N° d'ordre : 4210

THÈSE

PRÉSENTÉE A

L'UNIVERSITÉ BORDEAUX 1

ÉCOLE DOCTORALE DE MATHÉMATIQUES ET INFORMATIQUE

Par Alessandro FACCHINI

POUR OBTENIR LE GRADE DE

DOCTEUR

SPÉCIALITÉ : Informatique

A STUDY ON THE EXPRESSIVE POWER OF SOME FRAGMENTS OF THE MODAL μ-CALCULUS

Professeur: Jacques DUPARC
Directeur de recherche CNRS : Igor WALUKIEWICZ

Soutenue le : 3 Décembre 2010

Devant la commission d'examen formée de :

M. DUPARC, Jacques	Professeur	Université de Lausanne	Co-directeur de thèse
M. GARBINATO, Benoît	Professeur	Université de Lausanne	Examinateur
M. GRÄDEL, Erich	Professeur	Université d'Aix-la-Chapelle	Examinateur
M. JÄGER, Gerhard	Professeur	Université de Bern	Examinateur
M. NIWINSKI, Damian	Professeur	Université de Varsovie	Rapporteur
M. PIGNEUR Yves	Professeur	Université de Lausanne	Président du Jury
M. VAN BENTHEM, Johan	Professeur	Université d'Amsterdam	Rapporteur
M. WALUKIEWICZ, Igor	DR CNRS	Université de Bordeaux I	Directeur de thèse
M. ZEITOUN, Marc	Professeur	Université de Bordeaux I	Examinateur

Résumé

Dans ce travail nous étudions la complexité de certains fragments du μ-calcul selon deux points de vue: l'un syntaxique et l'autre topologique. Dans la première partie nous adoptons le point de vue syntaxique afin d'étudier le comportement du μ-calcul sur des classes restreintes de modèles. Parmi d'autres résultats, nous montrons en particulier que sur les modèles transitifs toute propriété définissable par une formule du μ-calcul est définissable par une formule sans alternance de points fixes. Pour ce qui concerne la perspective topologique, nous montrons d'abord que sur les modèles transitifs la logique modale correspond au fragment borélien du μ-calcul. Ensuite nous donnons une description effective des hiérarchies de Borel et de Wadge d'un sous-fragment sans alternance de cette logique sur les arbres binaires et vérifions que pour ce fragment les points de vue topologique et syntaxique coïncident.

Mots-clés : μ-calcul, alternance de points fixes, automates d'arbres, hiérarchie de Wadge, hiérarchie de Borel.

Abstract

In this work we study the complexity of some fragments of the modal μ-calculus from two points of view: the syntactical and the topological. In the first part of the dissertation we adopt the syntactical point of view in order to study the behavior of this formalism on some restricted classes of models. Among other results, we show that on transitive transition systems, every μ-formula is logically equivalent to an alternation free formula. For what concerns the topological point of view, we first prove that on transitive models, the modal logic is exactly the Borel fragment of the modal μ-calculus. Then we provide an effective description of the Borel and Wadge hierarchies of a sub-fragment of the alternation free fragment of the μ-calculus on binary trees. Finally we verify that for this fragment the syntactical point of view and topological point of view coincide.

Keywords : μ-calculus, fixpoint alternation, tree automata, Wadge hierarchy, Borel hierarchy.

BORDEAUX UNIVERSITY 1
Laboratoire Bordelais de Recherche en Informatique

UNIVERSITY OF LAUSANNE
Faculty of Business and Economics

A Study on the Expressive Power of Some Fragments of the Modal μ-Calculus

Alessandro Facchini

PhD Dissertation

Contents

Résumé (long)

Le μ-calcul est une extension de la logique modale par des opérateurs de point fixe. Dans ce travail nous étudions la complexité de certains fragments de cette logique selon deux points de vue, différents mais étroitement liés: l'un syntaxique (ou combinatoire) et l'autre topologique. Du point de vue syntaxique, les propriétés définissables dans ce formalisme sont classifiées selon la complexité combinatoire des formules de cette logique, c'est-à-dire selon le nombre d'alternances des opérateurs de point fixe. Comparer deux ensembles de modèles revient ainsi à comparer la complexité syntaxique des formules associées. Du point de vue topologique, les propriétés définissables dans cette logique sont comparées à l'aide de réductions continues ou selon leurs positions dans la hiérarchie de Borel ou dans celle projective.

Dans la première partie de ce travail nous adoptons le point de vue syntaxique afin d'étudier le comportement du μ-calcul sur des classes restreintes de modèles. En particulier nous montrons que:

(1) sur la classe des modèles symétriques et transitifs le μ-calcul est aussi expressif que la logique modale;

(2) sur la classe des modèles transitifs, toute propriété définissable par une formule du μ-calcul est définissable par une formule sans alternance de points fixes,

(3) sur la classe des modèles réflexifs, il y a pour tout n une propriété qui ne peut être définie que par une formule du μ-calcul ayant au moins n alternances de points fixes,

(4) sur la classe des modèles bien fondés et transitifs le μ-calcul est aussi expressif que la logique modale.

Le fait que le μ-calcul soit aussi expressif que la logique modale sur la classe des modèles bien fondés et transitifs est bien connu. Ce résultat est en effet la conséquence d'un théorème de point fixe prouvé indépendamment par De Jongh et Sambin au milieu des années 70. La preuve que nous donnons de l'effondrement de l'expressivité du μ-calcul sur cette classe de modèles est néanmoins indépendante de ce résultat. Par la suite, nous étendons le langage du μ-calcul en permettant aux opérateurs de point fixe de lier des occurrences négatives de variables libres. En montrant alors que ce formalisme est aussi expressif que le fragment modal, nous sommes en mesure de fournir une nouvelle preuve du théorème d'unicité des point fixes de Bernardi, De Jongh et Sambin et une preuve constructive du théorème d'existence de De Jongh et Sambin.

Pour ce qui concerne les modèles transitifs, du point de vue topologique cette fois, nous prouvons que la logique modale correspond au fragment borélien du μ-calcul sur cette classe des systèmes de transition. Autrement dit, nous vérifions que toute propriété définissable des modèles transitifs qui, du point de vue topologique, est une propriété borélienne, est nécessairement une propriéte modale, et inversement. Cette caractérisation du fragment modal découle du fait que nous sommes en mesure de montrer que, modulo EF-bisimulation, un ensemble d'arbres est définissable dans la logique temporelle EF si et seulement il est borélien. Puisqu'il est possible de montrer que ces deux propriétés coïncident avec une caractérisation effective de la définissabilité dans la logique EF dans le cas des arbres à branchement fini donnée par Bojanczyk et Idziaszek [24], nous obtenons comme corollaire leur décidabilité.

Dans une deuxième partie, nous étudions la complexité topologique d'un sous-fragment du fragment sans alternance de points fixes du μ-calcul. Nous montrons qu'un ensemble d'arbres est définissable par une formule de ce fragment ayant au moins n alternances si et seulement si cette propriété se trouve au moins au n-ième niveau de la hiérarchie de Borel. Autrement dit, nous vérifions que pour ce fragment du μ-calcul, les points de vue topologique et combinatoire coïncident. De plus, nous décrivons une procédure effective capable de calculer pour toute propriété définissable dans ce langage sa position dans la hiérarchie de Borel, et donc le nombre d'alternances de points fixes nécessaires à la définir. Nous nous intéressons ensuite à la classification des ensembles d'arbres par réduction continue, et donnons une description effective de l'ordre de Wadge de la classe des ensembles d'arbres définissables dans le formalisme considéré. En particulier, la hiérarchie que nous obtenons a une hauteur $(\omega^\omega)^\omega$. Nous complétons ces résultats en décrivant un algorithme permettant de calculer la position dans cette hierarchie de toute propriété définissable.

Introduction

In this dissertation we study the complexity of some fragments of the modal μ-calculus from two different but closely related points of view: the syntactical (or combinatorial) and the topological. The syntactical perspective means that properties definable in the modal μ-calculus are classified according to the combinatorial complexity of the defining formulae, that is to say according to the number of alternations of fixpoints. The comparison of two sets of models therefore comes to comparing the syntactical complexity of the associated formulae. On the other hand, from the topological perspective, properties definable in the modal μ-calculus are either compared through continuous reductions or by their positions in the Borel and projective hierarchies, such positions being based on how many times the operations of projection, countable unions and complementation must be used, starting from simple (open) sets, in order to obtain the considered collections. To get a first glimpse of the issues to be discussed, we shall introduce the main object of our investigations: the modal μ-calculus.

The modal μ-calculus is an extension of modal logic with least and greatest fixpoint operators. The term "μ-calculus" and the idea of extending modal logic with fixpoints appeared for the first time in the paper of Scott and De Bakker [110] and was further developed by other authors. Nowadays, the term "modal μ-calculus" stands for the formal system introduced by Kozen [73]. It is a powerful logic of programs subsuming (almost) all the most studied dynamic and temporal logics, like PDL, LTL, CTL and CTL*, and corresponds exactly to the bisimulation invariant fragment of monadic second order logic. Therefore these are very good reasons to consider the μ-calculus as the right *meta* formal system for reasoning about assertions concerning temporal properties of dynamic (reactive and parallel) systems with potentially infinite behavior.

What makes the modal μ-calculus very powerful in terms of expressivity is not simply the use of greatest and least fixpoint operators, but rather the possibility of nesting (alternate) them. This observation naturally leads to the introduction of a measure of this syntactical alternation, generating what is called the *fixpoint alternation hierarchy.* This hierarchy, which shape is represented in figure 1, attempts to capture the complexity of a certain class of models depending on the combinatorial complexity of the defining formulae. More precisely, the fixpoint hierarchy consists in the collection $\{\Sigma_n^{\mu\mathbb{T}} : n \in \mathbb{N}\} \cup \{\Pi_n^{\mu\mathbb{T}} : n \in \mathbb{N}\}$. The sigma class $\Sigma_n^{\mu\mathbb{T}}$ denotes the collection of models of formulae with at most n alternations of fixpoint operators, starting with a least fixpoint, whereas the dual pi class $\Pi_n^{\mu\mathbb{T}}$ denotes the collection of models of formulae with at most n alternations of fixpoint operators, but starting with a greatest fixpoint. When $n = 0$, both the corresponding pi and sigma classes coincide with the collection of models of modal formulae. The delta (ambiguous) classes are given by the intersection of the two dual classes immediately above in the hierarchy.

$$
\begin{tikzcd}
\Sigma_0^{\mu\mathbb{T}} \arrow[rr] \arrow[dr] & & \Sigma_1^{\mu\mathbb{T}} \arrow[rr] \arrow[dr] & & \Sigma_2^{\mu\mathbb{T}} & \cdots \\
|| & \Delta_1^{\mu\mathbb{T}} \arrow[ur] \arrow[dr] & & \Delta_2^{\mu\mathbb{T}} \arrow[ur] \arrow[dr] & & \cdots \\
\Pi_0^{\mu\mathbb{T}} \arrow[rr] \arrow[ur] & & \Pi_1^{\mu\mathbb{T}} \arrow[rr] \arrow[ur] & & \Pi_2^{\mu\mathbb{T}} & \cdots
\end{tikzcd}
$$

Figure 1: The fixpoint alternation hierarchy.

From this point of view, understanding the complexity of the modal μ-calculus is like having a complete description of this hierarchy, i.e. being able to answer the following questions:

- can we always find properties that require more and more complex formulae to be expressed (strictness of the hierarchy) ?
- can the position in the hierarchy of a definable language always be decided?
- and, eventually, can we give a (possibly effective) characterization of its levels?

For the general case, since the work of Bradfield and others [32, 33, 34, 7, 80], we know that the hierarchy is strict. However, at the moment, we can only decide the low levels of the hierarchy illustrated above [78, 129, 103]. Concerning the characterization of its levels, we know for example that the intersection of the two incomparable classes of the first level corresponds to the class of models of purely modal formulae, while the intersection of the two incomparable classes of the second level is the class of models of formulae without any alternation [77, 103].

The objective of the first part of this thesis is to describe the fixpoint alternation hierarchy when only models that satisfy a certain well-defined property are considered, such as, for instance, that of being a transitive model.

The modal μ-calculus is further strongly connected with automata theory, another important area of computer science. Automata are abstract models of machines that perform computations on an input by moving through a serie of states or configurations. Through automata, computer scientists are able to understand how machines compute functions and solve problems and, more importantly, what it means for a function to be defined as computable or for a question to be described as decidable. Initially, the considered input was just a finite string. But over the last few decades, several new types of automata have been introduced that extend finite automata in various directions. A natural extension is to consider infinite computations as a model for non-terminating reactive systems.

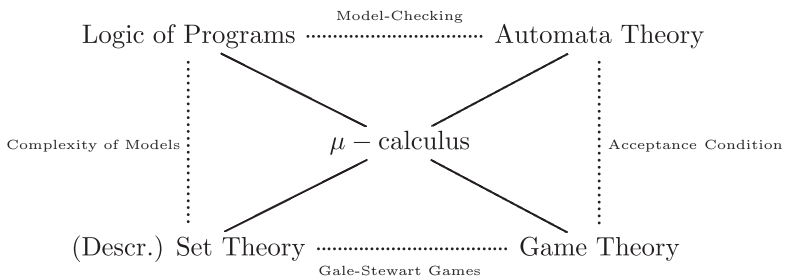
The connection between automata and logic was first established in the early 1960s in the work of Büchi and of Elgot, who showed that there is an effective correspondence between finite automata and monadic second-order logic interpreted over finite words. Later, in the work of Büchi, McNaughton and Rabin, such equivalence was also shown between automata and monadic second order logic over *infinite* words and trees. Moreover, thanks to this equivalence, Büchi and Rabin were able to prove that the monadic second order theory $S1S$, resp. $S2S$, of one, resp. two, successor functions were decidable.

Since those seminal works, the idea of translating a logic into appropriate models of finite-state automata on infinite words or infinite trees has become a central paradigm in the theory of system verification (cf. [133] for a nice recent overview). Through this translation, the model-checking problem is reduced to the non emptyness problem for automata. This is for instance the case for the modal μ-calculus, whose counterpart in terms of automata is given by alternating automata with the parity condition [53]. Roughly, alternating parity automata are non-deterministic automata whose set of states is partitioned between the set of existential states and the set of universal states and such that a natural number is associated to every state. The computation of an alternating parity automaton on a infinite words or trees is then given in terms of a two-players infinite game with perfect information, called a parity game. A player is in charge of the existential nodes, while the other is in charge of the universal nodes. The game proceeds in rounds. At the beginning of every round there is a copy of the alternating automaton in its own initial state in a certain node of the tree. During a round, the player in charge of the initial state sends a copy of the automaton to a successor in the tree and changes the initial state, all this done according to the transition function. Since a natural number was assigned to every state, at the end of a play, an infinite sequence of natural numbers is produced. A play is thence said to be won by the player in charge of the existential nodes if the maximum number occurring infinitely often in the sequence is even. An infinite word or infinite tree is said to be accepted by an alternating tree automaton if the existential player has a winning strategy in the corresponding parity game. Because we are able to effectively find an equivalent alternating parity automaton running on infinite words or trees for every μ-formula, we can thus successfully base model-checking algorithms for the μ-calculus on tools provided by that class of finite-state automata.

When considering recognizable sets of infinite words or trees, three hierarchies are classically used to measure their complexity: the Mostowski-Rabin index hierarchy, or the Borel/projective hierarchy and the Wadge hierarchy, which is an enormous refinement of the Borel/projective hierarchy. Since it reflects the depth of nesting of positive and negative conditions, the first hierarchy determines the combinatorial complexity of the recognizing automaton and is closely related to the previous fixpoint alternation hierarchy of the modal μ-calculus. The second and third hierarchies, on the other hand, capture the topological complexity of languages accepted by such devices. Indeed, from a topological point of view, an infinite tree is very similar to an infinite word, which is also very close to a real number. The Wadge hierarchy in particular enables a precise comparison of different models of computation. Consider for instance deterministic and weak alternating automata on trees. Knowing that there are deterministic languages that are not weakly recognizable and vice versa, and that therefore we cannot compare them by inclusion, how then can we decide what is the more powerful model of computation? The Wadge hierarchy, defined by the preorder induced on languages by simple (continuous) reductions, makes it possible. The sole heights (huge ordinals) of the Wadge hierarchy restricted to the classes under comparison provide more information then other logical techniques. It is for instance known that the Wadge hierarchy of deterministic tree languages has height $\omega^{\omega \cdot 3} + 3$, and $\omega^{\omega \cdot 3} + 2$ when restricted to Borel sets [92], whereas the height of the Wadge hierarchy of weakly recognizable tree languages, which are all Borel, is at least ε_0 [51]. But using the Wadge hierarchy enables also to compare models of computation on trees with models of computation on words. As another example, the Wadge hierarchies for the two previous models of computation on trees can be compared with the Wadge hierarchy for deterministic context-free word languages [49] or the Wadge hierarchy for word languages recognized by deterministic Turing machines [55], whose heights are respectively $(\omega^\omega)^\omega$ and $(\omega_1^{\mathsf{CK}})^\omega$, where ω_1^{CK} is the first non recursive ordinal, known also as the Church-Kleene ordinal. Moreover, evidences indicate that the two kinds of hierarchy (combinatorial vs topological) are closely related. Consider for instance once more the case of deterministic recognizable tree languages. On the one hand, Niwinski and Walukiewicz [101] have shown that a tree language recognized by a deterministic parity automaton is either hard for the co-Büchi level or is on a very low level in the hierarchy of weak alternating automata, and that this property has a precise topological counterpart. On the other hand, relying on the previous work, Murlak [93] was able to verify that for this class of languages the weak index and the Borel rank coincide and conjectured that this coincidence still holds for weakly recognizable tree languages.

In the second part of the dissertation we study the topological complexity of a subclass of weak alternating automata, or equivalently of a fragment of the alternation free fragment of the modal μ-calculus. In particular, this is done by describing the corresponding topological hierarchies and by verifying whether or not the correspondence between the combinatorial and topological complexities, known to hold for deterministic recognizable tree languages, extends to this class of automata.

To conclude, there seems to be a whole world of subtle and rich connections between (fixpoint) logics, (parity) automata, topology and games, as depicted in the next figure:



Hence, the study of the structure of the combinatorial and topological hierarchies of fragments of the modal μ-calculus, as well as that of their relationships, leads us to a rich web of connections between different fields, thus revealing the whole beauty and complexity of the theory underlying this logic. We hope that this dissertation contributes to this enterprise.

Outline of the dissertation

This dissertation consists of two parts preceeded by an introductory chapter, the first part (mainly) devoted to the syntactical approach, the other to the topological.

In **Chapter 1** we introduce all the relevant fundamental concepts and results that will be used in the two parts of the dissertation. In doing so, we always try to motivate and situate the introduced objects and results.

Chapters 2-4 constitute the **First Part** of our work. In **Chapter 2** we discuss the strictness of the fixpoint hierarchy on three restricted classes of models of the modal μ-calculus: symmetric and transitive models, transitive models, and finally reflexive models. We prove that:

(1) over symmetric and transitive models the modal μ-calculus collapses into its modal fragment,

(2) over transitive transition systems the modal μ-calculus collapses into its alternation free fragment,

(3) over reflexive models, the fixpoint alternation hierarchy is strict.

In **Chapter 3** we discuss the modal μ-calculus on well-founded and transitive models. In extending the language by allowing fixpoints to bound also negative occurrences of free variables and showing that it collapses on the modal fragment, we provide a new proof of the uniqueness theorem of Bernardi, De Jongh and Sambin and of the existence theorem of De Jongh, Sambin. For the last one we also give a simple algorithm which shows how the fixpoint can be computed. Chapters 2 and 3 are based on two journal papers joint with Luca Alberucci ([3, 4]). **Chapter 4** is a kind of "bridge" between the first and the second parts of the dissertation. Indeed, in this chapter we show that modal logic corresponds exactly to the Borel fragment of the modal μ-calculus on transitive models. This is done by providing a bunch of equivalent effective characterizations for the temporal logic EF on arbitrary trees. More specifically we prove that up to EF-bisimilarity, the property of being definable by an EF formula and the property of being a Borel set coincide for languages of both finite and infinite trees. Since we can verify that every WMSO-language is Borel, we also immediately obtain that the EF-bisimulation invariant fragment of weak monadic second order logic with the child relation is the logic EF. Because all these properties are proved to be equivalent also with a nice effective algebraic characterization of EF-definability for *finitely* branching trees given by Bojanczyk and Idziaszek [24], as a corollary we obtain their decidability. This chapter is based on a joint work with Balder ten Cate [40].

The shorter **Second Part** of the dissertation consists of Chapters 5 and 6. **Chapter 5** introduces and discusses the two topological hierarchies that will be investigated in the remaining chapters, as well as some useful new definitions concerning weak automata. In **Chapter 6**, we introduce a new subclass of weak alternating tree automata, linear game automata, and provide an effective characterization for all the three corresponding hierarchies: index hierarchy, Borel hierarchy and Wadge hierarchy. Moreover, we verify that for every language recognized by these automata, the Borel rank and the Mostowski-Rabin index coincide. This chapter is based on a conference paper joint with Jacques Duparc and Filip Murlak [50].

In the **Conclusion**, we summarize the main results and indicate directions for further research. Finally, the **Appendix A** contains the long combinatorial proof of a result used in Chapter 6.

Related works

Part 1

The strictness of the fixpoint alternation hierarchy has been first proven by Bradfield (cf. [32, 33]). Simultaneously, Lenzi [80] has proven a strictness theorem for the positive μ-calculus, that is, the fragment consisting of all formulae such that the propositional variables appear only positively. The same question, but restricted to full binary trees, has been independently solved by Bradfield [34] and Arnold [7].

The question whether the modal μ-calculus hierarchy collapses on special classes of transition systems has been addressed in various other works. A prominent subclass, coming from Gödel-Löb logic, is the class of transitive upward well-founded frames. As shown by Visser [123] and van Benthem [13] by using the De Jongh-Sambin fixpoint Theorem, the modal μ-calculus collapses to its modal fragment. An analogous result via the same technique but for the class of finite trees with the descendant relation is obtained in [39] by ten Cate, Fontaine and Litak. This collapse also follows from the main result proved by Bojanczyk and Walukiewicz [28].

Concerning the hierarchy on transitive frames, d'Agostino and Lenzi [44] propose a different purely combinatorial proof which explicitly uses Theorem 2.17 of this thesis. Moreover, by using some byproducts of their proof, the authors are able to show an unexpected behavior arising over finite transitive frames, namely that, despite the fact that the μ-calculus and modal logic do not coincide on finite transitive models, the bisimulation invariant fragments of first order and monadic second order logic coincide, meaning that the μ-calculus is included in first-order logic[1]. This inclusion also follows from a characterization of the bisimulation invariant fragment of MSO given by Dawar and Otto [46]. In this work, the authors extend the characterization given by van Benthem [12] of modal logic as the bisimulation invariant fragment of first order logic, and the characterization of the modal μ-calculus as the bisimulation invariant fragment of MSO given by Janin and Walukiewicz [67], to several subclasses of graphs, including transitive graphs, rooted graphs, finite rooted graphs, finite transitive graphs, well-founded transitive graphs, and finite equivalence graphs. From the above results the authors also obtain the collapse of the μ-calculus over transitive models.

[1] Note that this inclusion does not hold for the class of *all* transitive transition systems, a counterexample being well-foundedness.

In [45] D'Agostino and Lenzi extend the study of the expressiveness of the modal μ-calculus over some special classes of finite graphs characterized by having strongly connected components of size bounded by a finite constant. Notice that those classes generalize finite trees and (up to bisimulation) finite transitive graphs. In their paper, the authors show that on the considered classes of models, the μ-calculus collapses to the second semantical ambiguous class of the corresponding fixpoint alternation hierarchy[2]. These results also generalize an analogous result presented in [44] by the same authors but for the class of finite transition systems where every strongly connected component has at most one node for each possible label.

In the last years, thanks to an effort towards understanding fragments of CTL* and the successful use of what are called *forest algebra* (cf. [29]), effective characterizations of logics on trees have been obtained. Notably, this formalism has been used for obtaining decidable characterizations for the classes of tree languages definable in $\mathsf{EF} + \mathsf{EX}$ [28], $\mathsf{EF} + \mathsf{F}^{-1}$ [20, 106], $\mathsf{BC} - \Sigma_1(<)$ [26, 106], and $\Delta_2(\leq)$ [27, 106]. This approach has then been extended in the case of the temporal logic EF on infinite finitely branching trees by Bojanczyk and Idziaszek [24]. The decidability of the problem of knowing whether a formula of the modal μ-calculus is equivalent to a modal formula on transitive transition systems follows as an (almost) immediate corollary of the main result of this last work.

We already mentioned that at the level of transition systems, thanks to the work of Janin and Walukiewicz [67], the expressive power of the μ-calculus is well understood. In a series of articles [63, 64, 65], Janin and Lenzi highlight the fact that the relationship between the modal μ-calculus and monadic second order logic is richer than expected. This is done by comparing the first levels of the fixpoint alternation hierarchy with the bisimulation invariant fragments of the first levels of the quantifier alternation hierarchy of MSO. More precisely, while from [12] we already know that the bisimulation invariant fragment of the first level of the quantifier alternation hierarchy (that is first order logic) corresponds to the first level of the fixpoint alternation hierarchy (that is modal logic), the authors are able to prove that this correspondence holds up to the the level Σ_2 of the monadic hierarchy of formulae[3]. They also remark that this correspondence cannot hold higher in the hierarchy. Observe that the second level of the fixpoint alternation hierarchy of μ-formulae where the outermost fixpoint operator is a greatest fixpoint is known to be as expressible as Büchi tree automata [97, 66].

Part 2

Measuring the topological hardness of recognizable languages has a long tradition of research. In the case of infinite words, their understanding is almost complete since Wagner's 1977 paper [126]. In the same paper, Wagner was also the first to discover remarkable relations with the index hierarchy. Sub-

[2]Over the class of *all* transition systems, the second ambiguous class of the fixpoint alternation hierarchy is known to coincide with the alternation free fragment [77]. However this does not imply that the same result must hold over the considered subclasses, leaving open the question of whether the μ-calculus collapses to the alternation free fragment in those cases.

[3]This level is given by MSO formulae of the kind $\exists X_1 \cdots \exists X_n \forall Y_1 \cdots \forall Y_m \phi$, with ϕ a first order formula.

sequently, decision procedures determining an ω-regular language's position in both the topological and the index hierarchies were given by several authors [75, 100, 127]. On the contrary, the situation for trees is not so satisfactory. For example, although it is known that the index hierarchy is strict for deterministic [127], non deterministic [96], alternating [7, 34] and also for weak alternating tree automata [89], very little is known about the problem of computing the minimal index of a tree language. The only case examined satisfactorily is that of deterministic automata. On the one hand, Niwinski and Walukiewicz [101, 102] give algorithms to compute the deterministic and nondeterministic indices for deterministic languages, while Murlak [93] prove that for deterministic languages the Borel hierarchy and the weak index hierarchy coincide[4]. On the other hand, still Niwinski and Walukiewicz [102] show that the topological complexity of deterministic tree languages goes much higher than that of ω-regular languages. Indeed all recognizable set of infinite words are in the third Borel ambiguous class, while deterministic automata over trees can recognize either a language in the class $\mathbf{\Pi}^0_3$ in the Borel hierarchy or even $\mathbf{\Pi}^1_1$-complete languages. They are also able to show that it can be decided effectively which of the two cases takes place. In [90] Murlak solve the missing cases, thus providing a complete procedure calculating the position of a deterministic language in the Borel and projective hierarchies. For what concerns the Wadge hierarchy for deterministically recognizable sets of infinite trees, Murlak [92, 93] give a complete description of it. In particular he is able to provide an elementary procedure to decide if one deterministic tree language is continuously reducible to another and to show that the hierarchy has the height $\omega^{\omega\cdot 3}+3$, which should be compared with ω^ω for regular ω-languages [127], ω^{ω^2} for deterministic context-free ω-languages [49], $({\omega_1}^{\mathsf{CK}})^\omega$ for ω-languages recognized by deterministic Turing machines [111], or an unknown ordinal $\xi > (\omega_1^{\mathsf{CK}})^\omega$ for ω-languages recognized by non-deterministic Turing machines, and the same ordinal ξ for non-deterministic context-free languages [55]. For non-deterministic or alternating automata the only results obtained are strictness theorems for various classes [32, 33, 89, 96], and lower bounds for the heights of the hierarchies [51, 114].

It follows from the definition of the acceptance condition of non deterministic automata and from Rabin's Complementation Theorem that every regular set of trees is a $\mathbf{\Delta}^1_2$-set, (cf. [107, 104]). But there are only few known results on the complexity of non Borel regular tree languages. Simonnet [112] gives examples of $D_{\omega^n}(\mathbf{\Sigma}^1_1)$-complete regular tree languages. Arnold and Niwinski [10] show that the game tree languages form a infinite hierarchy of non Borel regular sets of trees with regard to the continuous (Wadge) reducibility. More recently, Finkel and Simonnet [56] investigate the topological complexity of non Borel recognizable tree languages with regard to the difference hierarchy of analytic sets. They prove for instance that for each integer $n > 0$, there is a $D_{\omega^n}(\mathbf{\Sigma}^1_1)$-complete tree language accepted by a non-deterministic Muller tree automaton, that a tree language recognized by an unambiguous Büchi tree automaton must be Borel, and that almost all game languages are not in any class $D_\eta(\mathbf{\Sigma}^1_1)$, for $\eta < \omega^\omega$.

When studying hierarchies, the first question is to know if the considered hierarchy is strict. However, another interesting question is to know whether

[4] In the same paper Murlak propose a procedure computing for a deterministic automaton an equivalent minimal index weak automaton.

the considered hierarchy also satisfies what is called the *separation property*, which roughly corresponds to the fact that given two disjoint sets on a certain level, they can be separated by a set from the lower level. In topology, it is well known that the separation property holds for the class of analytic sets, but fails for the class of co-analytic sets (c.f. [70]). In [11] Santocanale and Arnold thoroughly investigate the separation property within the μ-calculus and the index hierarchy. They show that separability fails in general starting from the third level. For the first level, a closer look at Rabin's original proof ([108]) of the fact that if a set of infinite trees can be defined both by an existential and by a universal MSO sentence then it can also be defined by a sentence of weak monadic second order logic, reveals that the (stronger) separation property holds for the class of Büchi recognizable languages, or equivalently for the second pi class in the fixpoint alternation hierarchy. The missing case of level 2 is solved by Hummel, Michalewski and Niwinski [61], by exhibiting two disjoint languages recognized by co-Büchi tree automata which cannot be separated by any Borel set. This result may be read as an evidence of a strong analogy between the class of Büchi recognizable tree languages and the class of analytic set. However in the same paper, the authors show that this analogy is not perfect. Indeed, whereas Rabin [108] already observed that all Büchi tree languages are definable by existential sentences of monadic second order logic, and therefore analytic, they verify that the converse is not true. This is done by exhibiting an analytic tree language, recognized by a parity tree automaton, but not by any Büchi automaton.

Acknowledgements

First and foremost, I want to thank my two supervisors, Jacques Duparc and Igor Walukiewicz. Throughout the last three years, they showed unwavering support and were always there when help was needed. Doing research under their supervision was both a great pleasure and a huge opportunity. I benefited greatly from their inspiring ideas, comments and suggestions, but also from the ways in which - scientifically - they complement one another.

Others who deserve a big and warm thanks are my friends and co-authors, with whom I spent time writing and doing research in joy, and sometimes a mild pain. I am immensely obliged to Luca Alberucci, Balder ten Cate and Filip Murlak. All chapters being the result of our fruitful collaborations, all the positive outcomes of this thesis are also collectively theirs. Mistakes and inaccuracies are only mine.

Because I was lucky enough to cross their paths during my academic life, I am very thankful to (in chronological order): Denis Miéville, Nadine Gessler, Pierre Joray, Arthur de Pury, Béatrice Godart, Cédric Degrange, Jean-Yves Béziau, Alexandre Costa Leite, Jérémie Cabessa, Ramon Jansana, Ignasi Jané, Stéphane Salaet, Miguel Angel Mota, Fabrice Correia, Christian W. Bach, Gerhard Jäger, Thomas Strahm, Giacomo Lenzi, Giovanna D'Agostino, Johan van Benthem, Yde Venema, Gaëlle Fontaine, Amélie Gheerbrant, Claire David, Raphael Carroy, Mikolaj Bojanczyk, Inanc Seylan and Phokion Kolaitis.

Last but not least, thanks to all my family and friends for their untiring support and love.

Financial support from the Swiss National Science Foundation (SNSF), grant number 100011- 116508 (Project "Topological Complexity, Games, Logic and Automata"), is gratefully acknowledged.

Chapter 1

Logics, Games and Automata

1.1 Trees and transition systems

In this section we introduce two kinds of structures widely used and studied in theoretical computer science and logic: trees and transition systems.

From the practical perspective of those disciplines, trees arise in several forms, for example as the parse trees obtained in the process of parsing a string of symbols according to a context-free grammar or as abstract models of discrete systems. On the other hand, it was advocated by many authors that transition systems provide a very good framework for describing for instance the behavior of programs. Both structures will provide semantics for the modal μ-calculus and some of its fragments studied in this work.

1.1.1 Trees and their undressing

Given a set, or alphabet, Δ, by Δ^* we denote the set of all finite words over Δ, by Δ^+ the set of nonempty sequences and Δ^ω denotes the set of all infinite words over Δ. The symbol ε stands for the empty word. The concatenation of a finite word $v \in \Delta^*$ with a possibly infinite word $w \in \Delta^* \cup \Delta^\omega$ is denoted by vw. This operation is naturally generalized for any sequence of finite words. We say that a finite word $v \in \Delta^*$ is a prefix of a word $w \in \Delta^* \cup \Delta^\omega$ if there is a word $u \in \Delta^* \cup \Delta^\omega$ such that $vu = w$. By Δ^n we denote the set of words over Δ of length n, and by $\Delta^{<n}$ we denote the set of all the words over Δ of length less than n. The set of all words over Δ of length at most n is then the set $\Delta^{\leq n} = \Delta^{<n} \cup \Delta^n$. Given $x \in \Delta^*$ and $n \in \mathbb{N}$, by $|x|$ we denote the length of the word x, and, if $|x| \geq n$, by $x[n]$ the prefix of x of length n. A set $S \subseteq \Delta^*$ is said to be *closed under prefixes* if for every $v \in S$, if w is a prefix of v, then $w \in S$.

A *tree* is any subset of Δ^* closed under the prefix relation. Given a tree $T \subseteq \Delta^*$, we call an element $x \in T$ a node. If x is a prefix of a node $y \in T$, then x is called an *ancestor* of y in T. A node that is not an ancestor of any other node is called a *leaf* of the tree. If $x, y \in T$ and there exists $w \in \Delta$ such that $x = yw$, we say that x is a *child* of y in T. In what follows we are interested in trees whose nodes have labels from a certain set Σ. Formally, a tree over Σ, or a Σ-tree, is any partial function $t : \Delta^* \to \Sigma$ such that $\mathsf{dom}(t)$ is a tree. By $t.x$ we denote the subtree of t rooted in $x \in \mathsf{dom}(t)$. A Σ-tree is called *regular* if, up to isomorphism, it has only finitely many subtrees. A branch in a tree t is a maximal chain for the prefix order in $\mathsf{dom}(t)$.

Those trees can have both infinite and finite branches. We call them *conciliatory.* A tree is called *full* if $\mathsf{dom}(t) = \Delta^*$. Thus, assuming that the underlying set Δ is known and fixed, by $T_\Sigma^{\mathfrak{c}}$ we denote the set of all conciliatory trees over Σ and by T_Σ the set of full trees over Σ. A conciliatory tree is said to be *well-founded* if it has no infinite branches. A Σ-tree t is called *finitely branching* if for every node $u \in \mathsf{dom}(t)$ the set of its children is finite.

Given a full Σ-tree t, by $t[n]$ we denote the Σ-tree given by restricting the domain of t to $\Delta^{\leq n}$. $t[n]$ is called the initial Σ-tree of height n of the tree t. If t' corresponds to a initial Σ-tree of height n of some full tree t, we just call t' an initial tree (of height n). Initial trees (of height n) are thus usually denoted by $t[n]$. In what follows, by $T_\Sigma^{(n)}$ we denote the the set of all initial Σ-trees of height n, and by $T_\Sigma^{(<\omega)}$ we denote the the set of all initial Σ-trees.

Given two sets L and T of conciliatory Σ-trees, the set $L \cdot T$ is the set of all trees obtained by replacing in a tree $t \in L$ each occurrence of a leaf by a tree from the set T. When L is a singleton $\{t\}$, we sometimes write $t \cdot T_\Sigma$ instead of $\{t\} \cdot T_\Sigma$. Finally, given a node $x \in \Delta^*$, and a language $L \in T_\Sigma$, by xL we denote the set of Σ-trees t such that $t.x \in L$.

When $\Delta = \{0, 1\}$ we call a tree *binary.* From now we always assume that Δ is either $\mathbb{N}$ or $\{0, 1\}$.

It will be useful to relate full and conciliatory trees one to another. Different perspectives can be adopted. Here we present a solution, given by Duparc and Murlak [51], that will be used when playing games with full trees. Another solution will be introduced and used in Chapter 4. The idea that will be pursue in this chapter is that given a full tree $t \in T_{\Sigma\cup\{s\}}$, we want to skip a node labeled by s and replace it with its leftmost child. That is, if we are in a node v such that $s = t(v)$, then we want to replace v with $v0$. But another problem is that we may encounter an infinite sequence of nodes labeled by s. This would keep us replacing the current node with its left child, and never get to a node not labeled with s. In that case, the undressing of t will simply not contain the node v. Formally, we have the following. Let $t \in T_{\Sigma\cup\{s\}}$ be a full tree, the *undressing* of t, denoted by $U(t)$, is the conciliatory tree defined as follows. Let x be the first node of t not labelled with s on the leftmost path of the tree (if there is no such node, then $U(t)$ is empty). Then for each $v \in \Delta^*$, consider two possibly infinite sequences:

- $w_0 = \varepsilon, v_0 = v$,
- for $v_i = bv'$, $w_{i+1} = w_i b, v_{i+1} = 0v'$, if $t.x(w_{i+1}) = s$ and $v_{i+1} = v'$ otherwise.

If $v_n = \varepsilon$ for some n, then $v \in \mathsf{dom}(U(t))$ and $U(t)(v) = t.x(w_n)$. Otherwise, $v \notin \mathsf{dom}(U(t))$. The undressing of conciliatory trees is defined analogously as for full trees[1].

[1] Formally, given a conciliatory $\Sigma \cup \{s\}$-tree t, it is enough to consider its full companion t^f over $\Sigma \cup \{s\}$ defined by the condition: if $x \in \mathsf{dom}(t)$, then $t^f(x) = t(x)$, otherwise $t^f(x) = s$. Thus, the undressing of t is defined as being the undressing of t^f.

A non-empty tree t over Σ can also be characterized as a relation structure over the signature $\langle <, (P_a : a \in \Sigma)\rangle$, where $<$ is a binary relation and all the P_a are unary relations:

$$\mathfrak{W}_t := \langle \mathsf{dom}(t), <^t, (P^t_a : a \in \Sigma)\rangle$$

with

- $<^t$ is the child relation: $<^t = \{(u, v) \in \mathsf{dom}(t) \times \mathsf{dom}(t) : v = wi, i \in \Delta\}$, and
- for all $a \in \Sigma$, $P^t_a = \{v \in \mathsf{dom}(t) : t(v) = a\}$.

1.1.2 Transition systems

A *transition system* $\mathcal{T}$ over a set Σ is of the form $(\mathsf{S}, \to^{\mathcal{T}}, \lambda^{\mathcal{T}})$ where S is a non empty set of *states* $\to^{\mathcal{T}}$ is a binary relation on S called the *accessibility relation* and the function $\lambda^{\mathcal{T}} : \mathsf{S} \to \Sigma$ is a labeling function, associating to every state s a label in Σ. When $\Sigma = \wp(A)$, with for example A being a set of propositional variables, it it useful to take $\lambda^{\mathcal{T}}$ as a *valuation* function from A into $\wp(\mathsf{S})$.

A transition system $\mathcal{T}$ with a distinguished state s is called a *pointed transition system* and denoted by $(\mathcal{T}, s)$. $\mathbb{T}(\Sigma)$ denotes the class of all pointed transition systems over Σ. Given any property P, by $\mathbb{T}^P(\Sigma)$ we denote the subclass of pointed transition systems over Σ satisfying the property P. In particular $\mathbb{T}^r(\Sigma)$ denotes all pointed reflexive transition systems, $\mathbb{T}^{st}$ all pointed symmetric and transitive transition systems over Σ, $\mathbb{T}^t(\Sigma)$ all pointed transitive transition systems and $\mathbb{T}^{rst}(\Sigma)$ denotes all pointed transition systems over Σ where the accessibility relation is an equivalence relation. Given any property P, with $\mathbb{T}^{Pf}(\Sigma)$ we denote the subclass of *finite* pointed transition systems over Σ satisfying the property P. For example, $\mathbb{T}^{tf}(\Sigma)$ denotes all *finite* pointed transition systems over Σ where the accessibility relation is transitive. When the set Σ is clear from the context, we omit to write the parameter Σ in all the previous notations.

Let $\mathcal{T} = (\mathsf{S}, \to^{\mathcal{T}}, \lambda)$ be a transition system and s, s' two states in S. A sequence $s_0, s_1, \ldots, s_n$ such that $s_i \to^{\mathcal{T}} s_{i+1}$, $s_0 = s$ and $s_n = s'$ is a *path of length* n connecting s to s'. We say that s' is *reachable* from s. A subset $\mathsf{S}' \subseteq \mathsf{S}$ of the set of states is called a *strongly connected component* if for all $s, s' \in \mathsf{S}'$ we have that s' is reachable from s. For each s by $\mathsf{scc}(s)$ we denote the greatest strongly connected component which contains s if there is one and $\mathsf{scc}(s) = \emptyset$ if s is not contained in any strongly connected component. Note, that the notion $\mathsf{scc}(s)$ is well-defined. Given a pointed transition system $(\mathcal{T}, s)$ and a state s' in it, we define the *depth of* s', $dp(s')$, to be the length of the shortest path from s to s'. Since parts which are non connected to the point s will be irrelevant in the sequel we assume that all transition system are connected and, therefore, that $dp(s')$ is defined for all s'.

Note that we can see a tree t over Σ as a pointed transition systems $(\mathcal{T}_t, \varepsilon)$ over the same set Σ by taking $\mathsf{S} = \mathsf{dom}(t)$, $\to^{\mathcal{T}_t} = <$ and $\lambda^{\mathcal{T}_t} = t$.

Clearly, as for trees, also a transition system $\mathcal{T}$ over Σ can be characterized as a relation structure over the signature $\langle <, (P_a : a \in \Sigma)\rangle$, where $<$ is a binary relation and all the P_a are unary relations:

$$\mathfrak{W}_{\mathcal{T}} := \langle \mathsf{S}, <^{\mathcal{T}}, (P^{\mathcal{T}}_a : a \in \Sigma)\rangle$$

with

- $<^{\mathcal{T}}$ corresponds to the accessibility relation $\rightarrow^{\mathsf{S}}$, and
- for every $a \in \Sigma$, $P_a^{\mathcal{T}} = \{s \in \mathsf{S} : \lambda(s) = a\}$.

1.2 Topology

Trees do not play only a central role in computer science but also in topology and descriptive set theory, thus providing a natural bridge between the two fields. In this section we introduce some basic notions and concepts that will be used in Chapter 4 and more extensively in the whole second part of this dissertation.

1.2.1 The Borel sets

A *topological space* is a set X together with a collection τ of subsets of X containing the empty set and X, and closed under arbitrary unions and finite intersections. The collection τ is usually called a *topology* on X and its elements are called the *open sets*. The complements of the elements in the topology are called *closed sets*. When the underlying topology is understood, we simply call X a topological space. A base B for a topology τ on X is a family of sets in τ such that every set in τ is an union of elements in B. A topological space X is called *second countable* if it has a countable base.

A *metric* on a set X is a function $d : X \times X \to \mathbb{R}$ such that for every $x, y, z \in X$:

(1) $d(x, y) = 0$ iff $x = y$

(2) $d(x, y) = d(y, x)$, and

(3) $d(x, z) \leq d(x, y) + d(y, z)$

A *metric space* is then a pair (X, d) where d is a metric on X. When the underlying metric is understood, we simply call X a metric space. Let (X, d) a metric space, $x \in X$ and $r > 0$. We put

$$B(x, r) = \{y \in X : d(x, y) < r\}$$

and call it the open ball with center x and radius r. Clearly the collection τ given by all subsets of X which are the union of a family of open balls of X is a topology on X. A metric space (X, d) is said to be *complete* if the limit of every sequence in X whose elements become arbitrarily close to each other with respect to d as the sequence progresses is still in X. A topological space whose topology is induced by a metric is called a *metrizable space*. It is called *completely metrizable* if its topology is induced by a complete metric. A metrizable space which is also second countable is called a *Polish space*.

Consider the class T_Σ of all full trees[2] over a finite set Σ. Let $d : T_\Sigma \times T_\Sigma \to \mathbb{R}$ the function defined by:

$$d(t, s) = \begin{cases} 0 & \text{if } t = s \\ 2^{-1-n} \text{ with } n = \min\{|w| : t(w) \neq s(w)\} & \text{otherwise.} \end{cases}$$

[2] The definition of the metric is the same either that T_Σ is the space obtained by taking $\Delta = \mathbb{N}$ or that it is the space obtained by taking $\Delta = \{0, 1\}$.

Clearly d is a complete metric on T_Σ.

REMARK 1.1. The open balls of the metric space T_Σ are given by fixing any initial tree over Σ and then taking all possible full trees extending it. This means that a subset L of T_Σ is an open sets if there is a set L' of initial Σ-trees such that $L = L' \cdot T_\Sigma$. This implies that if T_Σ is the space of all full *binary* trees, the space has a countable base and therefore is a Polish space.

Let (X, τ) be a topological space. The class of Borel sets $\mathbf{Borel}(X)$ is the smallest collection of subsets of X that contains the open sets and is closed under the set-theoretical operations of countable unions and complementation.

Given two topological spaces X and Y, a function $f : X \to Y$ is said to be continuous if for every open set $O \subseteq Y$, $f^{-1}(O) \subseteq X$ is also open. Here, by $f^{-1}(O)$, we mean the set $\{x \in X : f(x) \in O\}$. The next well known result states that the class of all Borel sets is closed by continuous pre-images (see [87]):

PROPOSITION 1.2. *Let X, Y be two topological spaces and $f : X \to Y$ a continuous function. Then for every $B \subseteq Y$, if $B \in \mathbf{Borel}(Y)$ then $f^{-1}(B) \in \mathbf{Borel}(X)$.*

Note that by the previous proposition, if we have a subset B of a topological space Y and a subset A of a topological space X which is known for not being Borel, together with a continuous function $f : X \to Y$ such that $f^{-1}(B) = A$, we can conclude that B is also not Borel.

The next lemma gives an example of a non Borel set of full binary trees.

LEMMA 1.3 ([101]). *The set W_a of all full binary trees over $\{a, b\}$ where in every branch there are only finitely many nodes with label a is not Borel.*

The class of Borel sets $\mathbf{Borel}(X)$ of a topological space X can naturally be spread in a hierarchy of length ω_1, called the *Borel hierarchy*. More precisely, for every space X we recursively define the classes $\mathbf{\Sigma}^0_\xi(X)$, $\mathbf{\Pi}^0_\xi(X)$ and $\mathbf{\Delta}^0_\xi(X)$ as follows:

- $L \in \mathbf{\Sigma}^0_1(X)$ iff L is an open set of X,
- if $1 \leq \xi < \omega_1$, $L \in \mathbf{\Pi}^0_\xi(X)$ iff $L^{\complement} \in \mathbf{\Sigma}^0_\xi(X)$,
- if $2 \leq \xi < \omega_1$, $L \in \mathbf{\Sigma}^0_\xi(X)$ iff there is a sequence $\langle L_n : n \in \omega \rangle$ of elements of $\bigcup_{\eta \in \xi} \mathbf{\Pi}^0_\eta(X)$ such that $L = \bigcup_{n \in \omega} L_n$,
- if $1 \leq \xi < \omega_1$, $L \in \mathbf{\Delta}^0_\xi(X)$ iff $L \in \mathbf{\Sigma}^0_\xi(X)$ and $L \in \mathbf{\Pi}^0_\xi(X)$.

where for every subset $L \subseteq X$, $L^{\complement} = \{t \in X : t \notin L\}$.

By convention, $\mathbf{\Pi}^0_0(X) = \{X\}$ and $\mathbf{\Sigma}^0_0(X) = \emptyset$. Clearly $\mathbf{Borel}(X) = \bigcup_{\xi < \omega_1} \mathbf{\Sigma}^0_\xi(X) = \bigcup_{\xi < \omega_1} \mathbf{\Pi}^0_\xi(X)$. It is well known that (see [87]):

THEOREM 1.4. *For X a topological space:*

(1) $\mathbf{\Pi}^0_\xi(X) \subseteq \mathbf{\Sigma}^0_{\xi+1}(X)$,

(2) $\mathbf{\Sigma}^0_\xi(X) \subseteq \mathbf{\Pi}^0_{\xi+1}(X)$, *and*

(3) if $\mathbf{\Pi}^0_1(X) \subseteq \mathbf{\Pi}^0_2(X)$*, then*

(a) $\mathbf{\Pi}^0_\xi(X) \subseteq \mathbf{\Pi}^0_{\xi+1}(X)$*,*

(b) $\mathbf{\Sigma}^0_\xi(X) \subseteq \mathbf{\Sigma}^0_{\xi+1}(X)$*, and*

(c) $\mathbf{\Pi}^0_\xi(X) \cup \mathbf{\Sigma}^0_\xi(X) \subseteq \mathbf{\Delta}^0_{\xi+1}(X)$*.*

The condition $\mathbf{\Pi}^0_1(X) \subseteq \mathbf{\Pi}^0_2(X)$ always holds in a metric space X, since

$$C = \bigcap_{n\in\omega} \{x : \exists y \in C, d(x,y) < (n+1)^{-1}\}$$

for C a closed set.

Whenever the space X is determined by the context, we omit it in the above notations and simply write $\mathbf{\Sigma}^0_n$, $\mathbf{\Pi}^0_n$, and so on.

$$\begin{tikzcd}
\mathbf{\Sigma}^0_0(X) \arrow[rr] \arrow[dr] & & \mathbf{\Sigma}^0_1(X) \arrow[rr] \arrow[dr] & & \mathbf{\Sigma}^0_2(X) \arrow[rr] \arrow[dr] & & \mathbf{\Sigma}^0_3(X)\cdots \\
& \mathbf{\Delta}^0_1(X) \arrow[ur] \arrow[dr] & & \mathbf{\Delta}^0_2(X) \arrow[ur] \arrow[dr] & & \mathbf{\Delta}^0_3(X) & \cdots \\
\mathbf{\Pi}^0_0(X) \arrow[rr] \arrow[ur] & & \mathbf{\Pi}^0_1(X) \arrow[rr] \arrow[ur] & & \mathbf{\Pi}^0_2(X) \arrow[rr] \arrow[ur] & & \mathbf{\Pi}^0_3(X)\cdots
\end{tikzcd}$$

Figure 1.1: The first levels of the Borel hierarchy in the case $\mathbf{\Pi}^0_1(X) \subseteq \mathbf{\Pi}^0_2(X)$. Arrows stand for set-theoretic inclusion.

1.2.2 The Wadge game

Let X and Y be two topological spaces, and $L \subseteq X$ and $M \subseteq Y$. We say that L *continuously reduces* or *Wadge reduces* to M whenever there is a continuous function $f : X \to Y$ such that $f^{-1}(M) = L$. If this is the case, "topologically" this means that L is less complicated than M and we write $L \leq_W M$. If $L \leq_W M$, but $M \not\leq_W L$, then L is strictly less complicated than M, and we write $L <_W M$. Whenever $L \leq_W M$ and $M \leq_W L$, we say that L and M are *Wadge-equivalent*, and write $L \equiv_W M$. The sets L and M are called *incomparable* when both $L \not\leq_W M$ and $M \not\leq_W L$ hold. In addition, a set L is called *self-dual* if $L \equiv_W L^{\complement}$, and *non self-dual* otherwise.

For us, the crucial observation is that, when considering full (binary or not) trees, the relation $\leq_W$ can be characterized by a two-players infinite game with perfect information, called the *Wadge game* ([124, 125]). This game is defined as follows. Let $L \subseteq T_\Sigma$ and $M \subseteq T_{\Sigma'}$ be two sets of full trees. The Wadge game $\mathcal{G}_W((L, T_\Sigma), (M, T_{\Sigma'}))$ is played by two players, Spoiler who is in charge of L, and Duplicator, who is in charge of the set M. At every round, Spoiler plays first, and both players add one more complete layer of nodes to the tree they constructed in the previous round. Duplicator is allowed to skip her turn whenever she wishes, postponing the choice of the next layer in his tree. However, she may not do so infinitely many times in a row (otherwise the tree constructed in the limit will not be full). Spoiler is not allowed to do so. Formally, skipping means adding an entire layer of nodes labelled by a "skip" label $s \notin \Sigma'$ to the tree, and playing a full tree means, for Duplicator, playing a full tree whose undressing is still a full tree. At the limit, Spoiler and Duplicator have respectively produced two full trees $t_I \in T_\Sigma$ and $t_{II} \in T_{\Sigma'}$, where formally t_{II} is given by the undressing with respect to s of the tree produced by Duplicator at the end of the game. Duplicator wins the game $\mathcal{G}_W((L, T_\Sigma), (M, T_{\Sigma'}))$ iff the condition $(t_I \in L \Leftrightarrow t_{II} \in M)$ holds. From this point onward, the Wadge game $\mathcal{G}_W((L, T_\Sigma), (M, T_{\Sigma'}))$ will be denoted by $\mathcal{G}_W(L, M)$ and the underlying spaces and alphabets involved will always be assumed to be known from the context.

Along the play, the finite sequence of the previous moves of a given player is called the current position of this player. A strategy for Spoiler is a mapping $\sigma : T^{(<\omega)}_{\Sigma' \cup \{s\}} \to T^{(<\omega)}_{\Sigma}$ such that if $t \in T^{(n)}_{\Sigma' \cup \{s\}}$ then $\sigma(t) \in T^{(n+1)}_{\Sigma}$ with the condition that if $\sigma(t[n]) = t'[n+1]$ and $\sigma(t[n+1]) = t''[n+2]$, then $t'[n+1] = t''[n+1]$, whereas a strategy for Duplicator is a mapping $\sigma : T^{(<\omega)}_{\Sigma} \setminus \emptyset \to T^{(<\omega)}_{\Sigma' \cup \{s\}}$ such that if $t \in T^{(n)}_{\Sigma' \cup \{s\}}$ then $\sigma(t) \in T^{(n)}_{\Sigma}$, with the analogous monotonicity condition stating that if $\sigma(t[n]) = t'[n]$ and $\sigma(t[n+1]) = t''[n+1]$, then $t'[n] = t''[n]$. A strategy is winning if the player following it must win, no matter what his opponent plays.

