship then persists.

Conversely, if we have any subrelation of the *move* relation with the given two properties, it is easy to see by induction on the depth of subtrees that all its moves survive each stage of the above main BI procedure, by the definition of the elimination step. □

[9] One could change the formal language for CF1 here to a more technical first-order one avoiding the closure operator – but for our main points, such variations are not important.

[10] We say "largest" in this formulation because in the presence of more than one best successor, different subrelations of the *move* relation might satisfy CF2. Note that there need not be a largest relation satisfying a given structural property, but in this particular case, it does.

We now make these same points about the procedure more syntactically, by inspecting the syntax of CF2. We can restate this in terms of the well-known formalism of first-order least fixed-point logic FO(LFP):[11]

Theorem 6.3.4. *The BI relation is definable in FO(LFP).*

Proof. Indeed, the definition involves just one *greatest fixed-point* in addition to the transitive closure operations. This fixed-point is in the language of FO(LFP), all occurrences of the predicate symbol X in the relevant formula are positive:

$$\mathsf{BI}(x,y) = [GFP_{X,x,y}(move(x,y) \wedge \bigwedge_i (turn_i(x) \rightarrow \forall z(move(x,z) \\ \rightarrow \exists u \exists v(end(u) \wedge end(v) \wedge X^*(y,v) \wedge X^*(z,u) \wedge u \leq_i v))))](x,y)^{12}$$

(Note that we use $X^*(z,u)$ as shorthand for $[TC_{z,u}X(z,u)](z,u)$, which is expressible in FO(LFP^1), see Chapter 2.) □

This definition will be our point of reference in what follows. Interestingly, it is both a static description of the BI relation and also a definition of a procedure computing it. For, we can now use the standard defining sequence for a greatest fixed-point, starting from the total *move* relation, and see that its successive decreasing approximation stages X^k are exactly the "active move stages" of the above algorithm. We will refer to these stages X^k at several places in what follows. In our view, fixed-point logics are attractive since they analyze both the statics and dynamics of game solution.

In the following sections, we extend this theme by looking at two further logical ways of construing the Backward Induction procedure that have been proposed in recent years.

6.4 A Dynamic-Epistemic Scenario: Iterated Announcement of Rationality

Here is another procedural line on Backward Induction as a rational process. [18] proposed an analysis in the spirit of current *dynamic-epistemic logics* that describe acts of information flow, such as public announcements or observations ([56], [21]). The following analysis of BI takes it to be a process of prior off-line deliberation about a game by players whose minds proceed in harmony - though they need not communicate in reality. [13]

[11] In terms of [15], the syntax of CF2 has dual "PIA form", guaranteeing that the union of all relations satisfying CF2 exists, while a small extra argument gives the existence.

[12] We can also replace the reflexive transitive closures X^* by definitions in FO(LFP).

[13] Compare also the dynamic agreement procedures studied in [68].

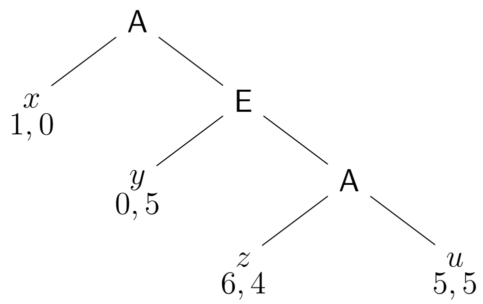
Solving games by announcements of rationality The following analysis uses the dynamic epistemic logic of public announcements of the form $!\varphi$, which say that some proposition φ is true. These transform a current epistemic model $\mathfrak{M}$ into its submodel $\mathfrak{M}|\varphi$ whose domain consists of just those worlds in $\mathfrak{M}$ that satisfy φ. [18] makes the solution process of extensive games itself the focus of a PAL style analysis:

Definition 6.4.1 (Node rationality)**.** As before, at a turn for player i, a move to a node x is *dominated* by a move to a sibling y of x if every history through x ends worse, in terms of i's preference, than every history through y. Now rat says that "at the current node, no player has chosen a strictly dominated move in the past coming here".

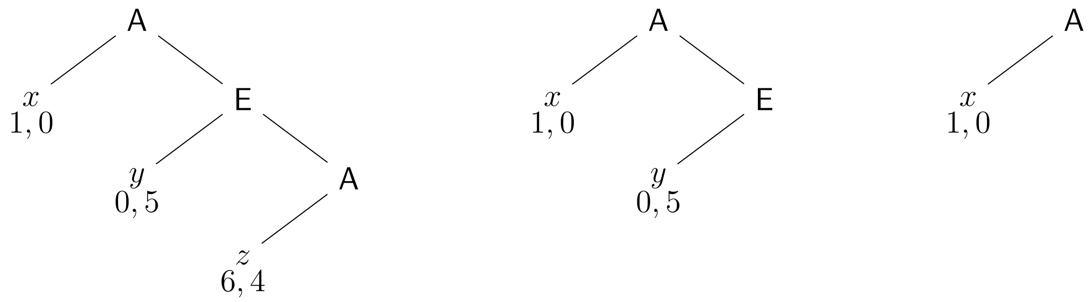
This makes an assertion about nodes in a game tree, viz. that they did not arise through playing a dominated move. Some nodes will satisfy this, others may not. Note that we do not say that every node in the game satisfies rat: we merely say that it is an informative property of nodes. Thus, announcing this formula as a fact about the players of a game is informative, and it will in general make the current game tree smaller.

But then we get a dynamics as in famous puzzles like the Muddy Children, where *repeated assertions* of ignorance eventually produce enough information to solve the whole puzzle. In our case, in the new smaller game tree, new nodes may become dominated, and hence announcing rat *again* (saying that it *still* holds after this round of deliberation) makes sense, and so on. This process of iterated announcement must always reach a limit, that is, a smallest subgame where no node is dominated any more:

Example 6.4.2 (Solving games through iterated assertions of Rationality)**.** Consider a game with three turns, four branches, and pay-offs for A, E in that order:



Stage 0 of the procedure rules out point u (the only point where Rationality fails), Stage 1 rules out z and the node above it (the new points where Rationality fails), and Stage 2 rules out y and the node above it. In the remaining game, Rationality holds throughout:



In such generic games, the BI solution emerges step by step. [18] shows that the actual Backward Induction path for extensive games is obtained by repeated announcement of the assertion rat to its limit. We repeat some relevant notions from dynamic-epistemic logic:

Definition 6.4.3 (Announcement limit)**.** For each epistemic model $\mathfrak{M}$ and each proposition φ that is true or false at points in the model, the announcement limit $(\varphi, \mathfrak{M})^{\#}$ is the first model reached by successive announcements $\varphi!$ that no longer changes after the last announcement is made.

That such a limit exists is clear for finite models, since the sequence of submodels is weakly decreasing.[14] There are two possibilities for the limit model. Either it is non-empty, in which case φ holds in all nodes, meaning that it has become common knowledge (the *self-fulfilling* case), or it is empty, meaning that the negation $\neg\varphi$ has become common knowledge (the self-refuting case). Both occur in concrete puzzles, though generally speaking, rationality assertions like rat tend to be self-fulfilling, while the ignorance statement that drives the Muddy Children is self-refuting: at the end, it holds nowhere.

Capturing BI by iterated announcements With general relational strategies, the iterated announcement scenario produces the earlier $\forall\forall\exists\exists$ version of Backward Induction:

Theorem 6.4.4. *In any game tree $\mathfrak{M}$, $(!rat, \mathfrak{M})^{\#}$ is the actual subtree computed by BI.*

This can be proved directly, but it also follows from our next observations. For a start, it turns out easier to change the definition of the driving assertion rat a bit. We now consider rat', wich only demands that the current node was not arrived at directly via a dominated move for one of the players. This does not eliminate nodes further down, and indeed, announcing this repeatedly will make the game tree fall apart into a forest of disjoint subtrees – as is easily seen in the above examples. These record more information.

[14] Announcement limits also exist in infinite models, if one takes intersection at limit ordinals.

Sets of nodes as relations Here is an obvious fact about game trees. Each subrelation R of the total *move* relation has an obvious unique corresponding set of nodes $reach(R)$ consisting of the set-theoretic range of R plus the root of the tree (we add the latter for convenience). And vice versa, each set X of nodes induces a unique corresponding subrelation of the *move* relation $rel(X)$ consisting of all moves in the tree that end in X. Incidentally this suggests that Theorem 6.3.4 can be slightly refined:

Theorem 6.4.5. *The* BI *relation is definable in* $\mathsf{FO(LFP^1)}$.

Proof. We simply put

$$\mathsf{BI}(x,y) = Move(x,y) \wedge [GFP_{X,y}\exists x(move(x,y) \wedge \bigwedge_i (turn_i(x) \to \forall z(move(x,z)$$
$$\to \exists u \exists v([TC_{zu}Move(z,u) \wedge X(u)](z,u) \wedge [TC_{yv}Move(y,v) \wedge X(v)](y,v)$$
$$\wedge end(u) \wedge end(v) \wedge u \leq_i v)))))](y)$$

□

With this simple connection, we can link the earlier approximation stages BI^k for Backward Induction (i.e., the successive relations computed by our earlier procedure) and the stages of our public announcement procedure. They are in harmony all the way:

Fact 6.4.6. For each k, in each game model $\mathfrak{M}$, $BI^k = rel((!\mathsf{rat}')^k, \mathfrak{M})$.

Proof. By induction on k. The base case is obvious: $\mathfrak{M}$ is still the whole tree, and the relation BI^0 equals *move*. Next, consider the inductive step. If we announce rat' again, we remove all points reached by a move that is dominated for at least one player. These are precisely the moves cancelled by the corresponding step of the BI algorithm. □

It follows also that, for each stage k,

$$reach(\mathsf{BI}^k) = ((!\mathsf{rat}')^k, \mathfrak{M}).$$

Either way, we conclude that the earlier algorithmic fixed-point definition of the BI procedure and van Benthem's iterated announcement procedure amount to the same thing. [15]

Thus, one might say that the deliberation scenario is just a way of "conversationalizing" the underlying mathematical fixed-point computation. Still, it is of interest in the following sense. Viewing a game tree as an epistemic model with nodes as worlds, we see how repeated announcement of Rationality eventually makes this property true throughout the remaining limit model: in this way, it has made itself into *common knowledge*.

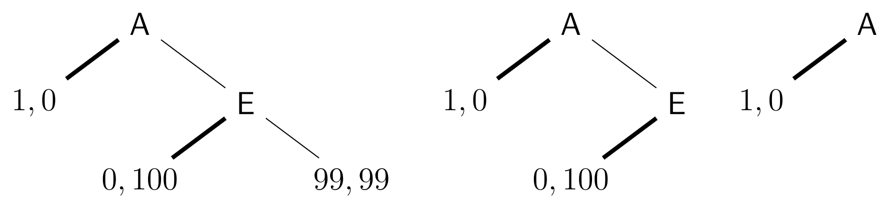
[15] We leave the technical question open to which extent this is a more general technical method for switching between different types of predicate arities with fixed-points.

6.5 Another Dynamic Scenario: Beliefs and Iterated Plausibility Upgrade

Next, in addition to knowledge, consider the equally fundamental notion of *belief*. Many foundational studies in game theory (cf. the extensive discussion and references for belief-based game theory in [137]) view Rationality as choosing a best action *given what one believes* about the current and future behavior of the players. Indeed, this may be the most widely adopted view of game solution in the epistemic foundations of game theory today. We will first state a logical analysis of game solution in these terms, and then relate it to our earlier account of Backward Induction.

Backward Induction in a soft light An appealing take on the BI strategy in terms of beliefs uses "soft update" that does not eliminate worlds as above for announcements $!\varphi$, but rearranges the plausibility order between worlds.[16] A typical example is the *radical upgrade* $\Uparrow \varphi$ that makes all current φ-worlds best, and then puts all $\neg\varphi$-worlds underneath, while keeping the old ordering inside these two zones. Now recall our earlier observation that Backward Induction really creates *expectations* for players. All the essential information produced by the algorithm is then in the binary *plausibility relations* that it creates inductively for players among non terminal nodes in the game, standing for complete histories. To see this, consider our running example once more:

Example 6.5.1 (The debatable BI outcome, hard and soft)**.** The hard scenario in terms of events !rat removes nodes x from the tree that are reached via moves which are strictly dominated by moves to siblings of x as long as this can be done, resulting in the following sequence of stages:



By contrast, a soft scenario does not remove nodes but modifies the plausibility relation. We start with all endpoints of the game tree incomparable (other versions would have them equiplausible). Next, at each stage, we compare sibling nodes, using this notion:

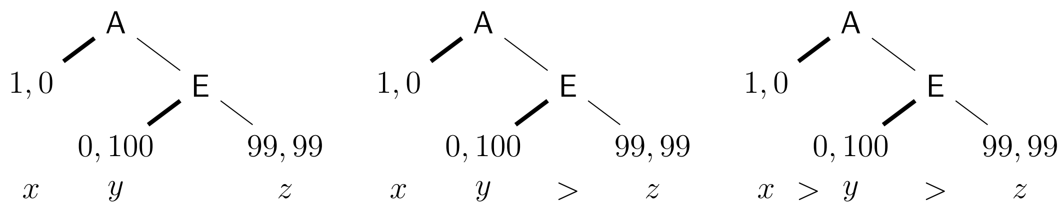
[16] This is a technique from current dynamic epistemic logics, where acts of knowledge update and belief revision are represented by transformations on domains of models, or their plausibility ordering of worlds. See [17], [9], [8] for some latest developments.

Definition 6.5.2 (Rationality in beliefs)**.** A move to a node x for player i *dominates* a move to a sibling y of x *in beliefs* if the most plausible end nodes reachable after x along any path in the whole game tree are all better for the active player than all the most plausible end nodes reachable in the game after y. Rationality* (rat^*) is the assertion that no player plays a move that is dominated in beliefs.

Now we perform a relation change that is like a radical upgrade $\Uparrow \mathsf{rat}^*$, except that plausibility upgrades may take place in subtrees, and hence one needs to work with submodels of the whole set of histories:

If a move to a node x dominates a move to a sibling y of x in beliefs, we make all end nodes reachable from x more plausible than those reachable from y, keeping the old order inside these zones.

This changes the plausibility order, and hence the dominance pattern, so that belief statements can change their truth values – and a genuine iteration can start. Here are the stages for this procedure in the above example, where we use the letters x, y, z to stand for the end nodes or histories of the game:



In the first game tree, going right is not yet dominated in beliefs for A by going left. rat^* only has bite at E's turn, and an upgrade takes place that makes $(0, 100)$ more plausible than $(99, 99)$. After this upgrade, however, going right has now become dominated in beliefs, and a new upgrade takes place, making A's going left most plausible.[17]

Here is a result stated without proof in [20]:

Theorem 6.5.3. *On finite trees, the Backward Induction strategy is encoded in the final plausibility order for end nodes created by iterated radical upgrade with rationality in belief.*

At the end of this procedure, players have acquired *common belief in rationality.* Let us now prove the result, using an idea from [10].

[17] Here the plausibility relation is defined over end nodes only. Another option would have been to define it over all nodes in the tree (see [13]).

Strategies as Special Plausibility Relations We present now in details a set-theoretical transformation which allows to go back and forth between strategies and plausibility relations in the case of finite extensive games. To do so, we characterize plausibility orders as a special sort of linear order over leaves, which satisfy a property that we call "tree compatibility".