The Wadge game was designed precisely in order to obtain that the Wadge reduction coincides with the existence of a winning strategy for Duplicator in a Wadge game:

Lemma 1.5 (Wadge). *Let $L \subseteq T_\Sigma$ and $M \subseteq T_{\Sigma'}$ be two sets of full trees. Then $L \leq_W M$ iff Duplicator has a winning strategy in the game $\mathcal{G}_W(L, M)$.*

Proof. Suppose Duplicator has a winning strategy σ in $\mathcal{G}_W(L, M)$. This strategy naturally induces the function σ from $T^{(<\omega)}_{\Sigma} \cup T_\Sigma$ into $T^{(<\omega)}_{\Sigma'} \cup T_{\Sigma'}$. We prove that the restriction of this function on full trees is continuous. Let $V \cdot T_{\Sigma'}$ be an open set of $T_{\Sigma'}$, for some set V of initial Σ'-trees. Then $\sigma^{-1}(V \cdot T_{\Sigma'}) = \sigma^{-1}(V) \cdot T_\Sigma$, showing that the pre-image of any open set is an open set. Moreover, the winning condition of the Wadge game states that $t \in M$ if and only if $\sigma^{-1}(t) \in L$, and therefore $\sigma^{-1}(M) = L$.

Conversely, suppose that $L \leq_W M$. Therefore there is a continuous function $f : T_\Sigma \to T_{\Sigma'}$ such that $f^{-1}(M) = L$. We describe a winning strategy for Duplicator in the game $\mathcal{G}_W(L, M)$. For that purpose, consider an enumeration $\{b_0, \ldots, b_n\}$ of the elements of Σ'. Since f is continuous, the sets $A_i = f^{-1}(b_i \cdot T_{\Sigma'})$ form a partition of T_Σ in open sets. Therefore, as long as Spoiler's play does not enter any of the sets A_i, Duplicator skips her turn. As soon as Spoiler's play enters a set A_{i_0}, for some $i_0 \leq n$, Duplicator plays on the leftmost path a node (root of a tree, with respect to the future undressing) labelled by b_0. Notice that since the sets A_i form a partition of T_Σ, Spoiler's play is forced to enter one of those sets after a finite amount of time. Then, consider an enumeration $\{t_0[1], \ldots, t_n[1], \ldots\}$ of all Σ'-trees of height 1 whose root is labelled by b_{i_0} and the associated sets $A'_i = f^{-1}(t_i[1] \cdot T_{\Sigma'})$, and proceed the same way. As long as Spoiler's play does not enter any of the sets A'_i, Duplicator skips her turn. As soon as Spoiler's play enters an A'_{i_1} , Duplicator plays in such a way that the undressing of her moves is $t_{i_1}[1]$. And so on and so forth. At the end of the play, the full trees t_I and t_{II} played respectively by Spoiler and Duplicator satisfy $f(t_I) = t_{II}$. Moreover, since $f^{-1}(M) = L$, the relation $(t_I \in L \Leftrightarrow t_{II} \in M)$ holds. This strategy is therefore winning for Duplicator in the game $\mathcal{G}_W(L, M)$. $\square$

The previous lemma is a very powerful result and will extensively be used all alongs this dissertation without explicitly referring to it.

We now states some other properties of the Wadge reduction especially useful for Part 2.

PROPOSITION 1.6. *The Wadge reduction is a pre-order.*

Proof. Reflexivity is clear since the copy-cat strategy is winning for Duplicator in $\mathcal{G}_W(L, L)$. For the transitivity, let L, M and N three languages of full trees such that $L \leq_W M, M \leq_W N$. Then Duplicator has two winning strategies σ_1 and σ_2 in the respective games $\mathcal{G}_W(L, M)$ and $\mathcal{G}_W(M, N)$. The composition $\sigma_2 \circ \sigma_1$ of these two strategies is a winning strategy for Duplicator in the game $\mathcal{G}_W(L, N)$, and therefore $L \leq_W N$. $\square$

Since $\leq_W$ is a pre-order, the Wadge-equivalence relation $\equiv_W$ is really an equivalence relation. Note that clearly, for every alphabet Σ, T_Σ and $\emptyset$ are incomparable and moreover any other subset $L \subsetneq T_\Sigma$ Wadge-reduces them. Some others easy properties of the Wadge-reduction are the following:

PROPOSITION 1.7. *Let $L \subseteq T_\Sigma$ and $M \subseteq T_{\Sigma'}$, then the following properties hold:*

(1) $L \leq_W M$ iff $L^{\complement} \leq_W M^{\complement}$,

(2) L and its complement $L^{\complement}$ are either Wadge-equivalent or incomparable,

(3) If $L <_W M$, then $M \nleq_W L$ and $M^{\complement} \nleq_W L$

Proof. The first point follows from the definition of the winning conditions of the Wadge game and the fact that $t \in L \Leftrightarrow t' \in M$ iff $t \in L^{\complement} \Leftrightarrow t' \in M^{\complement}$. Point two follows directly from the previous point, and point three from point two and the fact that the Wadge-reduction relation is a pre-order. $\square$

In the preliminaries of the second part of this dissertation we will discuss more deeply some nice and useful properties of the Wadge reduction on languages of full binary trees.

1.3 Monadic second order logics

Trees and transitions systems as relational structures can be described in monadic second order logic with the child relation. In addition to that logic, in this section we also introduce weak monadic second order logic, which is obtained by restricting the range of second order quantifiers to finite sets.

1.3.1 Full MSO

Let $\mathsf{Var}_1 = \{x, y, \dots\}$ be the set of first order variables, and $\mathsf{Var}_2 = \{X, Y, \dots\}$ the set of second order variables. Given a finite set Σ, the set of monadic second order (MSO) formulae over Σ is defined by the following grammar:

$$\phi ::= < (x, y) \mid x = y \mid P_a(x) \mid X(x) \mid \neg\phi \mid \phi \wedge \phi \mid \exists x \phi \mid \exists X \phi$$

for every $a \in \Sigma$, $x, y \in \mathsf{Var}_1$ and $X \in \mathsf{Var}_2$.

As usual, we introduce implication $\varphi \to \psi$ as $\neg\varphi \vee \psi$, disjunction as $\neg(\neg\varphi \wedge \neg\psi)$, and the universal first order and second order quantifiers as $\forall x\phi \equiv \neg\exists x \neg\phi$ and $\forall X\phi \equiv \neg\exists X \neg\phi$ respectively. A formula ϕ is called a sentence if it has no free variables.

We start with the case of trees. A valuation λ is given by a pair of functions

$$\begin{cases} \lambda_1 : \mathsf{Var}_1 \to W^* \\ \lambda_2 : \mathsf{Var}_2 \to \wp(W^*) \end{cases}$$

Let λ be a valuation, x a first order variable and v an element of W^*. The associate valuation $\lambda[x \mapsto v]$ is defined for all first order variables x' as follows:

$$\lambda[x \mapsto v](x') = \begin{cases} v & \text{if } x' = x, \\ \lambda(x') & \text{otherwise.} \end{cases}$$

Analogously for second order variables and pairs of sequences of first and second order variables.

We say that a valuation is consistent for a non-empty tree t if $\lambda_1(\mathsf{Var}_1) \subseteq \mathsf{dom}(t)$ and $\lambda_1(\mathsf{Var}_2) \subseteq \wp(\mathsf{dom}(t))$. Thus, a consistent valuation for a tree t assignes to every first-order variable an element of the domain and to each second order variable *any* set of elements of the domain.

The meaning of a MSO formula ϕ in a structure $\mathfrak{W}_t$, with t non-empty, and a consistent valuation λ for t is then defined recursively as follows:

- If ϕ is $< (x, y)$, then $\mathfrak{W}_t, \lambda \vDash \phi$ iff $<^t (\lambda(x), \lambda(y))$
- If ϕ is $x = y$, then $\mathfrak{W}_t, \lambda \vDash \phi$ iff $\lambda(x) = \lambda(y)$
- If ϕ is $P_a(x)$, then $\mathfrak{W}_t, \lambda \vDash \phi$ iff $\lambda(x) \in P_a^t$
- If ϕ is $X(x)$, then $\mathfrak{W}_t, \lambda \vDash \phi$ iff $\lambda(x) \in \lambda(X)$
- If ϕ is $\neg\psi$, then $\mathfrak{W}_t, \lambda \vDash \phi$ iff $\mathfrak{W}_t, \lambda \nvDash \psi$
- If ϕ is $\psi \wedge \gamma$, then $\mathfrak{W}_t, \lambda \vDash \phi$ iff $\mathfrak{W}_t, \lambda \vDash \psi$ and $\mathfrak{W}_t, \lambda \vDash \gamma$
- If ϕ is $\exists x\psi$, then $\mathfrak{W}_t, \lambda \vDash \phi$ iff there exists $v \in W^*$ such that $\mathfrak{W}_t, \lambda[x \mapsto v] \vDash \psi$
- If ϕ is $\exists X\psi$, then $\mathfrak{W}_t, \lambda \vDash \phi$ iff there exists a subset $V \subseteq W^*$ such that $\mathfrak{W}_t, \lambda[X \mapsto V] \vDash \psi$

We say that a formula ϕ is true in a structure $\mathfrak{W}_t$ if there is a valuation λ such that $\mathfrak{W}_t, \lambda \vDash \phi$.

Given a MSO sentence ϕ over the alphabet Σ, the tree language defined by ϕ is the set of non empty trees:

$$L(\phi) = \{t \in T^{\mathsf{c}}_{\Sigma} \setminus \{\emptyset\} : \exists \lambda \text{ such that } \mathfrak{W}_t, \lambda \vDash \phi\}$$

We say that a tree language L is MSO-definable if there is an MSO-formula ϕ such that $L = L(\phi)$.

The case of transition systems is treated exactly in the same way as the case of trees.

Given a MSO sentence ϕ over the alphabet Σ, the language of transition systems defined by ϕ is the set:

$$L(\phi) = \{\mathcal{T} \in \mathbb{T}(\Sigma) : \exists\lambda \text{ such that } \mathfrak{W}_{\mathcal{T}}, \lambda \vDash \phi\}$$

1.3.2 Weak MSO

Weak monadic second order logic differs from (full) monadic second order logic in the fact that it allows to quantify not over all subsets of the domain of the considered structure, but only on its *finite* subsets.

As before, let $\mathsf{Var}_1 = \{x, y, \dots\}$ be the set of first order variables, and $\mathsf{Var}_2 = \{X, Y, \dots\}$ the set of second order variables. Given a finite set Σ, the set of weak monadic second order (WMSO) formulae over Σ is defined by the following grammar:

$$\phi ::= < (x, y) \mid x = y \mid P_a(x) \mid X(x) \mid \neg\phi \mid \phi \wedge \phi \mid \exists x\phi \mid \exists^f X\phi$$

for every $a \in \Sigma$, $x, y \in \mathsf{Var}_1$ and $X \in \mathsf{Var}_2$. The concept of a sentence and of implication, of disjunction, of universal first order quantifier and *weak* universal second order quantifier are defined as expected.

As before we start by the case of trees. A weak valuation λ is given by a pair of functions

$$\begin{cases} \lambda_1 : \mathsf{Var}_1 \to W^* \\ \lambda_2 : \mathsf{Var}_2 \to \wp_{\mathsf{fin}}(W^*) \end{cases}$$

We say that a weak valuation is consistent for a non-empty tree t if $\lambda_1(\mathsf{Var}_1) \subseteq \mathsf{dom}(t)$ and $\lambda_1(\mathsf{Var}_2) \subseteq \wp_{\mathsf{fin}}(\mathsf{dom}(t))$. Thus, a consistent weak valuation for a tree t assignes to every first-order variable an element of the domain but to each second order variable a *finite* set of elements of the domain.

The meaning of a formula ϕ in a structure $\mathfrak{W}_t$, with t non-empty, and a consistent weak valuation λ for t is then defined recursively as before, with the new second order quantifier clause:

- If ϕ is $\exists^f X\psi$, then $\mathfrak{W}_t, \lambda \vDash \phi$ iff there exists a *finite* subset $V \subseteq W^*$ such that $\mathfrak{W}_t, \lambda[X \mapsto V] \vDash \psi$

We say that a formula ϕ is true in a structure $\mathfrak{W}_t$ if there is a weak valuation λ such that $\mathfrak{W}_t, \lambda \vDash \phi$.

Given a WMSO sentence ϕ over the alphabet Σ, the tree language defined by ϕ is the set of non empty trees:

$$L(\phi) = \{t \in T_\Sigma^{\mathsf{c}} \setminus \{\emptyset\} : \exists\lambda \text{ such that } \mathfrak{W}_t, \lambda \vDash \phi\}$$

We say that a tree language L is WMSO-definable if there is an WMSO-formula ϕ such that $L = L(\phi)$.

The case of transition systems is treated exactly in the same way as the case of trees.

Given a WMSO sentence ϕ over the alphabet Σ, the language of transition systems defined by ϕ is the set:

$$L(\phi) = \{\mathcal{T} \in \mathbb{T}(\Sigma) : \exists\lambda \text{ such that } \mathfrak{W}_{\mathcal{T}}, \lambda \vDash \phi\}$$

REMARK 1.8. The adjective "weak" is a bit misleading, since WMSO is in general not a fragment of MSO. Indeed, the class of finitely branching trees is not definable in MSO (because, as we will see in Chapter 4, every MSO-formula that is satisfiable on trees is true of some finitely branching tree) but is defined by the WMSO-formula $\forall x \exists^f X \forall y(x <_{im} y \to y \in X)$, where $x <_{im} y$ is shorthand for $x < y \wedge \neg\exists z(x < z \wedge z < y)$. The class of well-founded trees is definable in MSO but not in WMSO, as will follow from results we discuss below (in particular, the from the fact that the class of well-founded trees is not Borel). On finitely branching trees, WMSO *is* strictly less expressive than MSO. This follows from the fact that on finitely branching trees "X is a finite set" is expressed by the MSO-formula $\forall Y(\forall x(Yx \to Xx) \wedge \forall y(Yy \to \exists z(Yz \wedge y < z)) \to \neg\exists y(Yy))$ ("X does not contain an infinite path").

1.4 The modal μ-calculus

In this section we introduce the syntax and the semantics of the main object of study of this dissertation, the modal μ-calculus. After briefly introducing a variant of this logic, suitable for describing full binary trees, we discuss the relation between those formalisms and monadic second order logic.

1.4.1 Syntax

The *language of the modal μ-calculus*, $\mathcal{L}_\mu$, results by adding greatest and least fixpoint operators to propositional modal logic. More precisely, given a set Prop of *propositional variables*, the collection $\mathcal{L}_\mu$ of *modal μ-formulae* (or simply μ-formulae) is defined as follows:

$$\varphi ::= p \mid\ \sim p \mid \top \mid \bot \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \Diamond\varphi \mid \Box\varphi \mid \mu x.\varphi \mid \nu x.\varphi$$

where $p, x \in \mathsf{Prop}$ and x occurs only positively in $\eta x.\varphi$ ($\eta = \nu, \mu$), that is, $\sim x$ is not a subformula of φ. $\mathcal{L}_{\mathsf{M}}$ denotes the pure modal fragment of $\mathcal{L}_\mu$.

The fixpoint operators μ and ν can be viewed as quantifiers. Therefore we use the standard terminology and notations as for quantifiers and, for instance, $\mathsf{free}(\varphi)$ denotes the set of all propositional variables occurring free in φ and $\mathsf{bound}(\varphi)$ those occurring bound. Further, we define $\mathsf{var}(\varphi) = \mathsf{free}(\varphi) \cup \mathsf{bound}(\varphi)$. If ψ is a subformula of φ, we write $\psi \leq \varphi$. We write $\psi < \varphi$ when ψ is a proper subformula. $\mathsf{sub}(\varphi)$ is the set of all subformulae of φ.

Let $\varphi(x)$ and ψ be two μ-formulae, where x occurs only positively in φ. The substitution of all occurrences of x with ψ in φ is denoted by $\varphi[x/\psi]$ or sometimes simply $\varphi(\psi)$. Simultaneous substitution of all x_i by ψ_i ($i \in \{1, \ldots, n\}$) is denoted by $\varphi[x_1/\psi_1, \ldots, x_n/\psi_n]$. For serial substitution such as $(\varphi[x_1/\psi_1])[x_2/\psi_2]$ we often omit the parentheses and write $\varphi[x_1/\psi_1][x_2/\psi_2]$.

REMARK 1.9. Note, that if $\varphi(x), \psi \in \mathcal{L}_\mu$ then $\varphi[x/\psi]$ need not be a μ-formula, for example, if we set $\varphi \equiv \mu y.x$ and $\psi \equiv \sim y$ then we have $\varphi[x/\psi] \equiv \mu y. \sim y \notin \mathcal{L}_\mu$. Nevertheless, in this paper, if nothing else mentioned, an expression like $\varphi[x/\psi]$ will always denote a well defined μ-formula. For a formal introduction of substitution we refer to Alberucci [2].

The negation $\neg\varphi$ of a μ-formula φ is defined inductively such that $\neg p \equiv \sim p$ and $\neg(\sim p) \equiv p$, by using de Morgan dualities for boolean connectives and the usual modal dualities for $\Diamond$ and $\Box$. For μ, ν we define

$$\neg \mu x.\varphi(x) \equiv \nu x.\neg\varphi(x)[x/\neg x] \quad \text{and} \quad \neg \nu x.\varphi(x) \equiv \mu x.\neg\varphi(x)[x/\neg x].$$

As usual, we introduce implication $\varphi \to \psi$ as $\neg\varphi \vee \psi$ and equivalence $\varphi \leftrightarrow \psi$ as $(\varphi \to \psi) \wedge (\varphi \to \psi)$.

The *fixpoint alternation depth*, ad, of a formula is the number of non-trivial nestings of alternating least and greatest fixpoints. Formally, it is defined as follows.

Definition 1.10. *Let φ be a μ-formula. An* alternating μ-chain *in φ of length k is a sequence*

$$\varphi \geq \mu x_0.\psi_0 > \nu x_1.\psi_1 > \cdots > \mu/\nu x_{k-1}.\psi_{k-1}$$

where for every $i < k-1$ the variable x_i is free in every ψ such that $\psi_i \geq \psi \geq \psi_{i+1}$. The maximum length of an alternating μ-chain in φ is denoted by $\max^\mu(\varphi)$. ν-chains and $\max^\nu(\varphi)$ are defined analogously. The alternation depth *of a μ-formula φ, denoted by $\mathsf{ad}(\varphi)$, is the maximum of $\max^\mu(\varphi)$ and $\max^\nu(\varphi)$. If φ is a purely modal formula, we set $\mathsf{ad}(\varphi) = 0$.*

Given a μ-formula φ, for all set of bound variables $X \subseteq \mathsf{bound}(\varphi)$, the formula $\varphi^{\mathsf{free}(X)}$ is obtained from φ by eliminating all fixpoint operators binding a variable $x \in X$ and leaving the previously bound variables x as free occurrences. Further, if $X = \{x_i, \ldots, x_n\} \subseteq \mathsf{bound}(\varphi)$ then we define

$$\varphi^{-X} \equiv \varphi^{\mathsf{free}(X)}[x_1/\bot, \ldots, x_n/\bot].$$

Example 1.11. Let φ be the formula $\mu x.(\Diamond x \vee (\nu y.(\Box y \wedge \mu z.p \wedge \Diamond z)))$, and let $X = \{x, z\} \subseteq \mathsf{bound}(\varphi)$. Then $\varphi^{\mathsf{free}(X)}$ is the formula $\Diamond x \vee (\nu y.(\Box y \wedge (p \wedge \Diamond z)))$, and φ^{-X} is the formula $\Diamond\bot \vee (\nu y.(\Box y \wedge (p \wedge \Diamond\bot)))$

We say that a variable $x \in \mathsf{bound}(\varphi)$ is *well-bounded* in φ if no two distinct occurrences of fixpoint operators in φ bind x, and x occurs only once in $\varphi^{\mathsf{free}(\{x\})}$. A propositional variable p is guarded in a formula $\varphi \in \mathcal{L}_\mu$ if every occurrence of p in φ is in the scope of a modal operator. A formula φ of $\mathcal{L}_\mu$ is said to be *guarded* if and only if for every subformula of φ of the form $\eta x.\delta$, x is guarded in δ. The next definition is very important.

Definition 1.12. *A formula φ of $\mathcal{L}_\mu$ is said to be* well-named *if it is guarded and every $x \in \mathsf{bound}(\varphi)$ is well-bounded in φ.*

Notice that for all well-named[3] φ, if x is bound in φ then there is exactly one subformula $\eta x.\delta \leq \varphi$ which bounds x; this formula is denoted by φ_x.

Example 1.13. The formulae $\Box x \wedge \mu x.(\Diamond x \vee p)$ and $\mu x.(x \vee \Box y)$ are both non well-named. This is because on the one hand the variable x occurs twice in $\Box x \wedge (\Diamond x \vee p)$ and on the other hand x is not guarded in $\mu x.(x \vee \Box y)$. On the contrary, the formulae $\mu x.(\Diamond x \vee \nu y.\Box y)$ and $\Box y \wedge \mu x.(\Diamond x \vee p)$ are well-named.

[3]In the subsection of the semantics, we will see that any μ-formula φ is equivalent to a well-named formula $\mathsf{wn}(\varphi)$.

If $x \in \mathsf{bound}(\varphi)$ and x is in the scope of a $\Diamond$ operator in φ_x, resp. $\Box$ operator, then we say that x is *weakly existential in* φ, resp. *weakly universal in* φ. If $x \in \mathsf{bound}(\varphi)$ and x is in the scope only of $\Diamond$ operators in φ_x, resp. $\Box$ operators, then we say that x is *existential in* φ, resp. *universal in* φ. Let $\varphi(x)$ be a μ-formula. If x is free and occurs only positively in φ, then we define $\varphi^n(x)$ for all n inductively such that $\varphi^1(x) = \varphi(x)$ and such that

$$\varphi^{k+1}(x) \equiv \varphi[x/\varphi^k(x)].$$

We define $\varphi^n(\top) = \varphi^n[x/\top]$, and analogously for $\varphi^n(\bot)$.

In order to make proofs by induction on the structure of a formula, we now introduce a measure for the syntactical complexity of formulae of the modal μ-calculus. This is done by the so called *rank function*, which assigns to each formula a well-chosen ordinal number.

DEFINITION 1.14. *The* rank*,* $\mathsf{rank}(\varphi)$*, of a formula* φ *is an ordinal number defined inductively as follows:*

- $\mathsf{rank}(p) = \mathsf{rank}(\sim p) = 1$
- $\mathsf{rank}(\triangle\ \alpha) = \mathsf{rank}(\alpha) + 1$ *where* $\triangle \in \{\Box, \Diamond\}$
- $\mathsf{rank}(\alpha \circ \beta) = \mathsf{max}\{\mathsf{rank}(\alpha), \mathsf{rank}(\beta)\} + 1$ *where* $\circ \in \{\wedge, \vee\}$
- $\mathsf{rank}(\eta x.\alpha) = \mathsf{sup}\{\mathsf{rank}(\alpha^n(x)) + 1\ ;\ n \in \mathbb{N}\}$ *where* $\eta \in \{\nu, \mu\}$*.*

The fact that the definition of rank, originally introduced in [3], terminates is shown in [5] (see also [2]). It is an easy exercise to show that for all formulae φ we have that $\mathsf{rank}(\varphi) = \mathsf{rank}(\neg\varphi)$.

EXAMPLE 1.15.

(1) The rank of the formula $\Diamond(\Box p \vee \Diamond\Box q)$ is 5. In general, the rank of any modal formula is a finite ordinal.

(2) Let $\varphi_1 = \mu x.(\Diamond x \vee p)$. The rank of φ_1 is ω, while the rank of $\Diamond\varphi_1$ is $\omega + 1$.

(3) Let $\varphi_2 = \mu x.\mu y.((\Diamond x \vee p) \wedge y)$. The rank of φ_2 is ω^2.

(4) Let $\varphi_3 = \nu z.\mu w.(z \wedge (w \wedge \varphi_2))$. The rank of φ_3 is $\omega^2 \cdot 2$.

(5) Let $\varphi_4 = p \wedge (\top \vee \nu x.(\Diamond x \vee \varphi_3)$. The rank of φ_4 is $\omega^2 \cdot 2 + \omega + 2$.

REMARK 1.16. In [5], Alberucci and Krähenbühl are able to show that for every μ-formula φ, $\mathsf{rank}(\varphi) < \omega^\omega$ and that for every ordinal $\xi < \omega^\omega$ there is a formula whose rank is exactly ξ. Moreover they provide an algorithm to compute the rank of any formula by primitive recursion.

The axioms and inference rules below define the deduction system Koz, introduced by Kozen in [73]. As usual we write $\mathsf{Koz} \vdash \varphi$ if there is a derivation of φ in the system presented below.

Axioms: All classical propositional tautologies, the *Distribution Axiom* from modal logic

$$(\Box(\alpha \to \beta) \wedge \Box\alpha) \to \Box\beta$$

and the *Fixpoint Axioms*

$$\eta x.\varphi(x) \leftrightarrow \varphi(\eta x.\varphi(x)), \quad \eta \in \{\mu, \nu\}.$$

Inference Rules: Beside the classical *Modus Ponens*

$$\frac{\alpha \quad \alpha \to \beta}{\beta}$$

and the *Necessitation Rule*

$$\frac{\alpha}{\Box\alpha},$$

we have the *Induction Rule*

$$\frac{\varphi \to \alpha[x/\varphi]}{\varphi \to \nu x.\alpha}.$$

It can be shown that the fixpoint axioms can be replaced by the following weaker axioms:

$$\nu x.\varphi(x) \to \varphi(\nu x.\varphi(x)) \quad \text{and} \quad \varphi(\mu x.\varphi(x)) \to \mu x.\varphi(x).$$

1.4.2 Semantics

The semantics of modal μ-calculus is given by trees or transition systems over $\wp(\mathsf{Prop})$. Since the semantics given by trees can be seen, mutatis mutandis, as a special case of the semantics given by transition systems, we only present this last one.

Let $(\mathsf{S}, \to^{\mathcal{T}}, \lambda^{\mathcal{T}})$ be a transition system with valuation function $\lambda^{\mathcal{T}} : \mathsf{S} \to \wp(\mathsf{S})$, p a propositional variable and S' a subset of states S; the associated valuation $\lambda^{\mathcal{T}}[p \mapsto \mathsf{S}']$ is defined for all propositional variables p' as follows:

$$\lambda^{\mathcal{T}}[p \mapsto \mathsf{S}'](p') = \begin{cases} \mathsf{S}' & \text{if } p' = p, \\ \lambda^{\mathcal{T}}(p') & \text{otherwise.} \end{cases}$$

Given a transition system $\mathcal{T} = (\mathsf{S}, \to^{\mathcal{T}}, \lambda^{\mathcal{T}})$, then $\mathcal{T}[p \mapsto \mathsf{S}']$ denotes the transition system $(\mathsf{S}, \to^{\mathcal{T}}, \lambda^{\mathcal{T}}[p \mapsto \mathsf{S}'])$. This notions are generalized straightforwardly to $\lambda[x_1 \mapsto \mathsf{S}_1, \ldots, x_n \mapsto \mathsf{S}_n]$ and $\mathcal{T}[x_1 \mapsto \mathsf{S}_1, \ldots, x_n \mapsto \mathsf{S}_n]$. Given a transition system $\mathcal{T}$, the *denotation of* φ *in* $\mathcal{T}$, $\|\varphi\|_{\mathcal{T}}$, that is, the set of states satisfying a formula φ is defined inductively on the structure of φ. Simultaneously for all transition systems we set

- $\|p\|_{\mathcal{T}} = \lambda(p)$ and $\| \sim p\|_{\mathcal{T}} = \mathsf{S} \setminus \lambda(p)$ for all $p \in \mathsf{Prop}$,
- $\|\alpha \wedge \beta\|_{\mathcal{T}} = \|\alpha\|_{\mathcal{T}} \cap \|\beta\|_{\mathcal{T}}$,
- $\|\alpha \vee \beta\|_{\mathcal{T}} = \|\alpha\|_{\mathcal{T}} \cup \|\beta\|_{\mathcal{T}}$,
- $\|\Box\alpha\|_{\mathcal{T}} = \{s \in \mathsf{S} \mid \forall t((s \to^{\mathcal{T}} t) \Rightarrow t \in \|\alpha\|_{\mathcal{T}})\}$,
- $\|\Diamond\alpha\|_{\mathcal{T}} = \{s \in \mathsf{S} \mid \exists t((s \to^{\mathcal{T}} t) \wedge t \in \|\alpha\|_{\mathcal{T}})\}$,
- $\|\nu x.\alpha\|_{\mathcal{T}} = \bigcup\{\mathsf{S}' \subseteq \mathsf{S} \mid \mathsf{S}' \subseteq \|\alpha(x)\|_{\mathcal{T}[x \mapsto \mathsf{S}']}\}$, and
- $\|\mu x.\alpha\|_{\mathcal{T}} = \bigcap\{\mathsf{S}' \subseteq \mathsf{S} \mid \|\alpha(x)\|_{\mathcal{T}[x \mapsto \mathsf{S}']} \subseteq \mathsf{S}'\}$.

We say that a pointed transition system $(\mathcal{T}, s)$ is a model of a μ-formula if and only if $s \in \|\varphi\|_{\mathcal{T}}$. By $\|\varphi\|$ we denote the class of all models of φ and by $\|\varphi\|^P$ the class of all models of φ with property P. For example, by $\|\varphi\|^t$, we denote the class of all transitive models of φ. Whenever $\|\varphi\| = \mathbb{T}$, we write $\models \varphi$. For a formula $\varphi(x)$ and set of states $\mathsf{S}' \subseteq \mathsf{S}$ we sometimes write $\|\varphi(\mathsf{S}')\|_{\mathcal{T}}$ instead of $\|\varphi(x)\|_{\mathcal{T}[x \mapsto \mathsf{S}']}$. When clear from the context we use $\|\varphi(x)\|_{\mathcal{T}}$ for the function

$$\|\varphi(x)\|_{\mathcal{T}} : \begin{cases} \wp(\mathsf{S}) \to \wp(\mathsf{S}) \\ \mathsf{S}' \mapsto \|\varphi(\mathsf{S}')\|_{\mathcal{T}}. \end{cases}$$

By the Tarski-Knaster Theorem (c.f. [119]), $\|\nu x.\alpha(x)\|_{\mathcal{T}}$ is the greatest fixpoint and $\|\mu x.\alpha(x)\|_{\mathcal{T}}$ the least fixpoint of the operator $\|\alpha(x)\|_{\mathcal{T}}$.

We say that a class $L \subseteq \mathbb{T}^P$ of transition systems is definable by a μ-formula, or simply μ-definable, if there exists $\varphi \in \mathcal{L}_\mu$ such that $L = \|\varphi\|^P$.

Thanks to Walukiewicz, we have a completeness theorem for the modal μ-calculus.

Theorem 1.17 ([128]). *For all μ-formulae φ we have that*

$$\models \varphi \quad \textit{if and only if} \quad \mathsf{Koz} \vdash \varphi.$$

The next two lemmas state some basic properties of denotations. Their proofs are standard and left to the reader.

Lemma 1.18. *For all transition systems $\mathcal{T} = (\mathsf{S}, \to^{\mathcal{T}}, \lambda^{\mathcal{T}})$ and all formulae φ we have that*

(1) $\|\neg\varphi\|_{\mathcal{T}} = \mathsf{S} \setminus \|\varphi\|_{\mathcal{T}}$,

(2) $\|\eta x.\eta y.\varphi(x, y)\|_{\mathcal{T}} = \|\eta x.\varphi(x, x)\|_{\mathcal{T}}$, *where* $\eta \in \{\mu, \nu\}$,

(3) $\|\nu x.\varphi(x)\|_{\mathcal{T}} = \|\varphi(\top)\|_{\mathcal{T}}$, *if all occurrences of x are not guarded,*

(4) $\|\mu x.\varphi(x)\|_{\mathcal{T}} = \|\varphi(\bot)\|_{\mathcal{T}}$, *if all occurrences of x are not guarded.*

Lemma 1.19. *Let $\varphi, \alpha, \alpha_i, \beta, \beta_i, \psi, \psi_i \in \mathcal{L}_\mu$ be well-named μ-formulae, where $i \in \{1, \ldots, k\}$. For all transition systems $\mathcal{T}$ the following holds:*

(1) If $\mathsf{free}(\psi_i) \cap \mathsf{bound}(\varphi) = \emptyset$ *for all* $i \in \{1, \ldots, k\}$ *then*

$$\|\varphi[x_1/\psi_1, \ldots, x_k/\psi_k]\|_{\mathcal{T}} = \|\varphi\|_{\mathcal{T}[x_1 \mapsto \|\psi_1\|_{\mathcal{T}}, \ldots, x_k \mapsto \|\psi_k\|_{\mathcal{T}}]}.$$

(2) If $\psi \leq \varphi$ *and* $x_i \in \mathsf{free}(\psi) \cap \mathsf{bound}(\varphi)$, *with* $i = 1, \ldots, k$, *then*

$$\|\psi[x_1/\varphi_{x_1}, \ldots, x_k/\varphi_{x_k}]\|_{\mathcal{T}} = \|\psi\|_{\mathcal{T}[x_1 \mapsto \|\varphi_{x_1}\|_{\mathcal{T}}, \ldots, x_k \mapsto \|\varphi_{x_k}\|_{\mathcal{T}}]}.$$

(3) If $\mathsf{free}(\psi_i) \cap \mathsf{bound}(\alpha) = \mathsf{free}(\psi_i) \cap \mathsf{bound}(\beta) = \emptyset$ *and* $x_i \in \mathsf{free}(\alpha) \cap \mathsf{free}(\beta)$ *for all* $i \in \{1, \ldots, k\}$ *and for every transition system $\mathcal{T}$ we have that*

$$\|\alpha\|_{\mathcal{T}} = \|\beta\|_{\mathcal{T}}$$

then, for every transition system $\mathcal{T}$ we have that

$$\|\alpha[x_1/\psi_1, \ldots, x_k/\psi_k]\|_{\mathcal{T}} = \|\beta[x_1/\psi_1, \ldots, x_k/\psi_k]\|_{\mathcal{T}}.$$

(4) *Let* $\mathsf{free}(\alpha_i) \cap \mathsf{bound}(\varphi) = \mathsf{free}(\beta_i) \cap \mathsf{bound}(\varphi) = \emptyset$ *and let* $x_i \in \mathsf{free}(\varphi)$ *occur positively in* φ*, where* $i = 1, \ldots, k$*. If for every transition system* $\mathcal{T}$ *we have that*

$$\|\alpha_i\|_{\mathcal{T}} \subseteq \|\beta_i\|_{\mathcal{T}}, \text{ for every } i \in \{1, \ldots, k\},$$

then we have that

$$\|\varphi[x_1/\alpha_1, \ldots, x_k/\alpha_k]\|_{\mathcal{T}} \subseteq \|\varphi[x_1/\beta_1, \ldots, x_k/\beta_k]\|_{\mathcal{T}}.$$

To conclude, we prove a useful lemma showing that a formula and its well-named companion have the same meaning and the same alternation depth.

LEMMA 1.20. *For all* μ*-formulae* φ *there is a well-named formula* $\mathsf{wn}(\varphi)$ *such that for all* $\mathcal{T}$ *we have* $\|\varphi\|_{\mathcal{T}} = \|\mathsf{wn}(\varphi)\|_{\mathcal{T}}$ *and* $\mathsf{ad}(\varphi) = \mathsf{ad}(\mathsf{wn}(\varphi))$*.*

Proof. We have just to verify that the construction of $\mathsf{wn}(\varphi)$ given by parts 2 to 4 of Lemma 1.18 does not increase the alternation depth of the formula. But this is straightforward. □

Given Lemma 1.21 and Lemma 1.20, we can assume that wn is a function associating to every formula φ a well-named formula $\mathsf{wn}(\varphi)$ which has the same alternation depth and the same denotation in every transition system. Moreover, the next lemma shows that wellnaming iterated formulae which are already well-named does not affect the rank. It follows by the fact that since φ is well-named the well-named formula corresponding to $\varphi^n(\top)$ is given by simply renaming bound variables.

LEMMA 1.21. *For all well-named formulae* φ *such that* $x \in \mathsf{free}(\varphi)$ *appears only positively and all* $n \in \mathbb{N}$ *we have that*

$$\mathsf{rank}(\varphi^n(\top)) = \mathsf{rank}(\mathsf{wn}(\varphi^n(\top))).$$

Similarly for $\bot$*.*

From now on, if nothing else is mentioned, we assume that all μ-formulae are well-named.

1.4.3 Another formalism for the binary tree

In this subsection we introduce a slightly modified language for the modal μ-calculus interpreted over full binary trees. The only difference is that the underlying modal language allows two different existential modalities: one for the left child and one for the right child. The reason of introducing this version of the modal μ-calculus is because it has an exact counterpart in terms of automata that will be studied in details in the second part of this dissertation.

Formally, given a set Prop of propositional variables, the collection of formulae of the *bi-modal* μ*-calculus* is defined as follows:

$$\varphi ::= p \mid \neg\varphi \mid (\varphi \vee \varphi) \mid (\varphi \wedge \varphi) \mid \langle 0 \rangle \varphi \mid \langle 1 \rangle \varphi \mid \mu x.\varphi \mid \nu x.\varphi$$

where $p, x \in \mathsf{Prop}$ and x occurs only positively in $\eta x.\varphi$ ($\eta = \nu, \mu$), that is, x is the scope of an even number of negations.

Mutatis mutandis, the alternation depth of a formula, all the other standard syntactical notions and the semantics (with the obvious interpretations for the two existential modalities) on full binary trees are defined as for the "normal" modal μ-calculus, as well as the definitions of evaluation game, alternation free fragment, and of the syntactical and semantical fixpoint hierarchies that will follow in the next sections. Note that for every formula a semantically equivalent formula in positive normal form (the only negated subformulae are free variables) can be obtained by repetitive uses of de Morgan laws, the usual equivalences connecting fixpoints and the obvious equivalences: $\neg\langle i\rangle\varphi \equiv \langle i\rangle\neg\varphi$, for $i = 0, 1$. Finally, it is worth noticing that on full binary trees the standard modal μ-calculus can be seen as a fragment of the bi-modal μ-calculus just by defining $\Diamond\varphi \equiv \langle 0\rangle\varphi \vee \langle 1\rangle\varphi$.

1.4.4 MSO vs μ-calculus

Since the work of Niwiniski [96, 97], it was known that the modal μ-calculus with two existential modalities corresponds exactly to monadic second order logic[4] on binary trees.

But if we look to a transition system as representing the behavior of a process, we usually do not want to distinguish between operationally equivalent such structures. Since bisimulation relation is often considered to be the finest equivalence relation which is interesting in this context, this means that a specification language should not distinguish between bisimilar transition systems.

Properties of trees or transition systems are naturally expressed in quantified languages. But such languages can distinguish between bisimilar structures. This is for example the case of first-order logic, but interestingly not of modal logic. In fact, van Benthem [12] proved that modal logic express exactly the first order properties on transition systems which are invariant under bisimulation, meaning that from an operational point of view, all first order properties which make sense are modal. Unfortunately, first order logic is often not expressive enough. Indeed, it cannot express properties like liveness and safety, contrary to a second order language. So, what about monadic second order logic? And where does the modal μ-calculus stand? The question has been solved by Janin and Walukiewicz. In [66] they show that all monadic second order properties of transition systems which are bisimulation invariant are exactly the ones which are expressible in the modal μ-calculus, meaning that this logic is the "right" weakening of second order logic.

So, let's first recall when two transition systems are said to be bisimilar. A bisimulation relation between two pointed transition systems $(\mathcal{T}_0, s_0)$ and $(\mathcal{T}_1, s_1)$ over an arbitrary set Σ is a relation $R \subseteq \mathsf{S}^{\mathcal{T}_0} \times \mathsf{S}^{\mathcal{T}_1}$ such that $(s_0, s_1) \in R$ and for every $(s_0', s_1') \in R$:

- $\lambda^{\mathcal{T}_0}(s_0') = \lambda^{\mathcal{T}_1}(s_1')$,
- whenever $s_0' \to^{\mathcal{T}_0} s_0''$, for some $s_0'' \in \mathsf{S}^{\mathcal{T}_0}$, then there exists $s_1'' \in \mathsf{S}^{\mathcal{T}_1}$ such that $s_1' \to^{\mathcal{T}_1} s_1''$ and $(s_0'', s_1'') \in R$, and
- whenever $s_1' \to^{\mathcal{T}_1} s_1''$, for some $s_1'' \in \mathsf{S}^{\mathcal{T}_1}$, then there exists $s_0'' \in \mathsf{S}^{\mathcal{T}_0}$ such that $s_0' \to^{\mathcal{T}_0} s_0''$ and $(s_0'', s_1'') \in R$.

[4]With the child relation *and* the order on siblings.

Two pointed transition systems over Σ are said to be *bisimilar* if there exists a bisimulation relation between them.

The fine-structure of the bisimulation relation suggests the following notion of a *bisimulation game*. There are two players, Bob and Anne. Bob claims that the two pointed transition systems $(\mathcal{T}_0, s_0)$ and $(\mathcal{T}_1, s_1)$ over Σ are different, while Anne says they are bisimilar. The game proceeds in rounds. At the beginning of each round, the state in the game is a pair of states $(s'_0, s'_1) \in \mathsf{S}^{\mathcal{T}_0} \times \mathsf{S}^{\mathcal{T}_1}$. A round is played as follows. If the label of the two states determining the current position of the play are different, then the game stops and Bob wins. Otherwise first Bob selects one of the transition systems $\mathcal{T}_i$ and a $\rightarrow^{\mathcal{T}_i}$-successor of s'_i, for $i = 0, 1$. Then Anne selects a $\rightarrow^{\mathcal{T}_{1-i}}$-successor of s'_{1-i} in the other structure $\mathcal{T}_{1-i}$ and the round is finished. A new round is played with the position updated to (s''_0, s''_1). If a player cannot choose a successor when it is her turn in a round, she loses. If Anne can survive for infinitely many rounds, she wins. It is then easy to verify that $(\mathcal{T}_1, s_1)$ and $(\mathcal{T}_2, s_2)$ are bisimilar iff Anne has a winning strategy in the corresponding bisimulation game.

Given a class $\mathbf{C}$ of pointed transition systems over Σ, $\mathbf{C}$ is said to be *closed under bisimulation*, or bisimulation closed, if whenever $(\mathcal{T}, s) \in \mathbf{C}$ and $(\mathcal{T}', s')$ is bisimilar to $(\mathcal{T}, s)$, then $(\mathcal{T}', s') \in \mathbf{C}$. Janin and Walukiewicz characterization of the μ-calculus is therefore the following:

THEOREM 1.22 ([66]). *A bisimulation closed class of transition systems is definable in* MSO *iff it is definable by a μ-formula.*

Since for all μ-formulae ϕ, $\|\phi\|$ is closed under bisimulation, it turns out that the previous theorem means that properties definable in MSO which are closed under bisimulation are exactly the one definable with a formula of the modal μ-calculus.

1.5 The fixpoint alternation hierarchy

What gives the modal μ-calculus its great expressive power is not simply the use of fixpoint operators, but the possibility of nesting (alternate) greatest and least fixpoints in a "non trivial" way. This phenomenon is important from a practical point of view, as the alternation depth of a μ-formula is what appears to generate the computational complexity of model-checking (cf. sections 1.6 and 1.7). From a theoretical point of view, the classification naturally raises questions about the expressive power of the classes. In particular, the question whether the expressiveness for the modal μ-calculus is somewhat "bounded", that is just a bounded number of alternation are needed to express all possible definable properties.

In this section we study alternation in more detail by describing what are called the syntactical and semantical hierarchies for the modal μ-calculus.

Let $\Phi \subseteq \mathcal{L}_\mu$. For $\eta \in \{\nu, \mu\}$, $\eta(\Phi)$ is the smallest class of formulae such that:

- $\Phi, \neg\Phi \subset \eta(\Phi)$;
- If $\psi(x) \in \eta(\Phi)$ and x occurs only positively, then $\eta x.\psi \in \eta(\Phi)$;
- If $\psi, \varphi \in \eta(\Phi)$, then $\psi \wedge \varphi, \psi \vee \varphi, \Diamond\psi, \Box\psi \in \eta(\Phi)$;

- If $\psi, \varphi \in \eta(\Phi)$ and $\mathsf{free}(\psi) \cap \mathsf{bound}(\varphi) = \emptyset$ then $\varphi[x/\psi] \in \eta(\Phi)$

With the help of this definition, we introduce the syntactical hierarchy for the modal μ-calculus. For all $n \in \mathbb{N}$, we define the class of μ-formulae Σ^μ_n and Π^μ_n inductively as follows:

- $\Sigma^\mu_0 := \Pi^\mu_0 := \mathcal{L}_{\mathsf{M}}$;
- $\Sigma^\mu_{n+1} = \mu(\Pi^\mu_n)$;
- $\Pi^\mu_{n+1} = \nu(\Sigma^\mu_n)$;
- $\Delta^\mu_n = \Sigma^\mu_n \cap \Pi^\mu_n$

It is clear that $\mathcal{L}_\mu = \bigcup_{n\in\omega} \Sigma^\mu_n = \bigcup_{n\in\omega} \Pi^\mu_n$. Moreover from the definitions above, we can easily prove that this hierarchy is strict, that is to say, for every $n \in \omega$

$$\Sigma^\mu_n \cup \Pi^\mu_n \subsetneq \Sigma^\mu_{n+1} \quad \text{and} \quad \Sigma^\mu_n \cup \Pi^\mu_n \subsetneq \Pi^\mu_{n+1}$$

All Σ^μ_n and Π^μ_n classes form the *syntactical modal μ-calculus hierarchy*, also called the syntactical fixpoint alternation hierarchy.

$$\begin{tikzcd}
\Pi^\mu_0 \arrow[rr] \arrow[dr] \arrow[dd, equal] & & \Sigma^\mu_1 \arrow[rr] \arrow[dr] & & \Sigma^\mu_2 \arrow[rr] \arrow[dr] & & \Sigma^\mu_3 \;\cdots \\
 & \Delta^\mu_1 \arrow[ur] \arrow[dr] & & \Delta^\mu_2 \arrow[ur] \arrow[dr] & & \Delta^\mu_3 \;\cdots & \\
\Sigma^\mu_0 \arrow[rr] \arrow[ur] & & \Pi^\mu_1 \arrow[rr] \arrow[ur] & & \Pi^\mu_2 \arrow[rr] \arrow[ur] & & \Pi^\mu_3 \;\cdots
\end{tikzcd}$$

Figure 1.2: The syntactical fixpoint hierarchy. Arrows stand for set-theoretic inclusion.

It can be asked what is the relation between levels of the syntactical modal μ-calculus hierarchy and the fixpoint alternation depth. This relation is summarized in the following result:

Proposition 1.23 ([99]). *Let $k \in \mathbb{N}$ and $\varphi \in \mathcal{L}_\mu$. Then:*

(1) $\mathsf{ad}(\varphi) \leq k$ iff $\varphi = \phi[x_1/\psi_1, \ldots, x_l/\psi_l]$ for some formula $\phi \in \Sigma^\mu_0$ and formulae $\psi_i \in \Sigma^\mu_k \cup \Pi^\mu_k$,

(2) the formula φ belongs to Σ^μ_k iff $\mathsf{ad}(\varphi) \leq k$ and there is no subformula $\nu x.\psi \leq \varphi$ such that $\max^\nu(\nu x.\psi) = k$

(3) the formula φ belongs to Π^μ_k iff $\mathsf{ad}(\varphi) \leq k$ and there is no subformula $\mu x.\psi \leq \varphi$ such that $\max^\mu(\mu x.\psi) = k$

We therefore obtain that the alternation depth of $\varphi \in \mathcal{L}_\mu$ is equivalently given by

$$\mathsf{ad}(\varphi) = \inf\{k : \varphi \in \Delta^\mu_{k+1}\}.$$

In particular this means that the fixpoint alternation free fragment, that is the class of μ-formulae whose alternation depth is 1, corresponds to the second ambiguous syntactical class Δ^μ_2.

The *semantical modal μ-calculus hierarchy over* $\mathbb{T}$, also called the semantical fixpoint alternation hierarchy over $\mathbb{T}$, consists of all $\Sigma^{\mu\mathbb{T}}_n$ and $\Pi^{\mu\mathbb{T}}_n$, which are classes of pointed transition systems defined inductively as follows:

$$\Sigma^{\mu\mathbb{T}}_n = \{\|\varphi\| : \varphi \in \Sigma^\mu_n\}$$

$$\Pi^{\mu\mathbb{T}}_n = \{\|\varphi\| : \varphi \in \Pi^\mu_n\}$$

As usual, the ambiguous classes are defined by

$$\Delta^{\mu\mathbb{T}}_n := \Sigma^{\mu\mathbb{T}}_n \cap \Pi^{\mu\mathbb{T}}_n.$$

The *semantical modal μ-calculus hierarchy over* $\mathbb{T}^P$, for any property P (like being a transitive model), is defined analogously.

$$\begin{tikzcd}
\Pi^{\mu\mathbb{T}}_0 \arrow[rr] \arrow[dr] & & \Sigma^{\mu\mathbb{T}}_1 \arrow[rr] \arrow[dr] & & \Sigma^{\mu\mathbb{T}}_2 \arrow[rr] \arrow[dr] & & \Sigma^{\mu\mathbb{T}}_3 \quad \cdots \\
\| & \Delta^{\mu\mathbb{T}}_1 \arrow[ur] \arrow[dr] & & \Delta^{\mu\mathbb{T}}_2 \arrow[ur] \arrow[dr] & & \Delta^{\mu\mathbb{T}}_3 \quad \cdots & \\
\Sigma^{\mu\mathbb{T}}_0 \arrow[rr] \arrow[ur] & & \Pi^{\mu\mathbb{T}}_1 \arrow[rr] \arrow[ur] & & \Pi^{\mu\mathbb{T}}_2 \arrow[rr] \arrow[ur] & & \Pi^{\mu\mathbb{T}}_3 \quad \cdots
\end{tikzcd}$$

Figure 1.3: The semantical fixpoint hierarchy. Arrows stand for set-theoretic inclusion.

The strictness of the semantical fixpoint hierarchy over all transition systems has been first proven by Bradfield (cf. [32, 33]). Simultaneously, Lenzi in [80] has proven a strictness theorem for the positive μ-calculus, that is, the fragment consisting of all formulae such that the propositional variables appear only positively.

Theorem 1.24 (Bradfield). *The semantical modal μ-calculus hierarchy over $\mathbb{T}$ is strict.*

The same question, but restricted to full binary trees, has been independently solved by Bradfield [34] and Arnold[5] [7].

[5] In his original formulation, Arnold's result concerns the analogous hierarchy for alternating automata, the so-called index hierarchy, a hierarchy that will be introduced and discussed in section 1.9. The strictness of the fixpoint hierarchy on binary trees can then be derived from this result thanks to the equivalence between alternating automata and the bi-modal μ-calculus, cf. section 1.8.

A classical result by Rabin [108] states that if a set of infinite trees and its complement are both definable by an existential and by a universal MSO sentence, then these sets are definable in weak monadic second order logic. In the language of the modal μ-calculus, the theorem asserts the equality, on infinite trees, between the second ambiguous class of the corresponding semantical fixpoint alternation hierarchy and the class of tree models definable by an alternation free formula [8][6]. This result is surprisingly no more true as we climb higher in the hierarchy, as shown by Arnold and Santocanale [11].

From now on, when we write about the *modal μ-calculus hierarchy*, resp. the *fixpoint alternation hierarchy*, we always mean the *semantical* modal μ-calculus hierarchy, resp. the *semantical* fixpoint alternation hierarchy.

EXAMPLE 1.25. It is instructive to have a look at two typical μ-formulae. The first formula express the property of "always eventually p"

$$\nu x.(\mu y.(p \vee \Diamond y)) \wedge \Box x).$$

Indeed, it says that from any node of a model, we can reach a node where p holds. Since this formula is in Π_1^μ, this kind of property can be expressed without any alternation. Moreover, it can be shown that this formula cannot be reduced to a purely modal formula. The second formula defines the property of "there is a path where p holds infinitely often"

$$\nu x.\mu y.((p \vee \Diamond y)) \wedge \Diamond x).$$

It can be verified that the alternation is really needed, that is, that the class of models of this formula is in $\Pi_2^{\mu\mathbb{T}} \setminus \Sigma_2^{\mu\mathbb{T}}$.

1.6 Parity games

Since its appearance in the verification community, it was clear that new techniques were required for the understanding of the complexity of the modal μ-calculus. That's when parity games make their remarkable entry into the arena. By replacing old methods in proof and techniques but also by becoming a central topic of investigation in their own right, they are now an essential tool in the study of this formalism. In the present and next sections parity games and their link with the μ-calculus are presented and discussed.

Formally, a *game* $\mathcal{G}$ is defined in terms of an arena A and a winning condition W. In our case an arena is simply a bi-partite graph $A = \langle V_0, V_1, E\rangle$, where $V_0 \cap V_1 = \emptyset$ and the edge relation, or set of moves, is $E \subseteq (V_0 \cup V_1) \times (V_0 \cup V_1)$. Let $V = V_1 \cup V_2$ be the set of vertices, or positions, of the arena. Given two vertices $a, b \in V$, we say that b is a successor of a, if $(a, b) \in E$. The set of all successors of a is sometimes denoted by aE or $E(a)$. We say that b is reachable from a if there are $a_1, \dots, a_n \in V$ such that $a_1 = a$, $a_n = b$ and for every $0 < i < n$, $a_{i+1} \in a_iE$.

A *play* in the arena A can be finite or infinite. In the former case, the play is a non empty finite path $\pi = a_1 \dots a_n \in V^+$ such that for every $0 < i < n$, $a_{i+1} \in a_iE$ and $a_nE = \emptyset$. In the last case, the play consists in an infinite path $\pi = a_1 \dots a_n \dots \in V^\omega$ with $a_{i+1} \in a_iE$ for every $i > 0$. Thus a finite or infinite

[6]This identity can then be lifted from trees to transition systems [77].

play in a game can be seen as the trace of a token moved on the arena by two Players, Player 0 and Player 1, in such a way that if the token is in position $a \in V_i$, then Player i has to choose a successor of a where to move the token.

The set of *winning conditions* W is a subset of V^ω. Thus, given a game $\mathcal{G} = (A, W)$ a play π is winning for Player 0 iff

(1) if π is finite, then the last position a_n of the play is in V_1,

(2) if π is infinite, then it must be a member of W.

A play is winning for Player 1 if it is not winning for Player 0. In this framework we are interested in what is called a *parity winning condition.* This winning condition is nowadays recognized as the most fundamental one of all winning conditions for two-player infinite games. In particular, as we will see in the next section, it captures exactly what is needed to evaluate formulae of the modal μ-calculus. Formally, given a set of vertices V, we assume a coloring or ranking function $\Omega : V \to \omega$ such that $\Omega[V]$ is bounded. Then, the set W of winning conditions is defined as the set of all infinite sequences π such that the greatest priority appearing infinitely often in $\Omega(\pi)$ is even [7].

Let A be an arena. A strategy for Player i is simply a function $\sigma_i : V^*V_i \to V$, with $i = 1, 2$. A prefix $a_1 \dots a_n$ of a play is said to be *compatible* or *consistent* with σ_i iff for every j with $1 \leq j < n$ and $a_j \in V_i$, it holds that $\sigma_i(a_1 \dots a_j) = a_{j+1}$. A finite or infinite play is compatible or consistent with σ_i if each of its prefix which is in V^*V_i is compatible with σ_i. The strategy σ_i is said to be a *winning strategy* for Player i on W if every play consistent with σ_i is winning for Player i. A position $a \in V$ is winning for Player i in the parity game $\mathcal{G}$ iff there is a strategy σ for Player i such that every play compatible with σ which starts from a is winning for Player i. A winning strategy σ is called *memoryless* if $\sigma(a_1 \dots a_n) = \sigma(b_1 \dots b_n)$, when $a_n = b_n$.

Ther are two basic questions about games that usually need to be answered.

(1) (Determinacy) Does every game has a winner?

(2) How difficult is it to determine who wins a (finite) game (if there is any winner) ?

The first question has a positive answer. That is, for parity games we have a memoryless determinacy result.

Theorem 1.26 ([53, 89]). *In a parity game, one of the Players has a memoryless winning strategy from each vertex.*

Having in mind this theorem, in the sequel we assume that all winning strategies are memoryless, that is, a winning strategy in a parity games for Player 0 is a function $\sigma : V_0 \to V$, analogously for Player 1.

The decision problem of determining the winner of a parity game is formally given by:

(WINS) given a finite parity game $\mathcal{G}$, determine whether or not Player 0 has a winning strategy in the game $\mathcal{G}$

[7] A parity game is called a *weak* parity game if, as a winning condition, we say that Player 0 wins (either a finite or an infinite play) if and only if the greatest priority occurring in the play is even.

This problem is clearly P-hard, and moreover we have that[8]:

Theorem 1.27 ([116, 69]).

- *WINS is solvable in time*

$$\mathcal{O}\big(dm(\frac{n}{\lfloor \frac{d}{2} \rfloor})^{\lfloor \frac{d}{2} \rfloor}\big)$$

  where n is the number of vertices, m is the number of edges, and d is the maximum priority in a parity game

- *WINS is in UP $\cap$ co-UP,*

A big open problem is to know whether WINS is in P or not.

1.7 Evaluation games

Using the standard semantics on transition systems is not the only way, and probably not the most transparent way, of interpreting the modal μ-calculus. It is for instance possible to associate a meaning to formulae in terms of games. In this section we will see, given $\varphi \in \mathcal{L}_\mu$ and a pointed transition system $(\mathcal{T}, s_0)$, how to determine the corresponding parity game $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$, called also the *evaluation game of φ over $(\mathcal{T}, s_0)$*. Thanks to the a "game-theoretical" version of what is usually called the *Fundamental Theorem* of the semantic of the modal μ-calculus, first proved by Emerson and Street, the two semantics are shown to be equivalent.

Recall that $\mathcal{T} = (\mathsf{S}, \to^{\mathcal{T}}, \lambda^{\mathcal{T}})$. The arena of $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$ is the triple $\langle V_0, V_1, E\rangle$ which is defined recursively such that

$$\langle \varphi, s_0 \rangle \in V$$

(recall that $V = V_0 \cup V_1$) and such that if $\langle \psi, s\rangle \in V$ then we distinguish the following cases:

- If $\psi \equiv (\neg)p$ and $p \in \mathsf{free}(\varphi)$. In this case we set $E\langle \psi, s\rangle = \emptyset$ and

$$\langle \psi, s\rangle \in V_1 \text{ iff } \begin{cases} s \in \lambda^{\mathcal{T}}(\psi) & \text{if } \psi \equiv p \\ s \notin \lambda^{\mathcal{T}}(\psi) & \text{if } \psi \equiv \neg p. \end{cases}$$

- If $\psi \equiv x$ and $x \in \mathsf{bound}(\varphi)$. In this case we set

$$(\langle \psi, s\rangle, \langle \varphi_x, s\rangle) \in E$$

  and we have

$$\langle \psi, s\rangle \in V_0 \text{ iff } x \text{ is a } \mu\text{-variable.}$$

- If $\psi \equiv \alpha \wedge \beta$ then we have $\langle \psi, s\rangle \in V_1$, and if $\psi \equiv \alpha \vee \beta$ then we have $\langle \psi, s\rangle \in V_0$. In both cases it holds that

$$(\langle \psi, s\rangle, \langle \alpha, s\rangle) \in E \text{ and } (\langle \psi, s\rangle, \langle \beta, s\rangle) \in E$$

[8]The class UP is the complexity class of decision problems solvable in polynomial time on a non-deterministic Turing machine with at most one accepting path for each input. UP contains P and is contained in NP.

- If $\psi \equiv \Box\alpha$ then we have $\langle\psi, s\rangle \in V_1$, and if $\psi \equiv \Diamond\alpha$ then we have $\langle\psi, s\rangle \in V_0$. In both cases it holds that

$$(\langle\psi, s\rangle, \langle\alpha, s'\rangle) \in E \text{ for all } s' \text{ such that } s \to^{\mathcal{T}} s'.$$

- If $\psi \equiv \nu x.\alpha$ then we have $\langle\psi, s\rangle \in V_1$, and if $\psi \equiv \mu x.\alpha$ then we have $\langle\psi, s\rangle \in V_0$. In both cases it holds that

$$(\langle\psi, s\rangle, \langle\alpha, s\rangle) \in E.$$

We complete the definition of the parity game $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$ by defining the (partial) priority function $\Omega : V \to \omega$. The function is first defined on states of the form $\langle\eta x.\delta, s\rangle \in V$, where $\eta \in \{\mu, \nu\}$. In this case we have that:

$$\Omega(\langle\psi, s\rangle) = \begin{cases} \mathsf{ad}(\eta x.\delta) & \text{if } \eta = \mu \text{ and } \mathsf{ad}(\eta x.\delta) \text{ is odd, or} \\ & \quad \eta = \nu \text{ and } \mathsf{ad}(\eta x.\delta) \text{ is even;} \\ \mathsf{ad}(\eta x.\delta) - 1 & \text{if } \eta = \mu \text{ and } \mathsf{ad}(\eta x.\delta) \text{ is even, or} \\ & \quad \eta = \nu \text{ and } \mathsf{ad}(\eta x.\delta) \text{ is odd.} \end{cases}$$

For a state of the form $\langle x, s\rangle$, where $x \in \mathsf{bound}(\varphi)$, we set

$$\Omega(\langle x, s\rangle) := \Omega(\langle\varphi_x, s\rangle).$$

where φ_x is the unique subformula $\eta x.\delta \leq \varphi$ which bounds x.

For all the other states $\langle\alpha, s\rangle$ we distinguish two cases. If there is a least formula $\eta x.\delta \in \mathsf{sub}(\varphi)$ such that $\eta x.\delta > \alpha$ we set

$$\Omega(\langle\alpha, s\rangle) := \Omega(\langle\eta x.\delta, s\rangle).$$

If there is no such formula then we set

$$\Omega(\langle\alpha, s\rangle) = \begin{cases} 1 & \text{if } \varphi \text{ is a modal formula,} \\ \min\{\Omega(\eta x.\delta) : \eta x.\delta \leq \varphi\} & \text{else.} \end{cases}$$

It can easily be seen that if there is a formula $\eta x.\delta > \alpha$ then there is also a least one. Therefore, the second case refers to subformulae α of φ which can not be regenerated by a fixpoint application in a parity game. In the following we simply write $\min\Omega$ and $\max\Omega$ instead of $\min\{\Omega(\langle\alpha, s\rangle) : \langle\alpha, s\rangle \in V\}$ and of $\max\{\Omega(\langle\alpha, s\rangle) : \langle\alpha, s\rangle \in V\}$.

Recall that if the play π is finite, Player 0 wins iff the last vertex of the play belongs to V_1, and if the play π is infinite, Player 0 wins iff the greatest priority appearing infinitely often even.

Theorem 1.28 ([116]). *$(\mathcal{T}, s) \in \|\varphi\|$ iff Player 0 has a winning strategy for $\mathcal{E}(\varphi, (\mathcal{T}, s))$.*

This result can be seen as the "game-theoretical version" of what is usually called the *Fundamental Theorem* of the semantic of the modal μ-calculus.

Example 1.29. Evaluation game $\mathcal{E}(\nu x.\Box((p \vee \Box\bot) \wedge x), (\mathcal{T}, s_1))$. The transition system $\mathcal{T}$ is as in Figure 1.4, that is, it has states $\{s_1, s_2, s_3\}$ and p holds in s_1 and s_2, and the accessibility relation is as depicted in Figure 1.4.

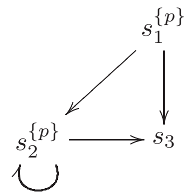


Figure 1.4:

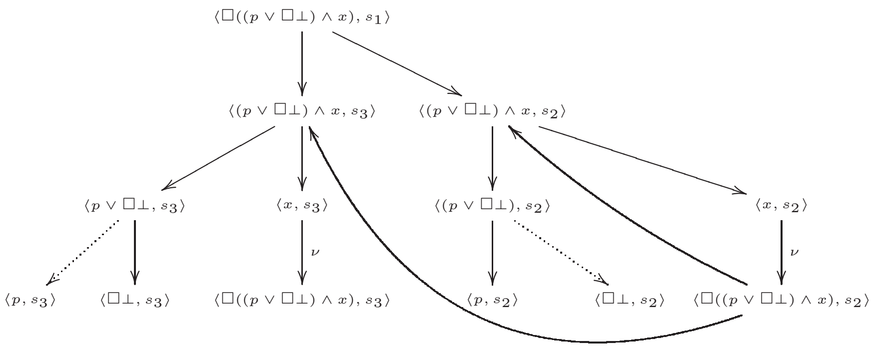


Figure 1.5:

In Figure 1.5, we have the arena of $\mathcal{E}(\nu x.\Box((p \vee \Box\bot) \wedge x), (\mathcal{T}, s_1))$. In order to simplify the picture we identified vertices of the form $\langle \nu x.\Box((p \vee \Box\bot) \wedge x), s \rangle$ with the vertices of the form $\langle \Box((p \vee \Box\bot) \wedge x), s \rangle$. Note, that this does not change essentially the evaluation game. Further, the graph given by the non-dotted edges represents the part of the arena which can be reached by a play given the strategy of Player 0 where he chooses, if there is the possibility, the non-dotted instead of the dotted move. Note, that it is a winning strategy. It is then easy to verify that $\nu x.\Box((p \vee \Box\bot) \wedge x)$ is valid if for all reachable states in a transition system we have that either, the state is terminal, or, p holds in the state.

The proof of the following lemma follows from the proof in Emerson and Street [116] of Theorem 1.28.

LEMMA 1.30. *Let $\mathcal{T} = (\mathsf{S}, \to^{\mathcal{T}}, \lambda^{\mathcal{T}})$ be a transition system and $\varphi(x_1, \dots, x_k)$ be a formula where all x_i occur positively. Let σ be a strategy for Player 0 in $\mathcal{E}(\varphi(x_1, \dots, x_k), (\mathcal{T}, s))$. Suppose that for all vertices of the form $\langle x_i, s' \rangle$ which are reachable by σ we have that $s' \in A_i \subseteq \mathsf{S}$, with $i = 1, \dots, k$. Then σ can be converted into a winning strategy for Player 0 in the evaluation game $\mathcal{E}(\varphi(x_1, \dots, x_k), (\mathcal{T}[x_1 \mapsto A_1, \dots, x_k \mapsto A_k], s))$.*

A first immediate consequence of Theorem 1.28 is a complexity bound for the model-checking problem of the modal μ-calculus. Formally, the problem is the following:

(Model-Checking) given a finite pointed transition system $(\mathcal{T}, s)$ and a formula $\varphi \in \mathcal{L}_\mu$, determine whether or not $(\mathcal{T}, s) \in \|\varphi\|$

From Theorem 1.27, this problem is in UP $\cap$ co-UP and is solvable in time

$$\mathcal{O}\big(dm(\frac{n}{\lfloor\frac{d}{2}\rfloor})^{\lfloor\frac{d}{2}\rfloor}\big)$$

where n and m are numbers depending on $(\mathcal{T}, s)$ and φ, and d is the alternation depth of φ.

Given a parity game $\mathcal{E}(\varphi,(\mathcal{T},s))$ for a formula φ we define the *pointed game transition system* $\mathcal{T}(\mathcal{E}(\varphi,(\mathcal{T},s))) = ((\mathsf{S}, \to^{\mathcal{T}}, \lambda^{\mathcal{T}}), s_0)$ such that the states S are the vertices V and the distinguished state $s_0 = \langle\varphi, s\rangle$, and such that the transition relation $\to^{\mathcal{T}}$ is the edge relation E of the parity game. If $\mathsf{ad}(\varphi) = n$ then the valuation $\lambda^{\mathcal{T}}$ is specified for the new propositional variables

$$\{c_i : 0 \le i \le n\} \cup \{d_i : 0 \le i \le n\}.$$

For all $\psi \in \mathsf{sub}(\varphi)$ we define our valuation for these propositional variables such that

$$\lambda^{\mathcal{T}}(d_i) = \{\langle\psi, s\rangle : \langle\psi, s\rangle \in V_0 \text{ and } \Omega(\langle\psi, s\rangle) = i\} \quad \text{and}$$

$$\lambda^{\mathcal{T}}(c_i) = \{\langle\psi, s\rangle : \langle\psi, s\rangle \in V_1 \text{ and } \Omega(\langle\psi, s\rangle) = i\}.$$

In the following we introduce the *game formulae* and show that with them it is possible to test the existence of a winning strategy for Player 0 in an evaluation game.

Definition 1.31. *For all $n \ge 1$ we define the Σ^μ_n game formula $W_{\Sigma^\mu_n}$ such that:*

$$W_{\Sigma^\mu_n} :\equiv \begin{cases} \mu x_{n-1}.\nu x_{n-2}.\dots\nu x_0\big(\bigvee_{i=0}^{n-1}(d_i \wedge \Diamond x_i) \vee \bigvee_{i=0}^{n-1}(c_i \wedge \Box x_i)\big) & n \text{ even} \\ \mu x_n.\nu x_{n-1}.\dots\mu x_1\big(\bigvee_{i=1}^{n}(d_i \wedge \Diamond x_i) \vee \bigvee_{i=1}^{n}(c_i \wedge \Box x_i)\big) & n \text{ odd} \end{cases}$$

The Π^μ_n game formula $W_{\Pi^\mu_n}$ is defined such that:

$$W_{\Pi^\mu_n} :\equiv \begin{cases} \nu x_n.\mu x_{n-1}.\dots\mu x_1\big(\bigvee_{i=1}^{n}(d_i \wedge \Diamond x_i) \vee \bigvee_{i=1}^{n}(c_i \wedge \Box x_i)\big) & n \text{ even} \\ \nu x_{n-1}.\mu x_{n-2}.\dots\nu x_0\big(\bigvee_{i=0}^{n-1}(d_i \wedge \Diamond x_i) \vee \bigvee_{i=0}^{n-1}(c_i \wedge \Box x_i)\big) & n \text{ odd} \end{cases}$$

For $n = 0$ we define

$$W_{\Sigma^\mu_0} :\equiv W_{\Pi^\mu_0} :\equiv W_{\Sigma^\mu_1}.$$

It is clear from definition that for all $n \ge 1$ we have that $W_{\Sigma^\mu_n} \in \Sigma^\mu_n$ and $W_{\Pi^\mu_n} \in \Pi^\mu_n$.

Proposition 1.32 ([53, 130]). *Let $\mathcal{G}$ an arbitrary parity game. Assume that $\min\Omega \in \{0,1\}$ and $\max\Omega = n$. We have that if n is even (resp. odd):*

(a) if $\min\Omega = 0$ then Player 0 has a winning strategy for $\mathcal{G}$ if and only if $\mathcal{T}(\mathcal{G}) \in \|W_{\Pi^\mu_{n+1}}\|$ (resp. $\mathcal{T}(\mathcal{G}) \in \|W_{\Sigma^\mu_{n+1}}\|$),

(b) if $\min\Omega = 1$ then Player 0 has a winning strategy for $\mathcal{G}$ if and only if $\mathcal{T}(\mathcal{G}) \in \|W_{\Pi^\mu_n}\|$ (resp. $\mathcal{T}(\mathcal{G}) \in \|W_{\Sigma^\mu_n}\|$)

From Proposition 1.32 and the definition of an evaluation game, it follows immediately that:

COROLLARY 1.33. *Let φ be a Π_n^μ-formula (resp. a Σ_n^μ-formula) and let $(\mathcal{T}, s)$ be an arbitrary pointed transition system. We have that Player 0 has a winning strategy for $\mathcal{E}(\varphi, (\mathcal{T}, s))$ if and only if $\mathcal{T}(\mathcal{E}(\varphi, (\mathcal{T}, s))) \in \|W_{\Pi_n^\mu}\|$ (resp. if and only if $(\mathcal{T}, s) \in \|W_{\Sigma_n^\mu}\|$).*

Therefore, by applying Proposition 1.28 and Corollary 1.33, we have the following result:

COROLLARY 1.34. *Let φ be a Π_n^μ-formula (resp. Σ_n^μ-formula) and let $(\mathcal{T}, s)$ be an arbitrary pointed transition system. We have that*

$$(\mathcal{T}, s) \in \|\varphi\| \text{ if and only if } \mathcal{T}(\mathcal{E}(\varphi, (\mathcal{T}, s))) \in \|W_{\Pi_n}\| \ \text{(resp. } (\mathcal{T}, s) \in \|W_{\Sigma_n}\|).$$

1.8 Automata for the modal μ-calculus

Another way of looking at μ-formulae is to consider automata. The kind of automata we introduce differ slightly depending on whether we are considering full binary trees or more generally arbitrary transition systems. After presenting both cases, we state the expected correspondences with the corresponding logical formalism (bi-modal μ-calculus for binary trees and standard modal μ-calculus for transition systems).

We refrain from giving a thorough introduction to automata on words and trees. For a comprehensive introduction to automata theory, we refer to the books [104, 57] and to the survey [118]. Throughout this section we assume that Prop is finite.

1.8.1 Automata on binary trees

Recall that full binary trees are total functions $t : \{0,1\}^* \to \Sigma$ with a prefix closed domain. From now on in this subsection T_Σ denote the space of full binary trees over Σ.

We choose to work with automata having the parity condition as acceptance condition. An *alternating parity tree automaton* over a finite input alphabet Σ $A = \langle Q, Q_\exists, Q_\forall, q_I, \delta, \Omega\rangle$ consists of a finite set Q of states partitioned into existential states $Q_\exists$ and universal states $Q_\forall$, an initial state q_I, a transition relation $\delta \subseteq Q \times \Sigma \times \{\varepsilon, 0, 1\} \times Q$ and a priority function $\Omega : Q \to \omega$. We can assume that $\iota \in \{0,1\}$.Sometimes we write $q \xrightarrow{\sigma,d} q'$ when $q' \in \delta(q, \sigma, d)$, or $q \xrightarrow{\sigma} q', q''$ when $(q', q'') \in \delta(q, \sigma)$.

The run of the alternating automaton A on an input infinite binary tree $t \in T_\Sigma$ is defined in terms of a parity game. More precisely, consider an alternating automaton A and an infinite binary tree $t \in T_\Sigma$. The corresponding parity game $\mathcal{G}(A, t)$ is then defined as follows.

- the set V_0 is $\{0,1\}^* \times Q_\exists$
- the set V_1 is $\{0,1\}^* \times Q_\forall$
- from each vertex (v, q) and for each $(q', a) \in \delta(q, t(v))$, $((v, q), (va, q')) \in E$,
- for every vertex (v, q), $\Omega((v, q)) = \Omega(q)$.

We say that A accepts t iff player 0 has a winning strategy in the parity game $\mathcal{G}(A,t)$. The language recognized by A, denoted $L(A)$ is the set of trees accepted by A. We call a tree language L *regular* if $L = L(A)$, for some alternating tree automaton A. For every state q which is not the initial state q_I of the automaton A, by A_q we denote the automaton corresponding exactly to A except the fact that the initial state now is q and not q_I.

A *weak* alternating parity tree automaton A is an alternating parity automaton, satisfying the condition that if a state q is reachable from the state q' in the graph of the automaton, then $\Omega(q') \leq \Omega(q)$[9].

Because of the next result, we are assured of an exact correspondence of the expressive power of alternating automata and the modal μ-calculus with two existential modalities on binary trees:

THEOREM 1.35 ([96, 53]). *For every set of full binary trees L, the following two statements are equivalent:*

- *there is an alternating automaton A such that $L = L(A)$,*
- *there is a bi-modal μ-formula φ such that $\{(\mathcal{T}_t, \varepsilon) : t \in L\} = \|\varphi\|$*

This leads to the consequence that over binary trees, MSO, the bi-modal μ-calculus and alternating automata have the same expressive power.

Regular languages are closed under union, intersection and complementation. Thus, for every pair of alternating automata A and B, by $A \sqcap B$ (resp. $A \sqcup B$), we denote the automaton recognizing $L(A) \cap L(B)$ (resp. $L(A) \cup L(B)$), and by $\overline{A}$ we denote the the automaton recognizing the language $L(A)^{\complement}$. Note that the previous closure property holds also when considering weakly recognizable languages only.

1.8.2 Automata on transition systems

The modal μ-calculus was originally defined over arbitrary transition systems and not confined to binary trees. It is therefore interesting to know which is the automata-theoretic counterpart of μ-formulae on those structures. In [66], Janin and Walukiewicz define a notion of an automaton, called μ-automaton, that operates on transition systems and that corresponds exactly to the modal μ-calculus.

Note that an analogous equivalent characterization of μ-formulae in terms of alternating parity automata admitting runs over arbitrary transition systems is given by Wilke in [132].

A μ-automaton A over finite input alphabet Σ is a tuple $\langle Q, q_0, \delta, \Omega \rangle$ where Q is a finite set of states, $q_0 \in Q$ is the initial state, $\delta : Q \times \Sigma \to \wp\wp(Q)$ is the transition function, and $\Omega : Q \to \mathbb{N}$ is a bounded parity function. Given a pointed transition system $(\mathcal{T}, s_0)$ with $\lambda : \mathsf{S} \to \Sigma$, an A-game in $(\mathcal{T}, s)$ with starting position (s_0, q_0) is played between two players, Duplicator and Spoiler. The game

[9] Another way of characterizing weak alternating automata is in terms of weak parity game. More precisely, we can say that a weak alternating parity tree automaton is defined exactly as an alternating parity automaton, except that the run is given by a weak parity game. Call a weak alternating automaton defined in this alternative way a second-weak alternating automaton. Then it can then be shown (cf. [91]) that a tree language is recognized by a weak alternating automaton of index $[\iota, \kappa]$ iff it is recognized by a second-weak alternating automaton of index $[\iota, \kappa]$ (cf. section 1.9 for the definition of the index of a parity automaton).

is defined recursively as follows: if we are in position $(s, q) \in \mathsf{S} \times Q$, Duplicator has to make a move. She chooses a marking $m : Q \to \wp(\{s' \in \mathsf{S} : s \to^{\mathcal{T}} s'\})$ and then a description D in $\delta(q, \lambda(s))$. If $s' \in m(q)$, we say that s' is marked with q. The marking and the description have to satisfy the two following properties. Firstly, if $q' \in D$, there exists a successor s' of s that is marked with q'. Secondly, if s' is a successor of s there exists $q' \in D$ such that s' is marked with q'. After Duplicator's choice of a marking m, Spoiler plays a position (s', q') such that $s' \in m(q')$.

If a play is finite, we say that the player who cannot make a move loses. An infinite play $(s_0, q_0), (s_1, q_1), \dots$ is won by Duplicator if the greatest element of $\{\Omega(q) : q \text{ appears infinitely often in } q_0, q_1, \dots\}$ is even. We say that a pointed transition system $(\mathcal{T}, s_0)$ is accepted by A if Duplicator has a winning strategy in the A-game in $(\mathcal{T}, s_0)$ with starting position (s_0, q_0).

The set of pointed transition system over Σ accepted by A is denoted by $L(A)$. We say that a set of transition systems L is recognized by A if $L = L(A)$.

The correspondence between logic and automata is then assured by the following theorem:

Theorem 1.36 ([66]). *For every μ-automaton A over $\wp(\mathsf{Prop})$, there is a μ-formula φ such that $L(A) = \|\varphi\|$. Conversely, for every μ-formula φ, there is a μ-automaton A over $\wp(\mathsf{Prop})$ such that $L(A) = \|\varphi\|$.*

1.9 The Mostowski-Rabin index hierarchy

As for the modal μ-calculus, there is a natural hierarchy resulting from the structure of automata with the parity acceptance condition, the *hierarchy of the Mostowski-Rabin indices* of parity automata. This hierarchy orders languages according to the nesting of positive and negative conditions checked by the recognizing automaton. It has two main versions: weak, when considering only weak alternating automata; and strong, when considering all alternating automata. This hierarchy is believed to reflect the inherent computational complexity of the language, and therefore has attracted a lot of attention encouraged by the expectations of the verification community [32, 33, 75, 96, 100, 101, 102].

Formally, the (Mostowski-Rabin) index of the automaton A is the pair $[\iota, \kappa]$, where ι is the minimal value and κ is the maximal value of the priority function of A. An automaton with index $[\iota, \kappa]$ is often called a $[\iota, \kappa]$-automaton. For an index $[\iota, \kappa]$ we shall denote by $\overline{[\iota, \kappa]}$ the dual index, i.e. $\overline{[0, \kappa]} = [1, \kappa + 1]$ and $\overline{[1, \kappa]} = [0, \kappa - 1]$. Let us define the following partial order on indices:

$$[\iota, \kappa] \sqsubseteq [\iota', \kappa']$$

$$\text{iff}$$

$$\text{either } \{\iota, \dots, \kappa\} \subseteq \{\iota', \dots, \kappa'\} \text{ or } \{\iota + 2, \dots, \kappa + 2\} \subseteq \{\iota', \dots, \kappa'\}$$

In other words, one index is smaller than another if and only if it uses less priorities. This means that dual indices are not comparable. The (Mostowski-Rabin) index hierarchy for a certain class of automata consists of ascending sets (levels) of languages recognized by $[\iota, \kappa]$-automata.

$$\begin{tikzcd}
 & {[0,1]} \arrow[rr] \arrow[dr] & & {[0,2]} \arrow[rr] \arrow[dr] & & {[0,3]\ \cdots} \\
{[0,1]\cap[1,2]} \arrow[ur] \arrow[dr] & & {[0,2]\cap[1,3]} \arrow[ur] \arrow[dr] & & {[0,3]\cap[1,4]\ \cdots} & \\
 & {[1,2]} \arrow[rr] \arrow[ur] & & {[1,3]} \arrow[rr] \arrow[ur] & & {[1,4]\ \cdots}
\end{tikzcd}$$

Figure 1.6: The Mostowski-Rabin index hierarchy. Arrows stand for set-theoretic inclusion. We omit the first two trivial levels $[0,0]$ and $[1,1]$.

In the case of alternating tree automata (or μ-automata, when dealing with transition systems) the hierarchy is called the *strong* index hierarchy. For weak alternating automata, the hierarchy is called the *weak* index hierarchy.

As usual, the first fundamental question about the hierarchy concerns its strictness, i.e. the existence of languages recognized by a $[\iota,\kappa]$-automaton, but not by a $\overline{[\iota,\kappa]}$-automaton. Because of a nice exact correspondence between the index of an μ-automata and the alternation depth of the equivalent μ-formula (cf. [132]), Bradfield's result on the strictness of the fixpoint alternation hierarchy on arbitrary transition systems implies the strictness of the strong index hierarchy on those structures. For full binary trees, we have already seen that the strong index hierarchy was shown to be strict by Arnold in [7] and independently by Bradfield in [34]. It turns out that Arnold's proof can also be applied to weak alternating automata, showing therefore the strictness of the weak index hierarchy, a result already obtained by Mostowski [88] but using a reduction to a hierarchy of weak monadic second-order quantifiers formerly examined by Thomas [117]. A direct proof of the strictness of the weak index hierarchy can also be found in [101].

In the remainder of the section we give another proof of Arnold's results for binary trees in terms of Wadge games by following the path taken by Arnold and Niwinski in a recent work [10].

Consider the alphabet $\Sigma_{[\iota,\kappa]} = \{0,1\} \times \{\iota,\dots,\kappa\}$ with $\iota \in \{0,1\}$ and $\iota \leq \kappa$. Then, to every full binary tree t over $\Sigma_{[\iota,\kappa]}$, we associate a parity game $\mathcal{G}(t)$ as follows: a node v in the tree is a position for player 0 iff the first component of the node is 0, and the rank of the node corresponds to its second component.

The set $W_{[\iota,\kappa]}$ corresponds to the class of trees in $T_{\Sigma_{[\iota,\kappa]}}$ for which Player 0 has a winning strategy in the corresponding parity game $\mathcal{G}(t)$. For every index $[\iota,\kappa]$, the set $W_{[\iota,\kappa]}$ is called the *the game language of index* $[\iota,\kappa]$. All those languages are regular. It is enough to consider to automaton $A_{[\iota,\kappa]}$ defined by:

- the set of states is the set $\{q^k_\sigma : \sigma \in \Sigma_{[\iota,\kappa]}, k \in \{0,1\}\} \cup \{q_I\}$ partitioned by
  - $Q_\exists = \{q^k_{(i,j)} \in Q : i = 0, k \in \{0,1\}\} \cup \{q_I\}$ and
  - $Q_\forall = \{q^k_{i,j} \in Q : i = 1, k \in \{0,1\}\}$,
- the transition relation δ is defined by

– $\delta(q_I, \sigma, \varepsilon) = q^0_\sigma$, for all $\sigma \in \Sigma_{[\iota,\kappa]}$,
– $\delta(q^0_\sigma, \sigma, d) = q^1_\sigma$ for all $\sigma \in \Sigma_{[\iota,\kappa]}$ and $d \in \{0,1\}$,
– $\delta(q^1_\sigma, \sigma', \varepsilon) = q^0_{\sigma'}$, for all $\sigma, \sigma' \in \Sigma_{[\iota,\kappa]}$.
– the priority function is defined by $\Omega(q_I) = \iota$ and $\Omega(q^k_{(i,j)}) = j$, for every $(i,j) \in \Sigma_{[\iota,\kappa]}$ and $k \in \{0,1\}$, meaning that the index of $A_{[\iota,\kappa]}$ is $[\iota,\kappa]$.

Clearly $L(A_{[\iota,\kappa]}) = W_{[\iota,\kappa]}$.

Moreover, it is easy to see that every parity game $\mathcal{G}(A,t)$ can be effectively encoded from the root into a full binary tree $t^W \in W_{[\iota,\kappa]}$, with $[\iota,\kappa]$ being the index of the automata A, in such a way that Player 0 has a winning strategy in $\mathcal{G}(A,t)$ iff she has a winning strategy in $\mathcal{G}(t^W)$. Moreover, this "encoding" is continuous. More precisely, we have the following:

Proposition 1.37. *Let A be any alternating parity tree automaton of index $[\iota,\kappa]$ over Σ. Then $L(A) \leq_W W_{[\iota,\kappa]}$.*

Proof. Let $W'_{[\iota,\kappa]}$ be the class of all full finitely branching trees over $\Sigma_{[\iota,\kappa]}$ for which Player 0 has a winning strategy in the corresponding parity game $\mathcal{G}(t)$ Then, the winning strategy for Duplicator in the Wadge Game $\mathcal{G}_W(L(A), W'_{[\iota,\kappa]})$ is given by just playing at every turn the arena of the parity game $\mathcal{G}(A,t)$, where t is the finite tree precisely played by Spoiler at that moment. Therefore we have that $L(A) \leq_W W_{[\iota,\kappa]}$. But clearly $W'_{[\iota,\kappa]} \leq_W W_{[\iota,\kappa]}$. Indeed, the winning strategy for Duplicator in the Wadge Game $\mathcal{G}_W(W'_{[\iota,\kappa]}, W_{[\iota,\kappa]})$ is the following. First, given the (finite) finitely branching tree $t : \{\epsilon, 0, 1, \ldots, n\} \to \Sigma$, with $\mathsf{dom}(t) = \{\epsilon, 0, 1, \ldots, n\}$, let $f(t) : \{0,1\}^* \to \Sigma$ be the *binary encoding* of t given by $f(t)(\varepsilon) = f(t)(1^j) = t(\varepsilon)$, with $1 \leq j < n$, $f(t)(1^n) = t(n)$ and $f(t)(1^j 0) = t(j)$, for $0 \leq j < n$. We say that $\epsilon \in \mathsf{dom}(f(t))$ corresponds to $\epsilon \in \mathsf{dom}(t)$, $1^n \in \mathsf{dom}(f(t))$ corresponds to $n \in \mathsf{dom}(t)$ and that for every $1 \leq j < n$, $1^j 0 \in \mathsf{dom}(f(t))$ corresponds to $j \in \mathsf{dom}(t)$. We then define inductively for every finite finitely branching tree t and for every node $v \in \mathsf{dom}(t)$, the binary encoding $f(t)$ of t and which is the (unique) node in $\mathsf{dom}(f(t))$ corresponding to v.

Consider the following strategy for Duplicator in $\mathcal{G}_W(W'_{[\iota,\kappa]}, W_{[\iota,\kappa]})$:

(1) at the first round, copy Spoiler's move,

(2) for every round $n \geq 2$, for every terminal node v of the tree constructed by Spoiler after round $n-1$, if t is the tree constructed after Spoiler's turn at round n, then replace the terminal node corresponding to v with the binary encoding of $t.v$.

By definition of the binary encoding, this is a well-defined *winning* strategy for Duplicator. By transitivity of $\leq_W$ we therefore obtain that $L(A) \leq_W W_{[\iota,\kappa]}$. □

It is worth noticing that the game languages witness the strictness of fixpoint alternation hierarchy, and therefore of the strong index hierarchy (cf. [33, 7]). This result was recently strengthened by Arnold and Niwinski in [10]. In their paper, the authors show that the class of game languages form a hierarchy with respect to the Wadge reducibility. In the remaining part of the section we give a very short proof of the the same results by exploiting the game-theoretical characterization of Wadge reducibility. But before this, we explicitly establish the link between the Wadge hierarchy and the index hierarchy.

LEMMA 1.38. *If the Wadge hierarchy of tree game languages is strict, then the index hierarchy of alternating parity tree automata is strict too.*

Proof. Suppose that the Wadge hierarchy of for tree game languages is strict, but that the index hierarchy for alternating parity tree automata collapses. Assume it collapses to the $[\iota, n]$ level. Consider the automata recognizing the game tree language $W_{[\iota,n+1]}$. By hypothesis, there is an alternating tree automaton A of index $[\iota, n]$ recognizing the same language. By Proposition 1.37, we are able to construct a winning strategy for Spoiler in the Wadge Game $\mathcal{G}_W(W_{[\iota,n+1]}, W_{[\iota,n]})$. But this contradicts the strictness of the Wadge hierarchy of tree game languages. □

We can now give another proof of Arnold and Niwinski's result [10]. First, we have the following lemma:

LEMMA 1.39. *Spoiler has a winning strategy in both $\mathcal{G}_W(W_{[1,n+1]}, W_{[0,n]})$ and $\mathcal{G}_W(W_{[0,n]}, W_{[1,n+1]})$, for every $n \geq 0$.*

Proof. We only prove that Spoiler has a winning strategy in the Wadge game $\mathcal{G}_W(W_{[1,n+1]}, W_{[0,n]})$, the other case being identical. This is done by describing the winning strategy for Spoiler in this game. As a first move, Spoiler plays a finite binary tree over $\{[1, 1]\}$. Then the strategy goes as follows:

(1) if Duplicator skips, then for every terminal node add two children labelled by $[1, 1]$

(2) otherwise, at every terminal node, add two dual copies of what Duplicator has already played as successors.

Clearly this is a winning strategy for Spoiler in $\mathcal{G}_W(W_{[1,n+1]}, W_{[0,n]})$. □

PROPOSITION 1.40 ([10]). *The Wadge hierarchy for tree game languages is strict.*

Proof. It is trivial to verify that, for every n, Duplicator has a winning strategy in both $\mathcal{G}_W(W_{[0,n]}, W_{[0,n+1]})$ and $\mathcal{G}_W(W_{[0,n]}, W_{[1,n+2]})$, and dually in both $\mathcal{G}_W(W_{[1,n+1]}, W_{[1,n+2]})$ and $\mathcal{G}_W(W_{[1,n+1]}, W_{[0,n+1]})$. Therefore, if we show that Spoiler has a winning strategy in $\mathcal{G}_W(W_{[0,n+1]}, W_{[0,n]})$ and $\mathcal{G}_W(W_{[1,n+2]}, W_{[0,n]})$ (and in the dual case) we are done. We only prove that Spoiler has a winning strategy in $\mathcal{G}_W(W_{[0,n+1]}, W_{[0,n]})$, the other cases being identical. We do this by describing the winning strategy for Spoiler in this game. As a first move, she plays an finite binary tree over the alphabet $\{(1, 0)\}$. Then the strategy goes as follows:

(1) if Duplicator skips, then for every terminal node add two nodes labelled by $(1, 0)$.

(2) otherwise, by Lemma 1.39, apply the winning strategy for Spoiler in the Wadge game $\mathcal{G}_W(W_{[1,n+1]}, W_{[0,n]})$.

Clearly this is a winning strategy for Spoiler in $\mathcal{G}_W(W_{[0,n+1]}, W_{[0,n]})$. □

By applying Proposition 1.40 to Lemma 1.38, we can therefore obtain an alternative, very easy, proof of the strictness of the index hierarchy for alternating tree automata.

PROPOSITION 1.41. *The index hierarchy for alternating parity tree automata is strict.*

Mutatis mutandis, by considering the *weak* counterpart of game languages, the same argument yields the strictness of the index hierarchy of *weak* alternating parity automata.

1.10 Summarizing remarks

The modal μ-calculus is an extension of modal logic with greatest and least fixpoint operators. This formalism, introduced in Section 1.4, is used to described properties of transition systems (labelled directed graphs) or trees. However using the standard semantics is not the only way, and probably not the most transparent way, of interpreting the μ-calculus. It is for instance possible to associate a meaning to formulae in terms of parity games (Section 1.6), highlighting the fact that the automata counterpart of this logic is given by alternating parity automata (Subsection 1.8.1 in the case of binary trees, and Subsection 1.8.2 in the case of transition systems).

We have also seen that another way of speaking about properties of trees and transition systems consists in using monadic second order logic[10] (Section 1.3). But curiously, at least at first sight, we do not necessarily gain so much with this logic in term of expressiveness when compared to the modal μ-calculus. Indeed, on the one hand, since the work of Niwiniski [96, 97] we know that on binary trees the modal μ-calculus with two existential modalities corresponds exactly to monadic second order logic[11]. On the other hand Janin and Walukiewicz [66] have shown that all bisimulation invariant monadic second order properties of transition systems are exactly the ones which are expressible in the μ-calculus.

We are interested here in understanding the expressive power of some fragments the modal μ-calculus, and thus, since they are equivalent to the latter, of subclasses of alternating automata. A first, natural, measure of complexity is given by what is called the *fixpoint alternation hierarchy* (Section 1.5), resp. the *Mostowski-Rabin (index) hierarchy* for parity automata (Section 1.9). This hierarchy reflects the depth of nesting of greatest and least fixpoints, resp. of positive and negative conditions. In the case of transition systems, the fixpoint hierarchy has been proved to be strict by Bradfield [32, 33]. Since Bradfield [34] and Arnold [7], this hierarchy is also known to be strict on binary trees. Chapter 2 and Chapter 3 study the behavior of the fxipoint alternation hierarchy of the μ-calculus when we restrict the class of the considered models.

Infinite trees also play a central role in topology and descriptive set theory. In these settings, sets of trees can be compared via continuous, or Wadge, reductions (Subsection 1.2.2) or be classified by using for instance the classical Borel hierarchy (Subsection 1.2.1). It is this therefore very natural to ask whether the two points of view (combinatorial vs topological) have some connections,

[10] Notice that in this dissertation we have decided to use monadic second order logic with only the child relation, meaning that, as models of this logic, trees are seen as *unordered* trees.

[11] With the child relation *and* the order on siblings.

or if they measure two different kinds of complexity. While this issue will be more extensively treated in the Second Part of the thesis, and partly already in Chapter 4, a first known link is presented in Section 1.9. Bradfield [33] and Arnold [7] have indeed observed that the so called game tree languages testify of the strictness of fixpoint, resp. index, hierarchy. More precisely, they proved that those languages form a strict hierarchy for the alternation of fixpoints, resp. the Mostowski-Rabin indices. This result has then been strengthened by Arnold and Niwinski [10], who have shown that the class of game languages form a hierarchy with respect to the Wadge reducibility. In order to exemplify the use of Wadge Games in proofs, in the last part of this introductory chapter we presented an easy proof by way of Wadge Games of the previous theorem of Arnold and Niwinski and of the fact that it almost immediately implies the strictness of the index hierarchy. In Chapter 6, the two points of view – topological and in terms of the combinatorial complexity of the recognizing automaton – are studied and discussed for a restricted class of alternating parity automata capturing a very weak form of alternation.

Part I

The μ-Calculus on Restricted Classes of Models

Chapter 2

The Fixpoint Hierarchy on Reflexive, Transitive, and Transitive-Symetric Models

This chapter is based on a joint work with Luca Alberucci [3].

2.1 Preliminary remarks

Many natural properties such as "there is an infinite path" can be expressed by a modal μ-formula. Further, most such properties are given by formulae with alternation depth two. Nevertheless, it is mathematically interesting to see whether the expressive power of the modal μ-calculus increases with the alternation depth. If this is the case then we have a strict hierarchy otherwise we have a collapse at some point.

We already know that the semantical fixpoint alternation hierarchy over arbitrary transition systems is strict. Having seen the strictness over arbitrary transition systems, it can be asked whether this hierarchy remains strict for restricted classes of transition systems such as those that are reflexive or those that are transitive. In the case of transitive systems, to our knowledge, the first attempt to answer this question was presented by Lenzi in [81]. There, he shows that on transitive frames every Büchi automaton is equivalent to a co-Büchi automaton, and conversely[1]. This implies that over transitive frames the modal μ-calculus collapses to the level of Büchi automata (and to co-Büchi automata). Since, for instance, well-foundedness is not definable in the modal fragment, the hierarchy is non trivial. Thus, since over arbitrary graphs the intersection of Büchi and co-Büchi automata corresponds to the alternation-free fragment, Lenzi conjectured that the full modal μ-calculus collapses to the alternation-free fragment ([82]). It is interesting to note that Visser has shown in [122] that in the case of reflexive and transitive models, where well-foundedness is false and therefore can be expressed by a modal formula, the non-triviality

[1]A complete proof of this fact, extended to the class of finite simple graphs (a class which contains - modulo bisimulation - the class of finite transitive graphs) can be found in [44].

of the fixpoint hierarchy is testified by the formula stating the existence of an infinite path alternately labelled with p, $\neg p$, p, $\neg p$, etc.

In this chapter we answer positively Lenzi's conjecture for the class of all transitive systems by giving an explicit syntactical translation of the full modal μ-calculus into the alternation-free fragment. This result is first showed for finite transition systems and then generalized, by proving a finite model theorem, to all transitive systems. We also verify, again by giving an explicit syntactical translation, that if we add symmetry to transitivity the modal μ-calculus collapses to the purely modal fragment. Further, by adapting Arnold's proof for the general case, we show that the hierarchy remains strict over reflexive frames.

In the next section some useful finite model theorems are proved. In Sections 2.3 and 2.4 the collapse of the fixpoint hierarchy over transitive-symmetric and over transitive systems are proved. In Section 2.5 we finally prove the strictness of the hierarchy over the class of all reflexive transition systems.

2.2 Finite model theorems

In this section finite model theorems for the modal μ-calculus over the class of all reflexive and over the class of all transitive transition systems are proved. Let us first state the well-known finite model theorem for general transition systems.

THEOREM 2.1 ([74, 116]). *For all modal μ-formulae φ for which there is a transition system $\mathcal{T}$ and a state s in $\mathcal{T}$ such that $s \in \|\varphi\|_{\mathcal{T}}$, there is a finite transition system $\mathcal{T}^F$ and a state s^F such that $s^F \in \|\varphi\|_{\mathcal{T}^F}$.*

2.2.1 Finite model theorem for reflexive transition systems

Let φ be a μ-formula. By induction on the structure of φ we define the formula φ_{ref} as follows:

- $(\sim)p_{ref} \equiv (\sim)p$,
- $(\alpha \circ \beta)_{ref} = \alpha_{ref} \circ \beta_{ref}$ where $\circ \in \{\wedge, \vee\}$,
- $(\Box\alpha)_{ref} = \Box\alpha_{ref} \wedge \alpha_{ref}$,
- $(\Diamond\alpha)_{ref} = \Diamond\alpha_{ref} \vee \alpha_{ref}$, and
- $(\eta x.\alpha)_{ref} = \eta x.\alpha_{ref}$ where $\eta \in \{\mu, \nu\}$.

The next Lemma is by induction on the structure of the formula.

LEMMA 2.2. *Let $\mathcal{T}$ be a transition system and let $\mathcal{T}^{ref}$ be its reflexive closure. For all μ-formulae φ the following holds*

$$s \in \|\varphi_{ref}\|_{\mathcal{T}} \quad \textit{if and only if} \quad s \in \|\varphi\|_{\mathcal{T}^{ref}}.$$

With the help of this lemma we can easily prove the finite model property for reflexive transition systems.

THEOREM 2.3. *For all modal μ-formulae φ for which there is a reflexive transition system $\mathcal{T}$ and a state s in $\mathcal{T}$ such that $s \in \|\varphi\|_{\mathcal{T}}$ there is a finite reflexive transition system $\mathcal{T}^F$ and a state s^F such that $s^F \in \|\varphi\|_{\mathcal{T}^F}$.*

Proof. Let φ be a μ-formula and $\mathcal{T}$ a reflexive transition system with a state s such that $s \in \|\varphi\|_{\mathcal{T}}$. Since $\mathcal{T}$ is reflexive we have that $\mathcal{T} = \mathcal{T}^{ref}$ and therefore by Lemma 2.2 we have that

$$s \in \|\varphi_{ref}\|_{\mathcal{T}}.$$

By the general Finite Model Theorem 2.1 we get that there is a finite transition system $\mathcal{T}^F$ and a state s_F such that

$$s_F \in \|\varphi_{ref}\|_{\mathcal{T}^F}.$$

If we define $\mathcal{T}^{Fref}$ to be the reflexive closure of $\mathcal{T}^F$ by applying again Lemma 2.2 we get

$$s_F \in \|\varphi\|_{\mathcal{T}^{Fref}}$$

and we have found the finite reflexive model and a state in it satisfying φ. □

2.2.2 Finite model theorem for transitive transition systems

Let φ be a μ-formula. By induction on the structure of φ we define the formula φ_{tr} as follows:

- $(\sim)p_{tr} \equiv (\sim)p$,
- $(\alpha \circ \beta)_{tr} = \alpha_{tr} \circ \beta_{tr}$ where $\circ \in \{\wedge, \vee\}$,
- $(\Box\alpha)_{tr} = \nu x.\Box(\alpha_{tr} \wedge x)$,
- $(\Diamond\alpha)_{tr} = \mu x.\Diamond(\alpha_{tr} \vee x)$, and
- $(\eta x.\alpha)_{tr} = \eta x.\alpha_{tr}$ where $\eta \in \{\mu, \nu\}$.

As in the reflexive case, the next Lemma is proved by induction on the structure of the formula.

LEMMA 2.4. *Let $\mathcal{T}$ be a transition system and let $\mathcal{T}^{tr}$ be its transitive closure. For all μ-formulae φ the following holds*

$$s \in \|\varphi_{tr}\|_{\mathcal{T}} \quad \textit{if and only if} \quad s \in \|\varphi\|_{\mathcal{T}^{tr}}.$$

By using Lemma 2.4, mutatis mutandis, the proof of the finite model property for transitive transition systems is exactly the same as for Theorem 2.3.

THEOREM 2.5. *For all modal μ-formulae φ for which there is a transitive transition system $\mathcal{T}$ and a state s in $\mathcal{T}$ such that $s \in \|\varphi\|_{\mathcal{T}}$ there is a finite transitive transition system $\mathcal{T}^F$ and a state s^F such that $s^F \in \|\varphi\|_{\mathcal{T}^F}$.*

2.3 The transitive and symmetric case

In this section, we prove the collapse of the semantical modal μ-calculus hierarchy over $\mathbb{T}^{st}$ to the purely modal fragment. Let us begin with the following easy result. Because the successors of two nodes belonging to the same strongly connected component of a transitive transition system are the same, we immediately obtain that:

LEMMA 2.6. *Let $\mathcal{T}$ be a transitive transition system and let $s' \in \mathsf{scc}(s)$. For all μ-formulae φ we have that*

$$s \in \| \vartriangle \varphi\|_{\mathcal{T}} \quad \textit{if and only if} \quad s' \in \| \vartriangle \varphi\|_{\mathcal{T}}$$

where $\vartriangle \in \{\Box, \Diamond\}$.

The next theorem assures us that over transitive and symmetric transition systems, fixpoints can be eliminated.

THEOREM 2.7. *Let $\mathcal{T}$ be a transitive and symmetric transition system. For every well-named μ-formula φ, we have that*

$$\|\nu x.\varphi(x)\|_{\mathcal{T}} = \|\varphi(\varphi(\top))\|_{\mathcal{T}}.$$

Proof. The $\subseteq$ inclusion is clear. For the $\supseteq$ inclusion, define $A = \|\varphi(\varphi(\top))\|_{\mathcal{T}}$; by definition of greatest fixpoint it is enough to show that we have

$$A \subseteq \|\varphi(A)\|_{\mathcal{T}}. \tag{2.1}$$

First recall that we assume that $\nu x.\varphi(x)$ is well-named. This means that in the formula $\varphi(x)$ the variable x is in the scope of a modal operator and occurs only once in φ. Therefore, we can assume that φ is of the form $\beta(\vartriangle \alpha(x))$ where $\vartriangle \in \{\Diamond, \Box\}$. Moreover we have that $\vartriangle \alpha(x)$ and $\vartriangle \alpha(\varphi(x))$ occur only once in the formula tree of $\varphi(\varphi(x))$. Let $s \in A$, by Proposition 1.33 there is a winning strategy σ for Player 0 in the evaluation game $\mathcal{E}(\varphi(\varphi(x)), (\mathcal{T}[x \mapsto \mathsf{S}], s))$. Let π be an arbitrary play consistent with σ. If π reaches a vertex of the form $\langle \vartriangle \alpha(x), s'\rangle$ then the same play reaches a vertex of the form $\langle \vartriangle \alpha(\varphi(x)), s''\rangle$. Since σ is a winning strategy for Player 0 by Proposition 1.33 we have that

$$s'' \in \| \vartriangle \alpha(\varphi(x))\|_{\mathcal{T}[x\mapsto \mathsf{S}]} \quad \text{and} \quad s' \in \| \vartriangle \alpha(x)\|_{\mathcal{T}[x\mapsto \mathsf{S}]}.$$

Since $\mathcal{T}$ is transitive and symmetric it clearly holds that $s'' \in \mathsf{scc}(s')$ and, by applying Lemma 2.6, we have

$$s' \in \| \vartriangle \alpha(\varphi(x))\|_{\mathcal{T}[x\mapsto \mathsf{S}]}.$$

Hence, we have shown that for all plays π consistent with σ, if π reaches a vertex of the form $\langle \vartriangle \alpha(x), s'\rangle$ then, by Theorem 1.28, there is a winning strategy for Player 0 in the evaluation game $\mathcal{E}(\vartriangle \alpha(\varphi(x)), (\mathcal{T}[x \mapsto \mathsf{S}], s'))$. *A fortiori*, this implies that if π reaches a vertex of the form $\langle \vartriangle \alpha(x), s'\rangle$ then there is a winning strategy $\sigma_{\langle \vartriangle\alpha(x), s'\rangle}$ for Player 0 in $\mathcal{E}(\vartriangle \alpha(x), (\mathcal{T}[x \mapsto \|\varphi(x)\|_{\mathcal{T}[x\mapsto \mathsf{S}]}], s'))$. Therefore, since $\|\varphi(x)\|_{\mathcal{T}[x\mapsto \mathsf{S}]} \subseteq \mathsf{S}$, the strategy σ^* given by following σ but switching to the corresponding $\sigma_{\langle \vartriangle\alpha(x), s'\rangle}$ when a position of the form $\langle \vartriangle \alpha(x), s'\rangle$ is reached, is winning for Player 0 in the parity game $\mathcal{E}(\varphi(\varphi(x)), (\mathcal{T}[x \mapsto \mathsf{S}], s))$.

Let $B := \|\varphi(x)\|_{\mathcal{T}[x\mapsto \mathsf{S}]}$. By construction of σ^* we have that for all vertices of the form $\langle x, v\rangle$ which are reachable by σ^* it holds that $v \in B$. Then, by applying Lemma 1.30, σ^* can be converted into a winning strategy for Player 0 in the evaluation game $\mathcal{E}(\varphi(\varphi(x), (\mathcal{T}[x \mapsto B], s))$. By Theorem 1.28, we have that

$$s \in \|\varphi(\varphi(B))\|_{\mathcal{T}}$$

which can be reformulated as $s \in \|\varphi(\varphi(\varphi(\top)))\|_{\mathcal{T}}$ or $s \in \|\varphi(A)\|_{\mathcal{T}}$. Therefore, we have proved Equation 2.1 and completed the proof. □

Based on the previous result, we can therefore define a syntactical translation which associates to every μ-formula a modal formula and preserves logical equivalence on transitive and symmetric transition systems.

Definition 2.8. *The syntactical translation $(\centerdot)^t : \mathcal{L}_\mu \to \mathcal{L}_M$ is defined as the identity for all propositional variables, $\perp$ and $\top$, such that it distributes over boolean and modal connectives, and such that*

$$(\mu x.\varphi)^t = \big(\mathsf{wn}(\varphi(\varphi(\perp)))\big)^t \quad \textit{and} \quad (\nu x.\varphi)^t = \big(\mathsf{wn}(\varphi(\varphi(\top)))\big)^t.$$

Note that $(\varphi)^t$ is defined via an application of $(\centerdot)^t$ either to a strict subformula ψ of φ, or to a formula whose rank, by Lemma 1.21, is strictly smaller than the rank of φ. Thus $(\centerdot)^t$ terminates and is well-defined.