Let a be a node in a finite tree. A *history containing* a is a path along the $Move$ relation which contains a, starts from the root and ends at a leaf. We denote by $RL(a)$ the set of leaves reachable via a history containing a. Note that every leaf in the tree determines a unique history (given a leaf, there is a unique path starting from the root and ending at this leaf).

Definition 6.5.4 (Relational strategy)**.** A relation $Best$ on a finite game tree $\mathfrak{M}$ is a *relational strategy* whenever:

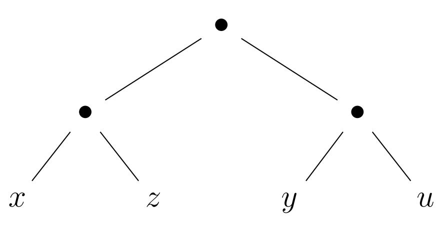
- $Best \subseteq Move$
- every non-terminal node is in the $Best$ relation with another node in the tree, i.e., $\mathfrak{M} \models \forall x(\neg end(x) \rightarrow \exists y Best(x, y))$

Some linear orders over leaves satisfy some special conditions which ground their "equivalence" (in a sense to be shown below) with relational strategies. Such linear orders satisfy a property that we call *tree-compatibility.*

Definition 6.5.5 (Ancestor-connected sets of leaves)**.** Let A, B be two sets of leaves in a finite tree. A and B are *ancestor-connected* if there exists a node x with two children z and y such that the set of z-reachable leaves is exactly A and the set of y-reachable leaves is exactly B.

Definition 6.5.6 (Tree-compatible linear order)**.** Let $\preceq$ be a linear order on the leaves of a finite tree. $\preceq$ is *tree-compatible* if for all ancestor-connected sets A, B in the tree, either every leaf in A is below every leaf in B relative to $\preceq$, or every leaf in B is below every leaf in A relative to $\preceq$.

As an example, there can be no criss-crossing as in the following tree:



with $x < y < z < u$

A very natural example of tree-compatible linear order in a finite tree is the left-right order over its leaves.

The following simple lemma will be useful in the proof of Theorem 6.5.8. It states that if one consider the set of least common ancestors of every two pairs of leaves formed out of three leaves, it cannot contain more than two distinct nodes.

Lemma 6.5.7. *Let u, v, w be three leaves in a finite tree, a the least common ancestor of u and v and a' the least common ancestor of v and w. Then, consider a'', the least common ancestor of u and w. Then either $a = a'$ or $a' = a''$.*

Proof. Consider the unique histories U, V determined respectively by u and v. The two paths start together at the root and there is a unique point where they diverge. This point is a, the least common ancestor of u and v. Then, consider the unique history W determined by w. Similarly, W crosses V on the unique point where they start to diverge, which is a'. Now there are two cases, either U and W start to diverge after U and V do, or they start before. In the first case $a' = a$ and in the second case, $a' = a''$. □

We first show that each relational strategy BI induces a tree-compatible linear order $\preceq_{Best}$ on leaves as follows: $x \preceq_{Best} y$ iff whenever y is reached via a *Best* move from the least common-ancestor of x and y, then so is x.

Theorem 6.5.8 (From strategies to plausibility orders)**.** *Let Best be a relational strategy and let $\preceq_{Best}$ be the smallest binary relation on the leaves such that for every $(a,b) \in Move$, $(a,c) \in Move$, $u \in RL(b)$ and $v \in RL(c)$:*

- *$(a,c) \notin Best$ implies $u \preceq_{Best} v$*
- *$(a,c) \in Best$ implies $v \preceq_{Best} u$*

The defined relation $\preceq_{Best}$ is a tree-compatible linear order and we say that it corresponds to Best.

Proof. Consider a relational strategy *Best* on a finite tree together with the corresponding relation $\preceq_{Best}$.

First assume that $\preceq_{Best}$ is not total. Then there are two leaves u and v which are not connected in $\preceq_{Best}$. Let a be their least common ancestor, b its immediate successor on the path to u and c its immediate successor on the path to v. Consider (a,b) and (a,c). There are four possible patterns as regards their inclusion in *Best*. In each case, by the definition of $\preceq_{Best}$, u and v stand together in the $\preceq_{Best}$ relation.

Now assume $\preceq_{Best}$ is not transitive. Then there are three leaves u, v, w such that $u \preceq_{Best} v$, $v \preceq_{Best} w$ and $w \prec_{Best} u$. Let a be the least common ancestor of u and v, b its immediate successor on the path to u and c its immediate successor on the path to v. Let a' be the least common ancestor of v and w, b' its immediate successor on the path to v and c' its immediate successor on the path to w. Let a'' be the least common ancestor of u and w, b'' its immediate successor on the path to u and c'' its immediate successor on the path to w. By definition of $\preceq_{Best}$, the following holds:

- $(a,c) \notin Best$ or $(a,b) \in Best$

- $(a', b') \in Best$ or $(a', c') \notin Best$
- $(a'', c'') \in Best$ and $(a'', b'') \notin Best$

Now we want to consider the least common ancestor of u, v and w. By Lemma 6.5.7, there are three cases. It could either be a, a' or a''. Without loss of generality (the reasoning is similar in the two other cases), let us assume it is a''. Then, again by Lemma 6.5.7, there are two cases, either $a'' = a$ (and $b'' = b$, $c'' = c$), i.e., it is also the least common ancestor of u and v, either $a'' = a'$ (and $b'' = b'$, $c'' = c'$), i.e., it is also the least common ancestor of v and w. Let us assume the latter (again, the former case is similar). It follows that:

- $(a'', b'') \in Best$ or $(a'', c'') \notin Best$
- $(a', c') \in Best$ and $(a', b') \notin Best$

But this contradicts our assumptions.

Finally, assume $\preceq_{Best}$ is not tree-compatible. So we can assume without loss of generality that there are two ancestor-connected sets of leaves A and B, with $u_1, u_2 \in A$ and $v_1, v_2 \in B$ such that $u_1 \prec_{Best} v_1$ and $v_2 \preceq_{Best} u_2$. Let a be the least common ancestor of the leaves in A and B, b its immediate successor on the paths to the leaves in A and c its immediate successor on the paths to the leaves in B. As $u_1 \prec_{Best} v_1$, by construction of $\prec_{Best}$, $(a, b) \in Best$ and $(a, c) \notin Best$. Similarly, by construction of $\preceq_{Best}$, as $v_2 \preceq_{Best} u_2$, it follows that $(a, b) \notin Best$, which is a contradiction. □

Now, conversely, any tree-compatible linear order $\preceq$ on leaves induces a relational strategy $Best_{\preceq}$ defined by selecting just those available moves at a node z that have the following property: their further available histories lead only to $\preceq$-minimal leaves in the total set of leaves that are reachable from z.

Theorem 6.5.9 (From plausibility orders to strategies)**.** *Let $\preceq$ be a tree-compatible linear order on a finite tree and let $Best_{\preceq}$ be the smallest binary relation on the tree which satisfies the following: for every $(a, b) \in Move$, $(a, b) \in Best_{\preceq}$ whenever there is a leaf u reachable from b such that for every leaf v reachable from a, $u \preceq v$. Then $Best_{\preceq}$ is a relational strategy and we say that it corresponds to $\preceq$.*

Proof. Consider a tree-compatible linear order over leaves $\preceq$ on a finite tree together with the corresponding relational strategy $Best_{\preceq}$. It is immediate that $Best_{\preceq} \subseteq Move$. Now suppose $Best_{\preceq}$ is not a relational strategy, i.e., there is an non-terminal node a which does not have any $Best_{\preceq}$-successor. By linearity of $\preceq$, there is a leaf u reachable from a such that for every leaf v reachable from a, $u \preceq v$. Consider now the immediate successor b of a on the history generated by u. By construction of $Best_{\preceq}$, $(a, b) \in Best_{\preceq}$. □

Theorem 6.5.10. *Let $\leq$ be a tree-compatible linear order and $Best_{\leq}$ the corresponding relational strategy. Then $\leq = \preceq_{Best_{\leq}}$.*

Proof. Let $(u, v) \in \leq$. Now let a be the least common ancestor of u and v and b the immediate successor of a on the history determined by u. By construction of $Best_{\leq}$, $(a, b) \in Best_{\leq}$ and by construction of $\preceq_{Best_{\leq}}$, $(u, v) \in \preceq_{Best_{\leq}}$.

Let $(u, v) \in \preceq_{Best_{\leq}}$. Similarly, let a be the least common ancestor of u and v and b the immediate successor of a on the history determined by u. By construction of $Best_{\leq}$, $(a, b) \in Best_{\leq}$ and $(u, v) \in \leq$.

□

Theorem 6.5.11. *Let S be a relational strategy and $\preceq_S$ the corresponding tree-compatible linear order. Then $S = Best_{\preceq_S}$.*

Proof. Let $(a, b) \in S$. Then, by construction of $\preceq_S$, for every $(a, c) \in Move$, $u \in RL(b)$ and $v \in RL(c)$, $(a, c) \notin S$ implies $u \preceq_S v$ and $(a, c) \in S$ implies $v \preceq_S u$. By construction of $Best_{\preceq_S}$, it follows that $(a, b) \in Best_{\preceq_S}$.

Let $(a, b) \in Best_{\preceq_S}$. Then, by construction of $\preceq_S$, for every $(a, c) \in Move$, $u \in RL(b)$ and $v \in RL(c)$, $(a, c) \notin S$ implies $u \preceq_S v$ and $(a, c) \in S$ implies $v \preceq_S u$. By construction of $Best_{\preceq_S}$, it follows that $(a, b) \in S$.

□

Via Theorems 6.5.11 and 6.5.10, the following definition gives a precise meaning to the assertion in [10] that "strategies are the same as plausibility relations".

Definition 6.5.12. We say that a tree-compatible linear order $\preceq$ and a relational strategy $Best$ are *equivalent* whenever $\preceq = \preceq_{Best}$, or, equivalently, $Best_{\preceq} = Best$.

Now we can relate the computation in our upgrade scenario for belief and plausibility to the earlier relational algorithm for BI. Things are in harmony stage by stage:

Fact 6.5.13. For any game tree $\mathfrak{M}$ and any k, $((\Uparrow \mathsf{rat}^*)^k, \mathfrak{M}))_{Best} = \mathsf{BI}^k$.

Proof. The key point is as demonstrated in the earlier example of a stepwise BI solution procedure. When computing a next approximation for the BI-relation according to CF2, we drop those moves that are dominated by another available one. But this has the same effect as making the leaves reachable from dominated moves less plausible than those reachable from surviving moves. And that was precisely the earlier upgrade step. □

This structural equivalence also yields immediate matching syntactic transformations as follows.

Proposition 6.5.14. *Let Best be a relational strategy, the corresponding plausibility relation $\preceq_{Best}$ can be defined as follows:*

$$y \preceq_{Best} z$$

$$=$$

$$\exists x(LCA(x,y,z)\wedge(\exists w(Best(x,w)\wedge move^*(w,z)) \rightarrow \exists w'(Best(x,w')\wedge move^*(w',y))))$$

where $LCA(x,y,z)$ stands for "x is the least common ancestor of the leaves y and z":

$$LCA(x,y,z) = move^*(x,y) \wedge move^*(x,z) \wedge end(y) \wedge end(z)\wedge$$
$$\neg\exists w(w \neq x \wedge move^*(x,w) \wedge move^*(w,y) \wedge move^*(w,z))$$

Proof. This follows by analyzing the proof of Theorem 6.5.8. □

Proposition 6.5.15. *Let $\preceq$ be a tree-compatible linear order, the corresponding relational strategy $Best_{\preceq}$ can be defined as follows:*

$$Best_{\preceq}(x,y) = move(x,y) \wedge \exists z(move^*(y,z) \wedge end(z) \wedge \forall z'$$
$$((end(z') \wedge move^*(x,z')) \rightarrow z \preceq z'))$$

Proof. This follows by analyzing the proof of Theorem 6.5.9. □

Note that whenever a relational strategy $Best$, or a plausibility order $\preceq$ are definable in a logic L extending $\mathsf{FO}(\mathsf{TC}^1)$, then it follows from the syntactic translations given in Proposition 6.5.14 and 6.5.15 that $\preceq_{Best}$ and $Best_{\preceq}$ are respectively also definable in L.

Theorem 6.5.16. *The binary relation $\preceq_{\mathsf{BI}}$ is definable in $\mathsf{FO}(\mathsf{LFP}^1)$.*

Proof. As BI is definable in $\mathsf{FO}(\mathsf{LFP}^1)$, it follows from Proposition 6.5.14 that:

$$y \preceq_{\mathsf{BI}} z$$

$$=$$

$$\exists x(LCA(x,y,z) \wedge (\exists w(\mathsf{BI}(x,w) \wedge move^*(w,z)) \rightarrow \exists w'(\mathsf{BI}(x,w') \wedge move^*(w',y))))$$

□

Remark 6.5.17. Let us point out that Theorem 6.5.16 could also be shown by providing a more direct definition in $\mathsf{FO}(\mathsf{LFP})$ as follows. For the sake of readability, let us first introduce the following shorthand ($RL(u,x)$ stands for x is a leaf which is reachable from the node u):

$$RL(u,x) :\approx Move^*(u,x) \wedge end(x)$$

Now we put:

$$x \preceq_{\mathsf{BI}} y$$

$$=$$

$$[GFP_{x,y,X} \bigvee_i \exists z \exists u \exists v (Turn_i(z) \wedge Move(z,u) \wedge Move(z,v) \wedge RL(u,x) \wedge RL(v,y) \wedge$$

$$\exists x_1 \exists y_1 (RL(u,x_1) \wedge RL(v,y_1) \wedge$$

$$\forall x_2 (RL(u,x_2) \to X(x_1,x_2)) \wedge$$

$$\forall y_2 (RL(v,y_2) \to X(y_1,y_2)) \wedge$$

$$y_1 \leq_i x_1) \vee (end(x) \wedge x = y)](x,y)$$

I.e, $x \preceq_{\mathsf{BI}} y$ whenever x and y respectively belong to two ancestor-connected sets of leaves X and Y (which least common-ancestor is a turn for player i) such that the most plausible leaf in X is better for i than the most plausible leaf in Y. Recall that $x \preceq_{\mathsf{BI}} y$ iff whenever y is reached via a BI move from the least common ancestor of x and y, then so is x. It follows that the relation defined here is the intended $\preceq_{\mathsf{BI}}$ relation.

⊣

We conclude that the algorithmic analysis of Backward Induction and its procedural doxastic analysis in terms of forming beliefs amount to the same thing. Still, as with the iterated announcement scenario, the iterated upgrade scenario also has some interesting features of its own. One is that, for logicians, it yields fine-structure to the plausibility relations that are usually treated as primitives in models for doxastic logic. Thus games provide an underpinning for possible worlds semantics of belief that seems of interest per se.

6.6 Midway Conclusion: Dynamic Foundations

Extensional equivalence, intensional difference We have now seen how three different approaches to analyzing Backward Induction turn out to amount to the same thing. To us, this means that the notion is stable, and that, in particular, its fixed-point definition can serve as a normal form. This motivates taking a closer look at fixed-point logics for game solution. Of course, as we have observed, extensionally equivalent definitions can still have interesting intensional differences in terms of what they suggest. For instance, we see the above analysis of strategy creation and plausibility change as one more concrete case study for a general conceptual issue: the fact that agents' beliefs and rational action are deeply entangled in the conceptual foundations of decision and game theory.

Dynamic instead of static foundations for game theory As we also said already, one key feature of our dynamic announcement and upgrade scenarios is this. In the terms of [18], they are *self-fulfilling*: ending in non-empty largest submodels where players have *common knowledge or common belief of rationality.*[18] [19] Thus, this dynamic style of game analysis is a big change from the usual static characterizations of Backward Induction in the epistemic foundations of game theory. Common knowledge or belief of rationality is not assumed, but *produced* by the logic.

6.7 Test Case: Variants of Backward Induction

We analyzed our standard BI relation and found a number of things: it is definable in fixed-point logic, it can also be analyzed alternatively as a subset of nodes and as a plausibility relation in tightly correlated ways. But are the preceding results just special effects for the notion of Backward Induction chosen here? As a test case, we will now look how all these themes work for the variant BI' defined in Section 6.2, where preference between sets of outcomes referred to ensuring a greater minimal value:

Definition 6.7.1. BI' is the largest subrelation of the *move* relation in a finite game tree satisfying the two properties that (a) the relation has a successor at each non terminal node, and (b) CF1 holds.[20]

It turns out that the $\forall\forall\forall\exists$-type syntactic definition CF1 can no longer be used for an immediate fixed-point definition of BI' in FO(LFP). We would get

$$\begin{array}{c} move(x,y) \wedge \bigwedge_i (turn_i(x) \rightarrow \forall u((end(u) \wedge X^*(y,u)) \\ \rightarrow \forall z(move(x,z) \rightarrow \exists v(end(v) \wedge X^*(z,v) \wedge v \leq_i u)))) \end{array}$$

where not all occurrences of the relation symbol X are positive. But there are two alternative candidates for general fixed-point logics that can be used: FO(PFP) and FO(IFP) (see Chapter 2). We will show how to do this.

6.7.1 Defining BI' in Partial Fixed-Point Logic

For a start it will be easier to compute the fixed-point we are interested in using FO(PFP). In FO(PFP) we usually focus on finite models (an extension of the

[18] We forego the issue of logical languages for explicitly *defining* the limit submodel.

[19] We also forego the further analysis of the limit behavior of upgrade actions on game models. For general models, [9] finds some curious phenomena, such as plausibility cycles, and they prove a general result stating when at least absolute beliefs stabilize in the limit. There are interesting general issues of fixed-point definability for predicates in limit models of dynamic epistemic procedures, that link up with our analysis. We leave this for further work.

[20] The fact that there is such a relation will follow from theorem 6.7.2.

framework to infinite structures can be found in [97]) and we can consider fixed-points of arbitrary formulas that are reached by a similar sequence of iteration stages as in the case of FO(LFP). We saw in Chapter 2 that the only difference is that the resulting operator is not necessarily monotone and hence, whenever a fixed-point is reached, it does not necessarily correspond to the least fixed-point of this operator. Additionally, whenever no fixed-point is reached, we simply evaluate the corresponding formula as false.

Theorem 6.7.2. *The relational BI'-strategy is definable in FO(PFP).*

Proof. Now we can use CF1 and prefix it with a PFP-operator, or equivalently (as we already noticed that each subrelation of the move relation has a unique corresponding set of nodes), we can put:

$$\mathsf{BI}'(x,y)$$
$$=$$
$$move(x,y) \wedge [PFP_{X,y} \exists x(move(x,y) \wedge \bigwedge_i (turn_i(x) \to \forall z(move(x,z) \to$$
$$\forall u((end(u) \wedge [TC_{yu} move(y,u) \wedge X(u)](y,u) \to$$
$$\exists v(end(v) \wedge [TC_{zu} move(z,u) \wedge X(v)](z,v) \wedge v \leq_i u))))](y)$$

In order to see that this formula defines a unique non-empty relation, let us rewrite the subformula that is inside the fixed-point operator, using only the non-reflexive counterpart TC^+ of the transitive closure operator TC, so that the variable X appears only in the scope of a TC^+-operator:

$$\exists x(move(x,y) \wedge \bigwedge_i (turn_i(x) \to \forall z(move(x,z) \to (\varphi_2 \vee \varphi_4 \vee \varphi_1 \vee \varphi_3))))$$

with

- $\varphi_1 := end(y) \wedge end(z) \wedge z \leq_i y$
- $\varphi_2 := \forall u((end(u) \wedge [TC^+_{yu} move(y,u) \wedge X(u)](y,u)) \to (end(z) \wedge z \leq_i u))$
- $\varphi_3 := end(y) \wedge \exists v(end(v) \wedge [TC^+_{z,v} move(z,v) \wedge X(v)](z,v) \wedge v \leq_i y$
- $\varphi_4 := \forall u(((end(u) \wedge [TC^+_{y,u} \wedge X(u)](y,u)) \to \exists v(end(v) \wedge [TC^+_{z,v} move(z,v) \wedge X(v)](z,v) \wedge v \leq_i u))$

The essence of the argument is that at any stage k of the fixed-point iteration, the computed set stabilizes at points whose siblings are along the child relation at greatest distance $\leq k$ to a leaf. By "stabilizes at some set of points at stage k" we mean that for every point in that set, the point belongs to the fixed-point approximant at stage k if and only if it belongs to the fixed-point approximant at every stage greater than k. At the first stage of the fixed-point iteration, the formula φ_1 (which does not contain any occurrence of X) determines once and for all whether leaves that have only leaves siblings belong to the current and later fixed-point approximants. Then, at stage k the same is similarly determined for points whose siblings are along the child relation at greatest distance $\leq k$ to a leaf. This is ensured by the syntactic shape of the formulas φ_2, φ_3 and φ_4, in which the variable X appears only in some restricted "guarded form" inside the formula $[TC^+_{yu} move(y,u) \wedge X(u)](y,u)$, where u refers to a point which is strictly lower down y or inside the formula $[TC^+_{zu} move(z,u) \wedge X(u)](y,u)$, where u refers to a point which is strictly lower down a sibling z of y.

□

6.7.2 Defining BI' in Inflationary Fixed-Point Logic

In the BI' partial fixed-point computation of the previous section, nothing was preventing nodes which were ruled out at some stage of the induction process to reappear at a later stage, which could at first sight suggest that the process is not inflationary. But Theorem 6.7.2 can still be refined and BI′ can be defined in the computationally better-behaved logic FO(IFP) as an inflationary process.[21] [22] The trick there is to use a simultaneous fixed-point induction, explicitly using an additional inductive "stable" predicate in order to progressively define the stables elements which can be safely added to the fixed-point at each stage of the computation. Note that this is a general idea (see for instance the use of simultaneous modal fixed-point formulas in [16]). Let us also recall that allowing simultaneous fixed-points does not increase the expressive power of FO(IFP) (see Chapter 2).

Theorem 6.7.3. *The relational BI'-strategy is definable in FO(IFP^1).*

Proof. We use a simultaneous fixed-point formula $[IFP\ X : S](y)$, where:

$$S := \begin{cases} Xy \leftarrow \Phi(X,y) \wedge Y(x) \\ Yw \leftarrow \Psi(Y,w) \end{cases}$$

with:

$$\Phi(X,y) := \exists x(move(x,y) \wedge \bigwedge_i (turn_i(x) \to \forall z(move(x,z) \to$$

[21]This is interesting, since [18] already observed how the limits of iterated public announcement procedures on modal models are definable in FO(IFP), and in fact, usually in the modal inflationary calculus, the extension of the modal μ-calculus by means of inflationary fixed-points ([49]).

[22]From the preceding fact, we can conclude (using [85], [98]) that there is an equivalent definition for BI' in FO(LFP) after all, though the latter may involve extra predicates, with a computation no longer matching the natural stages of our algorithm.

$$\forall u((end(u) \wedge [TC_{yu} move(y,u) \wedge X(u)](y,u) \rightarrow$$
$$\exists v(end(v) \wedge [TC_{zu} move(z,u) \wedge X(v)](z,v) \wedge v \leq_i u))))$$

and

$$\Psi(Y,w) := \exists x(move(x,w) \wedge end(w) \wedge \forall y(move(x,y) \rightarrow end(y))$$
$$\vee$$
$$(\forall y(move^+(w,y) \rightarrow Y(y)) \wedge$$
$$\exists x(move(x,w) \wedge \forall z \forall z'(move(x,z) \wedge move^+(z,z') \rightarrow Y(z'))))$$

$\Phi(X,y)$ is exactly the formula of which we considered the partial fixed-point in the proof of Theorem 6.7.2, whereas we use $\Psi(Y,w)$ in order to ensure that at any stage k of the fixed-point iteration, the fixed-point approximant can only contain points whose siblings are along the child relation at greatest distance $\leq k$ to a leaf (we noticed in the proof of Theorem 6.7.2 that these points are "stable" at stage k). It follows that this formula is equivalent to the formula in the proof of Theorem 6.7.2. □

As in the case of BI, now that we have a definition of BI', we can apply to it the general syntactic translation in Proposition 6.5.14 and obtain a definition of the associated plausibility order $\preceq_{\mathsf{BI'}}$.

Theorem 6.7.4. *$\preceq_{\mathsf{BI'}}$ is definable in FO(IFP^1).*

Proof. As BI' is definable in FO(IFP^1), it follows from Proposition 6.5.14 that:

$$y \preceq_{\mathsf{BI'}} z$$
$$=$$
$$\exists x(LCA(x,y,z) \wedge (\exists w(\mathsf{BI'}(x,w) \wedge move^*(w,z)) \rightarrow \exists w'(\mathsf{BI'}(x,w') \wedge move^*(w',y))))$$

□

It is time to conclude our analysis. We have shown how BI', our pilot example of an alternative game solution procedure, can indeed be defined in fixed-point logics of trees. We were even able to do this in different formalisms. [23]

6.7.3 Alternative: Recursion on Well-Founded Tree Order

We have now given a definition of BI' in standard fixed-point logics over general models. But let us mention that an alternative take is also possible. One major feature of game solution procedures like Backward Induction is their exploiting the inductive structure of extensive games, via the *well-founded tree dominance order* toward the leaves.[24] Such orderings allow for recursive definitions that yield

[23] This raises the issue of which fixed-point language is most congenial to analyzing games, something to which we return in later sections.

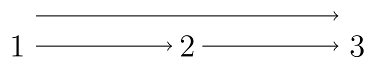
[24] But other recursions are possible, too. Both finite and infinite trees allow for recursive definitions over the well-founded tree order in the opposite *past direction* toward the root.

uniqueness even without positive occurrence:

Example 6.7.5 (Fixed-points in modal provability logic ([30]))**.** On finite trees, any modal formula of the form $p \leftrightarrow \varphi(p)$ where p occurs only "guarded" (that is, in the scope of at least one modality) in the formula φ, defines a unique proposition p. One proves this by induction on the well-founded tree order.[25]

This includes examples like the following:

Example 6.7.6 (Broader well-founded recursion)**.** Consider the definition $p \leftrightarrow \neg\Box p$. On a 3-nodes linear order



starting from any set as a value for p, this will stabilize with $p = \{2\}$. But it is easy to see that an inflationary bottom-up procedure for this formula stops in the pre-fixed point $\{1, 2\}$, and the deflationary top-down procedure stops in the post-fixed point $\emptyset$. Neither of these is even a fixed-point. What one can see more precisely in the straightforward approximation procedure, without forcing increasing or decreasing sets, is this: starting the iteration from any initial set will gradually get the predicate right, successively, at all nodes lying at increasing height from the leaves.

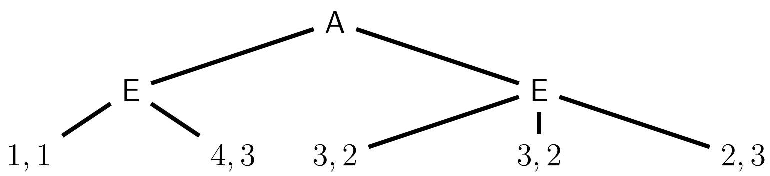
Of course, our analysis for Backward Induction did not use these simple modal languages, nor did it just use the simple tree dominance order. Still, by inspection of our earlier formulas and arguments the following result is easily seen to hold.

Fact 6.7.7. Stated as an equivalence, the Rationality principles CF1 and CF2 both define a unique subrelation of the *move* relation by recursion on a well-founded order on the nodes of finite trees: viz. the composition of the relations *sibling* and *dominance*.

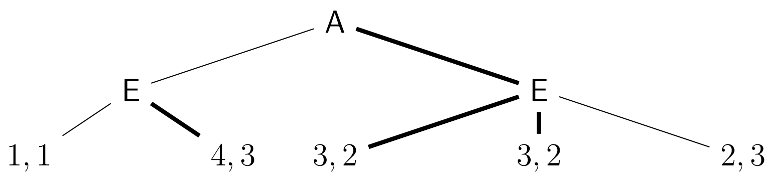
And this unique relation may also be computed for other versions which like CF1 lack positive syntax but do descend along the well-founded order. One can start with any subrelation of the *move* relation, and then compute according to the given instruction. At any stage k, the range of the fixed-point relation stabilizes at points whose siblings are along the child relation at a greatest distance $\leq k$ to a leaf.

Example 6.7.8 (Computing a fixed-point for CF1)**.** Consider this game, with values on leaves written as pairs (value for E, value for A):

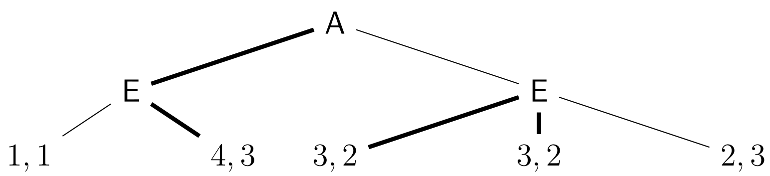
[25] The result was originally shown for transitive conversely well-founded frames, but the same argument applies to finite trees. Additionally, it is known as the de Jongh Sambien Theorem that such fixed-points are modally definable.