The next corollary proves that on transitive and symmetric models, the semantical hierarchy of the μ-calculus collapses to the class $\Delta_1^{\mu\mathbb{T}^{st}}$. Its proof goes by induction on the rank of a formula and uses Theorem 2.7.

Corollary 2.9. *On transitive and symmetric transition systems we have that*

$$\|\varphi\|_{\mathcal{T}} = \|\varphi^t\|_{\mathcal{T}}.$$

Example 2.10. If we look at our example from Section 3.4, for "always eventually p", we have that

$$\|\nu x.(\mu y.(p \vee \Diamond y)) \wedge \Box x)\|^{\mathbb{T}^{st}} = \|(p \vee \Diamond p) \wedge \Box(p \vee \Diamond p)\|^{\mathbb{T}^{st}}$$

and for "there is a path where p holds infinitely often", we have that

$$\begin{gathered}\|\nu x.\mu y.((p \vee \Diamond y)) \wedge \Diamond x)\|^{\mathbb{T}^{st}} \\ = \\ \|\big(p \vee \Diamond(p \wedge \Diamond((p \vee \Diamond(p \wedge \Diamond\top)) \wedge \Diamond\top)\big) \wedge \Diamond((p \vee \Diamond(p \wedge \Diamond\top)) \wedge \Diamond\top)\|^{\mathbb{T}^{st}}.\end{gathered}$$

Remark 2.11. Because the previous proof applies to any $\mathsf{S5}$ model, that is, for every $\mathcal{T} \in \mathbb{T}^{rst}$ we have that:

$$\|\varphi\|_{\mathcal{T}} = \|\varphi^t\|_{\mathcal{T}}$$

The fact that the modal μ-calculus hierarchy for $\mathsf{S5}$-models collapses to the pure modal fragment is indeed not surprising since for a $\mathsf{S5}$-formula φ there are only finitely many formulae with the same propositional variables which are not equivalent over $\mathbb{T}^{rst}$ and, therefore, it can easily be shown that for all $\nu x.\varphi(x) \in \mathcal{L}_\mu$ there is a $n \in \mathbb{N}$ such that $\|\varphi^n(\top)\|^{\mathbb{T}^{rst}} = \|\nu x.\varphi\|^{\mathbb{T}^{rst}}$. The existence of only finitely many non equivalent formulae follows from the fact that for all $\mathsf{S5}$-formulae φ there is a conjunctive modal normal form ψ such that $\psi \equiv \delta_1 \wedge \delta_2 \wedge .. \wedge \delta_n$ where $\delta \equiv \alpha \vee \Box\beta_1 \vee \Box\beta_2 \vee ...\Box\beta_n \vee \Diamond\gamma_1 \vee \Diamond\gamma_2 \vee .. \vee \Diamond\gamma_m$ and α, β_i and γ_j are propositional formulae[2].

[2]Cf. Chapter 5 in [60].

2.4 The transitive case

We show that the modal μ-calculus hierarchy over $\mathbb{T}^t$ collapses to the alternation-free fragment. This is done in four parts starting from subsection two. First, any modal μ-formula is reduced to a semantically equivalent formula $\tau(\varphi)$ such that normalized strategies on evaluation games – which will be introduced in the third subsection – have certain interesting properties. Then, we encode such normalized winning strategies in modal μ-formulae and, finally, we show the collapse for finite transitive transition system and, by using the previously proved finite model theorem, generalize it to all transitive transition systems.

In the next subsection some technical notions like the one of unfolding a formula in a model are introduced and some properties are proved.

2.4.1 Some technical preliminaries

Recall that we suppose all μ-formulae well-named. First we introduce the unfolding of a formula[3].

DEFINITION 2.12. *Let φ and ψ be μ-formulae such that $\{x_1, \ldots, x_n\} = X \subseteq \mathsf{bound}(\varphi)$. The* unfolding *of ψ in φ* over *X, $\mathsf{unf}^X_\varphi(\psi)$, is the formula defined recursively such that $\mathsf{unf}^\emptyset_\varphi(\psi) \equiv \psi$ and such that if X is of the form $\{x_1, \ldots, x_n\}$ then*

$$\mathsf{unf}^X_\varphi(\psi) \equiv \psi[x_1/\mathsf{unf}^{X^{-1}}_\varphi(\varphi_{x_1}), \ldots, x_n/\mathsf{unf}^{X^{-n}}_\varphi(\varphi_{x_n})]$$

where $X^{-i} = \{x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n\}$.

It can easily be seen that we have $X \cap \mathsf{free}(\mathsf{unf}^X_\varphi(\psi)) = \emptyset$.

In order to explain semantically the unfolding of a formula we introduce for each transition system $\mathcal{T}$ the *transition system induced by φ*, $\mathcal{T}^\varphi$. For every variable $x \in \mathsf{bound}(\varphi)$ we define a natural number $l(x)$ recursively such that $l(x) = 0$ if $\mathsf{free}(\varphi_x) \cap \mathsf{bound}(\varphi) = \emptyset$ and such that

$$l(x) = \max\{l(x_i) \;\; : \;\; x_i \in \mathsf{free}(\varphi_x) \cap \mathsf{bound}(\varphi)\} + 1$$

in the opposite case. For all transition systems $\mathcal{T}$ with valuation λ and for all $0 \leq i \leq \max\{l(x) \;\; : \;\; x \in \mathsf{bound}(\varphi)\} =: N$ we define new valuations λ^i and transition systems $\mathcal{T}^i$ such that $\lambda^0 = \lambda$ and $\mathcal{T}^0 = \mathcal{T}$, and such that $\mathcal{T}^{k+1}$ is identical to $\mathcal{T}^k$ except for the valuation λ^{k+1} which is defined as follows:

- $\lambda^{k+1}|_{\mathsf{Prop}\setminus\mathsf{bound}(\varphi)} = \lambda^k|_{\mathsf{Prop}\setminus\mathsf{bound}(\varphi)}$;
- if $x \in \mathsf{bound}(\varphi)$:

$$\lambda^{k+1}(x) = \begin{cases} \lambda^k(x) & \text{if } l(x) \neq k+1 \\ \|\varphi_x\|_{\mathcal{T}^k} & \text{if } l(x) = k+1. \end{cases}$$

We define $\mathcal{T}^\varphi$ to be $\mathcal{T}^N$ and $\lambda^\varphi = \lambda^N$. Note, that if we have a formula ψ such that $\mathsf{free}(\psi) \cap \mathsf{bound}(\varphi)$ is empty then, since the denotation of φ is independent of the valuation of the bound variables, we have $\|\psi\|_\mathcal{T} = \|\psi\|_{\mathcal{T}^\varphi}$. In particular, we have $\|\varphi\|_\mathcal{T} = \|\varphi\|_{\mathcal{T}^\varphi}$. Moreover note that for all $x_i \in \mathsf{bound}(\varphi)$ it holds that $\lambda^\varphi(x_i) = \|\varphi_{x_i}\|_{\mathcal{T}^\varphi}$.

[3] Since we restrict the unfolding of the free variables of a μ-formula to a certain fixed set of propositional variables, this notion generalizes in some sense the one of closure of a formula, introduced by Kozen in [73].

LEMMA 2.13. *For all formulae φ, all subformulae $\psi \leq \varphi$, all $X \subseteq \mathsf{bound}(\varphi)$, and all transition systems $\mathcal{T}$ we have that*

$$\|\psi\|_{\mathcal{T}^\varphi} = \|\mathsf{unf}_\varphi^X(\psi)\|_{\mathcal{T}^\varphi}.$$

Proof. By induction on the size of X. If X is empty, then by definition of unfolding we have that

$$\mathsf{unf}_\varphi^X(\psi) \equiv \psi$$

and the claim is trivial. For the inductive step, suppose that $X \cap \mathsf{free}(\psi)$ is the set $\{x_1, \ldots, x_m\}$. Hence, by definition we have

$$\mathsf{unf}_\varphi^X(\psi) \equiv \psi[x_1/\mathsf{unf}_\varphi^{X^{-1}}(\varphi_{x_1}), \ldots, x_m/\mathsf{unf}_\varphi^{X^{-m}}(\varphi_{x_m})].$$

Since $\mathsf{bound}(\psi) \cap \mathsf{free}(\varphi_{x_i}) = \emptyset$ and $\mathsf{free}(\mathsf{unf}_\varphi^{X^{-1}}(\varphi_{x_i})) \subseteq \mathsf{free}(\varphi_{x_i})$ for all i we get that $\mathsf{bound}(\psi) \cap \mathsf{free}((\mathsf{unf}_\varphi^{X^{-1}}(\varphi_{x_i})) = \emptyset$. Therefore, by induction hypothesis and Lemma 1.19.1 we get

$$\|\mathsf{unf}_\varphi^X(\psi)\|_{\mathcal{T}^\varphi} = \|\psi\|_{\mathcal{T}^\varphi[x_1 \mapsto \|\varphi_{x_1}\|_{\mathcal{T}^\varphi}, \ldots, x_m \mapsto \|\varphi_{x_m}\|_{\mathcal{T}^\varphi}]}.$$

Since for all x_i we have that $\lambda^\varphi(x_i) = \|\varphi_{x_i}\|_{\mathcal{T}^\varphi}$ we get

$$\|\mathsf{unf}_\varphi^X(\psi)\|_{\mathcal{T}^\varphi} = \|\psi\|_{\mathcal{T}^\varphi}.$$

□

The previous lemma tells us that on the transition system induced by φ the denotation of any subformula of φ and the denotation of any of its unfolding over φ are the same.

Other usefull properties of $\mathcal{T}^\varphi$ are summarized in the next lemma.

LEMMA 2.14. *Let $\mathcal{T} = (\mathsf{S}, \to^{\mathcal{T}}, \lambda^{\mathcal{T}})$ be a transition system, φ any μ-formula and $\psi \leq \varphi$. Then:*

(1) For every $X \subseteq \mathsf{bound}(\varphi)$ we have

$$\|\psi^{\mathsf{free}(X)}\|_{\mathcal{T}^\varphi} = \|\psi\|_{\mathcal{T}^\varphi}.$$

(2) For every $X_1, X_2 \subseteq \mathsf{bound}(\varphi)$, where $X_1 \cap X_2 = \emptyset$, we have

$$\|\mathsf{unf}_{\varphi^{\mathsf{free}(X_1)}}^{X_2} \psi^{\mathsf{free}(X_1)}\|_{\mathcal{T}^\varphi} = \|\psi\|_{\mathcal{T}^\varphi}.$$

(3) For every $X_1, X_2 \subseteq \mathsf{bound}(\varphi)$, where $X_1 \cap X_2 = \emptyset$, we have

$$\|\mathsf{unf}_\varphi^{X_2} \psi^{\mathsf{free}(X_1)}\|_{\mathcal{T}^\varphi} = \|\psi\|_{\mathcal{T}^\varphi}.$$

Proof. Part 1. By Lemma 1.19.2 and since $\|\varphi_x\|_{\mathcal{T}^\varphi} = \lambda^{\mathcal{T}^\varphi}(x)$ for every variable $x \in \mathsf{bound}(\varphi)$ we get

$$\|\psi^{\mathsf{free}(X)}\|_{\mathcal{T}^\varphi} = \|\psi^{\mathsf{free}(X)}[x_1/\varphi_{x_1}, \ldots, x_n/\varphi_{x_n}]\|_{\mathcal{T}^\varphi}.$$

The proof ends with a straightforward induction on the structure of ψ proving that for all transition systems $\mathcal{T}$ we have

$$\|\psi\|_{\mathcal{T}^\varphi} = \|\psi^{\mathsf{free}(X)}[x_1/\varphi_{x_1}, \ldots, x_n/\varphi_{x_n}]\|_{\mathcal{T}^\varphi}.$$

The only non trivial step is the one where ψ is of the form $\eta x.\alpha$ ($\eta \in \{\mu, \nu\}$). In this case, note that if any x_i appears free in α then x appears only bounded in φ_{x_i}.

Part 2. We prove the equation by induction on the size of X_2. If X_2 is empty, the equation holds by the previous point. For the inductive step, given $\{x_{i_1}, \ldots, x_{i_k}\} = X_2 \cap \mathsf{free}(\psi^{\mathsf{free}(X_1)})$, we have that by definition of unf the formula $\|\mathsf{unf}^{X_2}_{\varphi^{\mathsf{free}(X_1)}} \psi^{\mathsf{free}(X_1)}\|_{\mathcal{T}^\varphi}$ is equal to

$$\|\psi^{\mathsf{free}(X_1)}[x_{i_1}/\mathsf{unf}^{X_2^{-i_1}}_{\varphi^{\mathsf{free}(X_1)}}(\varphi^{\mathsf{free}(X_1)})_{x_{i_1}}, \ldots, x_{i_k}/\mathsf{unf}^{X_2^{-i_k}}_{\varphi^{\mathsf{free}(X_1)}}(\varphi^{\mathsf{free}(X_1)})_{x_{i_k}}]\|_{\mathcal{T}^\varphi}.$$

Since $\mathsf{free}(\mathsf{unf}^{X_2^{-i_1}}_{\varphi^{\mathsf{free}(X_1)}}(\varphi^{\mathsf{free}(X_1)})_{x_{i_1}}) \subseteq \mathsf{free}(\varphi^{\mathsf{free}(X_1)})_{x_{i_1}})$ and since we have that $\mathsf{free}(\varphi^{\mathsf{free}(X_1)})_{x_{i_1}}) \cap \mathsf{bound}(\psi^{\mathsf{free}(X_1)}) = \emptyset$ we get

$$\mathsf{free}(\mathsf{unf}^{X_2^{-i_1}}_{\varphi^{\mathsf{free}(X_1)}}(\varphi^{\mathsf{free}(X_1)})_{x_{i_1}}) \cap \mathsf{bound}(\psi^{\mathsf{free}(X_1)}) = \emptyset.$$

With Lemma 1.19.1 we get the equality with

$$\|\psi^{\mathsf{free}(X_1)}\|_{\mathcal{T}^\varphi[x_{i_1} \mapsto \|\mathsf{unf}^{X_2^{-i_1}}_{\varphi^{\mathsf{free}(X_1)}}(\varphi^{\mathsf{free}(X_1)})_{x_{i_1}}\|_{\mathcal{T}^\varphi}, \ldots, x_{i_k} \mapsto \|\mathsf{unf}^{X_2^{-i_k}}_{\varphi^{\mathsf{free}(X_1)}}(\varphi^{\mathsf{free}(X_1)})_{x_{i_k}}\|_{\mathcal{T}^\varphi}]}$$

and by induction hypothesis this expression is equal to

$$\|\psi^{\mathsf{free}(X_1)}\|_{\mathcal{T}^\varphi[x_{i_1} \mapsto \|\varphi_{x_{i_1}}\|_{\mathcal{T}^\varphi}, \ldots, x_{i_k} \mapsto \|\varphi_{x_{i_k}}\|_{\mathcal{T}^\varphi}]}.$$

Since in $\mathcal{T}^\varphi$ we have that $\lambda(x_{i_j}) = \|\varphi_{x_{i_j}}\|_{\mathcal{T}^\varphi}$ the last expression is equal to $\|\psi^{\mathsf{free}(X_1)}\|_{\mathcal{T}^\varphi}$.

Part 3. Suppose $\{x_{i_1}, \ldots, x_{i_k}\} = X_2 \cap \mathsf{free}(\psi^{\mathsf{free}(X_1)})$. Following the same argumentation as in part 2 we get that $\|\mathsf{unf}^{X_2}_\varphi \psi^{\mathsf{free}(X_1)}\|_{\mathcal{T}^\varphi}$ is equal to

$$\|\psi^{\mathsf{free}(X_1)}\|_{\mathcal{T}^\varphi[x_{i_1} \mapsto \|\mathsf{unf}^{X_2^{-i_1}}_\varphi \varphi_{x_{i_1}}\|_{\mathcal{T}^\varphi}, \ldots, x_{i_k} \mapsto \|\mathsf{unf}^{X_2^{-i_k}}_\varphi \varphi_{x_{i_k}}\|_{\mathcal{T}^\varphi}]}.$$

With Lemma 2.13 we get the equality with

$$\|\psi^{\mathsf{free}(X_1)}\|_{\mathcal{T}^\varphi[x_{i_1} \mapsto \|\varphi_{x_{i_1}}\|_{\mathcal{T}^\varphi}, \ldots, x_{i_k} \mapsto \|\varphi_{x_{i_k}}\|_{\mathcal{T}^\varphi}]},$$

and because in $\mathcal{T}^\varphi$ we have that $\lambda(x_{i_j}) = \|\varphi_{x_{i_j}}\|_{\mathcal{T}^\varphi}$ the last expression is equal to $\|\psi^{\mathsf{free}(X_1)}\|_{\mathcal{T}^\varphi}$ which by part 1 is equal to $\|\psi\|_{\mathcal{T}^\varphi}$. $\square$

LEMMA 2.15. *Let φ be a μ-formula and $\mathcal{T} = (\mathsf{S}, \to^{\mathcal{T}}, \lambda^{\mathcal{T}})$ be a transition system. For all $X \subseteq \mathsf{bound}(\varphi)$, all $x_k \in \overline{X} = \mathsf{bound}(\varphi) \setminus X$, all $\psi \leq \varphi$ and all $x \notin \overline{X}$ we have that*

(1) $\|\mathsf{unf}^X_{\varphi^{-\overline{X}}} \psi^{-\overline{X}}\|_{\mathcal{T}} \subseteq \|\mathsf{unf}^{X \cup \{x_k\}}_{\varphi^{-\overline{X}^{-k}}} \psi^{-\overline{X}^{-k}}\|_{\mathcal{T}}$,

(2) $\|\mathsf{unf}^X_{\varphi^{-\overline{X}}} \psi^{-\overline{X}}\|_{\mathcal{T}} \subseteq \|\mathsf{unf}^{\mathsf{bound}(\varphi)}_\varphi \psi\|_{\mathcal{T}}$,

(3) $\|\mathsf{unf}^X_{\varphi^{-\overline{X}}}(\varphi^{-\overline{X}})_x\|_{\mathcal{T}^\varphi} \subseteq \|\varphi_x\|_{\mathcal{T}^\varphi}$,

(4) $\|\psi^{-X}\|_{\mathcal{T}^{\varphi^{-X}}} \subseteq \|\psi\|_{\mathcal{T}^\varphi}$.

Proof. Suppose σ is a winning strategy for Player 0 in $\mathcal{E}(\mathsf{unf}^{X}_{\varphi^{-\overline{X}}}\psi^{-\overline{X}}, (\mathcal{T}, s))$. By definition, any winning play for Player 0 starting from $\langle \mathsf{unf}^{X}_{\varphi^{-\overline{X}}}\psi^{-\overline{X}}, s\rangle$ and compatible with σ do not reach a position of type $\langle \bot, s'\rangle$. Thus, this strategy determines a winning strategy for Player 0 in $\mathcal{E}(\mathsf{unf}^{X\cup\{x_k\}}_{\varphi^{-\overline{X}^{-k}}}\psi^{-\overline{X}^{-k}}, (\mathcal{T}, s))$. Part 1 is then obtained by applying Theorem 1.28. Part 2 follows by a finite reiteration of part 1. In order to obtain part *3* just apply Lemma 2.13 to part 2 and note that, since $x \notin \overline{X}$, $(\varphi^{-\overline{X}})_x \equiv (\varphi_x)^{-\overline{X}}$. Part *4* is also a consequence of an application of Lemma 2.13 to part 2. $\square$

2.4.2 A first reduction

We begin with a lemma whose proof is standard.

LEMMA 2.16. *Let $\mathcal{T}$ be a transitive transition system and let s, s' be two states such that $s \to^{\mathcal{T}} s'$. For all μ-formulae φ we have that*

$$s \in \|\Box\varphi\|_{\mathcal{T}} \quad\Longrightarrow\quad s' \in \|\Box\varphi\|_{\mathcal{T}} \quad \textit{and}$$

$$s' \in \|\Diamond\varphi\|_{\mathcal{T}} \quad\Longrightarrow\quad s \in \|\Diamond\varphi\|_{\mathcal{T}}.$$

The next theorem assures us that over transitive transition systems, a greatest fixpoint can be eliminated when the bounded variable is guarded by an universal modality.

THEOREM 2.17. *Let $\mathcal{T}$ be a transitive transition system and let $\nu x.\varphi(x)$ be a well-named formula such that x is weakly universal. We have that*

$$\|\nu x.\varphi(x)\|_{\mathcal{T}} = \|\varphi(\varphi(\top))\|_{\mathcal{T}}.$$

Proof. The $\subseteq$ inclusion is clear. For the $\supseteq$ inclusion, define $A = \|\varphi(\varphi(\top))\|_{\mathcal{T}}$; by definition of greatest fixpoint it is enough to show that we have

$$A \subseteq \|\varphi(A)\|_{\mathcal{T}}. \tag{2.2}$$

First, recall that we assume that $\nu x.\varphi(x)$ is well-named. This means that in the formula $\varphi(x)$ the variable x is in the scope of a modal operator and, therefore, we can assume that φ is of the form $\beta(\Box\alpha(x))$. Moreover x occurs only once in φ. This implies that $\Box\alpha(x)$ and $\Box\alpha(\varphi(x))$ occur only once in the formula tree of $\varphi(\varphi(x))$. Let $s \in A$, by Theorem 1.28 there is a winning strategy σ for Player 0 in the evaluation game $\mathcal{E}(\varphi(\varphi(x)), (\mathcal{T}[x \mapsto \mathsf{S}], s))$. Let π be an arbitrary play consistent with σ. If π reaches a vertex of the form $\langle\Box\alpha(x), s'\rangle$ then the same play passes a vertex of the form $\langle\Box\alpha(\varphi(x)), s''\rangle$, with $\Box\alpha(x) \leq \Box\alpha(\varphi(x))$ and s' reachable from s'' in $\mathcal{T}[x \mapsto \mathsf{S}]$. Since σ is a winning strategy for Player 0 by Proposition 1.33 we have that

$$s'' \in \|\Box\alpha(\varphi(x))\|_{\mathcal{T}[x\mapsto\mathsf{S}]} \quad \text{and} \quad s' \in \|\Box\alpha(x)\|_{\mathcal{T}[x\mapsto\mathsf{S}]}.$$

Since $\mathcal{T}[x \mapsto \mathsf{S}]$ is transitive we have that $s'' \to^{\mathcal{T}[x\mapsto\mathsf{S}]} s'$ and, by applying Lemma 2.16, we have

$$s' \in \|\Box\alpha(\varphi(x))\|_{\mathcal{T}[x\mapsto\mathsf{S}]}.$$

Hence, we have shown that for all plays π consistent with σ, if π reaches a vertex of the form $\langle\Box\alpha(x), s'\rangle$ then, by Theorem 1.28, there is a winning strategy for

Player 0 in the evaluation game $\mathcal{E}(\Box\alpha(\varphi(x)), (\mathcal{T}[x \mapsto \mathsf{S}], s'))$. A fortiori, this implies that if π reaches a vertex of the form $\langle \Box\alpha(x), s' \rangle$ then there is a winning strategy $\sigma_{\langle \Box\alpha(x), s' \rangle}$ for Player 0 in $\mathcal{E}(\Box\alpha(x), (\mathcal{T}[x \mapsto \|\varphi(x)\|_{\mathcal{T}[x\mapsto \mathsf{S}]}], s'))$. Therefore, since $\|\varphi(x)\|_{\mathcal{T}[x\mapsto \mathsf{S}]} \subseteq \mathsf{S}$, the strategy σ^* given by following σ but switching to the corresponding $\sigma_{\langle \Box\alpha(x), s' \rangle}$ when a position of the form $\langle \Box\alpha(x), s' \rangle$ is reached, is winning for Player 0 in the parity game $\mathcal{E}(\varphi(\varphi(x)), (\mathcal{T}[x \mapsto \mathsf{S}], s))$. Let $B := \|\varphi(x)\|_{\mathcal{T}[x\mapsto \mathsf{S}]}$. By construction of σ^* we have that for all vertices of the form $\langle x, v \rangle$ which are reachable by σ^* it holds that $v \in B$. Then, by applying Lemma 1.30, σ^* can be converted into a winning strategy for Player 0 in $\mathcal{E}(\varphi(\varphi(x), (\mathcal{T}[x \mapsto B], s))$. By Theorem 1.28, we have that

$$s \in \|\varphi(\varphi(B))\|_{\mathcal{T}[x\mapsto \mathsf{S}]}$$

which can be reformulated as $s \in \|\varphi(\varphi(\varphi(\top)))\|_{\mathcal{T}}$ or $s \in \|\varphi(A)\|_{\mathcal{T}}$. Therefore, we have proved Equation 2.2 and completed the proof. ☐

Definition 2.18. *The syntactical translation $\tau : \mathcal{L}_\mu \to \mathcal{L}_\mu$ is defined recursively on the structure of the formula such that $\tau(p) = p$, $\tau(\neg p) = \neg p$, $\tau(\bot) = \bot$ and $\tau(\top) = \top$, such that it distributes over boolean and modal connectives, and such that*

- $\tau(\mu x.\varphi) = \tau\big(\mathsf{wn}(\varphi(\varphi(\bot)))\big)$, *$x$ is weakly existential in φ*
- $\tau(\mu x.\varphi) = \mathsf{wn}(\mu x.\tau(\varphi))$, *$x$ is universal in φ*
- $\tau(\nu x.\varphi) = \tau\big(\mathsf{wn}(\varphi(\varphi(\top)))\big)$, *$x$ is weakly universal in φ*
- $\tau(\nu x.\varphi) = \mathsf{wn}(\nu x.\tau(\varphi))$, *$x$ is existential in φ.*

First, note that in each defining clause $\tau(\varphi)$ is defined via an application of τ to a formula whose rank, by Lemma 1.21, is strictly smaller than the rank of φ. Thus τ terminates and is well-defined. Note also, that it can be proved by induction on the structure of φ that all variables which are existential (resp. universal) in φ are weakly existential (resp. universal) in $\tau(\varphi)$ and that therefore for all $\mu x.\alpha \leq \tau(\varphi)$ we have that x is weakly universal and for all $\nu x.\alpha \leq \tau(\varphi)$ we have that x is weakly existential.

Corollary 2.19. *On transitive transition systems we have that*

$$\|\varphi\|_{\mathcal{T}} = \|\tau(\varphi)\|_{\mathcal{T}}.$$

Proof. By induction on $\mathsf{rank}(\varphi)$. If $\mathsf{rank}(\varphi) = 1$ or $\mathsf{rank}(\varphi)$ is a successor ordinal the proof is straightforward. If $\mathsf{rank}(\varphi)$ is a limit ordinal then φ is of the form $\eta x.\alpha$. We distinguish four cases. If φ is of the form $\nu x.\alpha$ and x is existential in φ the induction step is straightforward. Similarly for φ of the form $\mu x.\alpha$ and x is universal in φ. If φ is of the form $\nu x.\alpha$ and x is in the scope of a $\Box$ in φ the induction step follows from Theorem 2.17 and Lemma 1.20. In the third case, if φ is of the form $\mu x.\alpha$ and x is in the scope of a $\Diamond$ in φ then $\neg\varphi$ is of the form $\nu x.\neg\alpha[x/\neg x]$ and x is in the scope of a $\Box$ in $\neg\varphi$. Since in this case $\mathsf{rank}(\varphi) = \mathsf{rank}(\neg\varphi)$ we can apply the induction step as in the third case. ☐

2.4.3 Normalizing the winning strategies

Let $\mathcal{T}$ be a transitive transition system and φ a μ-formula. Consider an arbitrary (memoryless) strategy σ for Player 0, not necessarily winning. We define the *restriction of* $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$ *on* σ, denoted by $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$, as follows:

- The set of positions $V|_\sigma$ of the restriction is given by all nodes which are the positions of some play compatible with σ starting from position $\langle \varphi, s_0 \rangle$,
- The arena of $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$ is the triple $\langle V_0|_\sigma, V_1|_\sigma, E|_\sigma \rangle$ where:
  (1) $V_0|_\sigma = \emptyset$,
  (2) $V_1|_\sigma = V|_\sigma$,
  (3) if $\langle \psi, s \rangle \in V|_\sigma \cap V_1$ then $E|_\sigma(\langle \psi, s \rangle) = E(\langle \psi, s \rangle)$, and
  (4) if $\langle \psi, s \rangle \in V|_\sigma \cap V_0$ then $E|_\sigma(\langle \psi, s \rangle) = \{\sigma(\langle \psi, s \rangle)\}$.
- The ranking function $\Omega|_\sigma$ is given by the restriction of Ω on $V|_\sigma$.

Note, that if $\mathcal{T}$ is finite then $V|_\sigma$ is finite, too. We have that in $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$ the only Player who can move is Player 1. This can be done because the moves for Player 0 are already completely determined by the (memoryless) strategy σ. Clearly, any play in $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$ is a play in $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$ compatible with σ. We say that a play π in $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$ is winning for Player 0 if and only if the play π is winning for Player 0 in $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$. If σ is a winning strategy for Player 0 then any play in $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$ is winning for Player 0.

Example 2.20. Consider the arena of Example 1.29 depicted in Figure 1.5. The non-dotted part of the picture represents the arena of a restricted evaluation game.

We now formally define a measure, which roughly associates to every node of a restricted evaluation game the height of the corresponding strongly connected component in the scc tree.

Definition 2.21. *Let $\mathcal{T}$ be a finite transitive transition system and φ a μ-formula. Suppose there is a winning strategy σ for Player 0 in the parity game $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$. Then, for every position $\langle \psi, s \rangle$ of $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$, we define a measure $d(\langle \psi, s \rangle)$. We distinguish two cases in the definition, depending on whether the strongly connected component $\mathsf{scc}(\langle \psi, s \rangle)$ of $\langle \psi, s \rangle$ in $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$ is empty or not:*

(1) $\mathsf{scc}(\langle \psi, s \rangle) = \emptyset$:

$$d(\langle \psi, s \rangle) = \begin{cases} 0 & \textit{if } E|_\sigma(\langle \psi, s \rangle) = \emptyset \\ \max\{d(\langle \phi, s' \rangle) : \langle \phi, s' \rangle \in E|_\sigma(\langle \psi, s \rangle)\} + 1 & \textit{else} \end{cases}$$

(2) $\mathsf{scc}(\psi, s) \neq \emptyset$:

$$d(\langle \psi, s \rangle) = 0 \quad \textit{if} \quad \bigcup\{E|_\sigma(\langle \alpha, s \rangle) : \langle \alpha, s \rangle \in \mathsf{scc}(\psi, s)\} \setminus \mathsf{scc}(\psi, s) = \emptyset,$$

else

$$d(\langle \psi, s \rangle) = \max\{d(\langle \phi, s' \rangle) : \langle \phi, s' \rangle \notin \mathsf{scc}(\langle \psi, s \rangle) \textit{ and exists } \langle \xi, s'' \rangle \in \mathsf{scc}(\langle \psi, s \rangle) \textit{ with } \langle \phi, s' \rangle \in E|_\sigma(\langle \xi, s'' \rangle)\} + 1.$$

For all finite transition systems d is a well-defined measure. Indeed, if we have a finite transition system we obviously have a finite arena which can be collapsed to a finite and well-founded graph by identifying all vertices in the arena which are in the same strongly connected component. It is clear that on finite and well-founded graphs d is well-defined. By noticing that on the original arena the measure of a vertex corresponds to its measure of the collapsed arena we get that d is well-defined.

LEMMA 2.22. *Let $\mathcal{T}$ be a finite transitive transition system and $\varphi \in \Sigma_2^\mu$. Suppose there is a winning strategy σ for Player 0 in the parity game $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$. If $y \in \mathsf{bound}(\varphi)$ is a μ-variable, then for every position $\langle y, s\rangle \in V|_\sigma$, we have that $\mathsf{scc}(\langle y, s\rangle) = \emptyset$.*

Proof. If $\mathsf{scc}(\langle y, s\rangle) \neq \emptyset$ then Player 1 can find a play π in the restricted game $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$ where $\langle y, s\rangle$ occurs infinitely often, since in $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$ only Player 1 moves and therefore can stay as long as he wants in a strongly connected component. Recall that $\varphi \in \Sigma_2^\mu$. Thus, there is no ν-variable free in φ_y. Moreover, if $y \in \mathsf{free}(\varphi_x)$, where x is an arbitrary ν-variable, we have that $\Omega(\langle y, s\rangle)$ is strictly greater than the priorities of x and φ_x positions. Therefore, π is winning for Player 1. But since π is compatible with σ, the play must be winning for Player 0, too. A contradiction. □

Given the restriction of $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$ on a winning strategy σ and the measure d on $V|_\sigma$ we define the *normalization of* σ, denoted by σ^{N}, as follows:

- For all positions $\langle \Diamond\beta, s'\rangle \in V|_\sigma$ we have that
$$\sigma^{\mathsf{N}}(\langle \Diamond\beta, s'\rangle) = \sigma(\langle \Diamond\beta, s'\rangle),$$
if $d(\sigma(\langle \Diamond\beta, s'\rangle))$ is the minimum of the set $\{d(\langle \beta, \bar{s}\rangle) : \langle \beta, \bar{s}\rangle \in E(\langle \Diamond\beta, s'\rangle)\}$, where any $\langle \beta, \bar{s}\rangle \in E(\langle \Diamond\beta, s'\rangle)$ has to be reachable from $\langle \Diamond\beta, s'\rangle$ in $V|_\sigma$. Else
$$\sigma^{\mathsf{N}}(\langle \Diamond\beta, s'\rangle) = \langle \beta, s''\rangle,$$
where $\langle \beta, s''\rangle \in E(\langle \Diamond\beta, s'\rangle)$ is a vertex reachable from $\langle \Diamond\beta, s'\rangle$ in $V|_\sigma$ such that $d(\langle \beta, s''\rangle)$ is the minimum of the set $\{d(\langle \beta, \bar{s}\rangle) : \langle \beta, \bar{s}\rangle \in E(\langle \Diamond\beta, s'\rangle)\}$ where any $\langle \beta, \bar{s}\rangle \in E(\langle \Diamond\beta, s'\rangle)$ has to be reachable from $\langle \Diamond\beta, s'\rangle$ in $V|_\sigma$.
- If ψ is not of the form $\Diamond\beta$ then we simply set $\sigma^{\mathsf{N}}(\langle \psi, s\rangle) = \sigma(\langle \psi, s\rangle)$.

Intuitively, given a winning strategy σ for Player 0 on $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$, the normalized strategy σ^{N} for Player 0 is given by adapting σ such that for all vertexes of the form $\langle \Diamond\beta, s'\rangle$ Player 0 moves to a vertex $\langle \beta, s''\rangle$ whose measure is the minimal measure of all positions of the type $\langle \beta, \bar{s}\rangle$ reachable from $\langle \Diamond\beta, s'\rangle$ which are still winning in $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$. We have the following lemma.

LEMMA 2.23. *Let $\mathcal{T}$ be a finite transitive transition system. If σ is a winning strategy for Player 0 on $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$ then σ^{N} is a winning strategy for Player 0 on $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$, too.*

Proof. First we prove the following claim:

Claim : $E|_\sigma$ and $E|_{\sigma^{\mathsf{N}}}$ coincide on every non empty scc of $\mathcal{E}|_{\sigma^{\mathsf{N}}}(\varphi, (\mathcal{T}, s_0))$.

The proof of the claim goes as follows. If there is no position of the form $\langle \Diamond\beta, s\rangle$ in a scc of $\mathcal{E}|_{\sigma^{\mathsf{N}}}(\varphi, (\mathcal{T}, s_0))$, the claim is trivially verified. Consider now an arbitrary $\mathsf{scc}(\langle\Diamond\beta, s\rangle)$ of $\mathcal{E}|_{\sigma^{\mathsf{N}}}(\varphi, (\mathcal{T}, s_0))$. Let $\langle\psi, t\rangle \in \mathsf{scc}(\langle\Diamond\beta, s\rangle)$, in order to prove the claim we have to show that $E|_{\sigma^{\mathsf{N}}}(\langle\psi, t\rangle) = E|_{\sigma}(\langle\psi, t\rangle)$.

(a) If ψ is not of the form $\Diamond\alpha$, then $E|_{\sigma^{\mathsf{N}}}(\langle\psi, t\rangle) = E|_{\sigma}(\langle\psi, t\rangle)$.

(b) For the case where $\psi = \Diamond\alpha$ then suppose that $E|_{\sigma^{\mathsf{N}}}(\langle\psi, t\rangle) \neq E|_{\sigma}(\langle\psi, t\rangle)$ and that $E|_{\sigma^{\mathsf{N}}}(\langle\psi, t\rangle) = \{\langle\alpha, t'\rangle\}$. Note, that by construction of σ^{N} the position $\langle\alpha, t'\rangle$ is the only successor of $\langle\psi, t\rangle$. Since $E|_{\sigma^{\mathsf{N}}}(\langle\psi, t\rangle) \neq E|_{\sigma}(\langle\psi, t\rangle)$ it must hold that
$$d(\langle\alpha, t'\rangle) < d(\langle\psi, t\rangle) \tag{2.3}$$
where d is the depth defined on $E|_{\sigma}(\varphi, (\mathcal{T}, s_0))$. Since $\mathsf{scc}(\langle\Diamond\alpha, t\rangle) \neq \emptyset$ and since $\langle\alpha, t'\rangle$ is the only position reachable in one step from $\langle\Diamond\alpha, t\rangle$ we have that $\langle\alpha, t'\rangle \in \mathsf{scc}(\langle\Diamond\alpha, t\rangle)$ and therefore that $\langle\Diamond\alpha, t\rangle$ is reachable from $\langle\alpha, t'\rangle$ in $\mathcal{E}|_{\sigma^{\mathsf{N}}}(\varphi, (\mathcal{T}, s_0))$. Since reachability in $E|_{\sigma^{\mathsf{N}}}(\varphi, (\mathcal{T}, s_0))$ implies reachability in $E|_{\sigma}(\varphi, (\mathcal{T}, s_0))$ we can infer that $d(\langle\alpha, t'\rangle) \geq d(\langle\Diamond\alpha, t\rangle)$, where d is the depth defined on $E|_{\sigma}(\varphi, (\mathcal{T}, s_0))$. This is a contradiction to point 2.3 and therefore the claim is proved.

Consider an arbitrary play π in the graph of $\mathcal{E}|_{\sigma^{\mathsf{N}}}(\varphi, (\mathcal{T}, s_0))$. If π is finite, then by construction of the normalized arena the play is winning for Player 0. If π is infinite then from a certain position, say $\langle\alpha, t\rangle$, we are in a scc of $\mathcal{E}|_{\sigma^{\mathsf{N}}}(\varphi, (\mathcal{T}, s_0))$. But then by the previous claim after $\langle\alpha, t\rangle$ the strategies of σ and σ^{N} coincide. Since by construction of σ^{N} the position $\langle\alpha, t\rangle$ is winning in $\mathcal{E}|_{\sigma}(\varphi, (\mathcal{T}, s_0))$ the highest priority appearing infinitely often in π must be even and, therefore π is a winning play in $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$ for Player 0. $\square$

In the next lemma we prove that, when considering Σ_2^{μ}-formulae, normalized strategies have a nice and very usefull property.

Lemma 2.24. *Let $\mathcal{T}$ be a finite transitive transition system and $\varphi \in \Sigma_2^{\mu}$ such that all ν-variables are weakly existential. Let σ^{N} be a normalized winning strategy for Player 0 on $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$. Consider a position $\langle x, s_1\rangle$ in $\mathcal{E}|_{\sigma^{\mathsf{N}}}(\varphi, (\mathcal{T}, s_0))$ where $x \in \mathsf{bound}(\varphi)$ is a ν-variable. Then, if there is a position $\langle y, s_2\rangle$ reachable from $\langle x, s_1\rangle$ in $V|_{\sigma^{\mathsf{N}}}$, where $y \in \mathsf{bound}(\varphi)$ is a μ-variable, then there is no position $\langle x, s_3\rangle$ reachable from $\langle y, s_2\rangle$ in $V|_{\sigma^{\mathsf{N}}}$.*

Proof. Suppose there is a play π consistent with σ^{N} such that we have the following regenerations: $\langle x, s_1\rangle$ then $\langle y, s_2\rangle$ and then $\langle x, s_3\rangle$, where x is a ν-variable and y a μ-variable. Note that, since $\varphi \in \Sigma_2^{\mu}$, we have that $y \in \mathsf{free}(\varphi_x)$, and therefore $\varphi_x < \varphi_y$. This implies that in π we must have positions of the form $\langle\Diamond(\beta(x)), s_1'\rangle$ and $\langle\beta(x), s_1''\rangle$ before $\langle x, s_1\rangle$, and also positions of the form $\langle\Diamond(\beta(x)), s_3'\rangle$ and $\langle\beta(x), s_3''\rangle$ before $\langle x, s_3\rangle$ but after $\langle y, s_2\rangle$. By construction of normalized strategy and by the transitivity of the transition system $\mathcal{T}$ it holds that $d(\langle\beta(x), s_1''\rangle) = d(\langle\beta(x), s_3''\rangle)$ but also that $d(\langle\beta(x), s_1''\rangle) = d(\langle\beta(x), s_3''\rangle) = d(\langle y, s_2\rangle)$. This implies that $\mathsf{scc}(\langle y, s_2\rangle) \neq \emptyset$. Because σ^{N} is a winning strategy for Player 0, by Lemma 2.22 we get the desired contradiction. $\square$

We immediately can restate the previous lemma as the following theorem.

Theorem 2.25. *Suppose a finite transitive transition system $\mathcal{T}$, a formula $\varphi \in \Sigma_2^{\mu}$ such that all ν-variables are weakly existential and a normalized winning strategy, σ^{N}, of Player 0 in $\mathcal{E}(\varphi, (\mathcal{T}, s))$. If in a play π consistent with σ^{N} there* *is a regeneration of a ν-variable x then either there is no more regeneration of a μ-variable after the first regeneration of x or, if there is such a regeneration of a μ-variable, then after this position there is no more regeneration of x.*

2.4.4 Encoding normalized winning strategies

In the last theorem of the previous subsection, we essentially verified that given a finite transitive transition system $\mathcal{T}$, a formula $\varphi \in \Sigma_2^\mu$ such that all ν-variables are weakly existential and a normalized winning strategy of Player 0 in $\mathcal{E}(\varphi, (\mathcal{T}, s))$, every play consistent with this strategy does not have any "alternation" between positions with a least fixpoint formula and position with a greatest fixpoint formula. This fact suggests the idea that, if we are able to construct a formula ψ based on the original formula φ *and* on the general behavior of normalized winning strategies, on transitive models ψ should be proven to be equivalent to the original μ-formula *and* to be alternation free.

This is the approach we pursue in the present and next subsections. More precisely, in Definition 2.26 we define the formulae $\mathsf{NS}^-_\varphi(X', y)$ and $\mathsf{NS}^+_\varphi(x, X')$ that will be used to encode the main properties of the normalization of winning strategies of φ given by Theorem 2.25. Encoding, in this context, will be formalized in the two main Lemmas of the section, Lemmas 2.28 and 2.29. The intuition behind these formulae is the following:

- $\mathsf{NS}^-_\varphi(X', y)$ reflects the fact that we are regenerating y and any ν-variable regenerated afterwards will be an element of X',
- $\mathsf{NS}^+_\varphi(x, X')$ reflects the fact that we are regenerating x and if we regenerate any μ-variable then afterwards any ν-variable regenerated will be an element of X'.

Then in the next subsection we show how to construct from a given formula $\varphi \in \Sigma_2^\mu$ an equivalent alternation free formula based on the formulae $\mathsf{NS}^-_\varphi(X', y)$ and $\mathsf{NS}^+_\varphi(x, X')$

In the sequel, in order to ease notation, we write a formula of the form $\varphi_y^{\mathsf{free}(X)}$ instead of $(\varphi^{\mathsf{free}(X)})_y$.

DEFINITION 2.26. *Let φ be a Σ_2^μ-formula. Let $Y = \{y_1, \ldots, y_k\}$ be the set of all μ-variables in φ and X be the set of all ν-variables in φ. For all subsets of $X' \subset X$, all ν-variables x such that $x \in X/X'$ and all μ-variables y we define the formulae $\mathsf{NS}^+_\varphi(x, X')$ and $\mathsf{NS}^-_\varphi(X', y)$ recursively on the size of X' such that*

$$\mathsf{NS}^-_\varphi(\emptyset, y) \equiv \mathsf{unf}^Y_{\varphi^{-X}}((\varphi^{-X})_y)$$

and, such that

$$\mathsf{NS}^+_\varphi(x, \emptyset) \equiv (\mathsf{unf}^X_{\varphi^{\mathsf{free}(Y)}} \varphi_x^{\mathsf{free}(Y)})[y_1/\mathsf{NS}^-_\varphi(\emptyset, y_1), \ldots, y_k/\mathsf{NS}^-_\varphi(\emptyset, y_k)].$$

If $X' = \{x_{i_1}, \ldots, x_{i_l}\}$ and $\overline{X'} = X \setminus X'$, then

$$\begin{aligned}\mathsf{NS}^-_\varphi(X', y) \equiv \quad (\mathsf{unf}^Y_{(\varphi^{-\overline{X'}})^{\mathsf{free}(X')}} (\varphi^{-\overline{X'}})_y^{\mathsf{free}(X')})[\quad & x_{i_1}/\mathsf{NS}^+_{\varphi^{-\overline{X'}}}(x_{i_1}, X'^{-i_1}),\\ & \vdots \\ & x_{i_k}/\mathsf{NS}^+_{\varphi^{-\overline{X'}}}(x_{i_l}, X'^{-i_l})],\end{aligned}$$

and

$$\begin{aligned}\mathsf{NS}^{+}_{\varphi}(x, X') \equiv \quad (\mathsf{unf}^{\overline{X'}}_{\varphi^{\mathsf{free}(Y\cup X')}}\varphi^{\mathsf{free}(Y\cup X')}_{x})[\quad & y_1/\mathsf{NS}^{-}_{\varphi}(X', y_1),\\ & \vdots\\ & y_k/\mathsf{NS}^{-}_{\varphi}(X', y_k),\\ & x_{i_1}/\mathsf{NS}^{+}_{\varphi}(x_{i_1}, X'^{-i_1}),\\ & \vdots\\ & x_{i_l}/\mathsf{NS}^{+}_{\varphi}(x_{i_l}, X'^{-i_l})].\end{aligned}$$

Note that by construction we have that for every ν-variable x, every μ-variable y and every set of ν-variables X', $\mathsf{free}(\mathsf{NS}^{+}_{\varphi}(x, X')), \mathsf{free}(\mathsf{NS}^{-}_{\varphi}(X', y)) \subseteq \mathsf{free}(\varphi)$ and $\mathsf{bound}(\mathsf{NS}^{+}_{\varphi}(x, X')), \mathsf{bound}(\mathsf{NS}^{-}_{\varphi}(X', y)) \subseteq \mathsf{bound}(\varphi)$.

LEMMA 2.27. *Let $\varphi \in \Sigma^{\mu}_{2}$, y be μ-variable in φ and X' be a proper subset of the set of all ν-variables. Suppose that x_i is a ν-variable such that $x_i \notin X'$. We have that*

$$\mathsf{NS}^{-}_{\varphi}(X', y), \mathsf{NS}^{+}_{\varphi}(x_i, X') \in \Delta^{\mu}_{2}.$$

Proof. The proof goes by induction on the size of X'. If $X' = \emptyset$ then clearly $\mathsf{NS}^{-}_{\varphi}(X', Y) \in \Sigma^{\mu}_{1}$ and, by definition of the formula, $\mathsf{NS}^{+}_{\varphi}(x, X') \in \Delta^{\mu}_{2}$. The induction step follows from the definitions by noting that the class Δ^{μ}_{2} is closed under substitution of Δ^{μ}_{2} formulae if no new variable is bound. □

LEMMA 2.28. *Let φ be a Σ^{μ}_{2}-formula and X be the set of all ν-variables in φ. Suppose that all $x \in X$ are weakly existential. Let $(\mathcal{T}, s_0)$ be a finite transitive transition system such that there is a normalized winning strategy σ^{N} in the evaluation game $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$. The following holds for every $X' \subseteq X$ where $\overline{X'} = X/X'$:*

(1) If there is a play consistent with σ^{N} which reaches a position $\langle y, s\rangle$ (y a μ-variable in φ) such that on this play before $\langle y, s\rangle$ there are positions $\langle \overline{x}, s\rangle$ for all $\overline{x} \in \overline{X'}$ then it holds that

$$s \in \|\mathsf{NS}^{-}_{\varphi}(X', y)\|_{\mathcal{T}^{\varphi}}.$$

(2) If there is a play consistent with σ^{N} which reaches for the first time a position $\langle x, s\rangle$ (x a ν-variable in φ) such that on this play before $\langle x, s\rangle$ there are positions $\langle \overline{x}, s\rangle$ for all $\overline{x} \in \overline{X'} \setminus \{x\}$ then it holds that

$$s \in \|\mathsf{NS}^{+}_{\varphi}(x, X')\|_{\mathcal{T}^{\varphi}}.$$

Proof. Let $Y = \{y_1, \ldots, y_k\}$ be the set of all μ-variables in φ. We prove the two points simultaneously by induction on the size of X'. If $X' = \emptyset$ we have that $\mathsf{NS}^{-}_{\varphi}(\emptyset, y) \equiv \mathsf{unf}^{Y}_{\varphi^{-X}}((\varphi^{-X})_y)$. If there is a play consistent with σ^{N} reaching a position of the form $\langle y, s\rangle$ whereby for all ν-variables there has been a regeneration in this play before, then, since σ^{N} is a normalized strategy, by Theorem 2.25 there can not be any regeneration of a ν-variable after $\langle y, s\rangle$. Therefore σ^{N} determines a winning strategy in

$$\mathcal{E}(\mathsf{unf}^{Y}_{\varphi^{-X}}((\varphi^{-X})_y), (\mathcal{T}, s))$$

and with Theorem 1.28 we get the induction base for part 1. For part 2 recall that

$$\mathsf{NS}^+_\varphi(x,\emptyset) \equiv (\mathsf{unf}^X_{\varphi^{\mathsf{free}(Y)}}\varphi_x^{\mathsf{free}(Y)})[y_1/\mathsf{NS}^-_\varphi(\emptyset,y_1),\ldots,y_k/\mathsf{NS}^-_\varphi(\emptyset,y_k)].$$

Suppose that there is a play consistent with σ^{N} which reaches for the first time a position $\langle x,s\rangle$ $(x\in X)$ such that on this play before $\langle x,s\rangle$ there are positions $\langle \overline{x},s\rangle$ for all $\overline{x}\in X\setminus\{x\}$. Then, since σ^{N} is a normalized strategy, by Theorem 2.25 for every play extending this position which is compatible with σ^{N}, either there are only regenerations of ν-variables, or, if there is a regeneration of a $\langle y,s\rangle$, then after this regeneration there is no more regeneration of a ν-variable. Therefore σ^{N} determines a winning strategy in

$$\mathcal{E}((\mathsf{unf}^X_{\varphi^{\mathsf{free}(Y)}}\varphi_x^{\mathsf{free}(Y)})[y_1/\mathsf{NS}^-_\varphi(\emptyset,y_1),\ldots,y_k/\mathsf{NS}^-_\varphi(\emptyset,y_k)],(\mathcal{T},s))$$

and with Theorem 1.28 we get the induction base for part 2.

For the induction step of part 1, let $X'=\{x_{i_1},\ldots,x_{i_l}\}$ and let $\langle y,s\rangle$ be a position of a play consistent with σ^{N} such that all $\overline{x}\in\overline{X'}$ have been regenerated before. Then, by Theorem 2.25 for all ν-variables x_i regenerated afterwards in the play we have $x_i\in X'$. By construction for such a position $\langle x_i,s_i\rangle$ we will have that all ν-variables in $\overline{X'}$ are regenerated before this position. Define $X'^{-i}=X'\setminus\{x_i\}$. It can easily be seen that $\langle x_i,s_i\rangle$ satisfy the condition of part 2 and, since $x_i\in X'$, that $X'^{-i}\subsetneq X'$. Therefore, we can apply induction hypothesis of part 2 and get

$$s_i\in\|\mathsf{NS}^+(x_i,X'^{-i})\|_{\mathcal{T}}.$$

Recapitulating, we have that for all plays consistent with σ^{N} starting from $\langle y,s\rangle$ if a ν-variable x_i is regenerated by a position $\langle x_i,s_i\rangle$ then $s_i\in\|\mathsf{NS}^+(x_i,X'^{-i})\|_{\mathcal{T}}$ and otherwise we have only regenerations of μ-variables. But by Lemmas 1.19.1 and 1.30 this means that σ^{N} gives us a winning strategy in the evaluation game

$$\mathcal{E}(\gamma,(\mathcal{T},s))$$

where

$$\begin{aligned}\gamma\equiv\quad \mathsf{unf}^Y_{(\varphi^{-\overline{X'}})^{\mathsf{free}(X')}}((\varphi^{-\overline{X'}})_y^{\mathsf{free}(X')})[\quad & x_{i_1}/\mathsf{NS}^+_{\varphi^{-\overline{X'}}}(x_{i_1},X'^{-i_1}),\\ & \vdots\\ & x_{i_l}/\mathsf{NS}^+_{\varphi^{-\overline{X'}}}(x_{i_l},X'^{-i_l})].\end{aligned}$$

By noting that $\gamma\equiv\mathsf{NS}^-_\varphi(X',y)$ and using Theorem 1.28 we finish the induction step for part 1.

For the induction step of part 2 let $\langle x,s\rangle$ be a position of a play consistent with σ^{N} such that all $\overline{x}\in\overline{X'}$ have been regenerated before. There are only three disjoint classes of winning plays (consistent with σ^{N}) extending the position $\langle x,s\rangle$ and they are obtained by considering all possible regenerations of bound variables after this position:

(1) The class of plays in which afterwards we regenerate a $x_i\in X'$ in a position $\langle x_i,s_i\rangle$, and before this position there was no regeneration of a μ-variable. In this case we can apply the induction hypothesis for part 2 to the set X'^{-i} and get

$$s_i\in\|\mathsf{NS}^+_\varphi(x_i,X'^{-i})\|_{\mathcal{T}}.$$

(2) The class of plays in which afterwards we regenerate a μ-variable y in a position $\langle y, s_y \rangle$, and before this position there was no regeneration of a $x_i \in X'$. In this case, we can apply part 1, where the induction step is already done, and get

$$s_y \in \|\mathsf{NS}^-_\varphi(X', y)\|_\mathcal{T}.$$

(3) The class of plays in which there is no regeneration of $z \in X' \cup Y$, but there are eventually only regenerations of $x_i \in \overline{X'}$. Because these plays are consistent with σ^N, they are winning. Therefore, they are winning in the evaluation game $\mathcal{E}((\mathsf{unf}^{\overline{X'}}_{\varphi^{-(Y \cup X')}} \varphi_x^{-(Y \cup X')}), (\mathcal{T}, s))$, too.