Let R^0 be the whole *move* relation. Then R^1 is marked in black below:



This still gets the fixed-point relation wrong at the root, but in the next stage we get the stable solution:



Remark 6.7.9 (Recursion over other well-founded orders than the sibling-dominance order)**.** Let us note that we could also have shown Theorem 6.7.4 by providing a more direct definition of $\preceq_{\mathsf{BI}'}$ in $\mathsf{FO(IFP)}$. We think this is interesting, because it gives another example of well-founded order over finite trees than the sibling-dominance order. As in Remark 6.5.17, we use the following shorthand:

$$RL(u,x) :\approx Move^*(u,x) \wedge end(x)$$

Now we put $x \preceq_{\mathsf{BI}'} y = [IFP\ X : S](x,y)$, where:

$$S := \begin{cases} X(x,y) \leftarrow \Phi(X,Y,x,y) \\ Y(w) \leftarrow \Psi(Y,w) \end{cases}$$

with:

$$\Phi(X,x,y) :\approx \bigvee_i \exists z \exists u \exists v (Turn_i(z) \wedge Y(z) \wedge Move(z,u) \wedge Move(z,v) \wedge$$

$$RL(u,x) \wedge RL(v,y) \wedge$$

$$\forall x_1((RL(u,x_1) \wedge \forall x_2(RL(u,x_2) \rightarrow X(x_1,x_2))) \rightarrow$$

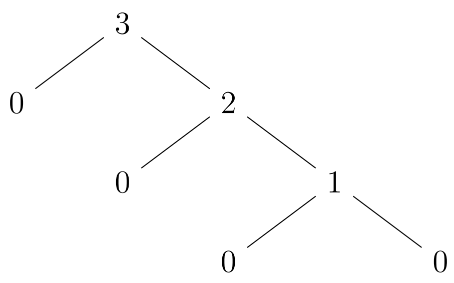
$$\exists y_1(RL(v,y_1) \wedge \forall y_2(RL(v,y_2) \rightarrow X(y_1,y_2)) \wedge y_1 \leq_i x_1)))$$

$$\vee$$
$$(end(x) \wedge x = y)$$

and

$$\Psi(Y, w) :\approx end(w) \vee \forall y(Move^{+}(w, y) \rightarrow Y(y))$$

$\Phi(X, x, y)$ corresponds to the BI’ variant of the formula in Proposition 6.5.16. Still, the different notion of preference encoded here entails that the first occurrence of the variable X is negative. Hence, as in the case of BI’, we need to rely on a trick in order to use an inflationary fixed-point computation. The simultaneous fixed-point formula given above ensures via the formula $\Psi(Y, w)$ that at any stage k of the fixed-point iteration, the computed relation stabilizes for couples of leaves which least common ancestor lies at maximal distance $\leq k$ to a leaf. To illustrate it via a simple example, this amounts to assigning numbers to nodes in the tree as follows:



Leaves are labeled 0 and nodes that are at a maximal distance n to a leaf are labeled n, so in particular the root is labelled with the length of the maximal path in the tree. On the tree pictured above for instance, the root is labelled 3. Hence it is only decided at stage 3 of the induction whether the first child of the root (which is a leaf) is more plausible or less plausible than all the other leaves in the tree. ⊣

In the next section, we will explore the idea of recursion exploiting the presence of well-founded tree orders further. We will introduce what we call “order-conform” operators, and show how these fit with fixed-point definitions in a natural way. This material forms a digression from the main game-oriented line of this chapter, which will be resumed in Section 6.9.

6.8 Excursion: Order-Conform Fixed-Point Logics

The analysis in Section 6.7.3 suggests the introduction of a logical formalism for games that can access the well-founded tree order directly.[26] But we also

[26]Relevant proposals in the literature include the more general “non monotone inductive definitions” of [54]. Such definitions need not have a fixed-point at all, even though some

believe that a more abstract analysis of conditions under which such recursions are successful could be interesting, not only from a game-theoretical point of view, but also from a more general logical perspective. In this section, we give a few preliminary definitions and easy results in this direction. We first introduce a general notion of "order-conform" operator. Such operators have the particularity to always yield a unique fixed-point. Then, we introduce and study some of the basic properties of the ι-calculus, a simple modal fixed-point logic on finite trees which extends the basic modal language with a unique fixed-point construct. The analyses in this section abstract from the later analysis of game solution concepts by first adopting a more general standpoint, where the object of interest are fixed-point operators defined not necessarily over finite game trees, but over some given class of finite structures. Then, as an example of a syntactically well-defined logic allowing only order-conform fixed-point constructs, we turn to the case of the ι-calculus, where we restrict attention to simple (i.e., without any preference order over the leaves) finite tree structures.

6.8.1 Order-Conform Operators

We will adopt a wider perspective than before, as we do not restrict attention to finite game trees and we are generally interested in recursion over some given well-founded order (we will mention the idea in connection with infinite games in Section 6.10). We will focus here for simplicity on finite structures. We will also restrict to monadic fixed-points. First of all, we need this very basic notion:

Definition 6.8.1 (Ordered partition of a finite set)**.** An ordered partition Π of a finite set X is an ordered sequence

$$\Pi(X) = X_1 < \ldots < X_k$$

of non-empty pairwise disjoint subsets of X, whose union is X.

But then, we are not only interested in isolated finite sets, but rather in *classes* of finite structures and in *ways* of ordering their domains. We formalize this using the following notion.

Definition 6.8.2 (Ordered partition mapping)**.** Let $\mathcal{C}$ be a class of finite structures. We call *ordered partition mapping on* $\mathcal{C}$ a mapping $\mathcal{P}$ which to every structure $\mathfrak{M} \in \mathcal{C}$ assigns an ordered partition $\mathcal{P}(\mathfrak{M}) = X_1 < \ldots < X_k$ of $dom(\mathfrak{M})$.

Now we can define abstractly what it means for an operator on the powerset of the domain of structures in $\mathcal{C}$ to "go along some order". Whenever we fix a structure and try to compute a fixed-point for such an operator, from whichever set we begin the computation with, the membership of points which are minimal

analysis of various conditions under which they correspond to a unique fixed-point is provided.

in the order will always be fixed in the same way at the first stage of the induction. Moreover, this will be done once and for all (i.e., these points will remain either inside or outside the set to be computed). Similarly, the membership of points which are at some level n in the order will be fixed once and for all at stage n of the induction.

Definition 6.8.3 (Order-conform operator)**.** Let $\mathcal{C}$ be a class of finite structures and F an operator on the powerset of the domain of the structures in $\mathcal{C}$ which for every $\mathfrak{M} \in \mathcal{C}$ assigns to each set $A \subseteq dom(\mathfrak{M})$ a set $F(A) \subseteq dom(\mathfrak{M})$.

Assume F is such that there is an ordered partition mapping $\mathcal{P}$ which to each $\mathfrak{M} \in \mathcal{C}$ assigns $\mathcal{P}(\mathfrak{M}) = X_1 < \ldots < X_k$ and for all $A, A' \subseteq dom(\mathfrak{M})$, for all n

$$F^n(A) \cap \bigcup_{i \leq n} X_i = F^n(A') \cap \bigcup_{i \leq n} X_i$$

then we call F an *order-conform operator* based on $\mathcal{P}$.

We now have as immediate consequence of our definition.

Theorem 6.8.4. *Order-conform operators always have a unique fixed-point.*

Proof. Let F_φ be an order-conform operator based on $\mathcal{P}$ on some class $\mathcal{C}$. Now let $\mathfrak{M} \in \mathcal{C}$ and consider $\mathcal{P}(\mathfrak{M}) = X_1 < \ldots < X_n$. It is immediate that $F^n_\varphi(\emptyset)$ is a fixed-point of F_φ. Now let Y and Y' be two fixed points of F_φ, i.e., $F_\varphi(Y) = Y$ and $F_\varphi(Y') = Y'$. As F_φ is an order-conform operator, by definition 6.8.3, $F^n_\varphi(Y) \cap \bigcup_{i \leq n} X_i = F^n_\varphi(Y') \cap \bigcup_{i \leq n} X_i$, i.e., $Y \cap dom(\mathfrak{M}) = Y' \cap dom(\mathfrak{M})$, i.e., $Y = Y'$. □

Let us now come back to operators that are yield by fixed-point logic formulas. Sometimes, an FO(IFP)-formula always has a unique fixed-point on some given class of structures, but we already noted that this fixed-point is generally not definable as its inflationary fixed-point. Still, we can notice that whenever the formula yields an order-conform operator, we can simply define this unique fixed-point in FO(PFP) (as FO(IFP) formulas are translatable in FO(PFP)):

Proposition 6.8.5. *Let $\Phi(X, x)$ be an FO(IFP)-formula and F_Φ an order-conform operator on some finite class of structures $\mathcal{C}$, then the unique fixed-point of Φ on $\mathcal{C}$ is definable in FO(PFP).*

But jumping to FO(PFP) is not very satisfying and we would like to remain whithin the expressive power of FO(IFP). Let us then restrict to cases where the ordered partition mapping is definable in FO(IFP).

Definition 6.8.6 (FO(IFP)-definable partition mapping)**.** Let σ be a relational vocabulary, $\mathcal{C}$ a class of finite σ-structures and $\mathcal{P}$ an ordered partition mapping on $\mathcal{C}$ such that there exists a total order $\leq_{\mathcal{P}}$ on the domains of the structures in $\mathcal{C}$ (with associated strict order $<_{\mathcal{P}}$) definable by an FO(IFP)-formula in vocabulary σ and such that for every $\mathfrak{M} \in \mathcal{C}$ with $\mathcal{P}(\mathfrak{M}) = X_1 < \ldots < X_k$, for every valuation g on $\mathfrak{M}$:

- $X_1 = \{a|\mathfrak{M}, g[a/y] \models \neg\exists x\ x <_{\mathcal{P}} y\}$ is the set of minimal elements in $<_{\mathcal{P}}$
- $\bigcup_{j\leq i+1} X_j = \{a|\mathfrak{M}, g[a/y] \models \forall x(x <_{\mathcal{P}} y \rightarrow x \in \bigcup_{j\leq i} X_j)\}$

then we say that $\mathcal{P}$ is an *FO(IFP)-definable ordered partition mapping.*

We do not address here the question of whether there are FO(IFP)-formulas yielding order-conform operators (on some given class of structures) that are based on non FO(IFP)-definable ordered partition mappings and we leave it open for further investigations. We only make the following easy observation.

Theorem 6.8.7. *Let $\Phi(X, x)$ be an FO(IFP)-formula in vocabulary σ such that F_Φ is an FO(IFP) order-conform operator on some class of finite σ-structures $\mathcal{C}$. Assume the ordered partition mapping on which F_Φ is based is FO(IFP)-definable. Then there exists an FO(IFP)-formula $\Psi(x)$ such that*

$$[PFP_{X,x}\Phi(X, x)^{\textsf{FO(PFP)}}](x) \leftrightarrow \Psi(x)$$

(where $\Phi(X, x)^{\textsf{FO(PFP)}}$ is the translation of Φ in FO(PFP))

Proof. The idea is similar to the one in the proof of Theorem 6.7.3 and $\Psi(x)$ can be written as the following simultaneous fixed-point formula:

$$[IFP\ X : S](x)$$

with

$$S := \begin{cases} Xx \leftarrow \Phi(X, x) \wedge Y(x) \\ Yy \leftarrow (\neg\exists x\ x <_{\mathcal{P}} y) \vee \forall x(x <_{\mathcal{P}} y \rightarrow Y(x)) \end{cases}$$

In the system S, we use the formula $(\neg\exists x\ x <_{\mathcal{P}} y) \vee \forall x(x <_{\mathcal{P}} y \rightarrow Y(x))$ in order to ensure that at any stage n of the fixed-point iteration, the fixed-point approximant of $[IFP\ X : S](x)$ can only contain points which are in $\bigcup_{i\leq n} X_i$ where X_i is the i^{th} set in the ordered sequence $\mathcal{P}(\mathfrak{M}) = X_1 < \ldots < X_k$. □

Now Theorem 6.8.7 gives a clean recipe to characterize non-positive variants of BI in FO(IFP): it is enough to mimic the simultaneous fixed-point system in the proof of Theorem 6.7.3. Only the way we define the set X will vary, but we will keep the same formula to define the set Y, as the recursion will always occurs along what we earlier identified as the "sibling dominance" order in the tree. Note that other game-theoretic notions might also be definable in FO(IFP) using the same idea, provided that the computation from which they arise be similarly based on some FO(IFP)-definable order on the nodes of the tree.

One further question is whether we could find natural syntactic characterizations of FO(IFP)-formulas yielding order-conform operators, for instance in the case of finite trees. The proof of Theorem 6.7.2 suggests a few patterns, but for now we will leave the question open. Rather, we will temporarily forget about the player's preferences labeling the leaves of finite game trees and we will turn to the case of modal fixed-point logics on plain finite trees. We view this excursion as a first step towards an understanding of the underlying action structure of prospective more elaborate fixed-point logics of finite game trees.

6.8.2 The Modal ι-Calculus

To show how order-conform formalisms make sense, we will briefly develop one particular modal fixed-point logic over finite Kripke models based on finite tree frames. While this system is clearly not rich enough to define the game solution algorithms we had earlier, it serves as a nice pilot example of what might be achieved with simple logical languages that exploit well-founded tree orders. We call this logic the ι-calculus (L_ι), by reference to Russell's definite description operator ιx, which is to be read as "the unique x such that". L_ι is a simple modal logic on finite trees which extends the basic modal language with a fixed-point operator which can be applied to formulas that are not necessarily positive but which satisfy a syntactic condition of "guardedness" which ensures that they yield order-conform operators. We show in particular that this logic has exactly the same expressive power as the modal μ-calculus on finite trees. We will restrict here to the class of finite trees Kripke-structures, i.e., finite Kripke structures based on a frame which belongs to the class of finite tree frames $\mathbf{T}$. The question of the ι-calculus was first raised to us by Johan van Benthem.

Definition 6.8.8 (L_ι)**.** Let σ be a propositional signature, and let $\mathcal{V} = \{x_1, x_2, \ldots\}$ be a disjoint countably infinite stock of *propositional variables.* The set of L_ι-formulas in vocabulary σ is generated by the following inductive definition:

$$\varphi, \psi, \xi := At \mid \top \mid \neg\varphi \mid \varphi \vee \psi \mid \varphi \wedge \psi \mid \varphi \rightarrow \psi \mid \Diamond\varphi \mid \iota x_i.\xi$$

where $At \in \sigma \cup \mathcal{V}$ and in the last clause, x_i occurs only guarded in ξ (i.e., within the scope of a $\Diamond$-operator). The satisfaction relation is inductively defined between L_ι-formulas and pointed structures $(\mathcal{T}, V, w)$ where $\mathcal{T} \in \mathbf{T}$ as in the case of the μ-calculus using an auxiliary assignment to interpret formulas with free variables. The only difference concerns the ι-operator and we interpret $\iota x.\varphi$ as the unique fixed-point of the operator F_φ.

The semantics is consistent, because of the following:

Lemma 6.8.9. *Let $\varphi(x)$ be a L_ι-formula in which x is guarded, then $\varphi(x)$ has a unique fixed-point.*

Proof. The proof is by induction on the number of nested fixed-point operators in $\varphi(x)$. We will show that F_φ is an order-conform operator on finite trees based on the following ordered partition mapping of the set of nodes:

$$x \in X_{i+1} \text{ iff } x \text{ lies at maximal distance } i \text{ to a leaf.}$$

The lemma will follow by Theorem 6.8.4. For the base case, let us assume $\varphi(x)$ is purely modal (i.e., does not contain any ι-operator). Let $\mathfrak{M}$ be a finite tree Kripke structure, we will show by induction on n that for all $A, A' \subseteq dom(\mathfrak{M})$, for all n:

$$F_\varphi^n(A) \cap \bigcup_{i \leq n} X_i = F_\varphi^n(A') \cap \bigcup_{i \leq n} X_i$$

The base case is $n = 1$, where X_1 is the set of all leaves in the tree (i.e. nodes which lie at maximal distance 0 to a leaf). As the variable x in φ is guarded, it occurs only in the scope of at least one $\Diamond$-operator. Let $a \in X_1$, the fact that $a \in F_\varphi(A)$ or $a \in F_\varphi(A')$ depends only on the fact that points that are lower down a in the tree belong to A or A' respectively, but points in X_1 being only leaves, there are no such points. Hence $a \in F_\varphi(A)$ if and only $a \in F_\varphi(A')$ and:

$$F_\varphi(A) \cap X_1 = F_\varphi(A') \cap X_1$$

Now assume the property holds for n, then the property follows for $n+1$ by a similar argument as in the base case. Let $a \in \bigcup_{i \leq n+1} X_i$, as the variable x in φ is guarded, the fact that $a \in F_\varphi^{n+1}(A)$ or $a \in F_\varphi^{n+1}(A')$ depends only on the fact that points that are lower down a in the tree belong to $F_\varphi^n(A)$ or $F_\varphi^n(A')$ respectively. By induction hypothesis, for every such point b, $b \in F_\varphi^n(A) \cap \bigcup_{i \leq n} X_i$ if and only if $b \in F_\varphi^n(A') \cap \bigcup_{i \leq n} X_i$. Hence $a \in F_\varphi^{n+1}(A)$ if and only if $a \in F_\varphi^{n+1}(A')$ and:

$$F_\varphi^{n+1}(A) \cap \bigcup_{i \leq n+1} X_i = F_\varphi^{n+1}(A') \cap \bigcup_{i \leq n+1} X_i$$

The inductive case for more complex formulas $\varphi(x)$ containing nested ι-operators is similar. □

Let us point out here that the argument used in the proof of Lemma 6.8.9 is similar to the one used in order to show the fixed-point lemma of provability logic (see [30]), recalled in Example 6.7.5. Note that provability logic interprets basic modal formulas on Kripke structures based on "transitive conversely well-founded frames", whereas we considered here finite trees. But we can simulate formulas of provability logic on finite trees in L_ι by considering the transitive closure of the $\Box$-operator of L_ι. Hence the fact recalled in Example 6.7.5 also follows from Lemma 6.8.9.

Now let us recall a result which immediately implies that the ι-calculus extends the μ-calculus on finite trees. We first need the general notion of guardedness for a modal fixed-point formula (by modal fixed-point formula we mean a formula in a logic extending basic modal logic with a fixed-point operator):

Definition 6.8.10. A modal fixed-point formula is guarded, if all propositional variables that are bound by and thus occur in the scope of a fixed-point operator are also in the scope of a modality that is itself in the scope of the fixed-point operator.

Proposition 6.8.11 ([99])**.** *Every formula of the μ-calculus is equivalent to a guarded formula.*

We will now show that L_ι and L_μ have exactly the same expressive power on finite trees. Moreover, we show that for every L_ι-formula there is an effective procedure which computes a L_μ-formula which is equivalent on finite trees, and vice versa.

Theorem 6.8.12. *The ι-calculus and the μ-calculus on finite trees are effectively equi-expressive.*

Proof. The fact that the ι-calculus extends the μ-calculus on finite trees follows from Proposition 6.8.11 because the ι-operator being a unique fixed-point operator, it can equivalently be replaced by a μ-operator whenever it is prefixed to a positive formula (whenever a formula has a unique fixed-point, then this fixed-point is also its least fixed-point). Hence, for every guarded formula φ of the μ-calculus there exists a formula of the ι-calculus which is equivalent to φ on finite trees (simply replace every μ-operator in φ by a ι-operator). But using the Janin Walukiewicz Theorem ([91], see Chapter 2), one can even refine this inclusion. It is immediate that on finite trees the ι-calculus is contained in the partial iteration calculus MPC (see Chapter 2), which is bisimulation-invariant (see [97] and Chapter 2), so it is also bisimulation-invariant. Note that the bisimulation-invariance of L_ι also follows from the fact that there is a recursive procedure which transforms any L_ι-formula φ into a MIC-formula φ_{MIC} which is equivalent to φ on finite trees. We can define the procedure by induction on the complexity of φ. The only interesting clause is $\iota x.\varphi_{\mathsf{MIC}}(x) := (ifp\ x : S)$ where:

$$S := \begin{cases} x \leftarrow \varphi_{\mathsf{MIC}}(x) \wedge y \\ y \leftarrow \Box\bot \vee \Box y \end{cases}$$

Moreover, the ι-calculus can be embedded in MSO. This can be shown by induction on the complexity of L_ι-formulas using a standard translation argument (see Chapter 2) where the new clause is $ST_x(\iota x_i.\varphi(x_i)) = \exists X_i((\forall y(ST_y(\varphi(x_i)) \leftrightarrow X_i(y)) \wedge X_i(x))$. The result then follows by the Janin Walukiewicz Theorem (see Theorem 2.2.13 in Chapter 2), which says that, in particular on finite trees, every bisimulation-invariant logic which is contained in MSO is also contained in the μ-calculus. Moreover, from the fact that the translation from bisimulation-invariant MSO-formulas to L_μ-formulas given by Theorem 2.2.13 is effective, it follows that the translation from L_ι-formulas to L_μ-formulas is effective. □

Corollary 6.8.13. *L_ι is decidable.*

Proof. To determine whether a L_ι-formula is satisfied in a finite tree model, first translate it in MSO, then construct an equivalent finite tree automata and check for emptiness of the language of the automata. For details on the relation in between finite tree automata and MSO-formulas on finite trees, see [127]. □

Let us now list a few questions. First, it would be interesting to look at the details of the translation procedure obtained in Theorem 6.8.12 via the Janin Walukiewicz Theorem. As it involves translations in MSO, it is likely that there might be more direct and efficient translation procedures. This question is closely related to the issue of the possible greater succinctness of L_ι as compared to the μ-calculus on finite trees[27], which we leave here as an open problem. Another question is the following. Consideration of the proof used in [40] in order to show the completeness of the μ-calculus on finite trees strongly suggests that a complete axiomatization of L_ι could easily be obtainable using a similar method. We leave the details of this question for further work. Then, it would also be interesting to determine more precisely general syntactic conditions satisfied by MIC formulas for which there is a L_ι-formula which is equivalent on finite trees. Such a criteria would allow to isolate a fragment of MIC which would be decidable on finite trees (remember that we noted in Chapter 2 that MIC is undecidable already on finite words). Also note that the decidability of L_ι immediately implies that the set of guarded MPC-formulas is decidable on finite trees.

Finally, let us conclude this section by saying that we hope our analysis of game solution concepts can also feed back to general fixed-point logics by raising logical questions that are of interest per se. One such question concerns logics featuring recursion along a well-founded order definable in structures of some given class; as a starting point for exploring the direction, we believe the simple modal ι-calculus introduced in the present section would deserve to be explored further.

6.9 Towards Well-Behaved Fixed-Point Logics on Finite Extensive Games

After our excursions into order-conform fixed-point logics on trees, we return to the main line of this chapter: defining solution concepts for games in the sense of game theory. Actually, we will not develop further theory here, but confine ourselves to raising a few key questions which arise at once.

[27]For the notion of compared succinctness of logics which have the same expressive power on a given class of structures, in particular on finite trees, we refer to [79] and [78].

Finding suitable fragments Of course, game solution procedures need not use the full power of logical languages that can define recursive procedures. Thus, there is a question *which fragments* are needed in our analysis. It might make sense to look at decidable fragments such as the modal μ-calculus - and indeed, [18] points out how the latter suffices, e.g., for defining the game solutions needed for Zermelo's Theorem. This may look too poor, since we often want to define relations on trees, and not just unary predicates. But we have already seen how subsets of the *move* relation are encoded by unary predicates, so a lot can be done in this way. Still, the intriguing issue is this. Crucially, game solution intertwines two different relations on trees: the *move* relation and the *preference* relations for players on endpoints. And the question is what happens to the known properties of computational logics when we add such preference relations. In particular, the following intriguing issue then arises.

Potential problem: the complexity of rationality In logics of *action* and *knowledge*, it is well-known that apparently harmless assumptions such as Perfect Recall for agents make the bimodal logic undecidable, and sometimes even Π^1_1-complete ([86]). The reason is that these assumptions generate commuting diagrams for actions *move* and epistemic uncertainty $\sim$ satisfying a "confluence" property of the form

$$\forall x \forall y((move(x, y) \wedge y \sim z) \rightarrow \exists u(x \sim u \wedge move(u, z))$$

that can serve as the basic grid cells in encodings of Tiling Problems in the logic. Thus, the logical theory of games with players that have perfect memory is more complex than that of forgetful agents ([24]).

But now consider the non-epistemic property of *Rationality* that mixes action and preference. The earlier properties CF1, CF2 have a similar flavor: they express the existence of a confluence diagram involving action and preference links. For instance, CF1 said this:

$$\forall x \forall y((turn_i(x) \wedge \sigma(x, y)) \rightarrow \forall z(move(x, z) \\ \rightarrow \forall u((end(u) \wedge \sigma^*(y, u)) \rightarrow \exists v(end(v) \wedge \sigma^*(z, v) \wedge v \leq_i u))))$$

So, what is the complexity of fixed-point logics for players with this kind of behavior? Can it be that Rationality, a widely used property meant to make behavior simple and predictable, actually makes its logical theory complex? Concrete instances of this open problem arise once we fix a sufficiently expressive logical language over trees.[28]

[28] *Model-checking complexity and definability on finite trees.* Balder ten Cate has reminded us of the potential use of *descriptive complexity theory* ([90]) for studying finite games. First, checking for game solutions is related to model checking logical formulas, say, stating the intended effects of players strategies. As an example, since both FO(LFP) and FO(IFP) capture PTIME on finite models (given an enumeration order on the tree), it should be close to defining

The Gist of it all: Modal Logics of Best Action We have made a plea for analyzing game solution procedures explicitly in rich logics. This follows the program of *making strategies explicit* advocated in [19]. But while this is useful in some cases, there is also the opposite direction of judiciously hiding information about the machinery of strategies when it is not needed. In practical reasoning, we are often only interested in our *best actions* without all details of their justification. Game solution procedures take a model with actions and preferences, and then compute a new relation of best action. As a mathematical abstraction, it would be good to extract a simple surface logic (a small modal fragment of complex fixed-point logics) for reasoning with best actions, while hiding most of the machinery:

Open problem Can we axiomatize the modal logic of finite game trees with a *move* relation and its transitive closure, turns and *preference* relations for players, and a new relation *best* as computed by Backward Induction?

We conjecture that we get a simple modal logic for the moves (these exist) plus a basic preference logic, while the modality $\langle best \rangle$ satisfies some obvious base laws plus one major bridge axiom that we already encountered earlier:[29]

Fact 6.9.1. The following modal axiom corresponds to CF2 by standard techniques:

$$(turn_i \wedge \langle best \rangle [best^*](end \to p)) \to [move]\langle best^* \rangle (end \wedge \langle pref_i \rangle p)$$

In this concrete setting, the earlier problem returns that the Rationality assumption built into this logic may be a grid property leading to undecidability. Is the modal logic of best action decidable?[30]

6.10 Further Issues in Extended Game Logics

In addition to the definability issues that we solved so far, game logics raise some other questions and there are many further lines for investigation following up on our stray observations. For instance, we want a more general view of possible

all "testable" properties of games. And other results in descriptive complexity theory may be game-theoretically relevant as well.

[29] [113] has some related thoughts on "logics of solved games".

[30] In our earlier analysis of Backward Induction, we look at either shrinking game trees, or smaller sets of "best moves" as the recursive procedure unfolds. This suggest that the logic of best moves may need a further modality: not just the above "absolute best" given the game as it is, but also "relative best" given some constraint on the set of nodes or moves considered. (This is similar to having conditional belief as a basic notion in doxastic logics.) If this is right, we may need in order to get completeness more operators than just the one shown in Fact 6.9.1.

representation languages, and on the notions of set preference that determine the dominance relation defining rationality. More generally, it would be of interest to connect our style of analysis for game solution more systematically with that found in epistemic game theory (c.f. [115], [52], [137]), where epistemic models are added describing what players know or believe about the course of the game.

In addition, some extensions to the games themselves seem natural:

Infinite games Can we extend our analysis to deal with *infinite games*? A transition to infinite ordinal sequences is easy to add to our iterated announcement or upgrade scenarios. Also, our general fixed-point definitions still make sense in this setting, though the special recursion over a well-founded tree dominance relation is no longer available. But there may be more to this generalization. Typically, in infinite trees, the reasoning changes direction, from "backward" to "forward". Here is an illustration:

Example 6.10.1 (Weak Determinacy)**.** The following principle holds in all infinite game trees, for any condition φ on histories:

> If player E has no strategy forcing φ at some stage s of the game, then A has a strategy for achieving a set of runs from s during all of which E never has a strategy forcing φ for the remaining game from then on.

In the notation of temporal game logics with *forcing modalities* $\{\}$, this says

$$\{\mathsf{E}\}\varphi \vee \{\mathsf{A}\}\mathsf{G}\neg\{\mathsf{E}\}\varphi$$

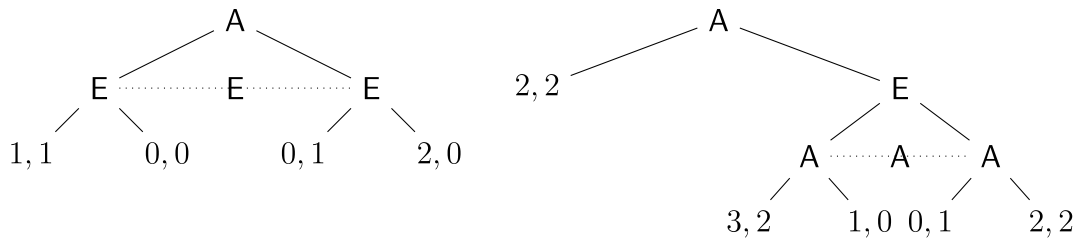
Here the reasoning is a typical inverse of Backward Induction. Suppose that $\neg\{\mathsf{E}\}\varphi$. A's strategy then arises as follows. If E is to move, then no successor available to her can guarantee a win, since she has no winning strategy now - and so A can just "wait and see". If A is to move, then there must be at least one possible move leading to a state where E has no winning strategy: otherwise, E has a winning strategy right now after all. Continuing this way, A is bound to produce runs of the kind described.[31]

How would our earlier analysis extend to a setting like this, where infinite histories themselves are the outcomes of the game, and players try to achieve global properties of these?

[31] This argument has a *co-algebraic* flavor, cf. [132], that we do not pursue here.

Dynamics in games with imperfect information Moreover, many if not most games have *imperfect information*, with uncertainties for players where they are in the game tree. Think of card games, or other games with restricted observation. Can't our analysis be extended to this area, where in general, Backward Induction no longer works? We merely illustrate the task ahead with two simple scenarios for the reader to ponder:

Example 6.10.2 (Strategic reasoning in imperfect information games)**.** In the following games, outcome values are written in the order ("A-value, E-value"):



[32]

The game to the left seems a straightforward extension of techniques for removing dominated moves, but that to the right raises tricky issues of what A would be telling E by moving right. We leave the question what should or will happen in both games to the reader: [52], [137] have more discussion.[33]

Language design and game equivalence As a final perspective, we mention that the choice of a best language for games is also correlated with the choice of an optimal notion of *structural equivalence* between games ([13]). The richer the equivalence, the stronger the language needed to capture its invariant properties. The options for languages that we have discussed here may also reflect the fact that there is no consensus yet on what such a structural notion of game equivalence should be. [34]

6.11 Coda: Alternatives to Backward Induction and True Game Dynamics

Our discussion in this chapter is basically complete, but we feel that we should mention one more issue that has received quite some attention in the literature

[32]The tree to the right is adapted from an example in an invited lecture by Robert Stalnaker at the Gloriclass Farewell Event, ILLC Amsterdam, January 2010.