By Lemmas 1.19.1 and 1.30 we have that

$$\begin{aligned} s \in \quad \|(\mathsf{unf}^{\overline{X'}}_{\varphi^{\mathsf{free}(Y \cup X')}} \varphi_x^{\mathsf{free}(Y \cup X')})[\quad & y_1/\mathsf{NS}^-_\varphi(X', y_1), \\ & \vdots \\ & y_k/\mathsf{NS}^-_\varphi(X', y_k), \\ & x_{i_1}/\mathsf{NS}^+_\varphi(x_{i_1}, X'^{-i_1}), \\ & \vdots \\ & x_{i_l}/\mathsf{NS}^+_\varphi(x_{i_l}, X'^{-i_l})]\|_\mathcal{T}. \end{aligned}$$

and this ends the induction step of part 2 and the proof. ☐

LEMMA 2.29. *Let φ be a Σ_2^μ-formula and X be the set of all ν-variables in φ. Suppose that all ν-variables are weakly existential. Then, for every finite transitive transition system $\mathcal{T}$ and for every $X' \subseteq X$ it holds that*

(1) For every $y \in Y$ we have

$$\|\mathsf{NS}^-_\varphi(X', y)\|_{\mathcal{T}^\varphi} \subseteq \|\varphi_y\|_{\mathcal{T}^\varphi}, \text{ and}$$

(2) for every $x \in \overline{X'} =: X/X'$ we have

$$\|\mathsf{NS}^+_\varphi(x, X')\|_{\mathcal{T}^\varphi} \subseteq \|\varphi_x\|_{\mathcal{T}^\varphi}.$$

Proof. Let $Y = \{y_1, \ldots, y_k\}$ be the set of all μ-variables. We prove the two points simultaneously by induction on the size of X'. Suppose X' is empty. Then we have that $\mathsf{NS}^-_\varphi(\emptyset, y) \equiv \mathsf{unf}^Y_{\varphi^{-X}}((\varphi^{-X})_y)$ and by Lemma 2.15.3 we obtain

$$\|\mathsf{unf}^Y_{\varphi^{-X}}((\varphi^{-X})_y)\|_{\mathcal{T}^\varphi} \subseteq \|\varphi_y\|_{\mathcal{T}^\varphi}.$$

Therefore we complete the base case of the induction for part 1. For part 2 recall that

$$\mathsf{NS}^+_\varphi(x, \emptyset) \equiv (\mathsf{unf}^X_{\varphi^{\mathsf{free}(Y)}} \varphi_x^{\mathsf{free}(Y)})[y_1/\mathsf{NS}^-_\varphi(\emptyset, y_1), \ldots, y_k/\mathsf{NS}^-_\varphi(\emptyset, y_k)].$$

Thus, by the induction base of part 1 and by Lemma 1.19.4, we have that

$$\begin{gathered} \|(\mathsf{unf}^X_{\varphi^{\mathsf{free}(Y)}} \varphi_x^{\mathsf{free}(Y)})[y_1/\mathsf{NS}^-_\varphi(\emptyset, y_1), \ldots, y_k/\mathsf{NS}^-_\varphi(\emptyset, y_k)]\|_{\mathcal{T}^\varphi} \\ \subseteq \\ \|(\mathsf{unf}^X_{\varphi^{\mathsf{free}(Y)}} \varphi_x^{\mathsf{free}(Y)})[y_1/\varphi_{y_1}, \ldots, y_m/\varphi_{y_m}]\|_{\mathcal{T}^\varphi}. \end{gathered}$$

But because in $\mathcal{T}^\varphi$ we have that $\lambda(y) = \|\varphi_y\|_{\mathcal{T}^\varphi}$ and by applying Lemma 1.19.1 and Lemma 2.14.2, it holds that

$$\|(\mathsf{unf}^X_{\varphi^{\mathsf{free}(Y)}}\varphi_x^{\mathsf{free}(Y)})[y_1/\varphi_{y_1},\dots,y_m/\varphi_{y_m}]\|_{\mathcal{T}^\varphi} \subseteq \|\varphi_x\|_{\mathcal{T}^\varphi}.$$

Therefore

$$\|(\mathsf{unf}^X_{\varphi^{\mathsf{free}(Y)}}\varphi_x^{\mathsf{free}(Y)})[y_1/\mathsf{NS}^-_\varphi(\emptyset,y_1),\dots,y_k/\mathsf{NS}^-_\varphi(\emptyset,y_k)]\|_{\mathcal{T}^\varphi} \subseteq \|\varphi_x\|_{\mathcal{T}^\varphi}.$$

This ends the induction base for both parts 1 and 2.

Let $X' = \{x_{i_1},\dots,x_{i_l}\}$. For the induction step of part 1, recall that

$$\begin{array}{lll}\mathsf{NS}^-_\varphi(X',y) \equiv & \mathsf{unf}^Y_{(\varphi^{-\overline{X'}})^{\mathsf{free}(X')}}((\varphi^{-\overline{X'}})^{\mathsf{free}(X')})_y[& x_{i_1}/\mathsf{NS}^+_{\varphi^{-\overline{X'}}}(x_{i_1},X'^{-i_1}),\\ & & \vdots\\ & & x_{i_l}/\mathsf{NS}^+_{\varphi^{-\overline{X'}}}(x_{i_l},X'^{-i_l})].\end{array}$$

By induction hypothesis, by Lemma 1.19.4 and because in $\mathcal{T}^{\varphi^{-\overline{X'}}}$ the evaluation of a variable $x_{i_j} \in X'$ is equal to $\|(\varphi^{-\overline{X'}})_{x_{i_j}}\|_{\mathcal{T}^{\varphi^{-\overline{X'}}}}$, we obtain

$$\begin{array}{ll}\|\mathsf{NS}^-_\varphi(X',y)\|_{\mathcal{T}^\varphi} = \| & \mathsf{unf}^Y_{(\varphi^{-\overline{X'}})^{\mathsf{free}(X')}}((\varphi^{-\overline{X'}})_y^{\mathsf{free}(X')})\\ & [x_{i_1}/\mathsf{NS}^+_{\varphi^{-\overline{X'}}}(x_{i_1},X'^{-i_1}),\\ & \vdots\\ & x_{i_l}/\mathsf{NS}^+_{\varphi^{-\overline{X'}}}(x_{i_l},X'^{-i_l})]\|_{\mathcal{T}}\\ & \subseteq \|\mathsf{unf}^Y_{(\varphi^{-\overline{X'}})^{\mathsf{free}(X')}}((\varphi^{-\overline{X'}})^{\mathsf{free}(X')})_y\|_{\mathcal{T}^{\varphi^{-\overline{X'}}}}.\end{array}$$

With Lemma 2.14.2 we obtain

$$\|\mathsf{unf}^Y_{(\varphi^{-\overline{X'}})^{\mathsf{free}(X')}}((\varphi^{-\overline{X'}})_y^{\mathsf{free}(X')})\|_{\mathcal{T}^{\varphi^{-\overline{X'}}}} = \|(\varphi^{-\overline{X'}})_y\|_{\mathcal{T}^{\varphi^{-\overline{X'}}}}.$$

Finally, because by Lemma 2.15.4 it holds that $\|(\varphi^{-\overline{X'}})_y\|_{\mathcal{T}^{\varphi^{-\overline{X'}}}} \subseteq \|\varphi_y\|_{\mathcal{T}^\varphi}$ we get

$$\|\mathsf{NS}^-_\varphi(X',y)\|_{\mathcal{T}^\varphi} \subseteq \|\varphi_y\|_{\mathcal{T}^\varphi}.$$

For the induction step of part 2 if $\overline{X} = X \setminus X'$, then by induction hypothesis and by part 1 we have for every finite transitive transition system $\mathcal{T}$

$$\begin{array}{rcl}\|\mathsf{NS}^-_\varphi(X',y_1)\|_{\mathcal{T}^\varphi} & \subseteq & \|\varphi_{y_1}\|_{\mathcal{T}^\varphi},\\ & \vdots & \\ \|\mathsf{NS}^-_\varphi(X',y_k)]\|_{\mathcal{T}^\varphi} & \subseteq & \|\varphi_{y_k}\|_{\mathcal{T}^\varphi},\\ \|\mathsf{NS}^+_\varphi(x_{i_1}X'^{-i_1})\|_{\mathcal{T}^\varphi} & \subseteq & \|\varphi_{x_{i_1}}\|_{\mathcal{T}^\varphi},\\ & \vdots & \\ \|\mathsf{NS}^+_\varphi(x_{i_l}X'^{-i_l})\|_{\mathcal{T}^\varphi} & \subseteq & \|\varphi_{x_{i_l}}\|_{\mathcal{T}^\varphi}.\end{array}$$

Therefore, by Lemmas 1.19.4 and 1.19.1, and because for every $z \in \mathsf{bound}(\varphi)$

we have that $\lambda(z) = \|\varphi_z\|_{\mathcal{T}^\varphi}$, we get

$$\begin{aligned}
\|(\mathsf{unf}^{\overline{X'}}_{\varphi^{\mathsf{free}(Y\cup X')}}(\varphi_x^{\mathsf{free}(Y\cup X')}))[\quad & y_1/\mathsf{NS}^-_\varphi(X', y_1),\\
& \vdots\\
& y_k/\mathsf{NS}^-_\varphi(X', y_k)\\
& x_{i_1}/\mathsf{NS}^+_\varphi(x_{i_1}, X'^{-i_1}),\\
& \vdots\\
& x_{i_l}/\mathsf{NS}^+_\varphi(x_{i_l}, X'^{-i_l})]\|_{\mathcal{T}^\varphi}\\
\subseteq \|\mathsf{unf}^{\overline{X'}}_{\varphi^{\mathsf{free}(Y\cup X')}}(\varphi_x^{\mathsf{free}(Y\cup X')})\|_{\mathcal{T}^\varphi}.
\end{aligned}$$

Thus, we can apply Lemma 2.14.2 and obtain

$$\|(\mathsf{unf}^{\overline{X'}}_{\varphi^{\mathsf{free}(Y\cup X')}}(\varphi_x^{\mathsf{free}(Y\cup X')}))\|_{\mathcal{T}^\varphi} \subseteq \|\varphi_x\|_{\mathcal{T}^\varphi}.$$

Because this implies that

$$\|\mathsf{NS}^+_\varphi(x, X')\|_{\mathcal{T}^\varphi} \subseteq \|\varphi_x\|_{\mathcal{T}^\varphi}$$

this ends the induction step of part 2 and the proof of the Lemma. □

2.4.5 The collapse over transitive models

Everything now is ready to prove the collapse of the μ-hierarchy over finite transitive transition systems.

DEFINITION 2.30. *For the formula $\varphi \in \Sigma_2^\mu$ such that $X = \{x_1, \ldots, x_m\}$ is the set of all ν-variables in φ. We define a new formula $\rho(\varphi) \in \Delta_2^\mu$ such that*

$$\rho(\varphi) \equiv \varphi^{\mathsf{free}(X)}[x_1/\mathsf{NS}^+_\varphi(x_1, X^{-1}), \ldots, x_m/\mathsf{NS}^+_\varphi(x_m, X^{-m})].$$

REMARK 2.31. By Lemma 2.27 it can easily be seen that $\rho(\varphi)$ is indeed a Δ_2^μ-formula.

THEOREM 2.32. *For all $\varphi \in \Sigma_2^\mu$ and all finite transitive transition systems $\mathcal{T}$ we have that*

$$\|\varphi\|_\mathcal{T} = \|\rho(\tau(\varphi))\|_\mathcal{T}.$$

Proof. First, we observe that $\tau(\varphi) \in \Sigma_2^\mu$ and that by Corollary 2.19 we have that $\|\varphi\|_\mathcal{T} = \|\tau(\varphi)\|_\mathcal{T}$. Thus, we can assume that each ν-variable in $\varphi \in \Sigma_2^\mu$ is weakly existential and any μ-variable weakly universal. If $X = \{x_1, \ldots, x_m\}$ is the set of all ν-variables in φ, by definition of ρ we have to prove that

$$\|\varphi\|_\mathcal{T} = \|\varphi^{\mathsf{free}(X)}[x_1/\mathsf{NS}^+_\varphi(x_1, X^{-1}), \ldots, x_m/\mathsf{NS}^+_\varphi(x_m, X^{-m})]\|_\mathcal{T}.$$

"$\supseteq$": Note that $\mathcal{T}[x_1 \mapsto \|\mathsf{NS}^+_\varphi(x_1, X^{-1})\|_\mathcal{T}, \ldots, x_m \mapsto \|\mathsf{NS}^+_\varphi(x_m, X^{-m})\|_\mathcal{T}\|]$ and $\mathcal{T}^\varphi[x_1 \mapsto \|\mathsf{NS}^+_\varphi(x_1, X^{-1})\|_{\mathcal{T}^\varphi}, \ldots, x_m \mapsto \|\mathsf{NS}^+_\varphi(x_m, X^{-m})\|_{\mathcal{T}^\varphi}\|]$ agree on the free variables of $\varphi^{\mathsf{free}(X)}$ because $\|\mathsf{NS}^+_\varphi(x_i, X^{-i})\|_\mathcal{T}$ and $\|\mathsf{NS}^+_\varphi(x_i, X^{-i})\|_{\mathcal{T}^\varphi}$ coincide for every $x_i \in X$. Therefore we have that

$$\begin{aligned}
&\|\varphi^{\mathsf{free}(X)}[x_1/\mathsf{NS}^+_\varphi(x_1, X^{-1}), \ldots, x_m/\mathsf{NS}^+_\varphi(x_m, X^{-m})]\|_\mathcal{T} =\\
&\quad \|\varphi^{\mathsf{free}(X)}[x_1/\mathsf{NS}^+_\varphi(x_1, X^{-1}), \ldots, x_m/\mathsf{NS}^+_\varphi(x_m, X^{-m})]\|_{\mathcal{T}^\varphi}
\end{aligned}$$

With Lemma 2.29 and, because all ν-variables appear positively in φ, by applying Lemma 1.19.4 we get that

$$\|\varphi^{\mathsf{free}(X)}[x_1/\mathsf{NS}^+_\varphi(x_1, X^{-1}), \dots, x_m/\mathsf{NS}^+_\varphi(x_m, X^{-m})]\|_{\mathcal{T}^\varphi}$$
$$\subseteq$$
$$\|\varphi^{\mathsf{free}(X)}[x_1/\varphi_{x_1}, \dots, x_m/\varphi_{x_m}]\|_{\mathcal{T}^\varphi}$$

By Lemma 1.19.4 and because in $\mathcal{T}^\varphi$ we have that $\lambda(x_i) = \|\varphi_{x_i}\|_{\mathcal{T}^\varphi}$, we obtain

$$\|\varphi^{\mathsf{free}(X)}[x_1/\varphi_{x_1}, \dots, x_m/\varphi_{x_m}]\|_{\mathcal{T}^\varphi} \subseteq \|\varphi^{\mathsf{free}(X)}\|_{\mathcal{T}^\varphi}.$$

Since by Lemma 2.14.2 we have that $\|\varphi^{\mathsf{free}(X)}\|_{\mathcal{T}^\varphi} = \|\varphi\|_{\mathcal{T}}$ we get this inclusion.

"$\subseteq$": Let $s \in \|\varphi\|_{\mathcal{T}}$. By Theorem 1.28 there is a winning strategy in $\mathcal{E}(\varphi, (\mathcal{T}, s))$ and by Theorem 2.25 it can be assumed to be normalized. Let π be any play consistent with the strategy starting from $\langle\varphi, s\rangle$. We have that if there is a (first) regeneration of a ν-variable x_i in a position $\langle x_i, s_i\rangle$ then by Lemma 2.28 we have that

$$s_i \in \|\mathsf{NS}^+(x_i, X^{-i})\|_{\mathcal{T}}$$

where X is the set of all ν-variables in φ. Therefore, there is a winning strategy for Player 0 in

$$\mathcal{E}(\varphi^{\mathsf{free}(X)}, (\mathcal{T}[x_1 \mapsto \|\mathsf{NS}^+(x_1, X^{-1})\|_{\mathcal{T}}, \dots, x_n \mapsto \|\mathsf{NS}^+(x_n, X^{-n})\|_{\mathcal{T}}], s))$$

By Theorem 1.28 we have that

$$s \in \|\varphi^{\mathsf{free}(X)}\|_{\mathcal{T}[x_1 \mapsto \|\mathsf{NS}^+(x_1, X^{-1})\|_{\mathcal{T}}, \dots, x_n \mapsto \|\mathsf{NS}^+(x_n, X^{-n})\|_{\mathcal{T}}]}$$

and with Lemma 1.19.1 we complete the proof. □

Corollary 2.33. *The modal μ-calculus hierarchy on finite transitive systems collapses to Δ_2^μ.*

Proof. By Theorem 2.32, $\Sigma_2^{\mu\mathbb{T}^{tf}} = \Delta_2^{\mu\mathbb{T}^{tf}}$. By duality, $\Pi_2^{\mu\mathbb{T}^{tf}} = \Delta_2^{\mu\mathbb{T}^{tf}}$. By this fact it is therefore very easy to verify inductively that for every $n > 0$, $\Sigma_{2+n}^{\mu\mathbb{T}^{tf}} = \Pi_{2+n}^{\mu\mathbb{T}^{tf}} = \Delta_2^{\mu\mathbb{T}^{tf}}$. □

Corollary 2.34. *The modal μ-calculus hierarchy on transitive systems collapses to Δ_2^μ.*

Proof. Suppose that the hierarchy does not collapse. Therefore, there is a formula φ such that for all formula $\psi \in \Delta_2^\mu$ there is a transitive system $\mathcal{T}$ such that $\mathcal{T}, s_0 \models \neg(\varphi \leftrightarrow \psi)$. By Theorem 2.5, there is a finite transitive model $\mathcal{T}^f$ such that $\mathcal{T}^f, s_i^f \models \neg(\varphi \leftrightarrow \psi)$. But this cannot be the case by Corollary 2.33. □

We end with the definition of a syntactical translation from $\mathcal{L}_\mu$ to Δ_2^μ preserving equivalence on transitive transition systems.

Definition 2.35. $R : \mathcal{L}_\mu \to \Delta_2^\mu$ *is defined as*

- $R(p) = p$ *and* $R(\neg p) = \neg p$
- $R(\bot) = \bot$ *and* $R(\top) = \top$

- $R(\alpha \circ \beta) = R(\alpha) \circ R(\beta)$, *where* $\circ \in \{\wedge, \vee\}$
- $R(\triangle\ \beta) = \triangle\ R(\beta)$, *where* $\triangle \in \{\Box, \Diamond\}$
- $R(\mu x.\varphi) = \mathsf{wn}\big(\rho(\tau(\mathsf{wn}(\mu x.(R(\varphi)))))\big)$
- $R(\nu x.\varphi) = \neg(R(\mu x.\neg\varphi[x/\neg x]))$

LEMMA 2.36. *For all μ-formula φ we have that*

(1) $R(\varphi)$ is well-defined, and

(2) $R(\varphi) \in \Delta_2^\mu$.

Proof. We prove both parts simultaneously by induction on the structure of φ. The induction cases for boolean and modal connectives are trivial. If φ is of the form $\mu x.\alpha$ we have that $R(\mu x.\alpha) = \mathsf{wn}\big(\rho(\tau(\mathsf{wn}(\mu x.(R(\alpha)))))\big)$. Because τ is a well-defined syntactical transformation, and neither wn nor τ increase the alternation depth of a formula, the application of ρ in the clause of $R(\mu x.\alpha)$ is well-defined by induction hypothesis. Thus, $R(\varphi)$ terminates and therefore it is well-defined too. The fact that $R(\mu x.\alpha) \in \Delta_2^\mu$ follows by induction hypothesis, by the fact that, by Remark 2.31, for all Σ_2^μ-formulae ψ we have that $\rho(\psi) \in \Delta_2^\mu$, and because we know that τ and wn do not increase the alternation depth. If φ is of the form $\nu x.\alpha$, on one hand $R(\nu x.\alpha)$ is well-defined because the clause for this form is defined via a reducing case $R(\mu x.\neg\varphi[x/\neg x])$, and, on the other hand $R(\varphi) \in \Delta_2^\mu$ because Δ_2^μ is closed under negation. $\Box$

THEOREM 2.37. *For all $\varphi \in \mathcal{L}_\mu$ and all finite transitive transition systems $\mathcal{T}$ we have that*

$$\|\varphi\|_\mathcal{T} = \|R(\varphi)\|_\mathcal{T}.$$

Proof. We prove the equivalence by induction on $\mathsf{rank}(\varphi)$ simultaneously for all finite transitive transition systems $\mathcal{T}$. The induction cases for boolean and modal connectives are trivial. If φ is of the form $\mu x.\alpha$ we have that

$$\begin{array}{rcll} \|R(\mu x.\alpha)\|_\mathcal{T} & = & \|\mathsf{wn}\big(\rho(\tau(\mathsf{wn}(\mu x.R(\alpha))))\big)\|_\mathcal{T} & \text{by definition of } R \\ & = & \|\mathsf{wn}(\mu x.R(\alpha))\|_\mathcal{T} & \tau(\mathsf{wn}(\mu x.R(\alpha))) \in \Sigma_2^\mu, \text{ and by} \\ & & & \text{Lemma 1.20 and Theorem 2.32} \\ & = & \|\mu x.\alpha\|_\mathcal{T} & \text{by Lemma 1.20 and induction} \\ & & & \text{hypothesis} \end{array}$$

If φ is of the form $\nu x.\alpha$ we do a similar induction step like above by using the equivalence $\|\nu x.\alpha\|_\mathcal{T} = \|\neg\mu x.\neg\alpha[x/\neg x]\|_\mathcal{T}$. $\Box$

We conclude by verifying that the syntactical translation R is also an explicit syntactical translation of all modal μ-formulae to the alternation free fragment preserving denotation in *every* transitive transition systems. The proof goes with similar argument as in Corollary 2.34 and it is left to the reader.

THEOREM 2.38. *For all $\varphi \in \mathcal{L}_\mu$ and all transitive transition systems $\mathcal{T}$ we have that*

$$\|\varphi\|_\mathcal{T} = \|R(\varphi)\|_\mathcal{T}.$$

Remark 2.39. Note, that due to the example of Visser in [122] mentioned in the introduction the alternation-free fragment is also the optimal bound if restrict ourselves to transition systems which are transitive *and* reflexive.

Example 2.40. Let's have a look at our example from Section 3.4. In the case of "always eventually", we have that

$$\|\nu x.(\mu y.(p \vee \Diamond y)) \wedge \Box x)\|^{\mathbb{T}^t} = \|(p \vee \Diamond p) \wedge \Box(p \vee \Diamond p)\|^{\mathbb{T}^t}.$$

For "infinitely often", it holds that

$$\|\nu x.\mu y.((p \vee \Diamond y)) \wedge \Diamond x)\|^{\mathbb{T}^t} = \|\nu x.(p \wedge \Diamond x)\|^{\mathbb{T}^t}.$$

But, because from footnote 4 of the introduction we know that $\nu x.(p \wedge \Diamond x)$ cannot be reduced to any purely modal formula, contrary to the transitive *and* symmetric case, over transitive transition systems "infinitely often" cannot be expressed by a Δ_1^μ formula.

2.5 The reflexive case

In this section we prove the strictness of the modal μ-calculus hierarchy on reflexive transition systems. This is done by following the argumentation of the proof of the strictness of the hierarchy on all binary transition systems of Arnold in [7]. First, we adapt the game transition system, introduced in Section 1.7, such that it is reflexive.

Let $\mathcal{E}(\varphi, (\mathcal{T}, s))$ be a parity game with priority function Ω and with corresponding pointed game transition system $\mathcal{T}(\mathcal{E}(\varphi, (\mathcal{T}, s)))$. We extend the edge relation E of the parity game to its reflexive closure $E^r = E \cup \{(s,s); s \in V_0 \cup V_1\}$, and change our priority function Ω to Ω^r such that for all vertices $\langle \psi, s\rangle$ where $\psi \equiv \eta x.\delta$ ($\eta \in \{\mu, \nu\}$) we have

$$\Omega^r(\langle \psi, s\rangle) = \Omega(\langle \psi, s\rangle) + 2$$

and such that for all other vertices we define:

- if $\min \Omega$ is even
$$\Omega^r(\langle \psi, s\rangle) = \begin{cases} 0 & \text{if } \langle \psi, s\rangle \in V_1 \\ 1 & \text{if } \langle \psi, s\rangle \in V_0. \end{cases}$$
- if $\min \Omega$ is odd
$$\Omega^r(\langle \psi, s\rangle) = \begin{cases} 2 & \text{if } \langle \psi, s\rangle \in V_1 \\ 1 & \text{if } \langle \psi, s\rangle \in V_0. \end{cases}$$

The new resulting "reflexive" parity game is denoted as $\mathcal{E}^r(\varphi, (\mathcal{T}, s))$. The following Lemma can be proved by unwinding the definition of winning strategy.

Lemma 2.41. *Player 0 has a winning strategy for $\mathcal{E}^r(\varphi, (\mathcal{T}, s))$ iff Player 0 has a winning strategy for $\mathcal{E}(\varphi, (\mathcal{T}, s))$.*

Given a "reflexive" parity game $\mathcal{E}^r(\varphi, (\mathcal{T}, s))$ the *pointed game transition system* $\mathcal{T}(\mathcal{E}^r(\varphi, (\mathcal{T}, s)))$ is defined analogously as above. Obviously, the pointed game transition system $\mathcal{T}(\mathcal{E}^r(\varphi, (\mathcal{T}, s)))$ is reflexive. We have that

PROPOSITION 2.42. *Let $(\mathcal{T}, s)$ be an arbitrary pointed transition system. For all $\varphi \in \Pi^\mu_n$ we have that*

$$\mathcal{T}(\mathcal{E}^r(\varphi, (\mathcal{T}, s))) \in \|W_{\Pi^\mu_{n+2}}\| \quad \textit{if and only if} \quad \mathcal{T}(\mathcal{E}(\varphi, (\mathcal{T}, s))) \in \|W_{\Pi^\mu_n}\|$$

and dually for $\varphi \in \Sigma^\mu_n$.

Proof. This follows directly by the definition of the "reflexive" parity game $\mathcal{E}^r(\varphi, (\mathcal{T}, s))$ and by applying Proposition 1.32 to Lemma 2.41. □

COROLLARY 2.43. *Let $(\mathcal{T}, s)$ be an arbitrary pointed transition system. For all $\varphi \in \Pi^\mu_n$ we have that:*

$$\mathcal{T}(\mathcal{E}^r(\varphi, (\mathcal{T}, s))) \in \|W_{\Pi^\mu_{n+2}}\| \textit{ if and only if } (\mathcal{T}, s) \in \|\varphi\|.$$

and dually for $\varphi \in \Sigma^\mu_n$.

Proof. By Proposition 2.42 and Corollary 1.34 we obtain our result. □

For all formulae φ we define a function f_φ (functional class) mapping a pointed transition system $(\mathcal{T}, s)$ to a reflexive transition system $f_\varphi(\mathcal{T}, s)$ such that

$$f_\varphi(\mathcal{T}, s) := \mathcal{T}(\mathcal{E}^r(\varphi, (\mathcal{T}, s))).$$

The proof of the next Lemma follows similar arguments as the proof of the same result for the class of all binary trees proved by Arnold in [7], and which has bee extended to arbitrary transition systems by Alberucci in [1].

LEMMA 2.44. *For all formulae $\psi \in \Sigma^\mu_n$ (resp. Π^μ_n), $n \in \mathbb{N}$, there is an equivalent formula $\varphi \in \Sigma^\mu_n$ (resp. Π^μ_n) such that the function f_φ has a fixpoint in $\mathbb{T}^r$, that is, a pointed reflexive transition system $(\mathcal{T}^F, s^F)$ such that*

$$f_\varphi(\mathcal{T}^F, s^F) = (\mathcal{T}^F, s^F).$$

Proof. First we remark that for any two pointed transition systems $(\mathcal{T}, s)$ and $(\mathcal{T}', s')$ which are identical up to depth m we have that $f_\psi(\mathcal{T}, s)$ and $f_\psi(\mathcal{T}', s')$ are identical up to depth m. This can be proved by induction on the structure of the formulae ψ and the proof is left to the reader. In order to prove the Lemma let ψ be an arbitrary formula. We define $\varphi \equiv \psi \wedge \psi$. By the first remark and the fact that $f_\varphi(\mathcal{T}, s)$ is basically the transition system $f_\psi(\mathcal{T}, s)$ with a new distinguished state $\langle \psi \wedge \psi, s \rangle$ from where you can reach the distinguished state of $f_\psi(\mathcal{T}, s)$ which is $\langle \psi, s \rangle$. We get the following

Claim: Given two pointed transition systems $(\mathcal{T}, s)$ and $(\mathcal{T}', s')$ which are identical up to depth m then $f_\varphi(\mathcal{T}, s)$ and $f_\varphi(\mathcal{T}', s')$ are identical up to depth $m + 1$.

For the construction of the fixpoint $(\mathcal{T}^F, s^F)$ we proceed as follows. We start from a one state pointed transition system $(\mathcal{T}, s) = (\{s\}, \{(s, s)\}, \lambda)$ where the valuation λ can be chosen arbitrarily. And define a sequence of pointed transition systems $\{(\mathcal{T}, s)_n\}_{n \in \mathbb{N}}$ as

$$(\mathcal{T}, s)_0 = (\mathcal{T}, s) \quad \text{and} \quad (\mathcal{T}, s)_{n+1} = f_\varphi((\mathcal{T}, s)_n).$$

By the claim above $\{(\mathcal{T}, s)_n\}_{n \in \mathbb{N}}$ is a monotone sequence of reflexive transition systems and it can easily be seen that beside being well defined $\lim(\{(\mathcal{T}, s)_n\}_{n \in \mathbb{N}})$ is a reflexive transition system. It is an easy exercise to show that we have $f_\varphi(\lim(\{(\mathcal{T}, s)_n\}_{n \in \mathbb{N}})) = \lim(\{(\mathcal{T}, s)_n\}_{n \in \mathbb{N}})$. □

Theorem 2.45. *For all natural numbers $n \in \mathbb{N} \setminus \{0\}$ we have that*

$$\Sigma_n^{\mathbb{T}^r} \subsetneq \Sigma_{n+1}^{\mathbb{T}^r} \quad \textit{and} \quad \Pi_n^{\mathbb{T}^r} \subsetneq \Pi_{n+1}^{\mathbb{T}^r}.$$

Proof. We proof the contrapositive. Assume that we have

$$\Sigma_{n+1}^{\mathbb{T}^r} \subseteq \Sigma_n^{\mathbb{T}^r} \quad \text{or} \quad \Pi_{n+1}^{\mathbb{T}^r} \subseteq \Pi_n^{\mathbb{T}^r}.$$

Without restriction of generality, assume $\Sigma_{n+1}^{\mathbb{T}^r} \subseteq \Sigma_n^{\mathbb{T}^r}$. Then, if $\|\varphi\| \in \Pi_{n+1}^{\mu}$ we have $\|\neg\varphi\| \in \Sigma_{n+1}^{\mu}$ and by assumption $\|\neg\varphi\| \in \Sigma_n^{\mu}$ and therefore $\|\varphi\| \in \Pi_n^{\mu}$. Therefore, assuming the contrapositive leads to

$$\Sigma_{n+1}^{\mathbb{T}^r} \subseteq \Sigma_n^{\mathbb{T}^r} \quad \text{and} \quad \Pi_{n+1}^{\mathbb{T}^r} \subseteq \Pi_n^{\mathbb{T}^r}.$$

Since from $\Sigma_{n+1}^{\mathbb{T}^r} \subseteq \Sigma_n^{\mathbb{T}^r}$, by definition, it can be inferred that $\Pi_n^{\mathbb{T}^r} \subseteq \Sigma_n^{\mathbb{T}^r}$, and from $\Pi_{n+1}^{\mathbb{T}^r} \subseteq \Pi_n^{\mathbb{T}^r}$, by definition, it can be inferred that $\Sigma_n^{\mathbb{T}^r} \subseteq \Pi_n^{\mathbb{T}^r}$, by assuming the contrapositive we get that $\Pi_{n+1}^{\mathbb{T}^r} = \Pi_n^{\mathbb{T}^r} = \Sigma_{n+1}^{\mathbb{T}^r} = \Sigma_n^{\mathbb{T}^r}$ and, obviously, we then have for all $k \in \mathbb{N}$ that

$$\Pi_{n+k}^{\mathbb{T}^r} = \Pi_n^{\mathbb{T}^r} = \Sigma_{n+k}^{\mathbb{T}^r} = \Sigma_n^{\mathbb{T}^r}. \tag{2.4}$$

Since $W_{\Sigma_{n+2}^{\mu}} \in \Sigma_{n+2}^{\mu}$ we have that $\neg W_{\Sigma_{n+2}^{\mu}} \in \Pi_{n+2}^{\mu}$ and with equation 2.4 we get

$$\|\neg W_{\Sigma_{n+2}^{\mu}}\|^{\mathbb{T}^r} \in \Sigma_n^{\mathbb{T}^r}.$$

By Lemma 2.44 there is a formula $\varphi \in \Sigma_n^{\mu}$ equivalent to $\neg W_{\Sigma_{n+2}}$ and a pointed transition system $(\mathcal{T}^F, s^F)$ such that

$$(\mathcal{T}^F, s^F) = f_{\varphi}(\mathcal{T}^F, s^F).$$

Since $f_{\varphi}(\mathcal{T}, s)$ is defined as $\mathcal{T}(\mathcal{E}^r(\varphi, (\mathcal{T}, s)))$, by Corollary 2.43, for all pointed transition systems $(\mathcal{T}, s)$ we have that $f_{\varphi}(\mathcal{T}, s) \in \|W_{\Sigma_{n+2}}\|$ if and only if $(\mathcal{T}, s) \in \|\varphi\|$. Since φ is equivalent to $\neg W_{\Sigma_{n+2}}$ we get that

$$(\mathcal{T}^F, s^F) \in \|\neg W_{\Sigma_{n+2}}\| \quad \text{iff} \quad (\mathcal{T}^F, s^F) \in \|W_{\Sigma_{n+2}}\|$$

which is a contradiction. □

Theorem 2.46.

(1) *The modal μ-calculus hierarchy is strict over reflexive transition systems.*

(2) *The modal μ-calculus hierarchy is strict over finite reflexive transition systems.*

Proof. Part 1 is a corollary of Theorem 2.45. For Part 2, let $\|\varphi\| \in \Sigma_n^{\mathbb{T}^r} \setminus \Pi_n^{\mathbb{T}^r}$. Then, by Part 1 we know that for every $\psi \in \Sigma_{n-1}^{\mu}$ it holds that $\neg(\varphi \leftrightarrow \psi)$ has a reflexive model. By Theorem 2.3, this model can be finite. Hence $\varphi \in \Sigma_n^{\mu}$ is not equivalent to any Σ_{n-1}^{μ} formula on finite reflexive transition systems. □

2.6 Summarizing remarks

Thanks to the use of least and greatest fixpoint operators, the modal μ-calculus is a powerful logic which can express many interesting properties of models. With least fixpoints one can express, for instance, liveness properties like "it is possible to reach a node where p holds", and with greatest fixpoints one expresses safety properties of the kind "p is true in all reachable nodes". On arbitrary transition systems, these two properties really need fixpoints in order to be expressed. That is to say, they are not modally expressible. This is because modal logic can only speak about *local* properties, in the sense that whether or not a modal formula is true in a certain node of a model only depends on the nodes accessible to the current one. But what gives to the μ-calculus all its power is the possibility of having nested least and greatest fixpoints. For instance, by one nesting it is possible to already capture fairness properties like "there is a branch where p holds infinitely often", while with several alternations one can even express the existence of a winning strategy in a parity game.

Although on arbitrary transition systems the number of alternations between different fixpoints generates a strict infinite hierarchy, the fixpoint hierarchy may not be infinite anymore if we restrict the semantics to subclasses of models. In this chapter, we were interested in the behavior of the modal μ-calculus on the class of transitive models, of reflexive models, and on the class of symmetric and transitive models. The obtained results are summarized in the next figure:

	reflexive	transitive	symmetric & transitive
fixpoint alternation hierarchy	strict	collapse to the alternation free fragment	collapse to the modal fragment

The natural missing case is the case of symmetric models. We conjecture that on this subclass of models the fixpoint alternation hierarchy is strict. Another missing prominent subclass, which is probably the most studied so far, is the class of transitive and upward well-founded models, also called the Gödel-Löb class in view of its relation with Gödel theorems and the logic of provability in Peano Arithmetic. This class is presented and studied in the next chapter.

Concerning the hierarchy on transitive models, d'Agostino and Lenzi [44] propose a different proof of its collapse which explicitly uses Theorem 2.17 of this thesis. This result has also independently been obtained by Dawar and Otto [46].

Chapter 3

The μ-Calculus vs the Gödel-Löb Logic

This chapter is based on a joint work with Luca Alberucci [4].

3.1 Preliminary remarks

The Gödel-Löb logic, GL for short, is a modal logic where the modal operator for necessity is interpreted as provability in a reasonably rich formal theory such as Peano arithmetic, and is thence used to investigate what arithmetical theories can express in a restricted language about their provability predicates. This modal logic has been studied since the early seventies, and has had important applications in the foundations of mathematics ([30, 113]). As a formal system, GL is obtained by adding the modal version of what is called the *Löb's theorem* to the minimal modal logic K. Beside the arithmetical interpretation there is also a semantics given by transition systems. The class of all transitive and upward well-founded systems, that is where there is no infinite chain of successor nodes, forms a complete semantics for GL: a formula is derivable in the system GL if and only if it is valid over the class of all transitive and upward well-founded models.

Fixpoints and fixpoint theorems play an important role in GL. The most famous one, the *existence of a fixpoint* for guarded formulae was proved by De Jongh and Sambin independently (see [113]). Even though it is formulated and proved by strictly modal methods, the fixpoint theorem still has great arithmetical significance. The *uniqueness of the fixpoint* was proved later by Bernardi, De Jongh and Sambin independently (see [113]).

Since the modal μ-calculus is a general framework to study fixpoints in modal logic, applying methods from the modal μ-calculus for the study of this modal logic is a promising work. This has been done by van Benthem in [13] and Visser in [123]. Both authors establish, by using the De Jongh-Sambin fixpoint theorem, that the modal μ-calculus over GL collapses to its modal fragment. But since they use the already known fixpoint theorem in order to establish this collapse, in [13] van Benthem writes:

> "Our [...] analysis does not explain why provability fixed-points

> are *explicitly definable* in the modal base language. Indeed, the general reason seems unknown."

In this chapter an answer to this question is given. More precisely, we prove the collapse of the modal μ-calculus over GL without using the De Jongh-Sambin Theorem by showing that fixpoints are reached after two iterations of well-named fixpoint formulae.

Fixpoint theorems in GL hold also for modal formulae where the variable appears guarded but not necessarily positively and, from this point of view, this first result is not completely satisfactory since modal μ-calculus allows fixpoint constructors only for syntactically positive propositional variables. Therefore, we also introduce the modal $\mu^{\sim}$-calculus which allows fixpoint constructors for formulae where the fixpoint variable appears guarded. As can be done also for the standard μ-calculus we define the semantics by way of games, in this case only over transitive and upward well-founded transition systems. By using game-theoretical arguments and providing an explicit syntactical translation of the modal $\mu^{\sim}$-calculus into GL which preserves logical equivalence, we show that the modal $\mu^{\sim}$-calculus collapses to the modal fragment. As a corollary of the collapse, we obtain a new version of the De Jongh-Sambin Fixpoint Theorem with a simple algorithm which shows how the fixpoint can be computed. In this sense we give an answer to a generalization of van Benthem's question. Summing up, the modal $\mu^{\sim}$-calculus allows us to apply techniques similar as those known from the standard μ-calculus to GL and could be regarded as a starting point for further studies in this direction.

Both the collapse of the modal μ-calculus over GL and the one of the $\mu^{\sim}$-calculus over the same class of models are proved by using techniques and results from the previous chapter where the collapse of the semantical fixpoint alternation hierarchy over transitive models was proved.

In the next section we formally introduce Gödel-Löb Logic GL and some results which are already known. In Section 2 we analyze the modal μ-calculus over GL and show that it collapses to the modal fragment. In the last section we introduce the modal $\mu^{\sim}$-calculus and show a collapse to the modal fragment. The result is then used to provide a new proof of the uniqueness theorem of Bernardi, De Jongh and Sambin and of the existence theorem of De Jongh, Sambin. For the last one we also give a simple algorithm which shows how the fixpoint can be computed.

3.2 Gödel-Löb Logic GL

3.2.1 Syntax and Semantics

We start from an infinite countable set Prop of *propositional variables*. Then the collection $\mathcal{L}_{\mathsf{GL}}$ of GL-*formulae* is given by the usual grammar for modal formulae:

$$\varphi ::= p \mid \ {\sim}p \mid \top \mid \bot \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \Diamond\varphi \mid \Box\varphi$$

where $p \in \mathsf{Prop}$. If all propositional variables occurring in φ are in $\mathsf{P} \subseteq \mathsf{Prop}$, we write $\varphi \in \mathcal{L}_{\mathsf{GL}}(\mathsf{P})$. If ψ is a subformula of φ, we write $\psi \leq \varphi$. We write $\psi < \varphi$ when ψ is a proper subformula. $\mathsf{sub}(\varphi)$ is the set of all subformulae of φ. The formula $\neg\varphi$ is defined by using de Morgan dualities for boolean connectives and the modal dualities for $\Diamond$ and $\Box$ and the law of double negation. As usual, we introduce implication $\varphi \to \psi$ as $\neg\varphi \vee \psi$ and equivalence $\varphi \leftrightarrow \psi$ as $(\varphi \to \psi) \wedge (\varphi \to \psi)$. We say that $p \in \mathsf{Prop}$ is *guarded* in φ if $p \leq \varphi$ and all occurrences of p are in the scope of a modal operator.

The axioms and inference rules below give a deduction system for GL. As usual we write $\mathsf{GL} \vdash \varphi$ if there is a derivation of φ in the system presented below.

Axioms: All classical propositional tautologies, the *Distribution Axiom* from modal logic and the *Löb Axiom*

$$\Box(\Box\alpha \to \alpha) \to \Box\alpha.$$

Inference Rules are the *Modus Ponens* and the *Necessitation Rule*.

As for all modal logics the semantics of GL is given by transition systems. Recall that a *transition system* $\mathcal{T}$ is of the form $(\mathsf{S}, \to^{\mathcal{T}}, \lambda^{\mathcal{T}})$ where S is a set of *states*, $\to^{\mathcal{T}}$ is a binary relation on S called the *accessibility relation* and $\lambda : \mathsf{Prop} \to \wp(\mathsf{S})$ is a *valuation* for all propositional variables. A transition system $\mathcal{T}$ with a distinguished state s is called a *pointed transition system* and denoted by $(\mathcal{T}, s)$. $\mathbb{T}$ denotes the class of all pointed transition systems. The accessibility relation is called *upward well-founded* if there is no infinite chain of the form

$$s_0 \to^{\mathcal{T}} s_1 \to^{\mathcal{T}} s_2 \to \ldots.$$

By $\mathbb{T}^{\mathrm{wft}}$ we denote the subclass of pointed transition systems such that the accessibility relation is transitive and upward well-founded.

The denotation of a GL-formula in a transition system $\mathcal{T}$ and the notion of being a model of a GL-formula are defined as for modal logic. When all pointed transition systems $(\mathcal{T}, s) \in \mathbb{T}^{\mathrm{wft}}$ are a model of φ, we write $\mathsf{GL} \models \varphi$. A proof of the next theorem can be found in [30].

THEOREM 3.1. *For all* GL*-formulae* φ *we have that*

$$\mathsf{GL} \vdash \varphi \quad \textit{if and only if} \quad \mathsf{GL} \models \varphi.$$

3.2.2 Embedding GL into the modal μ-calculus

In this subsection we present an embedding t from GL into the modal μ-calculus. First, we define the function $()^* : \mathcal{L}_{\mathsf{GL}}(\mathsf{P}) \to \mathcal{L}_\mu(\mathsf{P})$ recursively on the structure of the formula such that

- $(p)^* \equiv p$ and $({\sim}p)^* \equiv {\sim}p$,
- $(\alpha \wedge \beta)^* \equiv (\alpha)^* \wedge (\beta)^*$ and $(\alpha \vee \beta)^* \equiv (\alpha)^* \vee (\beta)^*$
- $(\Box\alpha)^* \equiv \nu x.\Box(x \wedge (\alpha)^*)$, and
- $(\Diamond\alpha)^* \equiv \mu x.\Diamond(x \vee (\alpha)^*)$.

The embedding $t : \mathcal{L}_{\mathsf{GL}}(\mathsf{P}) \to \mathcal{L}_\mu(\mathsf{P})$ is now defined as

$$t(\varphi) \equiv (\mu x.\Box x) \to (\varphi)^*.$$

The following theorem is due to van Benthem [13]. It shows that GL which semantically lives on transitive and upward well-founded transition systems can be translated into the modal μ-calculus over arbitrary transition systems. For the first equivalence van Benthem provides a syntactical proof without using completeness results.

THEOREM 3.2 ([13]). *For all formulae* $\varphi \in \mathcal{L}_{\mathsf{GL}}$ *we have that*

$$(\mathsf{GL} \vdash \varphi \quad \Leftrightarrow \quad \mathsf{Koz} \vdash t(\varphi)) \quad \textit{and} \quad (\mathsf{GL} \models \varphi \quad \Leftrightarrow \quad \models t(\varphi)).$$

The next lemma verifies that over upward well-founded transition systems, least fixpoints and greatest fixpoints coincide.

LEMMA 3.3. *Let* $\mathcal{T}$ *be an upward well-founded transition system. Then, for every* $\varphi(x) \in \mathcal{L}_\mu$ *such that* x *is guarded and positive it holds that*

$$\|\mu x.\varphi(x)\|_{\mathcal{T}} = \|\nu x.\varphi(x)\|_{\mathcal{T}}.$$

Proof. Note, that in an evaluation game there are no infinite regeneration of x since then we would have an infinite chain of the form

$$s_0 \to^{\mathcal{T}} s_1 \to^{\mathcal{T}} s_2 \ldots.$$

Therefore, a winning play for $\nu x.\varphi$ is also a winning play for $\mu x.\varphi$. With the Fundamental Theorem of the semantic of the modal μ-calculus (Theorem 1.28) we get the result.

□

3.3 The modal μ-calculus over GL

In this section we show that the expressivity of the modal μ-calculus over GL, that is, over transitive and upward well-founded transition systems, is the same as the one of the modal base language. In this sense we answer to van Benthem's question cited in the introduction.

In the previous chapter we showed that over transitive transition systems every μ-formula is equivalent to a μ-formula without alternation of fixpoint operators. Moreover, we showed that under certain conditions a fixpoint operator can be eliminated by regenerating the formula. More precisely, in Theorem 2.17 we proved that given a transitive transition system $\mathcal{T}$, and a well-named μ-formula $\varphi(x)$ such that $x \in \mathsf{free}(\varphi)$ and occurs only once:

(1) If x is in the scope of a $\Box$ in $\nu x.\varphi(x)$ then

$$\|\nu x.\varphi(x)\|_{\mathcal{T}} = \|\varphi(\varphi((\top))\|_{\mathcal{T}}.$$

(2) If x is in the scope of a $\Diamond$ in $\mu x.\varphi(x)$ then

$$\|\mu x.\varphi(x)\|_{\mathcal{T}} = \|\varphi(\varphi((\bot))\|_{\mathcal{T}}.$$

DEFINITION 3.4. *The translation* $\tau : \mathcal{L}_\mu^{\mathsf{wn}}(\mathsf{P}) \to \mathcal{L}_{\mathsf{GL}}(\mathsf{P})$ *is defined recursively on the* rank *of the formula such that* $\tau((\sim)p) \equiv (\sim)p$*, such that* τ *distributes over boolean and modal connectives and such that for all* $\eta \in \{\mu, \nu\}$ *we have*

$$\tau(\eta x.\varphi) = \begin{cases} \tau(\mathsf{wn}(\varphi(\varphi(\top)))) & x \textit{ is in the scope of a } \Box \textit{ in } \varphi, \\ \tau(\mathsf{wn}(\varphi(\varphi(\bot)))) & \textit{else.} \end{cases}$$

Obviously, by first well-naming a formula and then applying τ *we get a translation from* $\mathcal{L}_\mu(\mathsf{P})$ *to* $\mathcal{L}_{\mathsf{GL}}(\mathsf{P})$.

The fact that over well-founded transition systems greatest and least fixpoint coincide almost immediately leads us to the collapse of the modal μ-calculus over GL into its modal fragment.

THEOREM 3.5. *On transitive and upward well-founded transition systems we have that the following holds for every $\varphi \in \mathcal{L}_\mu$:*

$$\|\varphi\|_{\mathcal{T}} = \|\tau(\mathsf{wn}(\varphi))\|_{\mathcal{T}}.$$

Proof. By Lemma 1.18 we can assume that φ is well-named. The proof is by induction on $\mathsf{rank}(\varphi)$. The base case and the case where $\mathsf{rank}(\varphi)$ is a successor ordinal are straightforward. If $\mathsf{rank}(\varphi)$ is a limit ordinal then φ is of the form $\eta x.\alpha$ ($\eta \in \{\mu, \nu\}$). Assume that φ is of the form $\nu x.\varphi$. If x is in the scope of a $\Box$ in φ then the induction step follows from Theorem 2.17. Else, x is only in the scope of some $\Diamond$ in φ. In this case by Lemma 3.3 we have that

$$\|\nu x.\varphi\|_{\mathcal{T}} = \|\mu x.\varphi\|_{\mathcal{T}}$$

and by applying once more Theorem 2.17 we get the induction step. The case where φ is of the form $\mu x.\varphi$ is shown by analogous arguments. $\Box$

3.4 The modal $\mu^{\sim}$-calculus

In this section we introduce a new language, called the modal $\mu^{\sim}$-calculus, which, in some sense, can be seen as an extension of the guarded fragment of the modal μ-calculus. The main novelty is that we allow the μ-operator to bind negative (and guarded) occurrences of propositional variables. Therefore, the modal $\mu^{\sim}$-calculus allows us to refer explicitly, that is, in a μ-calculus style, to fixpoints of guarded formulae. For example, the fixpoint of the "equation" $p \leftrightarrow \alpha(p)$ where $\alpha(x)$ is a guarded formula can be directly denoted as $\mu x.\alpha(x)$. As it can be done for the modal μ-calculus the semantics of the modal $\mu^{\sim}$-calculus is defined by way of games over transitive and upward well-founded transition systems. We provide an explicit syntactical translation of the modal $\mu^{\sim}$-calculus into GL which preserves logical equivalence. As a corollary of the collapse, we obtain a new version of the De Jongh-Sambin Fixpoint Theorem. The modal $\mu^{\sim}$-calculus could be seen as a starting point for the application of tools of the standard μ-calculus, as for example games, to GL.

3.4.1 Basic notions and results

The *language of the modal $\mu^{\sim}$-calculus*, $\mathcal{L}_{\mu^{\sim}}$, is almost the same as the one for the modal μ-calculus with the only difference that we allow fixpoint constructors also when the bound variable is appearing negatively, that is, *modal $\mu^{\sim}$-formulae* (or simply $\mu^{\sim}$-formulae) are defined as follows:

$$\varphi ::= p \mid\ \sim p \mid \top \mid \bot \mid (\varphi \wedge \varphi) \mid (\varphi \vee \varphi) \mid \Diamond\varphi \mid \Box\varphi \mid \mu x.\varphi$$

where $p, x \in \mathsf{Prop}$ and where x appears guarded in φ. All syntactical notions, such as $\mathcal{L}_{\mu^{\sim}}(\mathsf{P})$, bound variable, rank of a formula, φ_x, $\varphi^{\mathsf{free}(X)}$ etc., are defined as for the modal μ-calculus. Without loss of generality, we always suppose that $\mathsf{bound}(\varphi) \cap \mathsf{free}(\varphi) = \emptyset$.

We say that a $\mu^{\sim}$-formula φ is in *normal form* if $\mathsf{bound}(\varphi) \cap \mathsf{free}(\varphi) = \emptyset$ and if for all subformulae of φ of the form $\mu x.\mu y.\alpha$ we have that

- α is not of the form $\mu z.\beta$, and
- x occurs only negatively in α and y has only positive occurrences in α.

For the substitution, if $x \in \mathsf{free}(\varphi)$, then $\varphi[x/\psi]$ is given by substituting $\neg\psi$ to every negative occurrence $\sim x$ and by substituting ψ to every positive occurrence x. As for the modal μ-calculus negation is defined by using de Morgan laws and the duality of $\Box$ and $\Diamond$, in addition we set

$$\neg\mu x.\alpha \equiv \mu x.\neg\alpha[x/\neg x].$$

The last equivalence can be rather surprising at a first look. It is motivated by the fact that the modal $\mu^{\sim}$-calculus will be interpreted over upward well-founded models, where least and greatest fixpoint coincide.

Note that, for every $y \in \mathsf{bound}(\mu x.\alpha)$, y is negative in $\mu x.\alpha$ if and only if y is negative in $\neg\mu x.\alpha$.

The semantics for the modal $\mu^{\sim}$-calculus over GL is given by evaluation games on pointed upward well-founded and transitive transition systems. These evaluation games are similar to the ones for the modal μ-calculus. Let $\varphi \in \mathcal{L}_{\mu^{\sim}}$ and $(\mathcal{T}, s) \in \mathbb{T}^{\mathrm{wft}}$.

- First we construct recursively the two arenas $\langle V_0^+, V_1^+, E^+\rangle$ from φ and $(\mathcal{T}, s)$ and $\langle V_0^-, V_1^-, E^-\rangle$ from $\neg\varphi$ and $(\mathcal{T}, s)$ as it is done for the modal μ-calculus, except the fact that, for each vertex of the form $\langle \sim x, t\rangle$ which was generated in the recursion defining the arena $\langle V_0^+, V_1^+, E^+\rangle$ we put the condition
$$\langle \sim x, t\rangle \in V_0^+ \text{ and } E^+(\langle \sim x, t\rangle) = \emptyset,$$
and if it was generated in the recursion defining the arena $\langle V_0^-, V_1^-, E^-\rangle$ we set
$$\langle \sim x, t\rangle \in V_0^- \text{ and } E^-(\langle \sim x, t\rangle) = \emptyset.$$
- Then the arena of $\mathcal{E}(\varphi, (\mathcal{T}, s))$ is the triple $\langle V_0, V_1, E\rangle$ defined by taking the disjoint union of the two previous arenas, with the following modification:
  - For every vertex of the form $\langle \sim x, t\rangle$ where $x \in \mathsf{bound}(\varphi)$ we set
$$E(\langle \sim x, t\rangle) = \begin{cases} \{\langle \neg\varphi_x, t\rangle\} \subseteq V^- & \text{if } \langle \sim x, t\rangle \in V_0^+ \\ \{\langle \varphi_x, t\rangle\} \subseteq V^+ & \text{if } \langle \sim x, t\rangle \in V_0^-. \end{cases}$$

Since we are on upward well-founded models and that all regenerated variables are guarded, all plays are finite. Therefore, we have that Player 0 wins if and only if the last vertex of the play belongs to Player 1. Because we do not have to care about priorities, the previous winning condition is admissible and the evaluation game for $\mu^{\sim}$-formulae is therefore well-defined[1].

We say that a pointed upward well-founded transitive transition system $(\mathcal{T}, s)$ is a model of a $\mu^{\sim}$-formula if and only if Player 0 has a winning strategy in $\mathcal{E}(\varphi, (\mathcal{T}, s))$. Further, we define

$$\|\varphi\|_{\mathcal{T}}^{\mathcal{W}} = \{s \in \mathsf{S} \mid (\mathcal{T}, s) \text{ is a model of } \varphi\}.$$

[1]Note that on non well-founded models a play can be infinite. Thus, since in this kind of plays it is possible that Player 0 and Player 1 "switch" their roles infinitely often, it is not clear how to extend our game-theoretical approach also to non well-founded models by adding a natural and uniform (parity) winning conditions for infinite plays.

By $\|\varphi\|^{\mathcal{W}}$ we denote the class of all upward well-founded and transitive models of φ, that is, all pointed transition systems $(\mathcal{T}, s)$, transitive and upward well-founded, such that $s \in \|\varphi\|^{\mathcal{W}}$.

EXAMPLE 3.6. Consider the formula $\mu x.\Diamond \sim x$. This formula says that Player 0 can always force the number of the states visited in a play to be even. Because the considered models are transitive, this implies that the formula says that the root of the models has at least one accessible state.

The next lemma states some basic properties of denotation.

LEMMA 3.7. *For all transition systems $\mathcal{T} = (\mathsf{S}, \to^{\mathcal{T}}, \lambda^{\mathcal{T}})$ and all $\mu^{\sim}$-formulae $\mu x.\varphi$ we have that*

(1) $\|\mu x.\mu y.\varphi(x, y)\|^{\mathcal{W}}_{\mathcal{T}} = \|\mu x.\varphi(x, x)\|^{\mathcal{W}}_{\mathcal{T}}$,

(2) $\|\mu x_1 \dots \mu x_n.\varphi\|^{\mathcal{W}}_{\mathcal{T}} = \|\mu x_{p(1)} \dots \mu x_{p(n)}.\varphi\|^{\mathcal{W}}_{\mathcal{T}}$, *where p is any permutation over $\{1, \dots, n\}$,*

(3) There is a well-named formula $\mathsf{wn}(\varphi)$ such that $\|\varphi\|^{\mathcal{W}}_{\mathcal{T}} = \|\mathsf{wn}(\varphi)\|^{\mathcal{W}}_{\mathcal{T}}$,

(4) There is a formula $\mathsf{nf}(\varphi)$ in normal form such that $\|\varphi\|_{\mathcal{T}} = \|\mathsf{nf}(\varphi)\|_{\mathcal{T}}$.

Proof. Part 1 is by definition of the evaluation game for the modal $\mu^{\sim}$-calculus. Part 2 is proved by an easy induction on the length of the prefix. Part 3 is a straightforward consequence of part 1. Part 4 is a straightforward consequence of part 1 and part 2.

□

The next lemma shows that over upward well-founded models, the positively bounded fragment of the modal $\mu^{\sim}$-calculus coincides with the guarded fragment of the standard modal μ-calculus.

LEMMA 3.8. *Let $\varphi \in \mathcal{L}_{\mu} \cap \mathcal{L}_{\mu^{\sim}}$. Then for every upward-well founded model $\mathcal{T}$*

$$\|\varphi\|^{\mathcal{W}}_{\mathcal{T}} = \|\varphi\|_{\mathcal{T}}.$$

Proof. This follows by applying Theorem 1.28 to the fact that, for every $\varphi \in \mathcal{L}_{\mu} \cap \mathcal{L}_{\mu^{\sim}}$, the evaluation games for $\mathcal{L}_{\mu^{\sim}}$ and the evaluation games for the μ-calculus coincide over upward well-founded models. □

The next lemma shows that negation behaves as expected.

LEMMA 3.9. *Let φ be a $\mu^{\sim}$-formula and $\mathcal{T} = (\mathsf{S}, \to^{\mathcal{T}}, \lambda^{\mathcal{T}})$ an upward well-founded transition system. We have that*

$$\|\neg\varphi\|^{\mathcal{W}}_{\mathcal{T}} = \mathsf{S} \setminus \|\varphi\|^{\mathcal{W}}_{\mathcal{T}}.$$

Proof. Consider the evaluation game $\mathcal{E}(\varphi, (\mathcal{T}, s))$ where Player 0 starts to play as Player 1 and vice versa. Clearly Player 0 (resp. Player 1) has a winning strategy in this modified game iff she has a winning strategy in $\mathcal{E}(\neg\varphi, (\mathcal{T}, s))$. From this fact we get the claim. □

The next lemma shows that in the modal $\mu^{\sim}$-calculus formulae of the form $\mu x.\varphi$ indeed define a fixpoint.

Lemma 3.10. *For every $\mu x.\varphi \in \mathcal{L}_{\mu^\sim}$ and every upward well-founded transition system $\mathcal{T}$ it holds that*

$$\|\mu x.\varphi\|_{\mathcal{T}}^{\mathcal{W}} = \|\varphi[x/\mu x.\varphi]\|_{\mathcal{T}}^{\mathcal{W}}.$$

Proof. This result follows straightforwardly by definition of the evaluation game for the modal $\mu^\sim$-calculus. □

3.4.2 The unicity of fixpoints

Let $\mathcal{T}$ be a upward well-founded and transitive transition system and φ a $\mu^\sim$-formula. Consider an arbitrary (memoryless) strategy σ for Player 0, not necessarily winning. We define the *restriction of* $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$ *on* σ, denoted by $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$, as follows:

- The set of positions $V|_\sigma$ of the restriction is given by all nodes which are the positions of some play compatible with σ starting from position $\langle \varphi, s_0 \rangle$,
- The arena of $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$ is the triple $\langle V_0|_\sigma, V_1|_\sigma, E|_\sigma \rangle$ where:
  (1) $V_0|_\sigma = \emptyset$,
  (2) $V_1|_\sigma = V|_\sigma$,
  (3) if $\langle \psi, s \rangle \in V|_\sigma \cap V_1$ then $E|_\sigma(\langle \psi, s \rangle) = E(\langle \psi, s \rangle)$, and
  (4) if $\langle \psi, s \rangle \in V|_\sigma \cap V_0$ then $E|_\sigma(\langle \psi, s \rangle) = \{\sigma(\langle \psi, s \rangle)\}$.

We have that in $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$ the only Player who can move is Player 1. This can be done because the moves for Player 0 are already completely determined by the (memoryless) strategy σ. Clearly, any play in $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$ is a play in $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$ compatible with σ. We say that a play π in $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$ is winning for Player 0 if and only if the play π is winning for Player 0 in $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$. If σ is a winning strategy for Player 0 then any play in $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$ is winning for Player 0.

In the next definition, we define a measure on the graph of an evaluation game restricted to a strategy for Player 0, which is essentially just the distance in the considered graph.

Definition 3.11. *Let $\mathcal{T}$ be a upward well-founded transitive transition system, φ a $\mu^\sim$-formula and σ any strategy for Player 0 in the parity game $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$. Then, for every position $\langle \psi, s \rangle$ of $\mathcal{E}|_\sigma(\varphi, (\mathcal{T}, s_0))$, we define a measure $d(\langle \psi, s \rangle)$:*

$$d(\langle \psi, s \rangle) = \begin{cases} 0 & \textit{if} \quad E|_\sigma(\langle \psi, s \rangle) = \emptyset \\ \sup\{d(\langle \psi, s' \rangle) + 1 : \langle \psi, s' \rangle \in E|_\sigma(\langle \psi, s \rangle)\} & \textit{otherwise.} \end{cases}$$

Note that, since $\mathcal{T}$ is upward well-founded, there cannot be an infinite chain of the form $\langle a_0, a_1, a_2, \dots \rangle$ such that for every $i \geq 0$, $\langle a_i, a_{i+1} \rangle \in E|_\sigma$. Therefore for all evaluation games $\mathcal{E}(\varphi, (\mathcal{T}, s_0))$ and all vertices $\langle \psi, v \rangle$ in the arena $d(\langle \psi, v \rangle)$ is a well-defined ordinal number, such that if a vertex $\langle \alpha, v' \rangle$ is reachable from a vertex $\langle \beta, v'' \rangle$ then we have that $d(\langle \alpha, v' \rangle) < d(\langle \beta, v'' \rangle)$.

The next theorem shows that a fixpoint formula $\mu x.\alpha(x)$ in the modal $\mu^\sim$-calculus defines a fixpoint, as proved in Lemma 3.10, and that any other fixpoint of a formula $\alpha(x)$ is identical to $\mu x.\alpha(x)$. In this sense it is an existence and uniqueness theorem, and it is the central result of the section.

THEOREM 3.12. *Let $\varphi(x) \in \mathcal{L}_{\mu^{\sim}}$ and $x \in \mathsf{free}(\varphi)$ a guarded variable. Let $\mathcal{T}$ be a upward well-founded transitive transition system. Then, for every $A \subseteq \mathsf{S}$ we have*

$$\|\varphi(A)\|_{\mathcal{T}}^{\mathcal{W}} = A \text{ if and only if } A = \|\mu x.\varphi(x)\|_{\mathcal{T}}^{\mathcal{W}}.$$

Proof. The implication from right to left follows from Lemma 3.10. In order to prove the implication from left to right, suppose $\|\varphi(A)\|_{\mathcal{T}}^{\mathcal{W}} = A$. From the definition of evaluation game we straightforwardly can derive the inclusion $A \subseteq \|\mu x.\varphi(x)\|_{\mathcal{T}}^{\mathcal{W}}$. For the other inclusion, suppose that $s \in \|\mu x.\varphi(x)\|_{\mathcal{T}}^{\mathcal{W}}$. We have that Player 0 has a winning strategy σ in $\mathcal{E}(\mu x.\varphi, (\mathcal{T}, s))$. Consider the restricted evaluation game $\mathcal{E}|_{\sigma}(\mu x.\varphi, (\mathcal{T}, s))$. For each vertex $\langle \alpha, v\rangle$ in $\mathcal{E}|_{\sigma}(\mu x.\varphi, (\mathcal{T}, s))$, we have that $d(\langle \alpha, v\rangle)$ is a well-defined measure. Clearly, with the following claim we finish the proof.

Claim: For all vertices of the form $\langle \alpha, s'\rangle$ in $\mathcal{E}|_{\sigma}(\mu x.\varphi, (\mathcal{T}, s))$ if $\alpha = \mu x.\varphi$ then $s' \in A$, and, if $\alpha = \mu x.\neg\varphi$ then $s' \notin A$.

The proof of the claim is by induction on d. Since $d(\langle \mu x.\varphi, s'\rangle) > 0$ and $d(\langle \mu x.\neg\varphi, s'\rangle) > 0$ the induction base is trivial. For the induction step assume first that we have a vertex $\langle \mu x.\varphi, s'\rangle$ in $\mathcal{E}|_{\sigma}(\mu x.\varphi, (\mathcal{T}, s))$. We distinguish two cases:

(1) If from $\langle \mu x.\varphi, s'\rangle$ there is no a reachable vertex of the form $\langle \mu x.\varphi, s''\rangle$ or $\langle \mu x.\neg\varphi, s''\rangle$ then we have that $s' \in \|\varphi(A')\|_{\mathcal{T}}^{\mathcal{W}}$ for all sets of states A' and, therefore, we also have $s' \in \|\varphi(A)\|_{\mathcal{T}}^{\mathcal{W}}$. Since by assumption $\|\varphi(A)\|_{\mathcal{T}}^{\mathcal{W}} = A$ we proved the claim.

(2) Otherwise we distinguish two subcases given by the first vertex reached which is of either the form $\langle \mu x.\varphi, s''\rangle$ or $\langle \mu x.\neg\varphi, s''\rangle$.

   (a) If the first vertex reached of such kind is $\langle \mu x.\varphi, s''\rangle$, then, since we have that $d(\langle \mu x.\varphi, s'\rangle) > d(\langle \mu x.\varphi, s''\rangle)$, by induction hypothesis we get $s'' \in A$.

   (b) If the first vertex reached of such kind is $\langle \mu x.\neg\varphi, s''\rangle$, then, since we have that $d(\langle \mu x.\varphi, s'\rangle) > d(\langle \mu x.\neg\varphi, s''\rangle)$, by induction hypothesis we get $s'' \notin A$.

   Therefore, for each play consistent with σ starting from $\langle \mu x.\varphi, s'\rangle$ it holds that if it reaches first a vertex of the form $\langle \mu x.\varphi, s''\rangle$ (or equivalently of the form $\langle x, s''\rangle$) we have that $s'' \in A$, and, if it reaches first a vertex of the form $\langle \mu x.\neg\varphi, s''\rangle$ (or equivalently of the form $\langle \sim x, s''\rangle$) we have that $s'' \notin A$. But it can easily be seen that this implies $s' \in \|\varphi(A)\|_{\mathcal{T}}^{\mathcal{W}}$. Since by assumption $\|\varphi(A)\|_{\mathcal{T}}^{\mathcal{W}} \subseteq A$ we finish the induction step for the case $\alpha = \mu x.\varphi(x)$.

The induction step for a vertex of the form $\langle \mu x.\neg\varphi, s'\rangle$ is verified in the same way by using the fact that by Lemma 3.9 we have that $\|\neg\varphi(A)\|_{\mathcal{T}}^{\mathcal{W}} = S \setminus \|\varphi(A)\|_{\mathcal{T}}^{\mathcal{W}}$ and, therefore, by assumption that $\|\neg\varphi(A)\|_{\mathcal{T}}^{\mathcal{W}} = S \setminus A$.

□

COROLLARY 3.13. *Let φ and ψ be two $\mu^{\sim}$-formulae. If for all upward well-founded transitive transition system $\mathcal{T}$ we have that $\|\psi\|_{\mathcal{T}}^{\mathcal{W}} = \|\varphi\|_{\mathcal{T}}^{\mathcal{W}}$ then for all variables x and all $\mathcal{T}$ we have that*

$$\|\mu x.\psi\|_{\mathcal{T}}^{\mathcal{W}} = \|\mu x.\varphi\|_{\mathcal{T}}^{\mathcal{W}}$$

Proof. By the "if" direction of Theorem 3.12 we have that

$$\|\mu x.\psi\|_{\mathcal{T}}^{\mathcal{W}} = \|\psi\|_{\mathcal{T}[x\mapsto\|\mu x.\psi\|_{\mathcal{T}}^{\mathcal{W}}]}^{\mathcal{W}}$$

and with the premise of the corollary we get

$$\|\mu x.\psi\|_{\mathcal{T}}^{\mathcal{W}} = \|\varphi\|_{\mathcal{T}[x\mapsto\|\mu x.\psi\|_{\mathcal{T}}^{\mathcal{W}}]}^{\mathcal{W}}.$$

Applying the "only if" direction of Theorem 3.12 we obtain that

$$\|\mu x.\psi\|_{\mathcal{T}}^{\mathcal{W}} = \|\mu x.\varphi\|_{\mathcal{T}}^{\mathcal{W}}.$$

□

The next theorem provides a new proof of Bernardi, De Jongh, Sambin Theorem (c.f. Chapter 8 in [30] or [113]) using our results on the modal $\mu^{\sim}$-calculus.

THEOREM 3.14. *Let $\varphi(x) \in \mathcal{L}_{\mathsf{GL}}$, where x is guarded. We have that*

$$\mathsf{GL} \vdash \Box^s(p \leftrightarrow \varphi(p)) \wedge \Box^s(q \leftrightarrow \varphi(q)) \rightarrow (p \leftrightarrow q)$$

where $\Box^s\varphi :\equiv \Box\varphi \wedge \varphi$.

Proof. By Theorem 3.2 it is enough to show that for all $\varphi(x) \in \mathcal{L}_{\mathsf{GL}}$ it holds that

$$\models \mu x.\Box x \rightarrow (((\Box^s)(p \leftrightarrow \varphi(p)) \wedge (\Box^s)(q \leftrightarrow \varphi(q))) \rightarrow (p \leftrightarrow q))^*.$$

And this can be done by showing for all $\varphi(x) \in \mathcal{L}_{\mathsf{GL}}$ that the following formula is valid for all transition systems

$$\mu x.\Box x \rightarrow (p \leftrightarrow \varphi(p) \wedge q \leftrightarrow \varphi(q) \wedge \Box^*(q \leftrightarrow \varphi(q)) \wedge \Box^*(q \leftrightarrow \varphi(q)) \rightarrow (p \leftrightarrow q)) \tag{3.1}$$

where $\Box^*\gamma \equiv \nu x.\Box(x \wedge \gamma)$.

Assume, that we have

$$s \in \|\mu x.\Box x \wedge p \leftrightarrow \varphi(p) \wedge q \leftrightarrow \varphi(q) \wedge \Box^*(q \leftrightarrow \varphi(q)) \wedge \Box^*(q \leftrightarrow \varphi(q))\|_{\mathcal{T}}.$$

Then, $(\mathcal{T}, s)$ is well-founded and we have for s and for all reachable states s' from s that $q \leftrightarrow \varphi(q)$ and $p \leftrightarrow \varphi(p)$. Therefore, if we assume that $\mathcal{T}$ consists of s and all reachable states from s, which is an admissible assumption, we get that we have

$$\lambda^{\mathcal{T}}(p) = \|\varphi(\lambda^{\mathcal{T}}(p))\|_{\mathcal{T}}^{\mathcal{W}} \quad \text{and} \quad \lambda^{\mathcal{T}}(q) = \|\varphi(\lambda^{\mathcal{T}}(q))\|_{\mathcal{T}}^{\mathcal{W}}.$$

By Theorem 3.12 we get that

$$\lambda^{\mathcal{T}}(p) = \|\mu x.\varphi(x)\|_{\mathcal{T}}^{\mathcal{W}} \quad \text{and} \quad \lambda^{\mathcal{T}}(q) = \|\mu x.\varphi(x)\|_{\mathcal{T}}^{\mathcal{W}}.$$

and therefore we obtain that

$$s \in \|p \leftrightarrow q\|_{\mathcal{T}}.$$

We have shown Equation 3.1 and finished the proof. □

3.4.3 Collapsing the modal $\mu^{\sim}$-calculus

In this subsection we provide an explicit syntactical translation of the modal μ-calculus into GL which preserves logical equivalence. As a corollary, we obtain a new proof of the De Jongh-Sambin Fixpoint Theorem which provides an explicit construction of the fixpoint formula based on the syntactical translation defining the collapse.

First of all, remember that, by Lemma 3.7.4, we can suppose that every $\mu^{\sim}$-formula is in normal form.

LEMMA 3.15. *Let $\alpha(x)$ be a modal formula such that x appears only negatively and guarded. Then, for every $\mathcal{T} \in \mathbb{T}^{wft}$ we have that*

$$\|\mu x.(\alpha[x/\alpha(x)])\|^{\mathcal{W}}_{\mathcal{T}} = \|\mu x.\alpha(x)\|^{\mathcal{W}}_{\mathcal{T}}.$$

Proof. Let A be $\|\mu x.\alpha(x)\|^{\mathcal{W}}_{\mathcal{T}}$. By the "if" direction of Theorem 3.12 we have that $\|\alpha(A)\|^{\mathcal{W}}_{\mathcal{T}} = A$. We can iterate this equivalence twice and get

$$\|\alpha[x/\alpha(A)]\|^{\mathcal{W}}_{\mathcal{T}} = A.$$

Applying the "only if" direction of Theorem 3.12 gives us

$$\|\mu x.((\alpha[x/\alpha(x)])\|^{\mathcal{W}}_{\mathcal{T}} = A$$

and therefore the proof of this lemma.

□

Note that, if $x \in \mathsf{bound}(\mu x.\alpha)$ appears only negatively, then x occurs only positively in $\mu x.(\alpha[x/\alpha(x)])$.

Everything is now set up in order to prove that the modal $\mu^{\sim}$-calculus over GL collapses to its modal fragment.

DEFINITION 3.16. *The syntactical translation $\mathcal{I} : \mathcal{L}_{\mu^{\sim}} \to \mathcal{L}_{\mathsf{GL}}$ uses the translation τ from $\mathcal{L}_{\mu}$ to $\mathcal{L}_{\mathsf{GL}}$ of Definition 3.4. It is defined recursively as follows:*

- $\mathcal{I}(p) = p$ *and* $\mathcal{I}(\sim p) =\sim p$.
- $\mathcal{I}(\bot) = \bot$ *and* $\mathcal{I}(\top) = \top$.
- $\mathcal{I}(\alpha \circ \beta) = \mathcal{I}(\alpha) \circ \mathcal{I}(\beta)$, *where* $\circ \in \{\wedge, \vee\}$.
- $\mathcal{I}(\triangle\, \beta) = \triangle\, \mathcal{I}(\beta)$, *where* $\triangle \in \{\Box, \Diamond\}$.
- *Assume that* $\mathsf{nf}(\mu x.\mathcal{I}(\alpha(x)))$ *is of the form* $\mu z.\mu y.\hat{\alpha}(z, y)$. *We set*

$$\mathcal{I}(\mu x.\alpha) = \tau(\mathsf{wn}(\mu z.\beta(z))),$$

where $\beta(z) \equiv \tau(\mathsf{wn}(\mu y.\hat{\alpha}(z, y)))[z/\tau(\mathsf{wn}(\mu y.\hat{\alpha}(z, y)))])$.

LEMMA 3.17. *The translation $\mathcal{I}$ is well-defined and, moreover, if*

$$\varphi \in \mathcal{L}_{\mu^{\sim}}(\mathsf{P}) \text{ then } \mathcal{I}(\varphi) \in \mathcal{L}_{\mathsf{GL}}(\mathsf{P}).$$

Proof. By induction on the structure of the formula. The only critical case is when $\varphi \equiv \mu x.\alpha$. By induction hypothesis, $\mathcal{I}(\alpha(x)) \in \mathcal{L}_{\mathsf{GL}}$. Therefore $\hat{\alpha}(y,z) \in \mathcal{L}_{\mathsf{GL}}$. By definition of normal form, z occurs only negatively and y occurs only positively in $\mu z.\mu y.\hat{\alpha}(y,z)$. Thus, $\mu y.\hat{\alpha}(y,z) \in \mathcal{L}_\mu$. This implies that $\mathsf{wn}(\mu y.\hat{\alpha}(y,z))$ is well-defined and by Theorem 3.5 that $\tau(\mathsf{wn}(\mu y.\hat{\alpha}(y,z))) \in \mathcal{L}_{\mathsf{GL}}$. Note that y occurs only positively in $\hat{\alpha}(y,z)$ and $\mathsf{wn}(\mu y.\hat{\alpha}(y,z))$ is given by duplicating and renaming y. Therefore, it follows that z occurs only positively in

$$\tau(\mathsf{wn}(\mu y.\hat{\alpha}(z,y)))[z/\tau(\mathsf{wn}(\mu y.\hat{\alpha}(z,y)))]).$$

This implies that $\mu z.\beta(z) \in \mathcal{L}_\mu$ and therefore that $\mathsf{wn}(\mu z.\beta(z))$ is well-defined. Thence by Theorem 3.5 we have that $\tau(\mathsf{wn}(\mu z.\beta(z))) \in \mathcal{L}_{\mathsf{GL}}$. ◻

THEOREM 3.18. *Let $\varphi \in \mathcal{L}_{\mu^\sim}$. On upward well-founded and transitive transition systems $\mathcal{T}$ we have that*

$$\|\varphi\|_{\mathcal{T}}^{\mathcal{W}} = \|\mathcal{I}(\varphi)\|_{\mathcal{T}}^{\mathcal{W}}.$$

Proof. The proof goes by induction on $\mathsf{rank}(\varphi)$. If $\mathsf{rank}(\varphi) = 1$ or $\mathsf{rank}(\varphi)$ is a successor ordinal the induction step is straightforward. If $\mathsf{rank}(\varphi)$ is a limit ordinal then φ is of the form $\mu x.\alpha$. In this case by Lemma 3.7 we have that

$$\|\mu x.\mathcal{I}(\alpha)\|_{\mathcal{T}}^{\mathcal{W}} = \|\mu z.\mu y.\hat{\alpha}(z,y)\|_{\mathcal{T}}^{\mathcal{W}}.$$

Since by induction hypothesis we have that $\|\mathcal{I}(\alpha)\|_{\mathcal{T}}^{\mathcal{W}} = \|\alpha\|_{\mathcal{T}}^{\mathcal{W}}$, with Corollary 3.13 we get that

$$\|\mu x.\mathcal{I}(\alpha)\|_{\mathcal{T}}^{\mathcal{W}} = \|\mu x.\alpha\|_{\mathcal{T}}^{\mathcal{W}}$$

and therefore that

$$\|\mu x.\alpha\|_{\mathcal{T}}^{\mathcal{W}} = \|\mu z.\mu y.\hat{\alpha}(z,y)\|_{\mathcal{T}}^{\mathcal{W}}. \tag{3.2}$$

Since by Lemma 3.17 and by construction of normal forms, $\hat{\alpha}$ is a modal formula we have that $\mu y.\hat{\alpha} \in \mathcal{L}_\mu$. With Theorem 3.5 and Lemma 3.7 we get that for all upward well-founded and transitive $\mathcal{T}$ we have that

$$\|\mu y.\hat{\alpha}\|_{\mathcal{T}}^{\mathcal{W}} = \|\tau(\mathsf{wn}(\mu y.\hat{\alpha}))\|_{\mathcal{T}}^{\mathcal{W}}.$$

By Corollary 3.13 it holds that

$$\|\mu z.\mu y.\hat{\alpha}\|_{\mathcal{T}}^{\mathcal{W}} = \|\mu z.\tau(\mathsf{wn}(\mu y.\hat{\alpha}))\|_{\mathcal{T}}^{\mathcal{W}}$$

and with Equation 3.2 that

$$\|\mu x.\alpha\|_{\mathcal{T}}^{\mathcal{W}} = \|\mu z.\tau(\mathsf{wn}(\mu y.\hat{\alpha}))\|_{\mathcal{T}}^{\mathcal{W}}. \tag{3.3}$$

Remember that y occurs only positively and z only negatively in $\hat{\alpha}$. Moreover $\mathsf{wn}(\mu y.\hat{\alpha}(y,z))$ is obtained by multiplying and renaming y. Therefore, since z appears only negatively in $\mu y.\alpha(y,z)$ it appears only negatively in $\mathsf{wn}(\mu y.\hat{\alpha}(y,z))$, too. Now, note that by definition of τ we are "regenerating" the formula only on positive occurrences and, therefore, we have that z appears only negatively in $\tau(\mu y.\hat{\alpha})$, too. By Lemma 3.15 it holds that

$$\|\mu z.(\tau(\mathsf{wn}(\mu y.\hat{\alpha})))\|_{\mathcal{T}}^{\mathcal{W}} = \|\mu z.\big(\tau(\mathsf{wn}(\mu y.\hat{\alpha}))[z/\tau(\mathsf{wn}(\mu y.\hat{\alpha}))](z)\big)\|_{\mathcal{T}}^{\mathcal{W}}.$$

With Equation 3.3 we get

$$\|\mu x.\alpha\|_{\mathcal{T}}^{\mathcal{W}} = \|\mu z.\big(\tau(\mathsf{wn}(\mu y.\hat{\alpha}))[z/\tau(\mathsf{wn}(\mu y.\hat{\alpha}))](z)\big)\|_{\mathcal{T}}^{\mathcal{W}}.$$

By Lemma 3.7 and Theorem 3.5 we finish the induction step. ◻

The last theorem of this chapter is a new version of the De Jongh-Sambin Fixpoint Theorem. Our version provides an explicit construction of the fixpoint formula based on the definition of $\mathcal{I}$.

THEOREM 3.19. *Let $\varphi(x) \in \mathcal{L}_{\mathsf{GL}}(\mathsf{P})$, where x is guarded. We have that*

$$\mathsf{GL} \vdash \mathcal{I}(\mu x.\varphi) \leftrightarrow \varphi(\mathcal{I}(\mu x.\varphi)).$$

Further if $\varphi \in \mathcal{L}_{\mathsf{GL}}(\mathsf{P})$ then we have that $\mathcal{I}(\mu x.\varphi) \in \mathcal{L}_{\mathsf{GL}}(\mathsf{P} \setminus \{x\})$.

Proof. The fact that $\mathcal{I}(\mu x.\varphi) \in \mathcal{L}_{\mathsf{GL}}(\mathsf{P} \setminus \{x\})$ follows from Lemma 3.17. For the provable equivalence, we show that $\mathsf{GL} \models \mathcal{I}(\mu x.\varphi) \leftrightarrow \varphi(\mathcal{I}(\mu x.\varphi))$. The proof then follows by Theorem 3.1. Let $\mathcal{T} \in \mathbb{T}^{\mathrm{wft}}$. We have

$$\begin{array}{rcll}
\|\mathcal{I}(\mu x.\varphi(x))\|_{\mathcal{T}} & = & \|\mu x.\varphi(x)\|^{\mathcal{W}}_{\mathcal{T}} & \text{Lemma 3.8 and Theorem 3.18} \\
& = & \|\varphi(\mu x.\varphi(x))\|^{\mathcal{W}}_{\mathcal{T}} & \text{Lemma 3.10} \\
& = & \|\varphi(x)\|^{\mathcal{W}}_{\mathcal{T}[x \mapsto \|\mu x.\varphi\|^{\mathcal{W}}_{\mathcal{T}}]} & \text{Definiton of evaluation game} \\
& = & \|\varphi(x)\|^{\mathcal{W}}_{\mathcal{T}[x \mapsto \|\mathcal{I}(\mu x.\varphi)\|^{\mathcal{W}}_{\mathcal{T}}]} & \text{Theorem 3.18} \\
& = & \|\varphi(x)\|_{\mathcal{T}[x \mapsto \|\mathcal{I}(\mu x.\varphi)\|_{\mathcal{T}}]} & \text{Lemma 3.8} \\
& = & \|\varphi(\mathcal{I}(\mu x.\varphi))\|_{\mathcal{T}} & \text{Definition of denotation}
\end{array}$$

□

We end with two examples where we apply our translation in order to solve a modal equation.

EXAMPLE 3.20. Consider the modal equation $x \leftrightarrow \neg\Box x$. It is the same as

$$x \leftrightarrow \Diamond \sim x. \tag{3.4}$$

By Theorem 3.19 the $\mu^{\sim}$-formula $\mu x.\Diamond \sim x$ is the solution of Equation 3.4. By definition of $\mathcal{I}$ we have that

$$\mathcal{I}(\mu x.\Diamond \sim x) = \tau(\mu x.\Diamond\neg\Diamond \sim x) = \tau(\mu x.\Diamond\Box x) = \Diamond\Box\Diamond\Box\top.$$

Note, that on upward-well-founded transitive transition system $\mathcal{T}$, it holds that $\|\Diamond\Box\Diamond\Box\top\|_{\mathcal{T}} = \|\neg\Box\bot\|_{\mathcal{T}}$.

EXAMPLE 3.21. Consider the modal equation $x \leftrightarrow (\Box(x \to q) \to \Box \sim x)$. This is the same as

$$x \leftrightarrow \Diamond(x\wedge \sim q) \vee \Box \sim x. \tag{3.5}$$

By Theorem 3.19 the formula $\mathcal{I}(\mu x.\Diamond(x\wedge \sim q) \vee \Box \sim x)$ is a solution of Equation 3.5. Let's trace the construction of the fixpoint given by Definition 3.16:

We have that

$$\hat{\alpha} \equiv \Diamond(x\wedge \sim q) \vee \Box \sim y$$

and that

$$\tau(\mu x.\hat{\alpha}) \equiv \Diamond((\Diamond(\bot\wedge \sim q) \vee \Box \sim y)\wedge \sim q) \vee \Box \sim y.$$

The formula $\tau(\mu x.\hat{\alpha})$ can be simplified by using the following equivalence

$$\|\tau(\mu x.\hat{\alpha})\|_{\mathcal{T}} = \|\Diamond(\Box \sim y\wedge \sim q) \vee \Box \sim y\|_{\mathcal{T}}.$$

Now, we calculate $\beta(y)$ of Definition 3.16 by using the simplified $\tau(\mu x.\hat{\alpha})$ above and get

$$\beta(y) \equiv \Diamond(\Box\neg(\Diamond(\Box \sim y \wedge \sim q) \vee \Box \sim y) \wedge \sim q) \vee \Box\neg(\Diamond(\Box \sim y \wedge \sim q) \vee \Box \sim y).$$

By definition of negation, we get

$$\beta(y) \equiv \Diamond(\Box(\Box(\Diamond y \vee q) \wedge \Diamond y) \wedge \sim q) \vee \Box(\Box(\Diamond y \vee q) \wedge \Diamond y).$$

Note that the following semantical equivalences hold

- $\|\Diamond(\Box(\Box(\Diamond y \vee q) \wedge \Diamond y) \wedge \sim q)\|_{\mathcal{T}} = \|\Diamond(\Box\bot \wedge \sim q)\|_{\mathcal{T}}$, and
- $\|\Box(\Box(\Diamond y \vee q) \wedge \Diamond y)\|_{\mathcal{T}} = \|\Box\bot\|_{\mathcal{T}}$.

Therefore, we get

$$\|\mu y.\beta(y)\|_{\mathcal{T}} = \|\Diamond(\Box\bot \wedge \sim q) \vee \Box\bot\|_{\mathcal{T}} = \|\Box(\Box\bot \to q) \to \Box\bot\|_{\mathcal{T}}.$$

Since $\mathcal{I}(\mu x.\Diamond(x \wedge \sim q) \vee \Box \sim x) \equiv \tau(\mathsf{wn}(\mu y.\beta(y)))$ it follows that the formula $\Box(\Box\bot \to q) \to \Box\bot$ is a solution of Equation 3.5.

3.5 Summarizing remarks

An important theorem in the study of the expressive power of the modal μ-calculus on restricted classes of transition systems is the De Jongh-Sambin Theorem. This theorem considers the modal logic GL, from Gödel-Löb, which besides an arithmetical interpretation, have also a complete semantics given by the class of all transitive and upward well-founded systems. The theorem says that fixpoint modal equations in GL have a unique solution. From this result it follows that over the previous class of models, the μ-calculus collapses to modal logic.

In this chapter, we decided to reverse this point of view. More precisely, we proved the collapse of the modal μ-calculus over GL *without* using the De Jongh-Sambin Theorem by showing that fixpoints are reached after two iterations of well-named fixpoint formulae.

However fixpoint theorems in GL hold also for modal formulae where the variable appears guarded but not necessarily positively. Thus, from this point of view, the previous collapse is not completely satisfactory since modal μ-calculus allows fixpoint constructors only for syntactically positive formulae. We therefore have extended the collapse to an extension of the μ-calculus, called the modal $\mu^{\sim}$-calculus, where fixpoint variables are not necessarily in positive positions in the formulae. This was done by providing an explicit syntactical translation of the $\mu^{\sim}$-calculus into GL which preserves logical equivalence. As a corollary of this result, we have then obtained a new version of the De Jongh-Sambin Fixpoint Theorem with a simple algorithm which shows how the fixpoint can be computed.

Notice that Lemma 3.3 immediately leads to the fact that on finite trees the modal μ-calculus collapses to the first ambiguous class of the fixpoint hierarchy.

Chapter 4

Characterizing the Modal Fragment on Transitive Models

This chapter is based on a joint work with Balder ten Cate [40].

4.1 Preliminary remarks

Determining effective characterizations of logics on trees is an interesting problem in theoretical computer science, the main motivation being the desire to understand the expressive power of logics like first-order logic (FO), or the temporal logic CTL*, on trees.

For finite words, this problem is well-studied and understood. The first-order definable regular languages, for instance, can be characterized as the class of star-free regular languages [84] or the class of regular languages whose syntactic monoid is aperiodic [109], a condition which can be tested effectively. Since the regular languages are precisely the string languages definable in monadic second-order logic with the child relation [36], this can also be seen as an effective characterization of FO as a fragment of MSO on strings. Similar effective characterizations have been obtained for various fragments of FO and for various temporal logics such as fragments of LTL (see [131] and [105] for references).

The list of results in the case of trees is much more frugal. The situation seems to have improved a little in the last years, thanks to an effort towards understanding fragments of CTL* and the successful use of what are called *forest algebra* (see [29]). Notably, this formalism has been used for obtaining decidable characterizations for the classes of tree languages definable in $\mathsf{EF}+\mathsf{EX}$ [28], $\mathsf{EF}+\mathsf{F}^{-1}$ [20, 106], $\mathsf{BC}-\Sigma_1(<)$ [26, 106], $\Delta_2(\leq)$ [27, 106]. This approach has then been extended in the case of the temporal logic EF on infinite but finitely branching trees by Bojanczyk and Idziaszek [24].

These results all demonstrate the importance of the algebraic approach in obtaining decidable characterizations of logics. In the case of infinite trees, it is natural to ask whether such logics also admit *topological* characterizations. Take for instance the logic EF. It follows from the results of Bojanczyk and Idziaszek's

[24] that the class of finite trees, that is the class of finitely branching tree models of the formula $\mu x.\Box x$, is not EF-definable (within the class of arbitrary finitely branching trees). This is because the syntactic ω-forest algebra of this tree language does not satisfy a certain equation. If we consider arbitrarily branching trees, there is another explanation for the fact that the formula $\mu x.\Box x$ defines a non-EF-definable tree language, which involves topology: $\mu x.\Box x$ defines the class of well-founded trees, while EF formulae can only define tree languages that are Borel. This raises the question whether EF, as a fragment of MSO, can be characterized by topological means.

We give a positive answer in the case of EF. Specifically, we prove that a (not necessarily finitely branching) tree language is EF-definable if and only if it is invariant for EF-bisimilarity and Borel. Since EF is a fragment of *weak monadic-second order logic* (WMSO) and all WMSO-definable tree languages are Borel, we obtain as an immediate corollary that EF is the EF-bisimulation invariant fragment of WMSO. Moreover, these characterizations are effective: given an MSO formula, one can effectively test whether it defines an EF-definable tree language. The proofs make crucial use of the results in [24]. As a corollary of these characterizations of EF on arbitrary trees, we obtain that on transitive transition systems, modal logic is the Borel fragment of the modal μ-calculus.

To sum up formally, we prove the following theorem:

THEOREM 4.1. *Let L be any* MSO*-definable tree language. The following conditions are equivalent and decidable:*

(1) L is EF*-definable*

(2) L is WMSO*-definable and closed under* EF*-bisimulation*

(3) L is Borel and closed under EF*-bisimulation*

(4) L is closed under EF*-bisimulation, and for every L-idempotent context c, and for every forest f, $c(f)$ and $(c + cf)^\infty$ are L-equivalent.*

The fourth condition, which is essentially the condition that was used in [24] to characterize EF on finitely branching trees, involves some algebraic notions that we will introduce in Section 4.2. Note that the previous equivalences do not hold for *finitely* branching trees. This is because on finitely branching trees, well-foundedness is equivalent to finiteness, which is Borel and closed under EF-bisimulation but not EF-definable.

Finally, remark that since any EF-definable tree language is also definable in first-order logic with the descendant relation, we obtain that all conditions of Theorem 4.1 are equivalent to the condition that L is first-order definable and closed under EF-bisimulation.

In the first part of Section 4.2, we introduce the basic notions of forest and context, as well as an analogous to the well known Myhill-Nerode equivalence relation but for tree languages. Then in its second part, after discussing some properties of tree languages definable in monadic second order logic and defining the logic EF, we present a characterization of this last formalism on finitely branching trees by Bojanczyk and Idziaszek [24], which, as we point out, in fact generalizes to arbitrarily branching trees. In Section 4.3 we present a natural (prefix) topology for trees, and give some examples of Borel and non Borel definable tree languages. The proof of Theorem 4.1 constitutes Section 4.4. In the last section of this chapter we obtain as a corollary of the main Theorem 4.1 that on transitive transition systems, modal logic is the Borel fragment of the modal μ-calculus.

4.2 Preliminaries

4.2.1 The beauty of forests

Forests, contexts. A forest over a finite set Σ is a sequence of conciliatory trees over Σ. Formally, we represent a forest by a partial function $f : \mathbb{N}^+ \to \Sigma$, where $\mathbb{N}^+$ is the set of all non-empty sequences of natural numbers, such that the domain of f is closed under non-empty prefixes. Given a forest f, and $x \in \mathsf{dom}(f)$, by $f.x$ we denote the subtree of f rooted in x. If $x \in \mathbb{N} \cap \mathsf{dom}(f)$, the subtree $f.x$ is called a *rooted subtree of the forest f*. With a slight abuse of notation, we will sometimes identify trees with forests that have a single root.

A context is a forest with a hole, which is a leaf but not a root. Formally, a context over Σ is a forest over the alphabet $\Sigma \cup \{\Box\}$ where exactly one node is labeled by $\Box$, and it is a leaf but not a root.

Operations on forests and contexts. We define two types of operations on forests and contexts: a (horizontal) *concatenation* operation, denoted by "+", and a (vertical) *composition* operation, denoted by "·". In spite of the notation we use, these operations are in general not commutative.

Given a sequence of forests $(f_i : i \in \alpha)$, with $\alpha \in \omega \cup \{\omega\}$, we want to concatenate these forests. Note that each forest can contain infinitely many rooted subtrees, and the length of the sequence itself, i.e., α, can be infinite. However, for every $i \in \alpha$, the set of all rooted subtrees of f_i is countable, and hence the set of all rooted subtrees of forests in the sequence is also countable. Let $(s'_k : k \in \omega)$ be any enumeration of the set of all rooted subtree of forests in the sequence. If each forest f_i consists of finitely many trees, we may assume that the enumeration is in fact the natural enumeration induced by the order of the sequence and the order on the rooted trees of each forest. The concatenation of the forests f_i is now defined as the forest $\sum_{i\in\alpha} f_i$ where for every $k \in \omega$, $(\sum_{i\in\alpha} f_i).k = s'_k$.

We allow also to concatenate a sequence of forests with a context. Clearly the result of such a concatenation is a context.

Concerning vertical composition, we want to compose a context p with a forest, resp. a context, f, and obtain as a result a forest, resp. a context , $c(t)$ by replacing the hole node of c by f in some way. Since either the set of rooted subtrees of f can be infinite, we distinguish between two cases. If this set is finite, the composition $c(f)$ is just obtained by replacing the hole of c with the forest f. Otherwise we proceed as follows. Let $(s_i : i \in \alpha)$ be any enumeration of all the subtrees starting in a sibling of the hole node y of c, and let x be the unique parent of y. Then the composition $c(f)$ is given by substituting in c the forest $\sum_{i\in\alpha} s_i + f$ to the forest given by all the subtrees having as a root a son of x.

Note that restricted to finitely branching trees, context and forests, the operations of concatenation and composition correspond to the ones in [24].

Remark 4.2. Admittedly, the definition of concatenation of trees and forests, and therefore also of vertical composition, is not nice. This is because it is neither associative nor commutative. However, throughout this chapter, we essentially work with languages definable in monadic second order logic with the child relation but not with an order on siblings. This implies that, when needed, the reader can safely think of trees and forests as *unordered* trees and forests, and thus of the operation of concatenation as an analogous of the set-theoretic operation of union for multisets, which is both associative and commutative. More formally, in the sequel we work in fact with an algebra consisting of equivalence classes under reordering of siblings (i.e., where two trees are equivalent if they are isomorphic as relational structures in a signature without the order relation on siblings) and the introduced operations are indeed well defined and well behaved operations on equivalence classes.

Myhill-Nerode equivalence. Given a tree language L, we want to define a notion of equivalence with respect to L, analogous to the well known Myhill-Nerode equivalence relation for finite words. Intuitively, two forests, or two contexts, are L-equivalent if they "*behave the same*" with respect to L. The precise definition of this equivalence relation is a delicate matter. It seems tempting to define two forests f, f' to be equivalent if for every context c, $c(f) \in L$ if and only if $c(f')$ in L, and similarly for contexts. However, it turns out that, in the setting of infinite trees, it is more convenient to work with a slightly finer grained notion of equivalence. The definition we use, which we will now present, is essentially the one in [24] (we will comment on the precise relationship between the two later).

The crucial notion here is that of a *template*. There are two kinds of templates, *forest-templates* and *context-templates*. A forest-template for an alphabet Σ is a forest over the alphabet $\Sigma \cup \{\star\}$ in which one or more leafs (possibly infinitely many) are labeled $\star$, and no non-leaf node is labeled $\star$. Similarly, a context-template for Σ is a forest over the alphabet $\Sigma \cup \{\star\}$ in which one or more nodes (possibly infinitely many) are labeled $\star$. Intuitively, the occurrences of $\star$ in a forest-template are placeholders for forests, and the occurrences of $\star$ in a context-template are placeholders for contexts.

In what follows, we will make use of the operation of replacing a subtree with another forest. We will not give a formal, precise definition of this operation. Just note that it can be defined in a straightforward manner by using horizontal concatenation.

Thus, given a forest-template f over $\Sigma \cup \{\star\}$ and a forest g over Σ, we denote by $f[\star \mapsto g]$ the forest obtained by replacing every node labeled $\star$ in f with the forest g. Similarly, given a context-template f over $\Sigma \cup \{\star\}$ and a context c over Σ, we denote by $f[\star \mapsto c]$ the forest obtained by replacing every subtree starting at a node labeled $\star$ in f with the forest obtained by composing the context c with the (possibly empty) forest given by all the subtrees rooted at a children of the considered node labeled $\star$. Recall that we require contexts to be guarded, so that $f[\star \mapsto c]$ is indeed a well-defined forest. In the special case where f is the infinite unary tree labeled by $\star$, then for every context c, $f[\star \mapsto c]$ will be denoted also by c^∞.

We call a forest-template *guarded* if no root is labelled by $\star$. Analogously for context-templates. Let $L \subseteq T^{\mathfrak{c}}_{\Sigma}$ be a tree language over an alphabet Σ. We say that two contexts, c_1, c_2, over Σ are L-equivalent (denoted by $c_1 \equiv_L c_2$) if for every guarded context-template t over alphabet $\Sigma \cup \{\star\}$, $t[\star \mapsto c_1] \in L$ if and only if $t[\star \mapsto c_2] \in L$. Similarly, two forests f_1, f_2 are L-equivalent (denoted by $f_1 \equiv_L f_2$) if for every guarded forest-template t over alphabet $\Sigma \cup \{\star\}$, $t[\star \mapsto f_1] \in L$ if and only if $t[\star \mapsto f_2] \in L$.

Note that since we are working with *tree* languages, we can safely ignore forests $t[\star \mapsto f]$ or $t[\star \mapsto c]$ that are not trees. Moreover, note that two trees (seen as forests) can be L-equivalent while they do not agree on membership in L.

It is easy to verify that:

PROPOSITION 4.3. *For every tree language L, L-equivalence is an equivalence relation (both on contexts and on forests).*

There is a natural generalization of forest-templates and context-templates, where the holes are marked. More precisely, multi-forest-templates and multi-context-templates for an alphabet Σ are forests over an extended alphabet $\Sigma \cup \{\star_1, \dots, \star_n\}$ (where n is a natural number), satisfying the same conditions as forest-templates and context-templates. Given a multi-forest-template t over alphabet $\Sigma \cup \{\star_1, \dots, \star_n\}$ and forests $f_1, \dots, f_n$, we denote by $t[\star_1 \mapsto f_1, \dots, \star_n \mapsto f_n]$ the forest obtained by replacing every node labeled $\star_i$ in t with the forest f_i, with $i = 1, \dots, n$. Similarly for multi-context-templates.

The next substitution lemma for L-equivalence and multi-forest-templates will be very useful:

LEMMA 4.4. *Let L be any MSO-definable Σ-tree language and f any multi-forest-template over the extended alphabet $\Sigma \cup \{\star_1, \dots, \star_n\}$. Consider a pair of finite sequences of Σ-forests $(g_1, \dots, g_n)$ and $(h_1, \dots, h_n)$ such that g_i and h_i are L-equivalent, for each $i = 1, \dots, n$. Then $f[\star_1 \mapsto g_1, \dots, \star_n \mapsto g_n]$ and $f[\star_1 \mapsto h_1, \dots, \star_n \mapsto h_n]$ are also L-equivalent.*

Proof. The proof is by induction on n. Let e be any guarded forest-template over alphabet $\Sigma \cup \{\star\}$. For the base case we reason as follows. Because $g \equiv_L h$, It follows that $(e[\star \mapsto f])[\star \mapsto g] \in L$ iff $(e[\star \mapsto f])[\star \mapsto h] \in L$. Because $(e[\star \mapsto f])[\star \mapsto g] = e[\star \mapsto (f[\star \mapsto g])]$ and $(e[\star \mapsto f])[\star \mapsto h] = e[\star \mapsto (f[\star \mapsto h])]$ we obtain that $f[\star \mapsto g] \equiv_L f[\star \mapsto h]$.

For the induction step, let f be a multi-forest-template over the extended alphabet $\Sigma \cup \{\star_1, \dots, \star_{n+1}\}$ and $(g_1, \dots, g_{n+1})$ and $(h_1, \dots, h_{n+1})$ be a pair of finite sequences of forests such that g_i and h_i are L-equivalent, for each $i = 1, \dots, n+1$. Denote by $e^n_{\overline{g}}$, resp. $e^n_{\overline{h}}$, the forest-template over $\Sigma \cup \{\star_{n+1}\}$ obtained by replacing every occurrence in $e[\star \mapsto f]$ of a node labeled $\star_i$ by g_i, resp. by h_i, for $i = 1, \dots n$ and denote by $e^{n+1}_{\overline{g}}$, resp. $e^{n+1}_{\overline{h}}$, the multi-forest-template over the extended alphabet $\Sigma \cup \{\star_1, \dots, \star_n\}$ obtained by replacing every occurrence in $e[\star \mapsto f]$ of a node labeled $\star_{n+1}$ by g_{n+1}, resp. by h_{n+1}.

First remark that, since the tree language L is MSO definable and therefore closed under reordering of siblings, the following holds for every guarded forest-template e', every multi-forest template f' over $\Sigma \cup \{\star_1, \dots, \star_{n+1}\}$ and every sequence of Σ-forests $(g'_1, \dots, g'_{n+1})$:

$$\begin{array}{rc}
e'^{n}_{g'}[\star_{n+1} \mapsto g'_{n+1}] \in L & \\
& \Leftrightarrow \\
e'[\star \mapsto (f'[\star_1 \mapsto g'_1, \ldots, \star_{n+1} \mapsto g'_{n+1}])] \in L & \\
& \Leftrightarrow \\
e'^{n+1}_{g'}[\star_1 \mapsto g'_1, \ldots, \star_n \mapsto g'_n] \in L &
\end{array}$$

Thence, we have

$$\begin{array}{rcl}
e[\star \mapsto (f[\star_1 \mapsto g_1, \ldots, \star_{n+1} \mapsto g_{n+1}])] \in L & & \\
& \Leftrightarrow & \\
e^{n}_{g}[\star_{n+1} \mapsto g_{n+1}] \in L & & \\
& \Leftrightarrow & (g_{n+1} \equiv_L h_{n+1}) \\
e^{n}_{g}[\star_{n+1} \mapsto h_{n+1}] \in L & & \\
& \Leftrightarrow & \\
e^{n+1}_{h}[\star_1 \mapsto g_1, \ldots, \star_n \mapsto g_n] \in L & & \\
& \Leftrightarrow & \text{(induction hypothesis)} \\
e^{n+1}_{h}[\star_1 \mapsto h_1, \ldots, \star_n \mapsto h_n] \in L & & \\
& \Leftrightarrow & \\
e[\star \mapsto (f[\star_1 \mapsto h_1, \ldots, \star_{n+1} \mapsto h_{n+1}])] \in L & &
\end{array}$$

This shows that $f[\star_1 \mapsto g_1, \ldots, \star_{n+1} \mapsto g_{n+1}]$ and $f[\star_1 \mapsto h_1, \ldots, \star_{n+1} \mapsto h_{n+1}]$ are also L-equivalent. □

The previous proposition can analogously be proved to hold for multi-context templates, but since we do not need it, we do not prove it.

From now on, when speaking about forest, resp. context, templates we always mean *guarded* forest, resp. context, templates.

4.2.2 More on monadic second order logics

For $n \geq 1$, we denote by $\equiv_n$ denote the equivalence relation that holds between two trees seen as relational structures if they cannot be distinguished by an MSO-sentence of quantifier depth n (where both first-order and second-order quantifiers are taken to contribute to the quantifier depth of a formula, i.e., $qd(\exists X.\phi) = qd(\exists x.\phi) = qd(\phi) + 1$). It is well known that, for each $n \geq 1$, over a finite signature, there are only finitely many $\equiv_n$-classes of structures (this can be shown by a straightforward induction, using also the fact that a sentence of quantifier depth n cannot contain more than n first-order and second-order variables). It follows that every structure can be completely described up to $\equiv_n$-equivalence by a single MSO-sentence of quantifier depth n. Furthermore, the equivalence relation $\equiv_n$ can be characterized using a variant of Ehrenfeucht-Fraïssé games. These games are defined as follows. Let $\mathfrak{M}_0$ and $\mathfrak{M}_1$ be two relational structures over the same signature τ and $n \in \omega$. The *n-round Ehrenfeucht-Fraïssé game* $\mathcal{G}_n(\mathfrak{M}_0, \mathfrak{M}_1)$ is played by two players, Spoiler and the Duplicator. Each player has to make n moves in the course of a play. The players take turns. In his k-th move, Spoiler first selects a structure, $\mathfrak{M}_0$ or $\mathfrak{M}_1$, and the kind of moves he want to perform. There are two possibilities: either he chooses an element of the domain of the chosen structure, or one of its subsets. If Spoiler picks an element of the domain of $\mathfrak{M}_i$, then Duplicator chooses an element of the domain of $\mathfrak{M}_{1-i}$, and if Spoiler selects a subset of the domain of $\mathfrak{M}_i$, then Duplicator also chooses a subset of the domain of $\mathfrak{M}_{1-i}$. At the end of the play, we have obtained two sequences $(a_1, \ldots, a_k, A_1, \ldots, A_l)$ and $(b_1, \ldots, b_k, B_1, \ldots, B_l)$, where $n = k + l$, each a_i is an element of the domain of $\mathfrak{M}_0$ and each A_i is a subset of the domain of $\mathfrak{M}_0$, each b_i is an element of the domain of $\mathfrak{M}_1$ and each B_i is a subset of the domain of $\mathfrak{M}_1$. In order for Duplicator to win the game, the resulting mapping $(a_1, \ldots, a_k) \mapsto (b_1, \ldots, b_k)$ should not only be a partial isomorphism with respect to the relations in the structures, but should also preserve membership in the chosen sets $A_1, \ldots A_l$ and $B_1, \ldots B_l$. It is then possible to prove that Duplicator has a winning strategy in the n-round game $\mathcal{G}_n(\mathfrak{M}_0, \mathfrak{M}_1)$ $(n \geq 1)$ if and only if the two structures $\mathfrak{M}_0$ and $\mathfrak{M}_1$ satisfy the same MSO-formulae of quantifier depth at most n. For more details, see [52, 83].