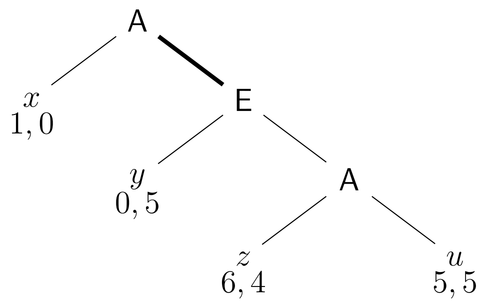
[33]Further challenges to our analysis include equilibria with *coalitions* of players, and *simultaneous moves*. Nothing in our logics prevent this: it has just not been done yet.

[34][13] proposes notions of game bisimulation where back and forth moves occur only when there is a switch of turns between players. This seems similar to the notion of "stuttering" encountered in Chapter 4 and 5, but we leave this analogy to further research.

(cf. [25]). It has been claimed that the very reasoning underlying Backward Induction, our hitherto unproblematic running example of a game solution method, is incoherent:

The paradox of Backward Induction

Example 6.11.1 (The "Paradox of Backward Induction")**.** Recall the style of reasoning toward a Backward Induction solution, as in:



Backward Induction tells us that A will go left at the start, on the basis of logical reasoning that is available to both players. But then, if A plays right (see the black line) what should E conclude? Does not this mean that A is not following the BI reasoning, and that all bets are off as to what he will do later on in the game? It seems that the very basis for the above computation collapses.

Responses to this conceptual difficulty vary - and many authors doubt that there is a genuine paradox here. The characterization result of [6] assumes that players know that rationality prevails throughout, something that [119] calls "rationality no matter what", a stubborn unshakable belief that players will act rationally later on, even if they have not done so up until now.[35] [10] essentially take the same tack, deriving the BI strategy from an assumption of "stable true belief" in Rationality, a gentler form of stubbornness stated in terms of dynamic-epistemic logic.

Logics of actions, preference, and agent types Personally, we are more inclined toward another analysis, in line with [125]. A richer game analysis should add an account of the types of agent that play a game. In particular, we need to represent the *belief revision* policies by the players, that determine what they will do when making a surprising observation contradicting their beliefs in the course of a game. There are many different options for such policies in the above example, such as

- "It was just an error, and A will go back to being rational",

[35] One can defend this by assuming that the other player only makes isolated "mistakes".

- "A is inviting me to go right, and I will be rewarded for that",
- "A is an automaton with a general rightward tendency", and so on ...[36]

Our logical analysis so far omits this type of information about players of the game, since our algorithms make implicit uniform assumptions about what they are going to do as the game proceeds.

Belief revision policies are not an explicit part of our models so far. Thus, our fixed-point logics tell only a limited story. Eventually, we may need a richer mathematical model for game solution, that can also deal with the dynamics of how players update knowledge and revise beliefs as a game proceeds.

6.12 Conclusion

We have shown how standard logical fixed-point languages can define game solution procedures and their resulting relations of "best action". We think that this is a good format for more general studies of game-theoretic notions, including finding alternatives to currently received views. But also, we hope to have shown that the game arena poses interesting problems for existing logics of computation, as one adds further structure that is typical for agents: preference, information, and eventually, even "processing types" for agents. All these contacts may eventually lead to legitimate children of logic and game theory. The chapter has mainly analyzed Backward Induction as a key to seeing how fixed-point logics can be used for game solution. From a logical perspective, the issue is now how to continue, what are the most useful fixed-point logics in the wide array that we have brought to bear and how expressive would be interesting well-behaved (in particular, axiomatizable) fragments of such logics. At the moment, we find it hard to choose, though we do think that both general fixed-point logics and special logics exploiting well-founded relations in trees make sense. We intend to investigate further game solution concepts to get a better sense which of the logics used in this chapter will stand the test of game-theoretic generalization.

[36]One reaction to these surprise events might even be a switch to an entirely new reasoning style about the game. That might require more finely-grained *syntax-based* views of revision.

Chapter 7

Conclusion

In this thesis we studied model-theoretic and proof-theoretic aspects of widely used logics on trees, including fixed-point extensions of first-order logic and linear-time temporal logic and μ-calculus.

In Chapter 2, we gave an overview of the area by presenting the different classes of structures and logics that we consider in the thesis. We discussed the issues of expressive power, decidability and complete axiomatization.

In Chapter 3 we presented complete axiomatizations for MSO, $\mathsf{FO}(\mathsf{TC}^1)$ and $\mathsf{FO}(\mathsf{LFP}^1)$ on finite node-labelled sibling-ordered trees. In order to prove completeness, we developed model-theoretic tools specifically geared towards Henkin models. We believe that these tools are of independent interest. Indeed, since the original publication of the results, both our axiomatizations and our proof techniques have been applied in other settings, namely in the context of coalition logic [27] and of the μ-calculus [40].

In Chapter 4, we essentially identified $\mathsf{LTL}(\mathsf{X})$, $\mu\mathsf{TL}$ and $\mu\mathsf{TL}(\mathsf{U})$ as the three only temporal fragments of $\mu\mathsf{TL}$ that satisfy Craig interpolation (moreover, they satisfy uniform interpolation). In this sense, our results singled out these three temporal logics as being exceptionally well-behaved. On finite and ω-words, $\mu\mathsf{TL}$ and $\mu\mathsf{TL}(\mathsf{U})$ have the same expressive power as MSO and its stutter-invariant fragment (with respect to initial semantics), and therefore these results can also be seen as identifying fragments of MSO that satisfy uniform interpolation on words. Well-behaved logics on trees are often characterized with respect to the trade-off that they provide between expressive power and complexity, but interpolation deserves to be explored further, as it is another interesting criteria to compare and classify these logics. In particular, extending our results to the case of MSO on trees and branching-time temporal logics is an interesting challenge.

In Chapter 5, we gave a complete axiom system for $\mu\mathsf{TL}(\mathsf{U})$, which was identified in the previous chapter as one of the three fragments of $\mu\mathsf{TL}$ with Craig interpolation, and which, it appears, has not been studied before. The results we gave were obtained through $\mu\mathsf{TL}(\Diamond_\Gamma)$, a new logic that has the same expressive power as μTL(U), but which is syntactically extremely close to μTL. We believe that $\mu\mathsf{TL}(\Diamond_\Gamma)$ could be reused as a tool to easily transfer results from μTL to μTL(U) and to other logics characterizing the stutter-invariant fragment of MSO on words.

In Chapter 6, we turned to an important special case of tree structures: finite extensive games with perfect information. Such games are finite trees enriched with additional relations representing the preferences of players. We showed how various fixed-point logics can define standard solution concepts from game theory while staying faithful to the underlying solution procedures. More generally, by making a link with current dynamic-epistemic logics of knowledge update and plausibility change, we showed how this approach can provide "dynamic foundations" for game analysis that supplement the usual style of thinking. In doing all this, we also studied fixed-point logics that exploited the special well-founded orders available in game trees. In addition, we touched upon the issue of the complexity of such logics, since one wants to know how the expressive power needed for game solution balances with the potential undecidability of trees with additional structure beyond the successor order. Finally, we identified a great number of further issues that arise when we take our fixed-point logic perspective to more sophisticated parts of game theory.

To conclude, let us emphasize a few general points and questions related to fixed-point logics on trees that became more visible through the results obtained in this thesis. First of all, the search for well-behaved systems appears as a recurring theme. Various criteria for identifying them were studied: expressivity, decidability, axiomatization, interpolation and, in connection with games, procedural aspects. Secondly, a general back and forth between modal and first-order languages also characterized the perspective adopted in the thesis. As we explained in Chapter 2, this is a distinctive feature of the landscape of fixed-point logics on trees. Indeed, as we mentioned above, the work described in Chapter 3 has inspired further work in modal logic [27, 40], whereas the results in Chapter 4 and 5, while formulated mostly in modal terms, shed light on well-behaved fragments of MSO. For instance, we identified μTL(U) as the stutter-invariant fragment of μTL, which is also the stutter-invariant fragment of MSO on words. Chapter 6 explicitly dealt with both modal and quantified logics. The issue of finding a balance between expressive power and complexity was especially highlighted, as the additional preference or knowledge structure carried by game trees typically increase complexity and call for the identification of well-behaved modal fragments of the usual logics on trees. As a final note, let us mention that similar issues are being addressed in the context of XML query languages, where one also need to add rich additional features to basic tree structures. Finite trees indeed serve as the standard theoretical abstraction of XML documents, but in this context it is often important to enriching the trees with additional "data structure" consisting of data values from an infinite alphabet, and enriching the logics with means to compare the data values associated to different nodes of a tree. This increases complexity dramatically, and FO is for instance no longer decidable on such structures. Some work has been done in order to identify decidable fragments of usual logics on such enriched tree structures, also known as *data trees* (see in particular [28, 65]). It might be interesting to take inspiration from the results obtained in this area in order to characterize interesting decidable fragments of fixed-point logics on finite extensive game trees.

Bibliography

[1] Loredana Afanasiev and Balder ten Cate. On Core XPath with Inflationary Fixed-Points. *Proceedings of FICS 2009*, 2009.

[2] Eyal Amir and Sheila A. McIlraith. Partition-based logical reasoning for first-order and propositional theories. *Artif. Intell.*, 162(1-2):49–88, 2005.

[3] Carlos Areces and Maarten de Rijke. Interpolation and bisimulation in temporal logic. In Ruy J. Guerra B. de Queiroz and Marcelo Finger, editors, *Proceedings of WoLLIC'98*, pages 15–21, 1998.

[4] Andre Arnold. *Finite Transition Systems: Semantics of Communicating Systems*. Prentice Hall, 1994.

[5] André Arnold and Damian Niwinski. *Rudiments of μ-Calculus*, volume 146 of *Studies in Logic and Foundations of Mathematics*. North-Holland, 2001.

[6] Robert Aumann. Backward induction and common knowledge of rationality. *Games and Economic Behavior*, 8:6–19, 1995.

[7] Rolf Backofen, James Rogers, and Krishnamurti Vijay-Shankar. A First-Order Axiomatization of the Theory of Finite Trees. *Journal of Logic, Language and Information*, 4(4):5–39, 1995.

[8] Alexandru Baltag and Sonja Smets. Conditional Doxastic Models: A Qualitative Approach to Dynamic Belief Revision. *Electr. Notes Theor. Comput. Sci.*, 165:5–21, 2006.

[9] Alexandru Baltag and Sonja Smets. Group belief dynamics under iterated revision: Fixed points and cycles of joint upgrades. *Proceedings of TARK 2009*, pages 41–50, 2009.

[10] Alexandru Baltag, Sonja Smets, and Jonathan Zvesper. Keep “hoping” for rationality: A solution to the backward induction paradox. *Synthese*, 169(2):301–333, 2009.

[11] Pablo Barceló and Leonid Libkin. Temporal Logics over Unranked Trees. In *LICS*, pages 31–40, 2005.

[12] John Barwise and Solomon Feferman. *Model-Theoretic Logics*. Springer, New York, 1985.

[13] Johan van Benthem. Extensive Games as Process Models. *Journal of Logic, Language and Information*, 11(3):289–313, June 2002.

[14] Johan van Benthem. Games in dynamic epistemic logic. *Bulletin of Economic Research*, 53(4):219–248, 2002.

[15] Johan van Benthem. Minimal predicates, fixed-points, and definability. *Journal of Symbolic Logic*, 70(3):696–712, 2005.

[16] Johan van Benthem. Modal Frame Correspondences and Fixed-Points. *Studia Logica*, 83(1):133–155, 2006.

[17] Johan van Benthem. Dynamic logic of belief revision. *Journal of Applied Non-Classical Logics*, 17(2):129–155, 2007.

[18] Johan van Benthem. Rational Dynamics and Epistemic Logic in Games. *International Game Theory Review (IGTR)*, 09(01):13–45, 2007.

[19] Johan van Benthem. In praise of strategies. to appear in J. van Eijck and R. Verbrugge, eds., Games, Actions and Social Software, College Publications, London, 2008.

[20] Johan van Benthem. The logic of rational agency. Invited Lecture, Third Indian ICLA Winter School, Chennai, 2009.

[21] Johan van Benthem. *Logical Dynamics of Information and Interaction*. Cambridge University Press, Cambridge, 2010.

[22] Johan van Benthem and Amélie Gheerbrant. Game Solution, Epistemic Dynamics and Fixed-Point Logics. *Fundam. Inform.*, 100(1-4):19–41, 2010.

[23] Johan van Benthem, Sieuwert van Otterloo, and Olivier Roy. *Modality Matters: Twenty-Five Essays in Honour of Krister Segerberg*, volume 53 of *Philosophical Studies*, chapter Preference Logic, Conditionals, and Solution Concepts in Games. Lagerlund, Lindstrm and Sliwinski, Uppsala, 2006.

[24] Johan van Benthem and Eric Pacuit. The tree of knowledge in action. *Proceedings Advances in Modal Logic*, pages 87–106, 2006.

[25] Cristina Bicchieri and Oliver Schulte. Common Reasoning about Admissibility. *Erkenntnis*, 45:229–325, 1997.

[26] Patrick Blackburn, Maarten de Rijke, and Yde Venema. *Modal Logic*. Cambridge University Press, Cambridge, 2000.

[27] Guido Boella, Dov M. Gabbay, Valerio Genovese, and Leendert van der Torre. Higher-order coalition logic. In *ECAI*, pages 555–560, 2010.

[28] Mikolaj Bojanczyk, Anca Muscholl, Thomas Schwentick, Luc Segoufin, and Claire David. Two-Variable Logic on Words with Data. In *LICS*, pages 7–16, 2006.

[29] Giacommo Bonanno. The logic of rational play in games of perfect information. *Economics and Philosophy*, 7:37–65, 1991.

[30] George Boolos. *The Logic of Provability*. Cambridge University Press, Cambridge, 1993.

[31] Egon Börger, Erich Grädel, and Yuri Gurevich. *The Classical Decision Problem*. Universitext. Springer, Berlin, Heidelberg, New York, 1997.

[32] Uwe Bosse. An "Ehrenfeucht-Fraïssé game" for fixpoint logic and stratified fixpoint logic. In *CSL '92: Selected Papers from the Workshop on Computer Science Logic*, pages 100–114, London, UK, 1993. Springer-Verlag.

[33] Michael C. Browne, Edmund M. Clarke, and Orna Grumberg. Characterizing Finite Kripke Structures in Propositional Temporal Logic. *Theor. Comput. Sci.*, 59:115–131, 1988.

[34] Véronique Bruyère and Olivier Carton. Automata on linear orderings. *J. Comput. System Sci.*, 73(1):1–24, 2007.

[35] J. Richard Büchi. Weak Second-Order Arithmetics and Finite Automata. *Z. Math. Logik Grundl. Math.*, 6:66–92, 1960.

[36] J. Richard Büchi and Dirk Siefkes. *Decidable Theories: Vol. 2: The Monadic Second Order Theory of All Countable Ordinals*. Lectures Notes in Mathematics. Springer, Berlin, Heidelberg, 1973.

[37] John P. Burgess. Axioms for tense logic. I. "Since" and "Until". *Notre Dame J. Formal Logic*, 23(4):367–374, 1982.

[38] A. Calo and Johann A. Makowsky. The Ehrenfeucht-Fraïssé games for transitive closure. *Lecture Notes in Computer Science*, 620:57–68, 1992.

[39] Balder ten Cate. *Model Theory for Extended Modal Languages*. PhD thesis, University of Amsterdam, 2005.

[40] Balder ten Cate and Gaëlle Fontaine. An Easy Completeness Proof for the Modal μ-Calculus on Finite Trees. In *FOSSACS*, pages 161–175, 2010.