Proposition 4.5. *If L is an* MSO*-definable tree language, then there are only finitely many L-equivalence classes of forests and contexts. Moreover, every L-equivalence class (of forests and of contexts) has a finitely branching and regular member.*

Proof. It is enough to verify that the proposition holds when considering only non empty forests. We first show that there are only finitely many L-equivalence classes (of non empty forests and of contexts) and that each is MSO-definable (to make sense of this statement, we view non empty forests and contexts also as relational structures, using an extra predicate $P_\Box$ to denote the hole of a context). Recall that for each $n \geq 1$, over a finite signature, there are only finitely many $\equiv_n$-classes of structures. Therefore, it is enough to show that n-equivalence implies L-equivalence for large enough n. We will pick n simply to be the quantifier depth of the MSO-formula that defines L. An easy Ehrenfeucht-Fraïssé game-argument shows that for all forests f_1, f_2 (viewed as relational structures), if $f_1 \equiv_n f_2$, then for every forest-template t, $t[\star \mapsto f_1] \equiv_n t[\star \mapsto f_2]$, hence $t[\star \mapsto f_1] \models \phi$ if and only if $t[\star \mapsto f_2] \models \phi$, hence, by definition, $f_1 \equiv_L f_2$. Every element of $t[\star \mapsto f_1]$ can be naturally identified either with an non-$\star$ element of t, or with a pair consisting of an $\star$-element of t and an element of f_1. Similarly for $t[\star \mapsto f_2]$. Duplicator's strategy simply copies elements of the first type, and for elements of the second type, copies the first coordinate and applies his known winning strategy to the second coordinate of the pair. A similar argument applies to contexts.

Next, note that every non-empty MSO-definable tree language contains a finitely branching and regular tree. This follows from the automata theoretic characterization of MSO given by Walukiewicz [130]. The same holds for MSO-definable classes of forests (just by considering the tree language consisting of all trees for which the forests consisting of all nodes except the root belongs to the given forest language). Since we just saw that every L-equivalence class of non empty forests is itself an MSO-definable class, we get the same for any L-equivalence class of non empty forests and, similarly, of contexts. $\Box$

Remark 4.6. It can be naturally asked what is the relation between the introduced notion of L-equivalence and the one originally introduced by Bojanczyk and Idziaszek. It follows by the previous Proposition 4.5 that if L-equivalence were defined using regular forest-templates and regular context-templates only, the result would be the same as L-equivalence the way we defined, assuming that

L is a MSO definable tree language. This means that our definition essentially coincides with the one used in [24] by the authors.

We still need a slightly more fine grained version of Proposition 4.5. Let c be a context and f a forest. By a *forest built from c and f*, we will mean a forest g for which there exists a forest s such that replacing each non-leaf node in s by a copy of c and replacing each leaf-node of s by a copy of f yields g. We then say that s is the *skeleton* of g.

PROPOSITION 4.7. *Let L be any MSO-definable tree language. Let c be a context and f a forest. Every forest g built from c and f is L-equivalent to a forest g' built from c and f, whose skeleton is regular and finitely branching. Moreover, if the skeleton of g satisfies some MSO-sentence ψ, then g' can be chosen so that its skeleton satisfies ψ as well.*

Proof. If g is the empty forest, the claim is easily verified. For the other case, we reason as follows. First, we introduce some terminology. Let g be a forest built from a context c and a forest f, whose skeleton is s. We can think of g has a pair (s, h_s), where h_s is a function assigning to each internal node of s the context c and to each leaf of s the forest f. Given a node x of the skeleton s of g, we call it a c-node or f-node, respectively, if $h_s(x) = c$ or $h_s(x) = f$.

- By a *region* of g we mean the set of points that correspond to some point of s. In other words, a region of g is some copy of f or of c (without the hole) occurring in g.

- Consider another forest g' built from c and f, whose skeleton is s'. We can think of a point move of a player in the n-round Ehrenfeucht-Fraïssé game $\mathcal{G}_n(g, g')$ as the player placing a pebble on a node of the chosen forest, and a set move as the player placing pebbles on a set of nodes of the chosen forest. So, by a *pebbling* of a tree, we mean a partial assignment of nodes or sets of nodes to sets of pebbles. This means that a configuration of the game $\mathcal{G}_n(g, g')$ after round k can be described by a pair $((g, (P_1, \dots, P_k), (g', (P'_1, \dots, P'_k))$, where, for every $i \leq k$, P_i and P'_i are either a set of nodes or a single node, and correspond to the positions of the new pebbles introduced at round i.

Given a forest f, we say that two pebblings $(P_1, \dots, P_k)$ and $(P'_1, \dots, P'_{k'})$ of f have same type if $k = k'$ and P_i is a set of nodes iff P'_i is a set of nodes. We say that two pebblings of the same type of f are $\equiv_n$-equivalent, if Duplicator wins the Ehrenfeucht-Fraïssé game of length n between f with the first pebbling and f with the second pebbling. Recall that for every type there are only finitely many equivalence classes of this equivalence relation. Similarly for contexts c.

Let n be the quantifier depth of the MSO-formula defining L, $N_f(k)$ be the finite number of all $\equiv_{(n-(k-1))}$-equivalence classes of types of pebblings of f of length $k \leq n$, and $N_c(k)$ be the finite number of all $\equiv_{(n-(k-1))}$-equivalence classes of types of pebblings of c of length $k \leq n$. Let $\xi_k = \sum_{i=1}^{k} \big((N_f(i) + N_c(i)) + 1) \cdot 3i\big)$. Assume s satisfies the formula ψ. Let ϕ_s be an MSO formula describing s completely, up to $\equiv_{\xi_n}$-equivalence. By Proposition 4.5, there is a finitely branching regular forest s' satisfying $\phi_s \wedge \psi$. In particular, $s \equiv_{\xi_n} s'$. Let g' be the forest built from c and f whose skeleton is s'. This concludes the construction of g'. In the remainder of the proof we show that $g \equiv_n g'$, and hence $g \equiv_L g'$.

Let's describe the main idea of the proof. While playing the n-round game $\mathcal{G}_n(g, g')$, Duplicator is maintaining on a scratch paper a ξ_n-round game on $\mathcal{G}_{\xi_n}(s, s')$ where she is applying her existing winning strategy σ. Every time Spoiler plays a move in $\mathcal{G}_n(g, g')$, say he plays something in g, then Duplicator takes the pebbling of g, and "projects" it onto the individual regions of g. In other words, Duplicator turns the pebbling of g into a function that assigns to each node of s a pebbling of f or of c (namely the intersection of the pebbling of g with this particular region). This means that after Spoiler's move at round k, every node x of s can be seen as being labeled by a sequence $(P_{k_1}, \dots, P_{k_j})$, and we may refer to it as a pair $(x, (P_{k_1}, \dots, P_{k_j}))$, where, whenever the sequence $(P_{k_1}, \dots, P_{k_j})$ is not empty, each P_{k_l} is either a set of nodes or a node of the region $h_s(x)$, and the index $k_l \leq k$ indicate at which round the new pebbles corresponding to P_{i_l} have been introduced over $h_s(x)$. We call the sequence $(k_1, \dots, k_j)$ the projected type of the projected pebbling of g onto the region $h_s(x)$ at round k. Thus, a player move at round k can be identified either with a pair or with a set of pairs $(x, (P_{k_1}, \dots, P_{k_j}))$ where $k_j = k$. Now, in order to determine her answer at round k, we want that Duplicator plays by standing with a certain set of rules. This rules specifies a certain class of "well-behaved" strategies for Duplicator and are as follows.

Well-behaved strategies : We distinguish between two cases depending on whether Spoiler's move in $\mathcal{G}_n(g, g')$ is a node or a set move. Assume that Spoiler chooses g at round k, the case when Spoiler chooses g' being (mutatis mutandis) the same:

(1) if Spoiler's move is a node move characterized by (x, P_k), where x is a node of s and P_k is a node of $h_s(x)$, then:

   i. Duplicator obtains the node $\sigma(x)$ of s' by playing x as a new Spoiler's move in the game $\mathcal{G}_{\xi_n}(s, s')$ on her scratch paper and then applying the winning strategy σ to this move (notice that $h_s(x) = h_{s'}(\sigma(x))$);

   ii. then she chooses a node P'_k in $h_{s'}(\sigma(x))$ and answers with the node of g' corresponding to $(\sigma(x), P'_k)$;

   iii. after playing her answer, Duplicator marks on her scratch paper the new round played in the game $\mathcal{G}_{\xi_n}(s, s')$, round k in the game $\mathcal{G}_n(g, g')$ and the relation $R_k = \{((x, P_k), (\sigma(x), P'_k)\}$. Intuitively R_k keeps trace of the relation between the new pebbling of g and the new pebbling of g' after round k.

(2) if Spoiler's move at round k is a set move characterized by $\{(x_l, P^l_k) : l \in L_k\}$, where L_k is the index set of the set and P^l_k is a set of nodes of $h_s(x_l)$, for each $l \in L_k$, then Duplicator's move is obtained as follows:

   - first the set is partitioned into two sets depending on whether a node x is a c-node or a f-node,
   - then, each set is partitioned by considering the projected type of each pair, and each set corresponding of a certain projected type is partitioned into $\equiv_{(n-(k-1))}$-equivalence classes.

Suppose that the partition is $A_{1_k}, \ldots, A_{i_k}$. Each set A_{l_k} determines a set A'_{l_k} of nodes of s. Notice that $A'_{1_k}, \ldots, A'_{i_k}$ are also pairwise disjoints. Then:

i. Duplicator determines the pairwise disjoint subsets $\sigma(A'_{1_k}), \ldots, \sigma(A'_{i_k})$ through her support game $\mathcal{G}_{\xi_n}(s, s')$ and her winning strategy σ by playing i_k new set moves.

ii. Thus, for every $l_k \leq i_k$, and every node $x' \in \sigma(A'_{l_k})$, Duplicator defines the pair (x', P'_i), where P'_i is a set of nodes of $h_{s'}(x')$. Thence, Duplicator's answer is determined by the union of all new pebblings over the obtained regions.

iii. After playing her answer, Duplicator marks on her scratch paper the new i_k rounds in the game $\mathcal{G}_{\xi_n}(s, s')$, the new round in the game $\mathcal{G}_n(g, g')$ and the new relation $R_k = \bigcup_{l_k \leq i_k} \{(x, P_k) : x \in A_{l_k}\} \times \{(x', P'_k) : n' \in \sigma(A_{l_k})\}$. In this case too, the relation R_k keeps trace of the relation between the new pebbling of g and the new pebbling of g' introduced at round k.

Suppose that Duplicator applies a strategy satisfying the previous rules. Then, every configuration after round k can be seen as a triple

$$\mathcal{C}_k = (C_1^k, C_2^k, (R_1, \ldots, R_k))$$

where:

(1) C_1^k is the actual configuration of the game $\mathcal{G}_n(g, g')$ after round k,

(2) C_2^k is the actual configuration of the game $\mathcal{G}_{\xi_n}(s, s')$ on her scratch paper after $i_1 + \cdots + i_k$ rounds (where each i_j depends on Spoiler's move in $\mathcal{G}_n(g, g')$ at round j),

(3) $(R_1, \ldots, R_k)$ is the sequence of relations written on her scratch paper.

Remark that if Duplicator can reach a configuration $\mathcal{C}_k$, she can survive for at least $\xi_{n-(k-1)}$ rounds from the actual configuration of the game $\mathcal{G}_{\xi_n}(s, s')$ written on her scratch paper. This means that for every $k < n$, if the game in $\mathcal{G}_n(g, g')$ after round k is in the configuration $\mathcal{C}_k$, then Duplicator can answer to each move of her opponent by applying a well-behaved strategy and thence reach a configuration $\mathcal{C}_{k+1}$.

Good configurations : Now, we want to show that there is a well-behaved strategy that is winning. In order to do so, we first define a notion of a *good* configuration. Intuitively a good configuration $\mathcal{C}_k$ implies that if the game on g and g' had to stop in the corresponding current configuration, then it would be winning for Duplicator, but it is stronger. More precisely, we show that in every good configuration, Duplicator has a well-behaved strategy that ensures that the configurations will always stay good. Because well-behaved strategies preserving good configurations are winning for Duplicator in the game $\mathcal{G}_n(g, g')$, we obtain the desired conclusion.

We say that a configuration $\mathcal{C}_k$ is good if:

(*) for every $j \leq k$, if $((x, P_j), (x', P'_j)) \in R_j$ then:

- the projected pebbling of $h_s(x)$ and the projected pebbling of $h_{s'}(x')$ have same projected type and are $\equiv_{(n-(j-1))}$-equivalent, and
- if $(P'_{l_1}, \ldots, P'_{l_y}, P_j)$ is the projected pebbling of $h_s(x)$, and $(P'_{l_1}, \ldots, P'_{l_y}, P_j)$ is the projected pebbling of $h_{s'}(x')$, then for every $l_z \in \{l_1, \ldots, l_y\}$, $((x, P_{l_z}), (x', P'_{l_z})) \in R_{l_z}$.

Notice that if $\mathcal{C}_k$ is good, $\mathcal{C}_j$ is good for every $j \leq k$. Now, suppose that Duplicator applies a well-behaved strategy and reaches a good configuration $\mathcal{C}_k$. Then, because the strategy σ she is applying in the support game $\mathcal{G}_{\xi_n}(s, s')$ written on her scratch paper is winning and because for every $j \leq k$ and every pair $((x, P_j), (x', P'_j)) \in R_j$ the projected pebbling of $h_s(x)$ and the projected pebbling of $h_{s'}(x')$ are $\equiv_{(n-(k-1))}$-equivalent, the mapping resulting from the pebbling of g and the pebbling of g' after round k is a partial isomorphism with respect to the relations in the structures which preserves membership in the chosen sets. Trivially, the initial configuration (the configuration before the game $\mathcal{G}_n(g, g')$ starts) is good. Thence, if we verify that from any good configuration $\mathcal{C}_k$ Duplicator can apply a well-behaved strategy in such a way that $\mathcal{C}_{k+1}$ is also good, we have shown that Duplicator has a winning strategy in $\mathcal{G}_n(g, g')$.

A well-behaved strategy preserving good configurations : Assume that Spoiler chooses g at round $k + 1 \leq n$, the case of g' being identical. We have two cases to consider, depending on whether Spoiler's move is either a node or a set move.

(1) Assume that Spoiler's move is a node move characterized by (x, P_{k+1}), where x is a node of s, P_{k+1} is a node of $h_s(x)$, and $(P_{l_1}, \ldots, P_{l_y}, P_{k+1})$ is the projected pebbling of the node $h_s(x)$ after his move. Then Duplicator obtains the new pebbling P'_{k+1} on $\sigma(x)$ as follows. If the projected pebbling of $h_s(x)$ is just (P_{k+1}), this means that $h_s(x)$ have no pebble on it before this round. Thus, Duplicator defines the pair (x', P'_{k+1}), where P'_{k+1} equals P_{k+1}. Otherwise, assume that $(P_{l_1}, \ldots, P_{l_y})$ is not empty. Because $\mathcal{C}_k$ is good and σ is a winning strategy, there is a pebbling $(P'_{l_1}, \ldots, P'_{l_y})$ of $h_{s'}(\sigma(x))$ such that:

- $((x, P_{l_z}), (x', P'_{l_z})) \in R_{l_z}$ for every $l_z \in \{l_1, \ldots, l_y\}$,
- the pebbling $(P_{l_1}, \ldots, P_{l_y})$ of $h_s(x)$ and the pebbling $(P'_{l_1}, \ldots, P'_{l_y})$ of $h_{s'}(\sigma(x))$ have same type and are $\equiv_{n-(k-1)}$-equivalent.

This means that there is a set of nodes P'_{k+1} such that the pebbling $(P_{l_1}, \ldots, P_{l_y}, P_{k+1})$ of $h_s(x)$ and the pebbling $(P'_{l_1}, \ldots, P'_{l_y}, P'_{k+1})$ of $h_{s'}(\sigma(x))$ are $\equiv_{(n-k)}$-equivalent. Thus Duplicator answers with the node $(\sigma(x), P'_{k+1})$.

(2) Suppose that Spoiler's move is characterized by $\{(x_l, P^l_{k+1}) : l \in L_{k+1}\}$, where P^l_{k+1} is a set of nodes of $h(x_l)$, for each $l \in L_{k+1}$. Let $\sigma(A'_{1_{k+1}}), \ldots, \sigma(A'_{j_{k+1}})$ be the pairwise disjoint sets obtained by Duplicator by playing j_{k+1} new set moves in her support game $\mathcal{G}_{\xi_n}(s, s')$. Thus, for every $l_{k+1} \leq j_{k+1}$, and every node $x' \in \sigma(A'_{l_{k+1}})$, Duplicator determines the set of nodes P'_{k+1} associated to $h_{s'}(x')$ as follows. Let x be any element of $A'_{l_{k+1}}$.

- If the projected pebbling of $h_s(x)$ is just (P_{k+1}), this means that $h_s(x)$ have no pebble on it before this round. Thus, Duplicator defines the pair (x', P'_{k+1}), where P'_{k+1} is a set of nodes of $h_{s'}(x')$ such that $h_{s'}(x')$ with pebbling P'_{k+1} is $\equiv_{(n-k)}$-equivalent with every member of $A_{l_{k+1}}$.
- Otherwise, assume that $(P_{l_1}, \dots, P_{l_y}, P_{k+1})$ is the projected pebbling of the node $h_s(x)$. Because $\mathcal{C}_k$ is good and σ is a winning strategy, there is a pebbling $(P'_{l_1}, \dots, P'_{l_y})$ of $h_{s'}(x')$ such that:
  - $((x, P_{l_z}), (x', P'_{l_z})) \in R_{l_z}$ for every $l_z \in \{l_1, \dots, l_y\}$,
  - the pebbling $(P_{l_1}, \dots, P_{l_y})$ of $h_s(x)$ and the pebbling $(P'_{l_1}, \dots, P'_{l_y})$ of $h_{s'}(x')$ have same type and are $\equiv_{(n-(k-1))}$-equivalent.

  This implies that there is a set of nodes P'_{k+1} such that the pebbling $(P_{l_1}, \dots, P_{l_y}, P_{k+1})$ of $h_s(x)$ and the pebbling $(P'_{l_1}, \dots, P'_{l_y}, P'_{k+1})$ of $h_{s'}(x')$ are $\equiv_{(n-k)}$-equivalent, meaning that the projected pebbling of $h_{s'}(x')$ is in the same $\equiv_{(n-k)}$-equivalence class as the projected pebbling of each member of $A'_{l_{k+1}}$.

In both cases the obtained configurations clearly satisfy condition $(*)$. $\square$

4.2.3 The logic EF and EF-bisimulation

Fix an alphabet Σ. The set of formulae of EF over Σ is defined by the grammar

$$\phi ::= a \mid \phi \wedge \phi \mid \neg\phi \mid \mathsf{EF}\phi \qquad (a \in \Sigma)$$

The semantics of EF over non empty trees is defined inductively as usual by saying that every EF-formula $a \in \Sigma$ is true in trees with root label a and that an EF formula $\mathsf{EF}\phi$ is true in trees that have a proper subtree where ϕ is true. For any EF formula ϕ, the class of trees where ϕ is true is denoted by $L(\phi)$. Given a tree language L, we say that L is EF-definable if there is an EF-formula ϕ such that $L = L(\phi)$. It is well-known that (see for example [31]):

Proposition 4.8. *Every EF-definable tree language is also* WMSO*-definable (and, in fact definable in first-order logic with the descendant relation).*

Following [24], we introduce a special bisimilarity game on forest, called the EF bisimulation game. We first define the game in the case of trees. Let t_0 and t_1 be two trees. The EF bisimulation game over t_0 and t_1 is played by two players: Bob and Anne. The game proceeds in rounds. At the beginning of each round, the state in the game is a pair of trees (t'_0, t'_1). A round is played as follows. First if the root labels a_0, a_1 of t'_0, t'_1 are different, then Bob wins the whole game. Otherwise Bob selects one of the trees t'_i, for $i = 0, 1$, and its subtree s_i. Then Anne selects a subtree s_{1-i} in the other tree t'_{1-i}. The round is finished, and a new round is played with the state updated to (s_0, s_1). If Anne can survive for infinitely many rounds in the EF bisimulation game on t_0 and t_1, then we say that the trees t_0 and t_1 are EF-bisimilar.

Note that clearly if two trees are bisimilar in the standard way, they also are EF-bisimilar. The converse need not to be true. Consider for example the tree t on the alphabet $\{a, b\}$ where the only nodes labelled by a are the nodes 0^{2k+1}, with $k > 0$, and the tree t' on the alphabet $\{a, b\}$ where the only nodes labelled by a are the nodes 0^{2k}, with $k > 0$. The two trees are EF-bisimilar but not bisimilar.

A tree language L is called *invariant, or closed, under EF-bisimulation* if it is impossible to find two trees, $t_0 \in L$ and $t_1 \notin L$ that are EF-bisimilar. From the previous remark on the interrelation between standard bisimilarity and EF bisimilarity, if a tree language is invariant under EF-bisimulation, then it is also invariant under standard bisimulation, but the converse is in general not true.

This game is so designed that all tree languages defined by an EF formula are invariant under EF-bisimulation. Formally, we have that:

PROPOSITION 4.9 ([24]). *Every EF-definable tree language is invariant under EF-bisimulation.*

The converse is not true. The typical counter-example is the language of all finite trees over a fixed finite alphabet. This language is invariant under EF-bisimulation but it cannot be defined by an EF-formula, as follows from a nice result of Bojanczyk and Idziaszek. Let us say that a context c is *L-idempotent* if the composition $c(c)$ is L-equivalent to c.

THEOREM 4.10. *A* MSO *definable tree language L can be defined by an EF formula if and only if*

1. *L is invariant under EF-bisimulation;*

2. *for every L-idempotent context c and for every forest f, $c(f) \equiv_L (c + c(f))^\infty$*

Moreover the previous two conditions are decidable.

REMARK 4.11. This characterization of the logic EF was proved by Bojanczyk and Idziaszek in [24] for the case of finitely branching trees. However, it follows from Proposition 4.5 that the result holds for arbitrarily branching trees as well. Also, strictly speaking, the result in [24] is different as it characterizes definability by EF-formulae that are Boolean combinations of formulae of the form $\mathsf{EF}\phi$. However, as explained in [24], this is not an essential restriction. More precisely, call a EF-formula ψ a *EF-forest formula* if ψ is a Boolean combinations of formulae of the form $\mathsf{EF}\phi$. Then, one can prove that every EF-formula ϕ is logically equivalent to $\bigwedge_{a \in A}(a \to \phi_a)$, where each ϕ_a is a EF-forest formula. This means that a tree language L is EF-definable iff, for every $a \in A$, the language L_a is definable by an EF-forest formula, where a tree t is said to be in L_a iff the tree obtained from t by relabeling its root with a is in L. Thus, the equivalence relation $\equiv_L$ in the previous theorem is, strictly speaking, given by the finite intersection of all $\equiv_{L_a}$. It is easily checked that all relevant properties of L-equivalence that we use in the proof of the main theorem still hold while thinking of this equivalence relation as given by the previous finite intersection.

We extend the notion of EF-bisimularity to forests by saying that two forests f_1, f_2 are EF-bisimilar if the trees obtained from f_1 and f_2 by adding a "fresh" root, are EF-bisimilar. More precisely, let f_1 and f_2 be two forests over Σ. Let t_1 and t_2 be any pair of trees over $\Sigma \cup \{a\}$, $a \notin \Sigma$, such that: (i) each forest f_i is obtained from the corresponding tree t_i by removing the root node and (ii) $t_1(\varepsilon) = t_2(\varepsilon) = a$. Then we say that f_1 and f_2 are EF-bisimilar if the trees t_1 and t_2 are EF-bisimilar. Note that for forests f_1 and f_2 consisting of a single tree t_1 and t_2 respectively, saying that f_1 and f_2 are EF-bisimilar is not the same as saying that t_1 and t_2 are EF-bisimilar.

The following lemma, relating EF-bisimilarity to L-equivalence, will come in handy later on. Call two contexts EF-bisimilar, if they are bisimilar when viewed as forests over an alphabet containing an additional label "$\square$".

LEMMA 4.12. *Let L be any EF-bisimulation-invariant tree language. Then every two EF-bisimilar forests are L-equivalent and every two EF-bisimilar contexts are L-equivalent.*

Proof. It follows from the fact that if f_1 and f_2 are EF-bisimilar, then for every (guarded) forest-template t (with a single root), $t[\star \mapsto f_1]$ is EF-bisimilar to $t[\star \mapsto f_2]$, and similarly in the case of contexts. $\square$

4.3 The complexity of conciliatory tree languages

Analogously to the space of full trees, the topology we will consider here on conciliatory trees is the *prefix topology*, where the open sets are, intuitively, those sets of trees for which membership of a tree is witnessed by a finite-depth prefix of the tree. Thus, for example the set of trees containing a node labeled a is an open set, but the complement is not.

We need first an analogous of the notion of initial tree but for the conciliatory case. For a conciliatory tree t over Σ and a natural number $n \geq 1$, the *depth-n prefix* of t, denoted by $t^{(n)}$, is the Σ-tree obtained by restricting the domain of t to the finite sequences of natural numbers of length at most $n-1$. We say that two trees t, t' are *equivalent up to depth n* if $t^{(n)} = t'^{(n)}$, i.e., for all finite sequence w of natural numbers of length at most $n-1$, (i) $w \in \mathsf{dom}(t)$ if and only if $w \in \mathsf{dom}(t')$, and (ii) $t(w) = t'(w)$ if t and t' are defined on w. We call a set X of conciliatory Σ-trees open if for each $t \in X$ there is a natural number $n \geq 1$ such that for all conciliatory trees t' over Σ, if t and t' are equivalent up to depth n then $t' \in X$. This indeed yields a topological space. When we say that a set of trees is Borel, we will mean that it is Borel with respect to this topology.

THEOREM 4.13. *Every* WMSO*-definable set of trees is Borel.*

Proof. The proof is by induction on the size of the WMSO-formula. For the induction, it is convenient to prove the following, slightly stronger, result: *for each* WMSO *formula ϕ and valuation λ for the free variables of ϕ, the set of trees that are consistent with λ and satisfy ϕ under valuation λ is Borel.*

First, observe that if λ is a valuation for a finite set of variables, then the class of trees that are consistent with λ is an open set. Indeed, if n is the depth of the longest element in the image of λ, then whether a tree is compatible with λ is determined by the depth-n prefix of the tree.

Next, the proof is by induction on the structure of ϕ. If ϕ is an atomic formula of the form $x < y$, $x = y$, or $X(x)$, then for all valuations λ, the set of trees consistent with λ and satisfying ϕ under valuation λ is either the empty set or the entire space, both of which are open sets. If ϕ is an atomic formula of the form $P_a(x)$, the truth of ϕ in a tree is determined by any prefix of the tree containing the node x, and hence, for every valuation λ, the set of trees consistent with λ and satisfying ϕ under valuation λ is an open (in fact, clopen) set.

The induction step for conjunction, disjunction, negation, first-order existential quantification, and weak second-order existential quantification, is straightforward, given that the number of elements of $\mathbb{N}^*$, as well as the number of finite subsets of $\mathbb{N}^*$, is countable. □

In the following subsection, we prove that the following is an example of a tree language that is *not* Borel. Recall that a tree is well-founded if it has no infinite branch.

THEOREM 4.14. *The set* WF *of all well-founded trees over the alphabet* $\{a\}$ *is not a Borel subset of the space* $T^{\mathsf{c}}_{\{a\}}$.

REMARK 4.15. Usually the set WF is shown to be not Borel by identifying each tree t over $\{a\}$ with its characteristic function, and therefore thinking the space $T^{\mathsf{c}}_{\{a\}}$ as a subset of the Cantor space 2^ω, see for example [70]. However it is not clear, at least for us, how to determine that this standard result of classical descriptive set theory immediately implies that the set WF, see as a subset of $T^{\mathsf{c}}_{\{a\}}$ and not of 2^ω, is not Borel. This is the reason why in the next subsection we present a proof for Theorem 4.14.

REMARK 4.16. Traditionally, the study of the topological complexity of tree language has focussed on full trees. Here, we are interested in arbitrary (conciliatory) trees. However, the two settings are not very different.

First of all, by Theorem 4.13, the set of full trees is itself a Borel set (with respect to the space of conciliatory trees). It follows that a set of full trees is Borel if and only if it is Borel within the subspace consisting of full trees only.

Secondly, we can associate in a natural way to each tree t over alphabet Σ the full tree t^s over alphabet $\Sigma \cup \{s\}$ (for some $s \notin \Sigma$) obtained by *padding* t with s, i.e., for each $w \in \mathbb{N}^*$, $t^s(w) = t(w)$ if $w \in dom(t)$, and $t^s(w) = s$ otherwise. Conversely, every full tree t over alphabet $\Sigma \cup \{s\}$ naturally gives rise to a (possibly empty) tree over the alphabet Σ, namely the subtree consisting of all nodes that are not labeled s and do not have an ancestor labeled s. We call this subtree the *undressing*^{s} of t. Now, for any tree language L over alphabet Σ, let us define L^s to be the set of full trees t over alphabet $\Sigma \cup \{s\}$ whose undressing^s belongs to L. By induction on the Borel rank, it can then be easily verified that if a set of conciliatory trees L is Borel then the set of full trees L^s is also Borel.

The next proposition, determining a sufficient condition for a function in order to be continuous, will be very useful.

PROPOSITION 4.17. *Let* Σ *and* Σ' *be two finite sets,* X *be either* T_Σ *or* T^{c}_Σ, Y *be either* $T_{\Sigma'}$ *or* $T^{\mathsf{c}}_{\Sigma'}$, *and* F *be a function from* X *into* Y. *Then, if for every depth-n prefix* $t^{(n)}$ *over* Σ *there exists a depth-n prefix* $t'^{(n)}$ *over* Σ' *such that* $F(t^{(n)} \cdot X) \subseteq t'^{(n)} \cdot Y$, F *is continuous.*

Proof. Consider a set $t'^{(n)} \cdot Y$, where $t'^{(n)}$ is a depth-n prefix over Σ'. If we show that the counterimage of $t'^{(n)} \cdot Y$ is an open subset of X we are done. Let P be the set of all depth-n prefix $t^{(n)}$ over Σ such that $F(t^{(n)} \cdot X) \subseteq t'^{(n)} \cdot Y$. Then, on the one hand because F is a function we have that $\bigcup_{t^{(n)} \in P} t^{(n)} \cdot X \subseteq F^{-1}(t'^{(n)} \cdot Y)$. On the other hand by definition of P and the property of F, we have that $\bigcup_{t^{(n)} \in P} t^{(n)} \cdot X \supseteq F^{-1}(t'^{(n)} \cdot Y)$, and therefore $\bigcup_{t^{(n)} \in P} t^{(n)} \cdot X = F^{-1}(t'^{(n)} \cdot Y)$, meaning that the counterimage of $t'^{(n)} \cdot Y$ is an open set. □

Note that the previous proposition clearly still holds if we consider X (or Y) as the topological space consisting of the set of all full, resp. conciliatory, binary trees over Σ (or Σ').

4.3.1 The set of well-founded trees is not Borel

In this subsection, we prove Theorem 4.14, i.e., we show that the set of well-founded trees over a singleton alphabet $\{a\}$, denoted by WF, is not Borel. As we explained in Remark 4.16, in order to show that a set of trees L is not Borel, it is enough to show that L^s is not Borel. Now, it is not hard to see that WF^s consists of all full trees $t : \mathbb{N}^* \to \{a, s\}$ with the property that on every branch there is at least a node labelled with s. We show that this set is not Borel.

We use the fact that, from Lemma 1.3, the set W_a of all full binary trees over $\{a, b\}$ where in every branch there are only finitely many nodes with label a is not Borel. We will construct a continuous function F from full binary trees over the alphabet $\{a, b\}$ to full trees over the alphabet $\{a, s\}$, such that $F^{-1}(\mathsf{WF}^s) = W_a$. This shows that $W_a \leq_W \mathsf{WF}^s$. Since W_a is not Borel, we can then conclude that WF^s (and hence also WF) is not Borel.

Intuitively, $F(t)$ will be a tree whose branches correspond to sequences of descendant-steps in t, possibly skipping over intermediate nodes, so that if t contains a branch in which a appears infinitely often, $F(t)$ contains an infinite branch containing entirely of a-nodes, and vice versa. In this way, we have $t \in W_a$ iff $F(t) \in \mathsf{WF}^s$. We have to take care also that F is a continuous function. It does not suffice to define $F(t)$ simply as the tree obtained from t by recursively adding, for each node w having a descendant v labeled a, an extra copy of v as a direct child of w, because this would not yield a continuous function. Instead, we employ a slightly more refined construction.

Let $<_{bf}$ be the breath-first order on $\{0, 1\}^*$, i.e., the ordering in which $\epsilon < 0 < 1 < 00 < 01 < 10 < 11 < 000 < \ldots$. Observe that this order has order type ω, i.e., $(\{0, 1\}^*, <_{bf})$ is isomorphic to $(\mathbb{N} \setminus \{0\}, <)$. Let ι be the isomorphism in question, i.e., $\iota(\epsilon) = 1, \iota(0) = 2, \iota(1) = 3, \iota(00) = 4$, etc. Similarly, for each $x \in \{0, 1\}^*$, let ι_x be the isomorphism between $(x \cdot \{0, 1\}^*, <_{bf})$ and $(\mathbb{N} \setminus \{0\}, <)$ mapping sequences starting with x bijectively onto the positive natural numbers in an order preserving way. Intuitively, for all $x, y \in \{0, 1\}^*$, $\iota_y(x)$ is the index of the node x within the binary subtree rooted by y, following the breadth-first ordering.

Now, define the function $I : \{0, 1\}^* \to \wp(\mathbb{N}^*)$ recursively on the length of all finite words over $\{0, 1\}$ as follows:

(1) $I(\varepsilon) = \{\iota(\varepsilon)^{\iota(\varepsilon)}\} = \{1\}$

(2) $I(x) = \{\iota(x)^{\iota(x)}\} \cup \{v\iota_y(x)^{\iota_y(x)} : y \text{ ancestor of } x, v \in I(y)\}$

Recall that given $x \in \mathbb{N}$, for all $n \in \mathbb{N}$, x^n, is defined as: $x^0 = \varepsilon$, $x^{n+1} = x^n x$. This means that, for each $x, y \in \{0, 1\}^*$ and $v \in \mathbb{N}^*$, $\iota(x)^{\iota(x)} = \underbrace{\iota(x) \cdots \iota(x)}_{\iota(x) \text{ times}}$ and therefore $v\iota_y(x)^{\iota_y(x)} = v \underbrace{\iota_y(x) \cdots \iota_y(x)}_{\iota_y(x) \text{ times}}$, where by definition $\iota(x), \iota_y(x) > 0$.

Finally, for any full binary $\{a, b\}$-tree t, we define $F(t)$ to be the tree $F(t) : \mathbb{N}^* \to \{a, s\}$ defined as follows:

(1) for every $x \in \{0,1\}^*$, and every $v \in I(x)$, $F(t)(v) = \begin{cases} a & \text{if } t(x) = a \\ s & \text{if } t(x) = b \end{cases}$,

(2) for every $w \notin \bigcup_{x\in\{0,1\}^*} I(x)$ which is an ancestor of an element of $\bigcup_{x\in\{0,1\}^*} I(x)$, $F(t)(w) = a$

(3) to all other nodes $u \in \mathbb{N}^*$, $F(t)(u) = s$

We verify that:

Claim 4.18. $t \in W_a$ *iff* $F(t) \in \mathsf{WF}^s$.

Proof of the Claim : First note that $\mathsf{dom}(F(t)) = \mathbb{N}^*$. If $t \notin W_a$, then there is a branch with infinitely many nodes labelled by a. Suppose those nodes, enumerated consistently with the prefix order, are $x_0, x_1, \ldots$. Every x_i is an ancestor of x_{i+1} in the full binary tree, meaning that

$$F(t)(\iota(x_1)^{\iota(x_1)}) = a$$

$$F(t)(\iota(x_1)^{\iota(x_1)}\iota_{x_1}(x_2)^{\iota_{x_1}(x_2)}) = a$$

$$F(t)(\iota(x_1)^{\iota(x_1)}\iota_{x_1}(x_2)^{\iota_{x_1}(x_2)}\iota_{x_2}(x_3)^{\iota_{x_2}(x_3)}) = a$$

$$\vdots$$

Thus there is an infinite branch in $F(t)$ where all nodes are labelled by a, and therefore $F(t) \notin \mathsf{WF}^s$. For the other direction, suppose that $F(t) \notin \mathsf{WF}^s$ and that there is a infinite branch π only labelled by a. By construction of $F(t)$, there are infinitely many nodes of π which are in $\bigcup_{x\in\{0,1\}^*} I(x)$. For every $n \in \mathbb{N}$, let $\pi(n), \pi(n+1) \in \bigcup_{x\in\{0,1\}^*} I(x)$. Therefore there are $x, y \in \{0,1\}^*$ such that $\pi(n) \in I(x)$ and $\pi(n+1) \in I(y)$, meaning that $t(x) = t(y) = a$. By construction of I, y is reachable from x, showing that $t \notin W_a$. *End of proof of claim.*

Claim 4.19. *F is a continuous function.*

Proof of claim: By construction of the function F, for every depth-n binary prefix $t^{(n)}$ over $\{a,b\}$ there exists a depth-n prefix $t'^{(n)}$ over $\{a,s\}$ such that $F(t^{(n)} \cdot X) \subseteq t'^{(n)} \cdot T_{\{a,s\}}$, where X is the space of all full binary trees over $\{a,b\}$. Thus F is continuous by Proposition 4.17. *End of proof of claim.*

Hence $W_a \leq_W \mathsf{WF}^s$ and W_a is not Borel, meaning that WF^s is not Borel either.

4.4 Characterizations of EF over infinite trees

Everything now is ready to prove the equivalence of the given characterizations of the logic EF on trees.

Theorem 4.1. Let L be any MSO-definable tree language. The following conditions are equivalent and decidable:

(1) L is EF-definable

(2) L is WMSO-definable and closed under EF-bisimulation

(3) L is Borel and closed under EF-bisimulation

(4) L is closed under EF-bisimulation, and for every L-idempotent context c, and for every forest f, $c(f)$ and $(c + cf)^\infty$ are L-equivalent.

Proof. We prove the equivalences in a round-robin fashion. The decidability follows from Theorem 4.10.

(1)⇒(2): If a tree language L is EF-definable, by Proposition 4.9 we know that L is invariant under EF-bisimulation and by Proposition 4.8 we know that it is also WMSO-definable.

(2)⇒(3): This follows from Theorem 4.13.

(3)⇒(4): The proof is by contraposition. Suppose there is an idempotent context c and a forest t such that $c(t)$ and $(c + c(t))^\infty$ are not L-equivalent. There are thence two possibilities:

(a) there is a (guarded) forest-template e such that

$$e[\star \mapsto c(t)] \in L \text{ and } e[\star \mapsto (c + c(t))^\infty] \notin L,$$

(b) there is a (guarded) forest-template e such that

$$e[\star \mapsto c(t)] \notin L \text{ and } e[\star \mapsto (c + c(t))^\infty] \in L.$$

We will show that both this two possibilities imply that L is not Borel, by giving a continuous reduction of the non-Borel set WF or its complement to L. Recall that WF is the set of all well-founded trees over an alphabet consisting of a single letter a.

For any tree t' over alphabet $\{a\}$, let $\widehat{t'}$ be the forest obtained as follows: if t' is the empty tree then $\widehat{t'}$ is ct, otherwise $\widehat{t'}$ is obtained by replacing every leaf node of t' by the forest ct, and replacing every non-leaf node by the context $(c + ct)$. Finally, let $F(t')$ be the tree $e(\widehat{t'})$, i.e., the result of replacing every $\star$ in e by $\widehat{t'}$. Because for every finite k, the first k levels of t determines the first k levels of $F(t)$, by Proposition 4.17 we know that F is continuous.

Claim 4.20.

(i) if $t' \in \mathsf{WF}$, then $\widehat{t'}$ is L-equivalent to ct

(ii) if $t' \notin \mathsf{WF}$, then $\widehat{t'}$ is L-equivalent to $(c + ct)^\infty$

This implies that L is not Borel. Indeed, suppose that case (a) holds. From the previous claim we obtain that $t' \in \mathsf{WF}$ if and only if $F(t') \in L$. This means that $F^{-1}(L) = \mathsf{WF}$, proving that $\mathsf{WF} \leq_W L$. Analogously, if case (b) holds, from the previous claim we obtain that $t' \in \mathsf{WF}^{\mathsf{C}}$ if and only if $F(t') \in L$, where WF^{C} is the complement of WF. Thus $F^{-1}(L) = \mathsf{WF}^{\mathsf{C}}$ and therefore $\mathsf{WF}^{\mathsf{C}} \leq_W L$. It remains only to prove the above claim.

We first need to introduce some terminology. Clearly, $\widehat{t'}$ is a forest that is built from the context c and the tree t. Let us denote the skeleton of $\widehat{t'}$ by s. Suppose that s is not well-founded. Let x any node in the domain of s such that the subtree $s.x$ is well-founded and there is no ancestor y of x such that

$s.y$ is also well-founded. Then we call the subforest of $\widehat{t'}$ built up from c and t whose skeleton is $s.x$ a *maximal well-founded subforest of* $\widehat{t'}$.

We first prove claim (i). Suppose that $t' \in \mathsf{WF}$. Then, clearly, s is well-founded. By Proposition 4.7, $\widehat{t'}$ is L-equivalent to a finitely branching, regular forest t'' built up from c and t whose skeleton s'' is well-founded (recall that well-foundedness is MSO-definable). Since every infinite finitely branching tree has an infinite branch, we then know that the skeleton s'' is in fact finite. Let k be the length of the longest branch of s''. It is easy to see that s'' is EF-bisimilar to a tree that has a single path whose labels (read from the root to the leaf) form the word $\mathsf{c}^{k+1}\mathsf{t}$. For convenience, we denote this tree itself by $\mathsf{c}^{k+1}\mathsf{t}$. Analogously, it is not hard to see that t'' is EF-bisimilar to the forest built up from c and t whose skeleton is $\mathsf{c}^{k+1}\mathsf{t}$, i.e., the forest $c^{k+1}t$. Since EF-bisimilarity implies L-equivalence (Lemma 4.12), this means that $\widehat{t'}$ is L-equivalent to $c^{k+1}t$. And since c is an L-idempotent context, we have that $\widehat{t'}$ is in fact L-equivalent to ct.

Next, we prove claim (ii). Suppose that $t' \notin \mathsf{WF}$. As before, clearly, s is non well-founded. By Proposition 4.7, $\widehat{t'}$ is L-equivalent to a regular (finitely-branching) forest t'' built up from c and t whose skeleton s'' is non well-founded (recall that non well-foundedness is MSO-definable). Because $\widehat{t''}$ is regular, there are only finitely many maximal well-founded subforests of $\widehat{t''}$ built up from c and t. Thus since we have already shown that every well-founded forest built from c and t is L-equivalent to ct, we can take all maximal well-founded subforests of $\widehat{t''}$ built up from c and t and replace each of these subforests by ct. Here, we use the substitution lemma (Lemma 4.4). Furthermore, using the fact that EF-bisimilarity implies L-equivalence (Lemma 4.12), if the resulting forest contains several copies of ct next to each other as siblings, they can be collapsed into one. This yields a new forest that can be viewed as a forest built up from the context $c + ct$ alone, whose skeleton has no leafs. But any such forest is EF-bisimilar, hence L-equivalent, to the forest $(c + ct)^\infty$.

(4)⇒(1): This is given by Theorem 4.10. □

4.5 Modal=Borel, on transitive models

Everything now is ready to show that on transitive transition systems, modal logic is the Borel fragment of the modal μ-calculus. Without loss of generality, from now we suppose that Prop is a finite set. Since models of EF are trees, we first have to relate transitive transition systems with trees. For a start, recall that a tree $t : \mathbb{N}^* \to \wp(\mathsf{Prop})$ can be seen as a pointed transition system $(\mathcal{T}_t, s)$, where $\mathsf{S} = \mathsf{dom}(t)$, the initial state s is just the root ϵ of t, the transition relation is the child relation, and the valuation $\lambda : \mathsf{Prop} \to \wp(\mathsf{S})$ is just given by $\lambda(p) = \{w \in \mathsf{dom}(t) : p \in t(w)\}$.

Given a pointed transition system $(\mathcal{T}, s)$, its tree unraveling is the set $T(\mathcal{T}, s)$ of trees t over $\wp(\mathsf{Prop})$ such that $(\mathcal{T}_t, \epsilon)$ is bisimilar with $(\mathcal{T}, s)$. In general, given a set P of pointed transition systems, by $T(P)$ we denote the set $\bigcup_{(\mathcal{T},s)\in P} T(\mathcal{T}, s)$.

In Subsection 2.2.2, we associated to every μ-formula φ, another μ-formula φ_{tr} such that for every transition system $\mathcal{T}$ it holds that

$$s \in \|\varphi_{tr}\|_{\mathcal{T}} \quad \text{if and only if} \quad s \in \|\varphi\|_{\mathcal{T}^{tr}},$$

where $\mathcal{T}^{tr}$ is the transitive closure of $\mathcal{T}$ (Lemma 2.4). This immediately leads to:

LEMMA 4.21. *For every μ-formula φ,*

$$\{(\mathcal{T}^{tr}, s) : (\mathcal{T}, s) \in \|\varphi_{tr}\|\} = \|\varphi\|^t$$

Thus, we say that the set of transitive models of a μ-formula φ is Borel iff the set of the tree unravelings of the models of the associated formula φ_{tr} is a Borel set. We also say that over transitive models a μ-formula is modal iff there is a modal formula $\psi \in \mathcal{L}_{\mathsf{M}}$ such that $\|\psi\|^t = \|\varphi\|^t$.

Before obtaining the desired characterization, we need a small lemma relating EF-bisimulation on trees with bisimulation on transitive models:

LEMMA 4.22. *Let φ be a μ-formula. Then the set $T(\|\varphi_{tr}\|)$ is closed under EF-bisimulation.*

Proof. We know that $\|\varphi\|^{tr}$ and $\|\varphi_{tr}\|$ are closed under bisimulation. Thus suppose that t' is EF-bisimilar to a tree $t \in T(\|\varphi_{tr}\|)$. This implies that $(\mathcal{T}_t, \epsilon) \in \|\varphi_{tr}\|$ and by Lemma 4.21 that $(\mathcal{T}_t^{tr}, \epsilon) \in \|\varphi\|^{tr}$. But the winning strategy for Anne in the EF game on t and t' is also a winning strategy for her in the bisimulation game on $(\mathcal{T}_{t'}^{tr}, \epsilon)$ and $(\mathcal{T}_t^{tr}, \epsilon)$. This means that $(\mathcal{T}_{t'}^{tr}, \epsilon)$ and $(\mathcal{T}_t^{tr}, \epsilon)$ are bisimilar and therefore that $(\mathcal{T}_{t'}^{tr}, \epsilon) \in \|\varphi\|^{tr}$. By applying Lemma 2.4 we obtain that $t' \in T(\|\varphi_{tr}\|)$. □

Finally, we have that:

COROLLARY 4.23. *For every μ-formula φ, on transitive models φ is modal iff $\|\varphi\|^t$ is Borel.*

Proof. Clearly for every EF-formula ϕ, there is a modal formula ψ such that for every tree t over $\wp(\mathsf{Prop})$, ϕ is true over t iff $(\mathcal{T}_t, \epsilon) \in \|\psi_{tr}\|$. Thus, if φ is modal, by Theorem 4.1, $T(\|\varphi_{tr}\|)$ is a Borel set. For the converse, suppose that $\|\varphi\|^t$ is Borel. By definition, this means that $T(\|\varphi_{tr}\|)$ is a Borel set. But on the one hand we have that $T(\|\varphi_{tr}\|)$ is recognizable by a μ-automaton and therefore MSO definable. On the other hand, by Lemma 4.22, this set is closed under EF-bisimulation, and therefore once more by Theorem 4.1, $T(\|\varphi_{tr}\|)$ is EF-definable, and therefore there is modal formula ψ such that $\|\psi\|^t = |\varphi\|^t$. □

4.6 Summarizing remarks

Understanding the expressive power of logics over words or trees is an important problem that can be found in many areas of Computer Science. Examples of such logics are temporal logics for system verification but also tree navigation languages for XML. If for (finite) words this type of problem is well-studied and understood, the same cannot be said for trees, the situation being even worse when considering infinite trees. A first result going in the direction of filling this gap has been obtained recently by Bojanczyk and Idziaszek [24]. In their work, they give a nice effective algebraic characterization on infinite finitely branching trees of the logic EF, a simple temporal logic which allows us to express statements such as "whenever a request is made it is eventually granted". In this chapter we have used Bojanczyk and Idziaszek result by providing several equivalent characterizations for this logic on arbitrary trees (finite and infinite, finitely and infinitely branching). More specifically we have proved that the properties of:

(1) being definable by an EF formula,

(2) being a Borel set and being closed under EF-bisimulation, and

(3) being definable by a formula of the weak monadic second order logic with the child relation and being closed under EF-bisimulation,

all coincide for regular languages of both finite and infinite trees. Because all the previous properties are proved to be equivalent to the characterization given by Bojanczyk and Idziaszek for the finitely branching case, we also get their decidability.

As an almost immediate corollary of the main result of the chapter, we have finally obtained that over transitive models, modal logic is the Borel fragment of the modal μ-calculus.

Part II

Hierarchical Questions for Tree Languages Definable Without Alternation

Chapter 5

Preliminaries

The Mostowski–Rabin (index) hierarchy, the Borel hierarchy, and the Wadge hierarchy are the most common measures of complexity of recognizable languages of infinite words or trees.

We have seen that the first one orders languages according to the nesting of positive and negative conditions checked by the recognizing automaton, and it has two main versions: weak and strong.

The classical Borel hierarchy is based on the nesting of countable unions and negations in the set theoretic definition of the language, starting from the simplest (open) sets. It drew attention of automata theorists as early as 1960s [79], and has continued to inspire research efforts ever since, mainly because of its intimate relations with the index hierarchy [61, 101, 114]. Finally, the Wadge hierarchy is the least known of the three. When restricted to Borel sets, it is an almost ultimate refinement of the Borel hierarchy. Defined by the preorder induced on languages by simple (continuous) reductions, it enables precise comparison of different models of computation by associating a certain ordinal (height) to the Wadge hierarchy restricted to the classes under comparison [49, 55, 92, 111].

Measuring hardness of recognizable languages of infinite trees is a long standing open problem. Unlike for infinite words, where the understanding is almost complete since Wagner's 1977 paper [126], for trees the only satisfyingly examined case is that of deterministic automata [90, 92, 93, 101, 102]. But the deterministic and non-deterministic case differ immensely for trees. The only results obtained for non-deterministic or alternating automata are strictness theorems for various classes [32, 33, 89, 96], and lower bounds for the heights of the hierarchies [51, 114]. To the best of our knowledge, the only nontrivial decidability results are the famous Rabin's theorem for non-deterministic tree automata, and decision procedures for low levels of the index hierarchy of alternating tree automata [129, 78, 103].

This part of the dissertation intents to change this situation, even if only very slightly for a start, by looking at the case of weakly recognizable languages, or equivalently to tree languages definable by a formula of the bi-modal μ-calculus without alternation [8, 96].

After introducing the projective hierarchy, the Borel Wadge hierarchy and some further basic notions concerning automata, we briefly discuss the Wadge hierarchy of weakly recognizable tree languages. Then in Chapter 6, we propose a subclass of weak alternating automata having all three hierarchies (weak index, Borel and Wadge) decidable and capturing a reasonable amount of non-determinism. The class we advocate, *linear game automata* (LGA), is obtained by taking linear automata (a.k.a. very weak automata), that emerged in the verification community, and restricting the alternation to the choice of a path in the input tree. Linear automata capture CTL [76], which is expressive enough for many applications. Though linear game automata are weaker, they retain most alternation related to the branching structure. Evidence for their expressivity is topological: they recognize sets of arbitrarily high finite Borel rank, and their Wadge hierarchy has the height $(\omega^\omega)^\omega$, much larger than $(\omega^\omega)^3 + 3$ for deterministic automata.

5.1 Topological hierarchies

In classical descriptive set theory, definable subsets of Polish spaces are classified according to the complexity of their definition in terms of projection, countable unions and complementation. The results of this classification are the Borel and projective hierarchies. A way to compare a pair of sets L and M is thence to compare their respective positions in those hierarchies. There exists however a more natural and refined complexity measure: the Wagde reduction, introduced in subsection 1.2.2. This order induces a hierarchy of sets that refines immensely the Borel hierarchy and, assuming a set-theoretic axiom of determinacy, also the projective hierarchy. From this point of view, to compare any pair of sets L and M is to compare their positions in the Wadge hierarchy, that is their Wadge degrees.

We have already seen that the space T_Σ of all full binary trees equipped with the initial segment topology is a Polish space. This means that we can use tools and methods from descriptive set theory in order to study the complexity of classes of recognizable tree languages. In order to do so for weakly recognizable languages, we first introduce the classical projective and Borel Wadge hierarchies[1]. Then, as the final subsection of this preliminary chapter, we discuss the Wadge hierarchy of tree languages recognized by weak alternating automata.

5.1.1 The topological complexity of regular tree languages

Given a Polish space X, recall that the class of Borel sets $\mathbf{Borel}(X)$ is obtained from the open sets of X by the set-theoretic operations of complementation and countable unions and that this class can then naturally be spread in a hierarchy of length ω_1, called the Borel hiearchy. But Borel sets are not the end of the story. Next we have the *projective sets*, which are obtained from the Borel sets by the operations of projection, or continuous image, and complementation. Indeed, although $\mathbf{Borel}(X)$ is closed by continuous *pre*-images, it is not closed under continuous images. Analogously to the case of Borel sets, the class $\mathbf{Proj}(X)$ of all projective sets of a Polish space X ramifies in a hierarchy of length ω. More precisely, we have the following. If $L \subseteq X \times Y$, the projection of L is $\pi''(L) =: \{x \in X : \exists y(x, y) \in L\}$. Note that π'' is continuous whenever X and Y are topological spaces. We call a subset of a Polish space X *analytic* if it is the projection of a Borel subset of $X \times \mathcal{N}$, where $\mathcal{N}$ is the Baire space

[1]The Borel hierarchy has already been introduced in Section 1.2.

and $X \times \mathcal{N}$ is equipped with the product topology. Thence, for every Polish space X, the class of projective sets $\mathbf{Proj}(X)$ is obtained from the analytic sets of X by the set-theoretic operations of complementation and projection. As for the Borel sets, also this class can naturally be spread in a hierarchy of length ω, called the projective hierarchy, where the class $\mathbf{\Sigma}^1_1(X)$ is constituted by all the analytic subsets of X. By a famous theorem of Suslin, it is well-known that the class of Borel subsets of a Polish space X is the class of all subsets of X which are both analytic and co-analytic.