[41] Balder ten Cate and Maarten Marx. Axiomatizing the Logical Core of XPath 2.0. In *ICDT*, pages 134–148, 2007.

[42] Alonzo Church. A Note on the Entscheidungsproblem. *The Journal of Symbolic Logic*, 1(1):Mar., 40-41 1936.

[43] Alonzo Church. An Unsolvable Problem of Elementary Number Theory. *American Journal of Mathematics*, 58(2):345–363, Apr. 1936.

[44] William Craig. Three Uses of the Herbrand-Gentzen Theorem in Relating Model Theory and Proof Theory. *Journal of Symbolic Logic*, 22(3):269–285, 1957.

[45] Giovanna D'Agostino. Interpolation in non-classical logics. *Synthese*, 164(3):421–435, 2008.

[46] Giovanna D'Agostino and Marco Hollenberg. Logical Questions Concerning the μ-Calculus: Interpolation, Lyndon and Lós-Tarski. *Journal of Symbolic Logic*, 65(1):310–332, 2000.

[47] Max Dauchet and Sophie Tison. The Theory of Ground Rewrite Systems is Decidable. In *LICS*, pages 242–248, 1990.

[48] Anuj Dawar. A Restricted Second-Order Logic for Finite Structures. *Inf. Comput.*, 143(2):154–174, 1998.

[49] Anuj Dawar, Erich Grädel, and Stephan Kreutzer. Inflationary Fixed-Points in Modal Logic. *ACM Transactions on Computational Logic*, 5:282–315, 2004.

[50] Christian Dax, Felix Klaedtke, and Stefan Leue. Specification Languages for Stutter-Invariant Regular Properties. In *ATVA*, pages 244–254, 2009.

[51] Boudewijn de Bruin. *Explaining Games: on the Logic of Game Theoretic Explanations*. PhD thesis, ILLC, University of Amsterdam, 2004.

[52] Cédric Dégremont. *The Temporal Mind. Observations on the Logic of Belief Change in Interactive Systems*. PhD thesis, ILLC, University of Amsterdam, 2010.

[53] Stéphane Demri and Philippe Schnoebelen. The Complexity of Propositional Linear Temporal Logics in Simple Cases. *Information and Computation*, 174(20):84–103, 2002.

[54] Marc Denecker and Eugenia Ternovska. A Logic of Nonmonotone Inductive Definitions. *ACM Trans. Comput. Logic*, 9(2):1–52, 2008.

[55] Wiebe van der Hoek and Marc Pauly. *Handbook of Modal Logic*, chapter Modal Logic for Games and Information, pages 1077–1148. Elsevier (add editors), Amsterdam, 2006.

[56] Hans van Ditmarsch, Wiebe van der Hoek, and Barteld Kooi. *Dynamic-Epistemic Logic*. Cambridge University Press, Cambridge, 2007.

[57] Kees Doets. *Completeness and Definability : Applications of the Ehrenfeucht Game in Second-Order and Intensional Logic*. PhD thesis, Universiteit van Amsterdam, 1987.

[58] Heinz-Dieter Ebbinghaus and Jörg Flum. *Finite Model Theory*. Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1995.

[59] E. Allen Emerson. Temporal and modal logic. In Jan van Leeuwen, editor, *Handbook of Theoretical Computer Science*, volume B, pages 995–1072. Elsevier, 1990.

[60] Herbert B. Enderton. *A Mathematical Introduction to Logic*. Academic Press, New York, NY, USA, 1972.

[61] Kousha Etessami. Stutter-Invariant Languages, omega-Automata, and Temporal Logic. In Nicolas Halbwachs and Doron Peled, editors, *Proceedings of CAV*, pages 236–248, London, UK, 1999. Springer-Verlag.

[62] Kousha Etessami, Moshe Y. Vardi, and Thomas Wilke. First-Order Logic with Two Variables and Unary Temporal Logic. In *LICS*, pages 228–235, 1997.

[63] Ronald Fagin, Larry J. Stockmeyer, and Moshe Y. Vardi. On Monadic NP vs. Monadic co-NP (Extended Abstract). In *Structure in Complexity Theory Conference*, pages 19–30, 1993.

[64] Solomon Feferman and Robert Vaught. The first-order properties of algebraic systems. *Fundamenta Mathematicae*, 47:57–103, 1959.

[65] Diego Figueira. Satisfiability of downward XPath with data equality tests. In *PODS*, pages 197–206, 2009.

[66] Gaëlle Fontaine and Thomas Place. Frame definability for classes of trees in the μ-calculus. In *MFCS*, pages 381–392, 2010.

[67] Dov Gabbay, Amir Pnueli, Saharon Shelah, and Jonathan Stavi. On the Temporal Analysis of Fairness. *Proceedings of the 7th ACM SIGPLAN-SIGACT symposium on Principles of programming languages*, pages 163–173, 1980.

[68] John Geanakoplos and Herakles Polemarchakis. We Can't Disagree Forever. *Journal of Economic Theory*, 28:192–200, 1982.

[69] Amélie Gheerbrant. Complete Axiomatization of the Stutter Invariant Fragment of the Linear-Time μ-Calculus. to appear in the proceedings of AiML 2010, 2010.

[70] Amélie Gheerbrant and Balder ten Cate. Complete Axiomatizations of MSO, FO(TC1) and FO(LFP1) on Finite Trees. In *LFCS*, pages 180–196, 2009.

[71] Amélie Gheerbrant and Balder ten Cate. Craig Interpolation for Linear Temporal Languages. In *CSL*, pages 287–301, 2009.

[72] Rob J. van Glabbeek. The Linear Time - Branching Time Spectrum (Extended Abstract). In *CONCUR*, pages 278–297, 1990.

[73] Rob J. van Glabbeek. The Linear Time - Branching Time Spectrum II. In *CONCUR*, pages 66–81, 1993.

[74] Kurt Gödel. Über eine bisher noch nicht benüzte erweiterung des finiten standpunktes. *Dialectica*, 12:280287, 1958.

[75] Georg Gottlob and Christoph Koch. Monadic datalog and the expressive power of languages for web information extraction. In *Proceedings of PODS 2002*, pages 17–28, 2002.

[76] Erich Grädel. On Transitive Closure Logic. In *CSL '91: Proceedings of the 5th Workshop on Computer Science Logic*, pages 149–163, London, UK, 1992. Springer-Verlag.

[77] Martin Grohe. *The Structure of Fixed-Point Logics*. PhD thesis, University of Freiburg, 1994.

[78] Martin Grohe and Nicole Schweikardt. Comparing the succinctness of monadic query languages over finite trees. *ITA*, 38(4):343–373, 2004.

[79] Martin Grohe and Nicole Schweikardt. The succinctness of first-order logic on linear orders. *Logical Methods in Computer Science*, 1(1), 2005.

[80] Jan Friso Groote and Frits W. Vaandrager. An Efficient Algorithm for Branching Bisimulation and Stuttering Equivalence. In *ICALP*, pages 626–638, 1990.

[81] Ron Gross. Invariance under Stuttering in Branching-Time Temporal Logic. Master's thesis, Israel Institute of Technology, Haifa, 2008.

[82] Yuri Gurevich. Elementary Properties of Oredered Abelian Groups (Russian). *Algebra i Logika*, 3(1):5–39, 1964.

[83] Yuri Gurevich. Elementary Properties of Oredered Abelian Groups. *American Mathematical Society Translations*, 46(1):165–192, 1965.

[84] Yuri Gurevich. *Model-Theoretic Logics*, chapter Chapter XIII: Monadic Second-Order Theories, pages 479–506. Springer-Verlag, New York, NY, USA, 1985.

[85] Yuri Gurevich and Saharon Shelah. Fixed-Point Extensions of First-Order Logic. *Annals of Pure and Applied Logic*, 32:265–280, 1986.

[86] Joseph Halpern and Moshe Vardi. The Complexity of Reasoning about Knowledge and Time, I: lower bounds. *Journal of Computer and System Sciences*, 38(1):195–237, 1989.

[87] Paul Harrenstein. *Logic in Conflict*. PhD thesis, Institute of Computer Science, University of Utrecht, 2004.

[88] Leon Henkin. Completeness in the Theory of Types. *The Journal of Symbolic Logic*, 15(2):81–91, 1950.

[89] Wilfrid Hodges. *A Shorter Model Theory*. Cambridge University Press, New York, NY, USA, 1997.

[90] Neil Immermann. *Descriptive Complexity*. Springer, Berlin, 1999.

[91] David Janin and Igor Walukiewicz. On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In *CONCUR*, pages 263–277, 1996.

[92] Roope Kaivola. *Using Automata to Characterise Fixed-Point Temporal Logics*. PhD thesis, University of Edinburgh, 1997.

[93] Hans Kamp. *Tense Logic and the Theory of Linear Order*. PhD thesis, UCLA, Los Angeles, 1968.

[94] Jerome Keisler. Logic with the quantifier "there exists uncountably many". *An. Math. Logic*, 1:1–93, 1970.

[95] Stephan Kepser. Querying Linguistic Treebanks with Monadic Second-Order Logic in Linear Time. *J. of Logic, Lang. and Inf.*, 13(4):457–470, 2004.

[96] Stephan Kepser. Properties of Binary Transitive Closure Logic over Trees. In Giorgio Satta Paola Monachesi, Gerald Penn and Shuly Wintner, editors, *Formal Grammar 2006*, pages 77–89, 2006.

[97] Stephan Kreutzer. *Pure and Applied Fixed-Point Logics*. PhD thesis, University of Aachen, 2002.

[98] Stephan Kreutzer. Expressive Equivalence of Least and Inflationary Fixed-Point Logic. *Annals of Pure and Applied Logic*, 130(1–3):61–78, 2004.

[99] Orna Kupferman, Moshe Y. Vardi, and Pierre Wolper. An automata-theoretic approach to branching-time model checking. *J. ACM*, 47(2):312–360, 2000.

[100] Grégory Lafitte and Jacques Mazoyer. Théorie des modèles et complexité. Technical report, Ecole Normale Supérieure de Lyon, septembre 1998.

[101] Leslie Lamport. What Good is Temporal Logic? In R. E. A. Mason, editor, *Proceedings of the IFIP 9th World Computer Congress*, pages 657–668. North-Holland/IFIP, 1983.

[102] Hans Lauchli and Christian Savioz. Monadic Second Order Definable Relations on the Binary Tree. *J. Symb. Log.*, 52(1), 1987.

[103] Gerard R. Renardel de Lavalette. Interpolation in computing science: the semantics of modularization. *Synthese*, 164(3):437–450, 2008.

[104] Leonid Libkin. *Elements of Finite Model Theory (Texts in Theoretical Computer Science. An Eatcs Series)*. SpringerVerlag, 2004.

[105] Fenrong Liu. *Changing for the Better, Preference Dynamics and Agent Diversity*. PhD thesis, Illc, University of Amsterdam, 2008.

[106] Johann A. Makowsky. Algorithmic uses of the Feferman Vaught Theorem. *Annals of Pure and Applied Logic*, 126(1–3):159–213, 2004.

[107] Larissa Maksimova. Temporal logics with "the next" operator do not have interpolation or the Beth property. *Siberian Mathematical Journal*, 32(6):989–993, 1991.

[108] María Manzano. *Extensions of First Order Logic*. Cambridge University Press, New York, NY, USA, 1996.

[109] Oliver Matz and Nicole Schweikardt. Expressive power of monadic logics on words, trees, pictures, and graphs. In Erich Grädel Jörg Flum and Thomas Wilke, editors, *Logic and Automata: History and Perspectives*, pages 531–552. Texts in Logic and Games, Amsterdam University Press, 2007.

[110] Louise E. Moser, P. M. Melliar-Smith, G. Kutty, and Y. S. Ramakrishna. Completeness and Soundness of Axiomatizations for Temporal Logics without Next. *Fundamenta Informatica*, 21(4):257–305, 1994.

[111] Damian Niwinski. Fixed Points vs. Infinite Generation. In *Proceedings of LICS*, pages 402–409, 1988.

[112] Martin Osborne and Ariel Rubinstein. *A Course in Game Theory.* The MIT Press, Cambridge (Mass.), 1994.

[113] Sieuwert van Otterloo. *A Strategic Analysis of Multi-Agent Protocols.* PhD thesis, ILLC, Amsterdam, 2005.

[114] Eric Pacuit and Olivier Roy. Interactive Rationality. Lectures Notes, University of Groningen and University of Tilburg, 2010.

[115] Eric Pacuit and Olivier Roy. Interactive Rationality. Lecture Notes, Departments of Philosophy, University of Groningen and University of Tilburg, 2010.

[116] Doron Peled and Thomas Wilke. Stutter-Invariant Temporal Properties are Expressible Without the Next-Time Operator. *Inf. Process. Lett.*, 63(5):243–246, 1997.

[117] Alexander Moshe Rabinovich. Expressive Completeness of Temporal Logic of Action. In *MFCS*, pages 229–238, 1998.

[118] James Rogers. *Descriptive Approach to Language-Theoretic Complexity.* CSLI Publications, Stanford, CA, USA, 1998.

[119] Dov Samet. Counterfactuals in Wonderland. *Games and Economic Behavior*, 51(2):537–541, 1997.

[120] Nicole Schweikardt. On the Expressive Power of Monadic Least Fixed-Point Logic. In *ICALP*, pages 1123–1135, 2004.

[121] Krister Segerberg. Modal logics with functional alternative relations. *Notre Dame Journal of Formal Logic*, 27(4):504–522, 1986.

[122] Reinhard Selten. Spieltheoretische Behandlung eines Oligopolmodells mit Nachfrageträgheit. *Zeitschrift für die gesamte Staatswissenschaft*, 121:301–324, 1965.

[123] Reinhard Selten. Reexamination of the Perfectness Concept for Equilibrium Points in Extensive Games. *International Journal of Game Theory*, 4:25–55, 1975.

[124] Dirk Siefkes. An Axiom System for the Weak Monadic Second Order Theory of Two Successors. *Israel Journal of Mathematics*, 30(3):264–284, 1978.

[125] Robert Stalnaker. Extensive and Strategic Form: Games and Models for Games. *research in Economics*, 53:293–319, 1999.

[126] Balder ten Cate and Luc Segoufin. Transitive closure logic, nested tree walking automata, and XPath. *J. ACM*, 57(3), 2010.

[127] Wolfgang Thomas. *Handbook of formal languages, vol. 3: beyond words*, chapter Languages, automata, and logic, pages 389–455. Springer-Verlag, New York, NY, USA, 1997.

[128] Hans-Jörg Tiede and Stephan Kepser. Monadic Second-Order Logic and Transitive Closure Logics over Trees. *Electron. Notes Theor. Comput. Sci.*, 165:189–199, 2006.

[129] Alan Turing. On computable problems, with an application to the entscheidungsproblem. *Proc. of the London Math. Soc.*, 42:230–265, 1937.

[130] Moshe Y. Vardi. A Temporal Fixpoint Calculus. In *Proceedings of POPL*, pages 250–259, 1988.

[131] Moshe Y. Vardi. From Philosophical to Industrial Logics. In *ICLA*, pages 89–115, 2009.

[132] Yde Venema. Algebras and coalgebras. In P. Blackburn, F. Wolter, and J. van Benthem, editors, *Handbook of Modal Logic*, pages 331–426. Elsevier, 2006.

[133] Igor Walukiewicz. A Note on the Completeness of Kozen's Axiomatization of the Propositional μ-Calculus. *The Bulletin of Symbolic Logic*, 2(3), 1996.

[134] Thomas Wilke. Classifying Discrete Temporal Properties. In *STACS*, pages 32–46, 1999.

[135] G. H. von Wright. *The Logic of Preference.* Edinburgh University Press, Edinburgh, 1963.

[136] Charles Zaiontz. Axiomatization of the Monadic Theory of Ordinals $< \omega^2$. *Mathematical Logic Quarterly*, 29(6):337–356, 1983.

[137] Jonathan Zvesper. *Playing with Information.* PhD thesis, ILLC, University of Amsterdam, 2010.

Samenvatting

In dit proefschrift bestuderen we bewijstheoretische en modeltheoretische aspecten van enkele veelgebruikte modale en gekwantificeerde dekpuntlogica's op bomen.

Hoofdstuk 2 behandelt basisprincipes van modale logica, temporele logica, dekpunt logica, en enkele eerste-orde en hogere-orde logica's over boomstructuren.

In hoofdstuk 3 beschouwen we eindige bomen, met labels op knopen, en een ordening op directe opvolgers. We presenteren axiomatisaties van de bijhorende theorieën: monadische tweede-orde logica (MSO), monadische transitieve afsluitingslogica ($\mathsf{FO}(\mathsf{TC}^1)$) en de monadische kleinste dekpuntlogica ($\mathsf{FO}(\mathsf{LFP}^1)$). Via modeltheoretische technieken tonen we met een uniform argument aan dat deze axiomatisaties volledig zijn. Met andere woorden, elke formule die waar is op alle bomen kan bewezen worden met behulp van de gegeven axioma's.

In hoofdstuk 4 bestuderen we tal van fragmenten en uitbreidingen van propositionele temporele logica (LTL). Deze fragmenten zijn gedefinieerd door ofwel de toegestane temporele operatoren te beperken, ofwel door toevoegen van kleinste dekpuntoperatoren aan de taal. Voor elk van deze fragmenten identificeren we de kleinste uitbreiding die de Craig interpolatie eigenschap heeft. We doen dit met techieken uit de abstracte modeltheorie. Afhankelijk van de toegelaten temporele operatoren verkrijgen we op deze manier drie logica's: het fragment van LTL met slechts de "Next" operator, de uitbreiding van LTL met een kleinste dekpuntoperator μ (beter bekend als de lineaire-tijd μ-calculus), en tenslotte $\mu\mathsf{TL}$, de uitbreiding met een kleinste dekpuntoperator van het fragment van LTL dat slechts de "Until" operator bevat.

In hoofdstuk 5 concentreren we ons vervolgens op de logica $\mu\mathsf{TL}(\mathsf{U})$. Dit fragment werd in het vorige hoofdstuk geïdentificeerd als het stotter-invariante fragment van de lineare-tijd μ-calculus $\mu\mathsf{TL}$. Daarenboven was dit fragment één van de drie temporele fragmenten van $\mu\mathsf{TL}$ die aan de Craig interpolatie eigenschap voldoet. Hoewel een volledige axiomatisatie voor de twee andere fragmenten reeds bekend is, is dit niet het geval voor $\mu\mathsf{TL}(\mathsf{U})$. Daarom geven we volledige axiomatisaties voor $\mu\mathsf{TL}(\mathsf{U})$ op zowel de klasse van eindige woorden als de klasse van ω-woorden. Hiertoe introduceren we een nieuwe temporele logica, $\mu\mathsf{TL}(\Diamond_\Gamma)$, een variant van $\mu\mathsf{TL}$ waarin de “Next” operator vervangen is door de corresponderende familie van stotter-invariante operatoren. Deze logica heeft dezelfde uitdrukkingskracht als $\mu\mathsf{TL}(\mathsf{U})$. Gebruikmakende van reeds bekende resultaten over $\mu\mathsf{TL}$, tonen we eerst de volledigheid aan van $\mu\mathsf{TL}(\Diamond_\Gamma)$, waaruit dan de vo! lledigheid van $\mu\mathsf{TL}(\mathsf{U})$ wordt verkregen.

Tenslotte brengen we in hoofdstuk 6 onze analysemethoden, gebruikmakend van modale en temporele dekpuntlogica's, over naar de speltheorie. De huidige oplossingsmethoden voor spelen bevatten een vorm van “procedurele rationaliteit” en deze vraagt om een logische analyse op zichzelf. Meer in het bijzonder bestudeert dit hoofdstuk de speciale casus van “Terugwaartse Inductie” voor extensieve spelen. We analyzeren een aantal recente versies van dit algoritme in termen van kennis en geloofsherziening in logica's die tevens voorkeuren van spelers kunnen beschrijven. We tonen aan dat elk van deze analyses wiskundig equivalent is vanuit het oogpunt van dekpuntlogica's op bomen. We generaliseren het aldus ontstane perspectief op spelen tot een exploratie van dekpuntlogica's op eindige bomen die het best passen bij speltheoretische evenwichtsbegrippen. We besluiten het hoofdstuk met een meer algemeen onderzoeksprogramma dat ten doel heeft een synthese te vormen tussen! computationele logica's en speltheorie.

Abstract

In this thesis, we study proof-theoretic and model-theoretic aspects of some widely used modal and quantified fixed-point logics on trees.

Chapter 2 includes basics of modal logic, temporal logic, fixed-point logics, and some first-order and higher-order logics of tree structures.

In Chapter 3, we consider the class of finite node-labelled sibling-ordered trees. We present axiomatizations of its monadic second-order logic (MSO), monadic transitive closure logic ($\mathsf{FO(TC^1)}$) and monadic least fixed-point logic ($\mathsf{FO(LFP^1)}$) theories. Using model-theoretic techniques, we show by a uniform argument that these axiomatizations are complete, i.e., each formula which is valid on all finite trees is provable using our axioms.

In Chapter 4 we consider various fragments and extensions of propositional linear temporal logic (LTL), obtained by restricting the set of temporal connectives or by adding a least fixed-point construct to the language. Using techniques from abstract model-theory, for each of these logics we identify its smallest extension that has Craig interpolation. Depending on the underlying set of temporal operators, this framework turns out to be one of the following three logics: the fragment of LTL having only the Next operator; the extension of LTL with a least fixed-point operator μ (known as linear time μ-calculus); and $\mu\mathsf{TL(U)}$, the least fixed-point extension of the "Until-only" fragment of LTL.

In Chapter 5, we focus on the logic $\mu\mathsf{TL(U)}$, that we identified in the previous chapter as the stutter-invariant fragment of the linear-time μ-calculus $\mu\mathsf{TL}$. We also identified this logic as one of the three only temporal fragments of $\mu\mathsf{TL}$ that satisfy Craig interpolation. Complete axiom systems were known for the two other fragments, but this was not the case for $\mu\mathsf{TL(U)}$. We provide complete axiomatizations of $\mu\mathsf{TL(U)}$ on the class of finite words and on the class of ω-words. For this purpose, we introduce a new logic $\mu\mathsf{TL}(\Diamond_\Gamma)$, a variation of $\mu\mathsf{TL}$ where the "Next time" operator is replaced by the family of its stutter-invariant counterparts. This logic has exactly the same expressive power as $\mu\mathsf{TL(U)}$. Using known results for $\mu\mathsf{TL}$, we first prove completeness for $\mu\mathsf{TL}(\Diamond_\Gamma)$, which then allows

us to obtain completeness for $\mu\mathsf{TL}(\mathsf{U})$.

Finally, in Chapter 6 we take our style of analysis via modal and temporal fixed-point logics to games. Current methods for solving games embody a form of "procedural rationality" that invites logical analysis in its own right. This chapter is a case study of Backward Induction for extensive games. We consider a number of analyses from recent years in terms of knowledge and belief update in logics that also involve preference structure, and we prove that they are all mathematically equivalent in the perspective of fixed-point logics of trees. We then generalize our perspective on games to an exploration of fixed-point logics on finite trees that best fit game-theoretic equilibria. We end with a broader program for merging computational logics to the area of game theory.

Titles in the ILLC Dissertation Series:

ILLC DS-2001-01: **Maria Aloni**
Quantification under Conceptual Covers

ILLC DS-2001-02: **Alexander van den Bosch**
Rationality in Discovery - a study of Logic, Cognition, Computation and Neuropharmacology

ILLC DS-2001-03: **Erik de Haas**
Logics For OO Information Systems: a Semantic Study of Object Orientation from a Categorial Substructural Perspective

ILLC DS-2001-04: **Rosalie Iemhoff**
Provability Logic and Admissible Rules

ILLC DS-2001-05: **Eva Hoogland**
Definability and Interpolation: Model-theoretic investigations

ILLC DS-2001-06: **Ronald de Wolf**
Quantum Computing and Communication Complexity

ILLC DS-2001-07: **Katsumi Sasaki**
Logics and Provability

ILLC DS-2001-08: **Allard Tamminga**
Belief Dynamics. (Epistemo)logical Investigations

ILLC DS-2001-09: **Gwen Kerdiles**
Saying It with Pictures: a Logical Landscape of Conceptual Graphs

ILLC DS-2001-10: **Marc Pauly**
Logic for Social Software

ILLC DS-2002-01: **Nikos Massios**
Decision-Theoretic Robotic Surveillance

ILLC DS-2002-02: **Marco Aiello**
Spatial Reasoning: Theory and Practice

ILLC DS-2002-03: **Yuri Engelhardt**
The Language of Graphics

ILLC DS-2002-04: **Willem Klaas van Dam**
On Quantum Computation Theory

ILLC DS-2002-05: **Rosella Gennari**
Mapping Inferences: Constraint Propagation and Diamond Satisfaction

ILLC DS-2002-06: **Ivar Vermeulen**
A Logical Approach to Competition in Industries

ILLC DS-2003-01: **Barteld Kooi**
Knowledge, chance, and change

ILLC DS-2003-02: **Elisabeth Catherine Brouwer**
Imagining Metaphors: Cognitive Representation in Interpretation and Understanding

ILLC DS-2003-03: **Juan Heguiabehere**
Building Logic Toolboxes

ILLC DS-2003-04: **Christof Monz**
From Document Retrieval to Question Answering

ILLC DS-2004-01: **Hein Philipp Röhrig**
Quantum Query Complexity and Distributed Computing

ILLC DS-2004-02: **Sebastian Brand**
Rule-based Constraint Propagation: Theory and Applications

ILLC DS-2004-03: **Boudewijn de Bruin**
Explaining Games. On the Logic of Game Theoretic Explanations

ILLC DS-2005-01: **Balder David ten Cate**
Model theory for extended modal languages

ILLC DS-2005-02: **Willem-Jan van Hoeve**
Operations Research Techniques in Constraint Programming

ILLC DS-2005-03: **Rosja Mastop**
What can you do? Imperative mood in Semantic Theory

ILLC DS-2005-04: **Anna Pilatova**
A User's Guide to Proper names: Their Pragmatics and Semanics

ILLC DS-2005-05: **Sieuwert van Otterloo**
A Strategic Analysis of Multi-agent Protocols

ILLC DS-2006-01: **Troy Lee**
Kolmogorov complexity and formula size lower bounds

ILLC DS-2006-02: **Nick Bezhanishvili**
Lattices of intermediate and cylindric modal logics

ILLC DS-2006-03: **Clemens Kupke**
Finitary coalgebraic logics

ILLC DS-2006-04: **Robert Špalek**
Quantum Algorithms, Lower Bounds, and Time-Space Tradeoffs

ILLC DS-2006-05: **Aline Honingh**
The Origin and Well-Formedness of Tonal Pitch Structures

ILLC DS-2006-06: **Merlijn Sevenster**
Branches of imperfect information: logic, games, and computation

ILLC DS-2006-07: **Marie Nilsenova**
Rises and Falls. Studies in the Semantics and Pragmatics of Intonation

ILLC DS-2006-08: **Darko Sarenac**
Products of Topological Modal Logics

ILLC DS-2007-01: **Rudi Cilibrasi**
Statistical Inference Through Data Compression

ILLC DS-2007-02: **Neta Spiro**
What contributes to the perception of musical phrases in western classical music?

ILLC DS-2007-03: **Darrin Hindsill**
It's a Process and an Event: Perspectives in Event Semantics

ILLC DS-2007-04: **Katrin Schulz**
Minimal Models in Semantics and Pragmatics: Free Choice, Exhaustivity, and Conditionals

ILLC DS-2007-05: **Yoav Seginer**
Learning Syntactic Structure

ILLC DS-2008-01: **Stephanie Wehner**
Cryptography in a Quantum World

ILLC DS-2008-02: **Fenrong Liu**
Changing for the Better: Preference Dynamics and Agent Diversity

ILLC DS-2008-03: **Olivier Roy**
Thinking before Acting: Intentions, Logic, Rational Choice

ILLC DS-2008-04: **Patrick Girard**
Modal Logic for Belief and Preference Change

ILLC DS-2008-05: **Erik Rietveld**
Unreflective Action: A Philosophical Contribution to Integrative Neuroscience

ILLC DS-2008-06: **Falk Unger**
Noise in Quantum and Classical Computation and Non-locality

ILLC DS-2008-07: **Steven de Rooij**
Minimum Description Length Model Selection: Problems and Extensions

ILLC DS-2008-08: **Fabrice Nauze**
Modality in Typological Perspective

ILLC DS-2008-09: **Floris Roelofsen**
Anaphora Resolved

ILLC DS-2008-10: **Marian Counihan**
Looking for logic in all the wrong places: an investigation of language, literacy and logic in reasoning

ILLC DS-2009-01: **Jakub Szymanik**
Quantifiers in TIME and SPACE. Computational Complexity of Generalized Quantifiers in Natural Language

ILLC DS-2009-02: **Hartmut Fitz**
Neural Syntax

ILLC DS-2009-03: **Brian Thomas Semmes**
A Game for the Borel Functions

ILLC DS-2009-04: **Sara L. Uckelman**
Modalities in Medieval Logic

ILLC DS-2009-05: **Andreas Witzel**
Knowledge and Games: Theory and Implementation

ILLC DS-2009-06: **Chantal Bax**
Subjectivity after Wittgenstein. Wittgenstein's embodied and embedded subject and the debate about the death of man.

ILLC DS-2009-07: **Kata Balogh**
Theme with Variations. A Context-based Analysis of Focus

ILLC DS-2009-08: **Tomohiro Hoshi**
Epistemic Dynamics and Protocol Information

ILLC DS-2009-09: **Olivia Ladinig**
Temporal expectations and their violations

ILLC DS-2009-10: **Tikitu de Jager**
"Now that you mention it, I wonder…": Awareness, Attention, Assumption

ILLC DS-2009-11: **Michael Franke**
Signal to Act: Game Theory in Pragmatics

ILLC DS-2009-12: **Joel Uckelman**
More Than the Sum of Its Parts: Compact Preference Representation Over Combinatorial Domains

ILLC DS-2009-13: **Stefan Bold**
Cardinals as Ultrapowers. A Canonical Measure Analysis under the Axiom of Determinacy.

ILLC DS-2010-01: **Reut Tsarfaty**
Relational-Realizational Parsing

ILLC DS-2010-02: **Jonathan Zvesper**
Playing with Information

ILLC DS-2010-03: **Cédric Dégremont**
The Temporal Mind. Observations on the logic of belief change in interactive systems

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**
Algebra and Topology. Investigations into canonical extensions, duality theory and point-free topology.