Let $\mathbf{C}$ be a class of sets in Polish spaces. A set L is $\mathbf{C}$-hard if $M \leq_W L$ for all $M \in \mathbf{C}$, and $\mathbf{C}$-complete if it $\mathbf{C}$-hard and $L \in \mathbf{C}$.

What we know about the topological complexity of regular tree languages? First of all, they are all in the second projective ambiguous class $\mathbf{\Delta}^1_2$ (cf. Chapter 10 of [104] for a proof of this fact), while if we look just at weakly recognizable languages, they are all Borel and inhabit all the finite levels of the Borel hierarchy [94, 88, 8], and nothing more [51]. On the other hand, Niwinski and Walukiewicz [101] provided a so called "Gap Theorem", which implies that a deterministic tree language is either on the level $\mathbf{\Pi}^0_3$ of the Borel hierarchy, or it is $\mathbf{\Pi}^1_1$-complete, and hence non Borel[2]. The Gap Theorem actually gives an effective criterion for this dichotomy: a deterministic automaton recognizes a $\mathbf{\Pi}^0_3$ tree language iff its transition graph does not contain a certain forbidden pattern. Inspired by this result, Murlak [90] provided analogous forbidden patterns for the remaining five Borel classes and thus proved decidability of the Borel hierarchy for deterministic recognizable tree languages. The non-deterministic case lacks decision procedures for the Borel and projective hierarchies. That is to say we still do not have a general, effective, procedure for determining for every regular tree language its position in the Borel and projective hierarchies. As we will see in the next subsections, this is still the case if we restrict the class of regular tree languages to those which are weakly recognizable.

5.1.2 The Wadge hierarchy of Borel sets of finite rank

The *Wadge hierarchy* consists of the collection of all sets of full binary trees ordered by the Wadge reduction, and the Borel Wadge hierarchy is the restriction of the Wadge hierarchy to the Borel tree languages. In the sequel, since we focus on weakly recognizable tree languages, we are particularly interested in the Borel Wadge hierarchy.

In Chapter 1 we saw that the Wadge reduction $\leq_W$ is a pre-order. But can we say something more about this order? First of all, as for parity games, it can be asked whether when $L \subseteq T_\Sigma$ and $M \subseteq T_{\Sigma'}$ are Borel sets, the Wadge game $\mathcal{G}_W(L, M)$ is determined. Because a Wadge game is a special case of what is called a Gale-Stewart game (cf. [70]), the positive answer comes from the determinacy of Gale-Stewart games whose winning conditions are Borel, a very famous result proved by Martin [85].

Theorem 5.1 (Borel Wadge Determinacy). *$L \subseteq T_\Sigma$ and $M \subseteq T_{\Sigma'}$ be two Borel tree languages. Then $\mathcal{G}_W(L, M)$ is determined*

[2]Interestingly, since Büchi automata can recognize only $\mathbf{\Sigma}^1_1$-sets of trees, this result provide a very simple topological argument of the fact that these automata cannot recognize all regular tree languages.

This result is the cornerstone of the description of the Borel Wadge hierarchy. Indeed, the determinacy of Borel Wadge games induces the following corollaries: the $\leq_W$- antichains have length at most two, and the only incomparable tree languages are - up to Wadge equivalence - of the form L and $L^{\complement}$, for L non-self-dual. Since the Wadge reduction can be shown to be well-founded on Borel sets, this means that - up to complementation and Wadge equivalence - the $\leq_W$ provides a well-ordering on Borel sets. Formally, we have the following:

Proposition 5.2. *$L \subseteq T_\Sigma$ and $M \subseteq T_{\Sigma'}$ be two Borel tree languages. The following properties hold.*

1. *(Wadge's Lemma) Either $L \leq_W M$, or $M \leq_W L^{\complement}$.*
2. *If L and M are incomparable, then $L \equiv_W M^{\complement}$.*
3. *The $\leq_W$-antichains have length at most two.*

Proof.

1. Either $L \leq_W M$, or $L \not\leq_W M$. Suppose $L \not\leq_W M$. Then Duplicator has no winning strategy in $\mathcal{G}_W(L, M)$. By determinacy, this means that Spoiler has a winning strategy σ in this game. From this, we describe a winning strategy for Duplicator in $\mathcal{G}_W(M, L^{\complement})$ as follows. On her first move, regardless Spoiler's move, Duplicator answers by σ applied to a skip move. Then, Duplicator answers to every current position $t[n]$ of Spoiler by the move $\sigma(t[n])$. At the end of the play, the definition of σ ensures that the full tree t played by Spoiler belong to M if and only if the tree played by Duplicator by applying σ doesn't belongs to L. Hence, Duplicator wins the game $\mathcal{G}_W(M, L^{\complement})$, and $M \leq_W L^{\complement}$.
2. If $L \not\leq_W M$ and $M \not\leq_W L$, then the previous point implies that $M \leq_W L^{\complement}$ and $L \leq_W M^{\complement}$. Therefore, $M \leq_W L^{\complement}$ and $L^{\complement} \leq_W M$, and finally $L^{\complement} \equiv_W M$.
3. Let L, M and N be three tree languages such that $L \not\leq_W M$ and $M \not\leq_W N$. Then the first point implies that $M \leq_W L^{\complement}$ and the second that $N \leq_W M^{\complement}$. Therefore, $N \leq_W M^{\complement}$ and $M^{\complement} \leq_W L$, and by transitivity $N \leq_W L$.

□

Proposition 5.3 (Martin, Monk). *The strict Wadge reduction $<_W$ is well-founded on Borel tree languages.*

Propositions 5.2 and 5.3 show that - up to complementation and Wadge equivalence - the Borel Wadge hierarchy is a well-ordering. Therefore, there exists a unique ordinal, called the *height of the Borel Wadge hierarchy*, and a mapping d_W from the Borel Wadge hierarchy onto its height, called the *Wadge degree*, such that $d_W(L) < d_W(M)$ if and only if $L <_W M$ and $d_W(L) = d_W(M)$ if and only if $L \equiv_W M$ or $L \equiv_W M^{\complement}$, for every pairs of Borel tree languages L and M. The Borel Wadge hierarchy is thus a (huge) refinement of the Borel hierarchy.

Recall that we have two kinds of tree languages: the self-dual and the non self-dual ones. Another nice consequence of the Borel determinacy of Wadge games is that every self-dual set can be described by translations of strictly $\leq_W$-smaller non self-dual sets (cf. [47, 48]). This means that we can concentrate only on the non self-dual sets and recursively define the Wadge degree in such a way that sticks every self-dual set on a non self-dual set lower at just one level in the hierarchy as follows:

$$d_W(X) = \begin{cases} 1 & \text{if } X = \emptyset \text{ or } X = \emptyset^c, \\ \sup\{d_W(Y) + 1 \mid Y \text{ n.s.d. and } Y <_W X\} & \text{if } X \text{ is non-self-dual}, \\ \sup\{d_W(Y) \mid Y \text{ n.s.d. and } Y <_W X\} & \text{if } X \text{ is self-dual}. \end{cases}$$

Finally, one can show that the Borel Wadge hierarchy consists of an alternating succession of non self-dual and self-dual sets, as depicted in figure 5.1.

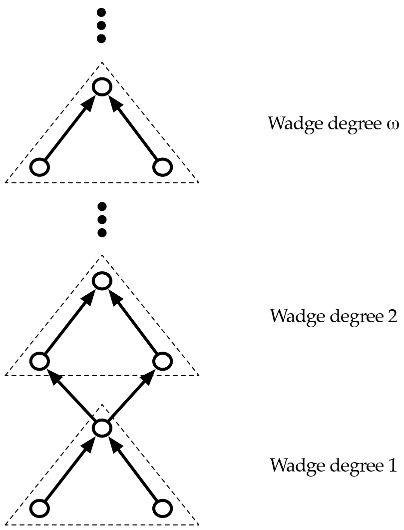


Figure 5.1: The Borel Wadge hierarchy: circles represent Wadge-equivalence classes and arrows stand for the strict Wadge reduction between those. The non self-dual sets and the self-dual ones located just one level above share the same Wadge degree.

If we consider only the class $\mathbf{\Delta}^0_\omega$ of Borel sets of finite rank, the height of the corresponding Wadge hierarchy is

$$\sup_{n\in\omega} \underbrace{\omega_1^{\omega_1^{\cdot^{\cdot^{\cdot^{\omega_1}}}}}}_{n \text{ times}} = {}^{\omega_1}\epsilon_0,$$

the least fixpoint of the ordinal exponentiation of base ω_1 (cf. [47, 48]).

Since we are interested in sets of trees which are recognizable (or equivalently definable by a μ-formula), the question now is: can we described and possibly decide the Borel Wadge hierarchy restricted to recognizable tree languages? Thanks to a very nice work of Murlak [92], we know that the answer is positive for deterministic tree languages. Unsurprisingly, the general problem is a very difficult one and still remains widely open.

5.2 Weak alternating tree automata

In this part of the dissertation we will work with weak alternating automata on full binary trees. Recall from Chapter 1 that a weak alternating parity tree automaton is an alternating parity automaton, satisfying the condition that if a state q is reachable from the state q' in the graph of the automaton, then the parity associated to q' is less or equal to the parity associated to q. For a start, we introduce some basic definitions that we also use in the next chapter. In the second and third subsections we present and discuss the three corresponding hierarchies: weak index hierarchy, Borel hierarchy and Wadge hierarchy.

5.2.1 Paths and loops

Given a weak alternating tree automaton A, a path in A is a sequence of states and transitions

$$q_0 \xrightarrow{\sigma_0,d_0} q_1 \xrightarrow{\sigma_1,d_1} q_2 \cdots q_n \xrightarrow{\sigma_n,d_n} q_{n+1}$$

If there is such a path with $q = q_0$ and $q' = q_{n+1}$, we say that q' is *reachable* from q'. We say that a path is a *loop* if $q_{n+1} = q_0$. If there is a loop from a state q, we say that this state is looping. If q is looping and $\Omega(q)$ is even (resp. odd) we say that the loop in q is positive (resp. negative). Finally, we say that a state p is *replicated* by q if there is a path $q \xrightarrow{\sigma_0,d_0} q_1 \cdots q_n \xrightarrow{\sigma_n,d_n} p$ and a transition $q \xrightarrow{\sigma_0,\bar{d_0}} q$. If q is reachable from the initial state q_I, then we say that A_q is sub-automaton of A, denoted by $A > A_q$. Clearly, the set $\{A_q : A > A_q\}$ is finite.

Given a loop $q \xrightarrow{\sigma_0,d_0} q_1 \xrightarrow{\sigma_1,d_1} q_2 \cdots q_n \xrightarrow{\sigma_n,d_n} q$, if it holds that $q_i = q$, for all $1 \leq i \leq n$, we say that the loop is trivial. An automaton where all the loops are trivial is called *linear*. An automaton without any loop is called *strict*.

Without loss of generality (except for the very special case of strict automata) we may assume that all states in the considered automata are productive save for one all-rejecting trivial-looping state $\perp$ of odd priority, that every state has an outgoing transition, and that all transitions are either productive or are of the form $q \xrightarrow{\sigma} \perp, \perp$. Sometimes it will also be useful to have an all accepting trivial-looping state $\top$ of even priority.

5.2.2 Weak index vs Borel rank

We have already seen that the index hierarchy of weak alternating tree automata, or weak index hierarchy, is strict. Concerning the topological hierarchies of weakly recognizable languages, we know since a long time that:

Proposition 5.4 ([88]). *For every weak alternating automaton with index* $(0, n)$ *(resp.* $(1, n+1)$*), it holds that* $L(A) \in \mathbf{\Pi}^0_n$ *(resp.* $L(A) \in \mathbf{\Sigma}^0_n$*).*

We can immediately ask whether the other implication also holds, and therefore whether for weakly recognizable tree languages the weak index hierarchy and the Borel hierarchy coincide level by level. The answer is still unknown, but in [93], it was conjectured that it should be positive, i.e., that:

(Weak index vs Borel rank conjecture) For every weakly recognizable tree language L,

- $L \in \boldsymbol{\Sigma}^0_n$ iff there is a weak alternating automaton A of index $(1, n+1)$ such that $L = L(A)$, and
- $L \in \boldsymbol{\Pi}^0_n$ iff there is a weak alternating automaton A of index $(0, n)$ such that $L = L(A)$.

It was recently proved that the conjecture holds when restricted to languages which are in addition deterministically recognizable [93]. But at our knowledge, since then no other step towards a solution for Murlak's conjecture has been made.

$$\begin{tikzcd}
 & \boldsymbol{\Sigma}^0_1 = [1,2] \arrow[rr] \arrow[dr] & & \boldsymbol{\Sigma}^0_2 = [1,3] \arrow[rr] \arrow[dr] & & \boldsymbol{\Sigma}^0_3 = [1,4]\cdots \\
\boldsymbol{\Delta}^0_1 = [0,1] \cap [1,2] \arrow[ur] \arrow[dr] & & \boldsymbol{\Delta}^0_2 = [0,2] \cap [1,3] \arrow[ur] \arrow[dr] & & \boldsymbol{\Delta}^0_3 = [0,3] \cap [1,4] & \cdots \\
 & \boldsymbol{\Pi}^0_1 = [0,2] \arrow[rr] \arrow[ur] & & \boldsymbol{\Pi}^0_2 = [0,1] \arrow[rr] \arrow[ur] & & \boldsymbol{\Pi}^0_3 = [0,3]\cdots
\end{tikzcd}$$

Figure 5.2: Representation of Murlak's conjecture on the correspondence between weak index and Borel rank for weakly recognizable tree languages.

5.2.3 The Wadge hierarchy of weak alternating automata

What about the Borel Wadge hierarchy restricted to weakly recognizable languages? In [51], the authors provided a lower bound for its height. It is interesting and useful to explain a little bit more about this result. In order to do so, let's come back to the Borel Wadge hierarchy. Because of a remarkable correspondence between ordinal and set theoretical operations, Duparc has shown in his PhD thesis [47] how to construct for any α strictly smaller than the least fixpoint of the ordinal operation of exponentiation of base ω_1, a Borel set $\Omega(\alpha)$ of Wadge degree exactly α whose definition is isomorphic to the Cantor normal form of base ω_1 of this very same ordinal α. This means that we are able to generate the Wadge hierarchy of Borel sets of finite rank from scratch, that is starting from the empty set. This result can be important for describing the Wadge hierarchies of some classes of word and tree automata. This is because, on the one hand, in order to obtain lower bounds for the Wadge hierarchy of the considered class of automata, it is often enough to check that some operations are definable in the class, as is done by Duparc and Murlak [51] for weak alternating automata. On the other hand, it is sometimes possible to verify that a certain pattern in the graph of the considered automaton corresponds exactly to a certain set-theoretic operation. Because of the correspondence between set-theoretic and ordinal operations, the Wadge degree of an automaton can thence be obtained by induction on its structure. This strategy is for instance the one successfully used by Murlak [92] in order to give an effective description of the Wadge hierarchy of deterministic recognizable tree languages, and it is also the one we use in the next chapter in the same aim but for the class of tree languages recognized by, what we call, linear game automata.

Thus, let us have a look at some of these set-theoretic operations[3].

We start defining four basic operations on sets of trees, that will also be useful in the next chapter. Let $L, M \subseteq T_\Sigma$, and assume that Σ contains at least two letters, a and b. Define *alternative* ($\vee$), *disjunctive product* ($\diamond$), and *conjunctive product* ($\square$) as

$$\begin{aligned} L \vee M &= \{t\colon t(\varepsilon) = a\,, t.0 \in L \text{ or } t(\varepsilon) \neq a\,, t.0 \in M\}\,, \\ L \diamond M &= \{t\colon t.0 \in L \text{ or } t.1 \in M\}\,, \\ L \square M &= \{t\colon t.0 \in L \text{ and } t.1 \in M\}\,. \end{aligned}$$

Multifold alternatives and parallel compositions are performed from left to right, e.g., $L_1 \vee L_2 \vee L_3 \vee L_4 = (((L_1 \vee L_2) \vee L_3) \vee A_4)$. It is easy to see that these three operations define associative and commutative operations on Wadge equivalence classes.

To allow easier reading, we sometimes write $L^{\langle k \rangle}$ for the set $\underbrace{L \diamond \cdots \diamond L}_{\text{k times}}$, and analogously $L^{[k]}$ for $\underbrace{L \square \ldots \square L}_{\text{k times}}$.

Another useful operation on sets if the following. Let $L, M \subseteq T_\Sigma$. We define the set $L \to M$ as the set of trees $t \in T_{\Sigma \cup \{a\}}$, with $a \notin \Sigma$, satisfying any of the following conditions:

- if $a = t(11^n)$ for all n, then $t.0 \in L$,
- if 11^n is the first node on the path 11^* not labeled by a, then $t.11^n0 \in M$.

A player in charge of $L \to M$ is like a player in charge of L endowed with an extra move, which can be used only once, that erases everything played before. Then she can restart the play being in charge of M. We say that a non-self dual set $L \subseteq T_\Sigma$ is *initializable* when $L \geq_W L \to L$.

Sum and supremum : Suppose that $L, M \subseteq T_\Sigma$. We define the set $M + L$ as $L \to M \vee M^{\complement}$. From the point of view of the player in charge of the set $M + L$ in a Wadge Game, everything goes as if she was starting the game being in charge of L. So, provided she plays in such a way that a always holds in the rightmost branch of the tree, the question whether the resulting infinite tree she will have produced at the end of the run belongs to $M + L$ or not reduces to the question whether the tree starting from the left son of the root belongs to L or not. But at any moment of the run she can play a node 11^n not labelled with a. Then, everything looks like the whole (finite) tree played since the beginning of the game is erased and he is now in charge of: M if a is the label of the node (11^n1), $M^{\complement}$ else.

The following remark ensures that the set-theoretic operation $+$ is the exact counterpart of the ordinal sum on Wadge degrees of Borel sets.

Remark 5.5 ([49, 47]). Let $L, M \subseteq T_\Sigma$ be two non self-dual Borel sets of full binary trees. Then $d_W(L + M) = d_W(L) + d_W(M)$.

[3] Duparc originally defined these operations for sets of infinite words. The operations on tree languages we present here are their straightforward translation for trees given in [51].

As for alternative, it is easy to see that sum defines associative and commutative operations on Wadge equivalence classes.

The next operation is a generalization of $\vee$ and $+$. Let $\lambda < \omega_1$, and $L_\kappa \subseteq T_{\Sigma\cup\{b\}}$ for any $\kappa < \lambda$. Fix any $1-1$ map $f : \omega \to \lambda$. Thus, define $\sup^-_{\kappa<\lambda} L_\kappa$ as the set of trees $t \in T_{\Sigma\cup\{b\}}$ satisfying the following conditions for some k:

- 0^k is the first node on 1^* labeled with b,
- $t.0^k1 \in L_{f(k)}$.

Intuitively, a player in charge of $\sup^-_{\kappa<\lambda} L_\kappa$ is given the choice between the L_κ's. The decision is determined by the number of labels different from b played on the leftmost branch of the tree before the first b. If the player keeps not playing b forever on the leftmost branch, the tree will be rejected.

Define also $\sup^+_{\kappa<\lambda} L_\kappa$ as $\sup^-_{\kappa<\lambda} L_\kappa \cup \{t : \forall_n \ t(1^n) \neq b\}$. The difference from the previous operation is that now, when the player does not plays b on the leftmost branch, the obtained tree is accepted. Note that the operations are dual:

$$\left(\overset{+}{\sup_{\kappa<\lambda}} L_\kappa\right)^{\complement} = \overset{-}{\sup_{\kappa<\lambda}} \left(L_\kappa^{\complement}\right)$$

The following fact ensures that, on Borel sets, the set-theoretic sup is the counterpart of the ordinal supremum on Wadge degrees.

Remark 5.6 ([49, 47]). Let $(L_\kappa)_{\kappa<\lambda}$ and $(M_\kappa)_{\kappa<\lambda}$ be two countable families of non self-dual Borel sets of full binary trees. Then

$$d_W(\overset{+}{\sup_{\kappa<\lambda}} L_\kappa) = d_W(\overset{-}{\sup_{\kappa<\lambda}} L_\kappa) = \sup_{\kappa<\lambda} d_W(L_\kappa).$$

Countable multiplication : With the help of ordinal sum and countable supremum we easily define the set-theoretic counterpart of the countable multiplication as an iterated sum. Let $L \subseteq T_\Sigma$. Inductively we define:

- $L \bullet 1 = L$,
- $L \bullet (\alpha + 1) = (L \bullet \alpha) + L$,
- $L \bullet \lambda = \sup^+_{\kappa\in\lambda} L \bullet \kappa$ when λ is some limit countable ordinal.

As for the previous operations, we remark that the set-theoretic countable multiplication is the exact counterpart of the ordinal operation of countable product on Wadge degrees.

Remark 5.7 ([49, 47]). Let $L \subseteq T_\Sigma$ be a non self dual Borel set. Then for every countable ordinal λ, $d_W(L \bullet \lambda) = d_W(L) \centerdot \lambda$

From the player's point of view when involved in Wadge Games, being in charge of a set of the form $L \bullet \lambda$ is like a player being in charge of L with the additional option to restart the run at any moment being in charge of its complement $L^{\complement}$ instead of L and start again and again replacing alternatively $L^{\complement}$ by L and L by $L^{\complement}$, provided that at every such changing the player decreases the ordinal λ. Therefore, during the run, this procedure will produce a decreasing finite sequence of ordinals, preventing her from initializing the game indefinitely.

The last operation, called the **action of a closed set**, is much more subtle and involved than the previous ones and will not explained here (cf. [49, 47, 51]). What is important is that this operation on Borel sets was shown to correspond (almost exactly) to the exponentiation of base ω_1: $\alpha \mapsto \omega_1^\alpha$.

By showing that the operations of sum, multiplication by ω and the action of a closed set are all definable by a weak alternating tree automaton, Duparc and Murlak [51] were thus able to infer that the Wadge hierarchy of weakly recognizable tree languages has height at least ϵ_0. It is still unknown whether the bound is tight, not to mention the problem of deciding the Wadge degree of a weakly recognizable language.

5.3 Summarizing remarks

In the next chapter we will study the index, Borel and Borel Wadge hierarchies of a subclass of weak alternating automata capturing a very weak form of alternation. In this aim, in this introductory chapter we briefly discussed those three hierarchies in the case of weakly recognizable tree languages.

We already know that the weak index hierarchy is strict and that weak alternating automata can recognize languages arbitrary high in the finite Borel hierarchy. In particular, it is known that a weak automaton of index $(0, n)$, resp. $(1, n+1)$, is in the Borel class $\mathbf{\Pi}^0_n$, resp. in $\mathbf{\Sigma}^0_n$. It is thus immediate to ask whether the other implication also holds, and therefore whether for weakly recognizable tree languages the weak index and the Borel rank coincide. The answer is still unknown, but Murlak [93], after verifying that this correspondence holds when restricted to languages which are in addition deterministically recognizable, conjecture that it should be positive. In the next chapter we show that the conjecture is true for the considered subclass of weak alternating automata, which is orthogonal to the the class of deterministic tree automata but recognize languages of arbitrary high finite Borel rank. Moreover we verify that both hierarchies, weak index and Borel, are decidable.

In Subsection 5.1.2, we introduced the Borel Wadge hierarchy. Thanks to the work of Wadge [125] and Duparc [47, 49], we know that this hierarchy can be generated from the empty set by way of a finite number of set-theoretic operations, which correspond exactly to some ordinal operations (Subsection 5.2.3). This means that for any α strictly smaller than the height of the Borel Wadge hierarchy, we can construct a Borel set $\Omega(\alpha)$ of Wadge degree exactly α whose definition is isomorphic to the Cantor normal form of base ω_1 of this very same ordinal α. This observation can be important for the study of the hierarchy when restricted to tree languages recognized by a certain class of automata. For example, by showing that some operations are definable by weak alternating tree automata, Duparc and Murlak [51] were able to infer that the Wadge hierarchy of weakly recognizable tree languages has height at least ϵ_0. In the next chapter, some properties of the correspondence between operations on sets and operations on ordinals are exploited in order to give an effective characterization of the Wadge hierarchy of the considered class of automata.

Chapter 6

Decidable Hierarchies for Linear Game Automata

This chapter is based on a joint work with Jacques Duparc and Filip Murlak [50].

6.1 Preliminary remarks

Alternating tree automata (or equivalently the modal μ-calculus), are notorious for the lack of decision procedures for classical hierarchies like the Mostowski-Rabin hierarchy (resp. the fixpoint alternation hierarchy), the Borel hierarchy, or the Wadge hierarchy. The reason for this is that when we move from infinite words to infinite trees, deterministic and non-deterministic modes of computation highly diverge. Topologically this is shown by the fact that all recognizable tree languages are in $\mathbf{\Delta}^1_2$, while deterministic recognizable tree languages are either $\mathbf{\Pi}^1_1$-complete or in the third level of the Borel hierarchy, as shown by Niwinski and Walukiewicz [101].

In this chapter we propose a novel class of automata capturing an interesting aspect of alternation and prove that all corresponding hierarchies mentioned above are decidable. The class we advocate, *linear game automata*, LGA for short, is obtained by taking linear automata and restricting the alternation to the choice of a path in the input tree. Even if linear game automata are weaker than linear automata, they preserve most alternations related to the branching structure. Evidence for the expressivity of LGA comes from their topological properties: they recognize sets of arbitrarily high finite Borel rank, and their Wadge hierarchy has the height $(\omega^\omega)^\omega$.

As we have already pointed out, these automata are very weak; so this is just a very first step on the way to the complete understanding of alternating automata, or the modal μ-calculus, on infinite trees. This notwithstanding, computing the Wadge degree of a LGA is much more involved than calculating the Wadge degree of an ω-word automaton and even a deterministic tree automaton. If the shape of these automata seems to bring down the calculation to the decomposition of nested chains, alternation (even if very weak) makes everything much harder to compute, more expressive and complicated. We also believe that the notion of game automata is well suited to take us further. Indeed, the next step is to consider weak game automata, then strong game automata, which inhabit every level of the (strong) index hierarchy and subsume deterministic languages.

The structure of the chapter is the following. In the next section we formally introduce the class of linear game automata. After providing a normal form for those automata, in Section 6.3 we show the the problem of determining the Borel rank of a LGA-recognizable language is decidable. In Section 6.4, the same problem but for the weak index is solved by an exact reduction to the Borel rank problem. Then, before the concluding remarks, we give a complete effective description of the corresponding Wadge hierarchy.

6.2 Linear game automata

A *linear game automaton* (LGA) is a linear alternating automaton where the transition relation is a (total) function $\delta : Q \times \Sigma \times \{0, 1\} \to Q$.

In the remaining of the chapter, we often write $q \xrightarrow{\sigma} q_0, q_1$ if $\delta(q, \sigma, 0) = q_0$ and $\delta(q, \sigma, 1) = q_1$. Recall that without loss of generality, we assume that:

- there is no trivial state, i.e., if $q \in Q$ is such that $A_q \equiv \top$ (resp. $A_q \equiv \bot$), then $q = \top$ (resp. $q = \bot$),
- there is no trivial transition, i.e., if $p \in Q_\forall$, and $p \xrightarrow{\sigma} q, \bot$, then $q = \bot$ (dually for $p \in Q_\exists$).

Moreover, by convention, $\top$ is a looping state of even rank, and $\bot$ is a looping state of odd rank. Recall also that if q is looping and the priority $\Omega(q)$ is even (resp. odd) we say that q is a positive (resp. negative) looping state.

LGA are closed under complementation. The usual complementation procedure, that increases the ranks by one and swaps existential and universal states turns LGA into LGA. However, LGA are not closed under union nor intersection. Given $\sigma \in \Sigma$, the language $L_\sigma = \{t \in T_\Sigma : t(0) = t(1) = \sigma\}$ is LGA-recognizable, but $L_\sigma \cup L_{\sigma'}$ is not.

Linear game automata clearly do not recognize all deterministic tree languages. However, not every tree language recognized by an LGA automaton is deterministic recognizable. An argument for this fact comes from the topological complexity of LGA-recognizable languages. Indeed, while from the Gap Theorem [101], a deterministic tree language is either on the level $\mathbf{\Pi}^0_3$ of the Borel hierarchy or it is not Borel, linear game automata can recognize tree languages of any finite Borel rank. This is because, as will be noticed in Section 6.3, some canonical LGA-recognizable languages coincide with the sets used by Skurczynski to prove the existence of weakly recognizable languages of each finite Borel rank.

6.2.1 A normal form

We now provide a useful normal form of LGA. Let L, M be tree languages over Σ containing at least two letters, a and b. The family of languages recognized by LGAs is closed under the operations of alternative ($\vee$), disjunctive product ($\diamond$), and conjunctive product ($\,\square\,$) introduced in Chapter 5. In particular, the operations have natural counterparts on automata. We write $A \vee B$ to denote the automaton recognizing $L(A) \vee L(B)$, and similarly for $\diamond$ and $\square$. Multifold alternatives are performed from left to right, e.g., $A_1 \vee A_2 \vee A_3 \vee A_4 = (((A_1 \vee A_2) \vee A_3) \vee A_4)$. It is easy to see that these three operations define associative and commutative operations on Wadge equivalence classes. To allow easier reading, we sometimes write $A^{\langle k \rangle}$ for the automaton $\underbrace{A \diamond \cdots \diamond A}_{\text{k times}}$, and analogously $A^{[k]}$ for $\underbrace{A \square \ldots \square A}_{\text{k times}}$.

Lemma 6.1. *Each* LGA *is Wadge equivalent to an* LGA *over the alphabet* $\{a, b\}$.

Proof. We proceed by induction on the number of states. Let C be an LGA. If C has only one state, the claim follows trivially. Suppose C has several states. We may assume w.l.o.g. that its initial state of C, q_0, is existential. Suppose that the transitions of C starting in q_0 are $q_0 \xrightarrow{a_i} p_i, p'_i$, $q_0 \xrightarrow{b_j} q_0, r_j$ and $q_0 \xrightarrow{c_k} q_0, q_0$ with $\Sigma = \{a_1, \ldots, a_\ell; b_1, \ldots, b_m; c_1, \ldots, c_n\}$. Then C is Wadge equivalent to

$$\begin{tikzcd}
 & \langle q_0 \rangle \arrow[loop left, "c_k{,}*"] \arrow[loop above, "b_j{,}0"] \arrow[r, "b_j{,}1"'] \arrow[dl, "a_i{,}0"] \arrow[dr, "a_i{,}1"] & C_{r_1} \vee \cdots \vee C_{r_m} \\
\bot & & (C_{p_1} \diamond C_{p'_1}) \vee \cdots \vee (C_{p_\ell} \diamond C_{p'_\ell})
\end{tikzcd}$$

By induction hypothesis, there exist automata A_i, A'_i, B_j over $\{a, b\}$, such that $A_i \equiv_W C_{p_i}$, $A'_i \equiv_W C_{p'_i}$, and $B_j \equiv_W C_{r_j}$. Let $A = (A_1 \diamond A'_1) \vee \cdots \vee (A_\ell \diamond A'_\ell)$ and $B = B_1 \vee \cdots \vee B_m$. Further, we see that if $A \vee B \equiv_W \top$, then C is Wadge equivalent to the automaton on the left below and otherwise to the one on the right:

$$\begin{tikzcd}
 & \langle q_0 \rangle \arrow[loop left, "b{,}0"] \arrow[r, "b{,}1"'] \arrow[dl, "a{,}0"] \arrow[dr, "a{,}1"'] & \bot \\
\top & & \top
\end{tikzcd}
\qquad
\begin{tikzcd}
 & \langle q_0 \rangle \arrow[loop left, "b{,}0"] \arrow[r, "b{,}1"'] \arrow[dl, "a{,}0"] \arrow[dr, "a{,}1"'] & B \vee A \\
\bot & & A
\end{tikzcd}$$

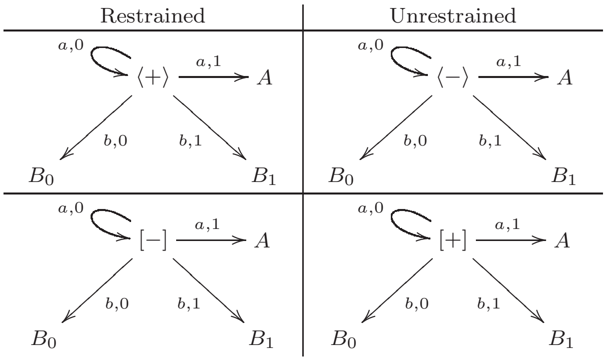
$\square$

From now on we work with automata over $\{a, b\}$, unless explicitly stated otherwise.

A looping state q of a linear game automaton A is

- *restrained* if it is an existential positive state or a universal negative state,
- *unrestrained* if it is an existential negative state or a universal positive state.

Examining the proof of Lemma 6.1, we see that in fact, each nontrivial looping state falls into exactly one of the categories shown below (+ means even rank, − means odd rank).



The term "restrained" is associate to existential positive looping states and universal negative looping states because, as we will see, the associated gadget acts like a sort of automata counterpart of the set-theoretic operation of sum. On the contrary, the term "unrestrained" is associate to existential negative looping states and universal positive looping states because the associated gadget can act like the automata counterpart of the more powerful set-theoretic operation of exponentiation of base ω_1.

A node q of each of the above kinds may be seen as an action over triples of LGAs; we denote by $q(A, B_0, B_1)$ the automaton being the result of the action q on A, B_0, B_1, e.g., $[+](A, B_0, B_1)$ or $\langle-\rangle(A, B_0, B_1)$. Often we use a shorthand $[\epsilon](A, B) = [\epsilon](A, B, \top)$, $\langle\epsilon\rangle(A, B) = \langle\epsilon\rangle(A, B, \bot)$ for $\epsilon = +$ or $\epsilon = -$.

6.3 Deciding the Borel hierarchy

In this section we show that the problem of determining the Borel rank of a tree language recognized by a linear game automaton is decidable. We approach this issue by using the technique of *difficult patterns* in the graph of an automaton. The general idea is the following. For every Borel class of finite rank we identify a certain pattern satisfying the following condition: if the pattern appears in an automaton A, then it provides a reduction of some difficult language to $L(A)$; otherwise $L(A)$ is in the considered class. This technique was used by Wagner [126] in his solution of the general problem of continuous reductions between ω-languages, and it has been successfully extended to the case of deterministic recognizable tree languages by Niwinski and Walukiewicz [101, 102], and Murlak [90].

6.3.1 Patterns menagerie

The basis for the procedure computing the Borel rank of a given LGA-recognizable language is a characterization in terms of difficult patterns. We define $(0, n)$-*pattern*, and $(1, n + 1)$-*pattern* by induction on n:

- a $(0, 1)$-pattern is a negative looping state reachable from a positive looping state,

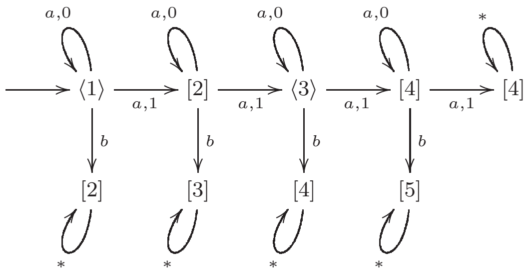
- a $(1, 2)$-pattern is a positive looping state reachable from a negative looping state,

- a $(0, n+1)$-pattern is a $(1, n+1)$-pattern replicated by a universal positive state,
- a $(1, n+2)$-pattern is a $(0, n)$-pattern replicated by an existential negative state.

We also define canonical automata, K_n^Σ and K_n^Π, corresponding to the patterns:

$$K_1^\Pi = [+](\top, \bot, \bot), \qquad K_{n+1}^\Pi = [+](K_n^\Sigma, \bot, \bot),$$
$$K_1^\Sigma = \langle - \rangle(\bot, \top, \top), \qquad K_{n+1}^\Sigma = \langle - \rangle(K_n^\Pi, \top, \top).$$

As an example, the next figure shows the structure of the canonical automaton K_4^Σ (instead of the signs "+" and "-", we write the parity associated to the states, and instead of $\bot$ and $\top$, we sketch explicitly the all-accepting and all-rejecting looping states):



The tree languages recognized by the above canonical automata coincide with the sets used by Skurczynski to prove the existence of weakly recognizable languages of each finite Borel rank.

Proposition 6.2 ([114]). *For every $n > 0$,*
$L(K_n^\Sigma)$ *is* $\mathbf{\Sigma}_n^0$*-complete and* $L(K_n^\Pi)$ *is* $\mathbf{\Pi}_n^0$*-complete.*

Skurczynski's result follows by straightforward induction from the following easy lemma. For $v \in \{0,1\}^*$ and $U \subseteq T_\Sigma$, let $vU = \{t \in T_\Sigma \colon t.v \in U\}$.

Lemma 6.3. *For each $n > 0$*

(1) if U_i is $\mathbf{\Sigma}_n^0$-hard for $i < \omega$, $\bigcap_{i \in \omega} 0^i 1 U_i$ is $\mathbf{\Pi}_{n+1}^0$-hard;

(2) if V_i is $\mathbf{\Pi}_n^0$-hard for $i < \omega$, $\bigcup_{i \in \omega} 0^i 1 V_i$ is $\mathbf{\Sigma}_{n+1}^0$-hard.

6.3.2 Effective characterization

Since any Borel class is closed under finite unions and finite intersections, we have:

Proposition 6.4. *Let K be a complete set for some Borel class of finite rank. For every k, if $U_i \leq_W K$ for $0 \leq i \leq k$, then*

(1) $\bigcup_{i=0}^k 0^i 1 U_i \leq_W K$, (2) $\bigcap_{i=0}^k 0^i 1 U_i \leq_W K$,

and if $V_i <_W K$ for $0 \leq i \leq k$, then

(3) $\bigcup_{i=0}^k 0^i 1 V_i <_W K$, (4) $\bigcap_{i=0}^k 0^i 1 V_i <_W K$.

Analogously, since $\mathbf{\Sigma}^0_n$ is closed under countable unions, and $\mathbf{\Pi}^0_n$ is closed under countable intersections, we obtain the following result.

PROPOSITION 6.5.

(1) Let K be a $\mathbf{\Sigma}^0_n$-complete set. If for every $i \in \omega$ it holds that $U_i \leq_W K$, then $\bigcup_{i\in\omega} 0^i 1 U_i \leq_W K$.

(2) Let K be a $\mathbf{\Pi}^0_n$-complete set. If for every $i \in \omega$ it holds that $U_i \leq_W K$, then $\bigcap_{i\in\omega} 0^i 1 U_i \leq_W K$.

We now apply these properties to characterize the topological power of looping nodes in an LGA.

LEMMA 6.6. *Let A, B_0, B_1, C be* LGA *such that $C = q(A, B_0, B_1)$. Then*

(1) if q is a negative looping node and $L(A), L(B_i) \leq_W L(K^\Sigma_1)$, then $L(C) \leq_W L(K^\Sigma_1)$;

(2) if q is a positive looping node and $L(A), L(B_i) \leq_W L(K^\Pi_1)$, then $L(C) \leq_W L(K^\Pi_1)$.

Proof. We prove the first claim, the second follows by duality. Assume that q is a negative looping node and $L(A), L(B_i) \leq_W L(K^\Sigma_1)$. Let us describe a winning strategy for Duplicator in $\mathcal{G}_W(L(C), L(K^\Sigma_1))$.

Suppose that $q = \langle - \rangle$. As long as Spoiler plays a on the leftmost branch, from the point of view of a player in a Wadge game, he is like a player being in charge of a countable union of open sets, and therefore Duplicator have just to apply the winning strategy given by Proposition 6.5 (1). If Spoiler finally plays a b in the kth round, he is like a player in charge of a finite union of open sets. Thus Duplicator switches to playing rejecting in every subtree rooted in $0^i 1$ for $i < k$, and in the subtree rooted in 0^k applies the winning strategy given by Proposition 6.4 (1). Hence, $L(C) \leq_W L(K^\Sigma_1)$.

Suppose that $q = [-]$. As long as Spoiler plays a on the leftmost branch, Duplicator plays rejecting in every subtree rooted in $0^i 1$. If Spoiler finally plays a b in the kth round, from the point of view of a player in a Wadge game, he is like a player in charge of a finite intersection of open sets. Thus in the subtree rooted in 0^k Duplicator applies the winning strategy given by Proposition 6.4 (2). Hence, in this case too $L(C) \leq_W L(K^\Sigma_1)$. ☐

LEMMA 6.7. *Let A, B_0, B_1, C be* LGA *such that $C = q(A, B_0, B_1)$, and q is a restrained looping node. For $n \geq 2$*

(1) if $L(A), L(B_i) <_W L(K^\Sigma_n)$, then $L(C) <_W L(K^\Sigma_n)$;

(2) if $L(A), L(B_i) <_W L(K^\Pi_n)$, then $L(C) <_W L(K^\Pi_n)$.

Proof. As for the previous lemma, it is enough to prove the first claim, the second follows by duality. Suppose that $q = \langle + \rangle$. We describe a winning strategy for Duplicator in $\mathcal{G}_W(L(C), L(K^\Sigma_n))$. If Spoiler plays a on the leftmost branch, Duplicator plays accepting in the subtrees rooted in nodes $0^i 1$. If Spoiler finally plays a b in the kth round, Duplicator switches to playing rejecting in every subtree rooted in $0^i 1$ for $i < k$, and in the subtree rooted in 0^k applies the winning strategy given by Proposition 6.4 (1). Hence, $L(C) \leq_W L(K^\Sigma_n)$.

To obtain strictness of the inequality, we describe a winning strategy for Spoiler in $\mathcal{G}_W(L(K_n^\Sigma), L(C))$. As long as Duplicator skips or plays a on the leftmost branch, Spoiler plays rejecting in the subtrees rooted in 0^i1. If in the kth round Duplicator finally plays b on the leftmost branch, Spoiler continues playing rejecting in every subtree rooted in 0^i1 for $i \leq n$, and in the subtree rooted in 0^{n+1} applies the winning strategy given by Proposition 6.4 (3).

For $q = [-]$ the proof is analogous, only uses Proposition 6.4 (2) and (4). $\square$

Lemma 6.8. *Let A, B_0, B_1, C be* LGA *such that $C = q(A, B_0, B_1)$, and q is an unrestrained looping node. Let $n \geq 2$. If $q = \langle - \rangle$, then*

1. *if $L(A) \leq_W L(K_{n-1}^\Sigma)$, and $L(B_i) <_W L(K_n^\Sigma)$, then $L(C) <_W L(K_n^\Sigma)$;*
2. *if $L(A), L(B_i) \leq_W L(K_{n-1}^\Sigma)$, then $L(C) \leq_W L(K_{n-1}^\Sigma)$;*
3. *if $L(A) \geq_W L(K_{n-1}^\Pi)$, then $L(C) \geq_W L(K_n^\Sigma)$;*

and if $q = [+]$, then

4. *if $L(A) \leq_W L(K_{n-1}^\Pi)$, and $L(B_i) <_W L(K_n^\Pi)$, then $L(C) <_W L(K_n^\Pi)$;*
5. *if $L(A), L(B_i) \leq_W L(K_{n-1}^\Pi)$, then $L(C) \leq_W L(K_{n-1}^\Pi)$;*
6. *if $L(A) \geq_W L(K_{n-1}^\Sigma)$, then $L(C) \geq_W L(K_n^\Pi)$.*

Proof. Use an argument similar to the proof of Lemma 6.7 to infer (1) and (4) from Proposition 6.5, and (3) and (6) from Lemma 6.3, and use an argument similar to the proof of Lemma 6.6 to infer (2) and (5) from Propositions 6.4 and 6.5. $\square$

The main theorem of this part follows from Lemmas 6.6, 6.7 and 6.8 by induction on the structure of the automaton. And as a corollary, we obtain the first decidability result.

Theorem 6.9. *For every $n \geq 1$ and every linear game automaton C*

(1) $L(C)$ is $\mathbf{\Pi}_n^0$-hard iff C contains a $(0, n)$-pattern.

(2) $L(C)$ is $\mathbf{\Sigma}_n^0$-hard iff C contains a $(1, n+1)$-pattern;

Proof. Recall that we always assume that all LGA have no trivial states and no trivial transitions. Moreover we assume that C is over the alphabet $\{a, b\}$ and it is in the normal form given by the proof of Lemma 6.1. The reasoning for the general case is analogous. We proceed by induction on the structure of the automaton C. If C has only one state, both claims follow trivially. Suppose that C has more than one state. Then, observe that if C does not contain neither a $(0, 1)$-pattern nor $(1, 2)$-pattern, $L(C) \in \mathbf{\Delta_1^0}$. Hence, we may assume that C has either a $(0, 1)$ or $(1, 2)$-pattern. If the initial state of C is not looping, the claims follow easily by applying the induction hypothesis. Let $C = q(A, B_0, B_1)$. We first verify Point (1).

⇒: Suppose that $L(C)$ is $\mathbf{\Pi}_n^0$-hard. Consider first the case when $n = 1$. We reason towards a contradiction. So let assume that C does not contain any $(0, 1)$-pattern but that it contains a $(1, 2)$-pattern, meaning that the parity of q is odd. By induction hypothesis, $L(A)$, $L(B_1)$ and $L(B_2)$ can at most be $\mathbf{\Sigma}^0_1$-complete. But by Lemma 6.6 (1), this implies that $L(C)$ cannot be $\mathbf{\Pi}^0_1$-hard, a contradiction. Therefore C contains a $(0,1)$-pattern. Assume now that $n > 1$. If q is a restrained looping node, then by Lemma 6.7 (2), either A or B_1 or B_2 is $\mathbf{\Pi}^0_n$-hard. By induction hypothesis C contains a $(0,n)$-pattern. If q is an unrestrained node we have two cases to consider: (i) q is an universal positive node, (ii) q is an existential negative node. In case (i), by Lemma 6.8 (4) either $L(A) >_W L(K^\Pi_{n-1})$ or $L(B_i)$ is $\mathbf{\Pi}^0_n$-hard, with $i \in \{0,1\}$. If either $L(B_0)$ or $L(B_1)$ is $\mathbf{\Pi}^0_n$-hard, by applying the induction hypothesis we have that C contains a $(0,n)$-pattern. Suppose that neither $L(B_0)$ nor $L(B_1)$ is $\mathbf{\Pi}^0_n$-hard, but $L(A) >_W L(K^\Pi_{n-1})$. Because by Proposition 6.2 $L(K^\Pi_{n-1})$ is $\mathbf{\Pi}^0_{n-1}$-complete, this means that $L(A)$ is $\mathbf{\Sigma}^0_{n-1}$-hard. By induction hypothesis A contains a $(1,n)$-pattern, and therefore C contains a $(0,n)$-pattern. For case (ii) we reason as follows. If either A or B_0 or B_1 contains a $(0,n)$-pattern, then C contains a $(0,n)$-pattern too. On the contrary, assume that neither A nor B_0 nor B_1 contain a $(0,n)$-pattern. By induction hypothesis neither $L(A)$ nor $L(B_0)$ nor $L(B_1)$ are $\mathbf{\Pi}^0_n$-hard. But this implies that $L(A)$, $L(B_0)$ and $L(B_1)$ can at most be $\mathbf{\Sigma}^0_n$-complete. Since by assumption $L(C)$ is $\mathbf{\Pi}^0_n$-hard, by Lemma 6.8 (2) we obtain a contradiction. Therefore either A or B_0 or B_1 must contain a $(0,n)$-pattern.

$\Leftarrow$: For the other direction, assume that C contains a $(0,n)$-pattern. Suppose first that either A or B_0 or B_1 contains a $(0,n)$-pattern. Then by induction hypothesis, either $L(A)$ or $L(B_0)$ or $L(B_1)$ is $\mathbf{\Pi}^0_n$-hard, and so is $L(C)$. Suppose that neither A nor B_0 nor B_1 contain a $(0,n)$-pattern. Assume that $n = 1$. Given the assumptions, q must be a positive looping node, otherwise either A or B_0 or B_1 would contain a $(0,1)$-pattern. Thence, one of the following possibilities holds:

- either: q is a positive universal node, $L(B_0 \,\square\, B_1) \geq_W \emptyset$ and $L(A) \geq_W \emptyset^{\complement}$,
- or: q is a positive universal node, $B_0 \,\square\, B_1 = \emptyset^{\complement}$ and $L(A) \geq_W \emptyset \vee \emptyset^{\complement}$,
- or: q is a positive existential node, and $L(B_0 \diamond B_1), L(A) \geq_W \emptyset$.

In each case clearly $L(C) \geq_W L(K^\Pi_1)$ and therefore by Proposition 6.2 $L(C)$ is $\mathbf{\Pi}^0_1$-hard. Assume now that $n > 1$. If q is a restrained looping node or an unrestrained existential negative looping node, then either A or B_0 or B_1 must contain a $(0,n)$-pattern, a contradiction. Thus, q is an unrestrained universal positive looping node. This means that either B_0 or B_1 contains a $(0,n)$-pattern, or A contains a $(1,n)$-pattern. In the first case, by induction hypothesis either $L(B_0)$ or $L(B_1)$ is $\mathbf{\Pi}^0_n$-hard, and therefore $L(C)$ is also $\mathbf{\Pi}^0_n$-hard. In the second case, by induction hypothesis $L(A)$ is $\mathbf{\Sigma}^0_{n-1}$-hard, and by applying Lemma 6.8 (6) we obtain that $L(C)$ is $\mathbf{\Pi}^0_n$-hard.

The proof for Point (2) is analogous. □

Corollary 6.10. *The problem of calculating the exact position in the Borel hierarchy of a language recognized by a linear game tree automaton is decidable (in polynomial time if the productive states are given).*

6.4 Weak index of LGA-recognizable sets

In [93] it was conjectured that for weakly recognizable tree languages the weak index hierarchy and the Borel hierarchy coincide, i.e., that a weakly recognizable tree language is in $\mathbf{\Sigma}^0_n$ (resp. $\mathbf{\Pi}^0_n$) iff it can be recognized by a weak alternating automaton of index $(1, n+1)$ (resp. $(0, n)$). It has long been known that one implication holds [88], that is if a weak alternating automaton A have index $(0, n)$ (resp. $(1, n+1)$), then $L(A) \in \mathbf{\Pi}^0_n$ (resp. $L(A) \in \mathbf{\Sigma}^0_n$). It was also proved recently that the conjecture holds when restricted to languages which are in addition deterministically recognizable [93]. We refine this result by showing that the conjecture also holds for languages recognizable by LGA.

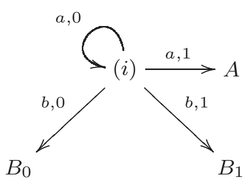
Theorem 6.11. *For languages recognizable by* LGA*, the Borel hierarchy and the weak index hierarchy coincide.*

Proof. For simplicity we assume that all automata are in the normal form given by the proof of Lemma 6.1. Extending the proof to the general case is easy.

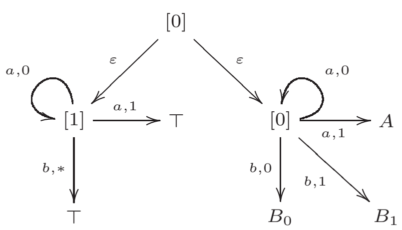
By duality it is enough to consider $\mathbf{\Pi}^0_n$ classes. By Proposition 5.4 it is suffices to show that for each LGA C with $L(C) \in \mathbf{\Pi}^0_n$ there exists an equivalent weak alternating automaton of index $(0, n)$. We proceed by induction on the structure of the automaton.

The case $n = 0$ is trivial. Suppose that $n = 1$. By Theorem 6.9, C does not contain an accepting loop reachable from a rejecting loop. It is enough to set the rank of all states reachable from odd looping states to 1 and the rank of the remaining states to 0 to obtain an equivalent automaton of index $(0, 1)$.

Suppose that $n \geq 2$. If the initial state of C is not looping, the claim follows easily from the induction hypothesis. Suppose that q_0 is a looping node, and C is of the form

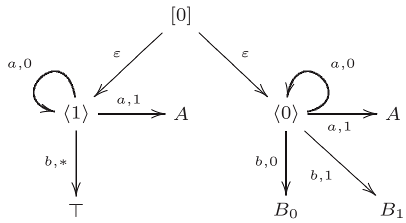


We can treat C as a weak alternating automaton and transform it into an equivalent one of index $(0, n)$. Clearly, it must hold that $L(A), L(B) \in \mathbf{\Pi}^0_n$ and by induction hypothesis we may assume that A, B_0, B_1 have index $(0, n)$. If $i = 0$, the claim follows trivially. For $(i) = [1]$, the equivalent weak alternating automaton of index $(0, n)$ is shown below.



To prove the equivalence, observe that the left-hand component checks that finally b occurs on the leftmost branch, and the right-hand component checks the condition A until the first b occurs, and after that checks the conditions B_0 and B_1.

Finally, suppose that $(i) = \langle 1 \rangle$. By Theorem 6.9, C contains $(1, n+1)$-pattern, which implies that A contains no $(0, n-1)$-pattern. By induction hypothesis we may assume that A has index $(1, n)$. Recall that B_0 and B_1 have index $(0, n)$. The corresponding equivalent weak alternating automaton is shown below.



The left-hand component takes care of the situation, when b never occurs on the leftmost path. If b does occur, this component is trivially accepting, but the right-hand component provides the appropriate semantics. □

Combining Theorem 6.9 and Theorem 6.11 we obtain the second decidability result.

Corollary 6.12. *The problem of calculating the exact position in the weak index hierarchy of a language recognized by a* LGA *is decidable (in polynomial time if the productive states are given).*

6.5 The Wadge hierarchy of linear game automata

In this section we provide a complete description of the Borel Wadge hierarchy restricted to LGA recognizable tree languages. In particular we show that the problem of determining the Wadge degree of a tree language recognized by a linear game automaton is decidable. In this aim, we first introduce the classical finite Hausdorff-Kuratowski, or difference, hierarchy and state some of its basic properties. This hierarchy is important because the whole Wadge hierarchy of linear game automata is, in some sense, built upon it. Then, in Subsection 6.5.2 we provide a family of canonical LGA recognizable tree languages. In Subsection 6.5.3 we verify that those languages are complete for every level of a certain normal form, thus obtaining a lower bound for the corresponding Wadge hierarchy. That the bound is tight is also proved in Subsection 6.5.3. This is done by showing that the previous degrees are closed under all the ordinal operations corresponding to an action of a node. By relying on this fact, we finally verify that the Wadge degree of any LGA-recognizable tree language can be computed.

6.5.1 The difference hierarchy

For a Borel class $\mathbf{\Sigma}^0_n$, the finite Hausdorff-Kuratowski, or difference, hierarchy is defined as $\mathrm{D}_1(\mathbf{\Sigma}^0_n) = \mathbf{\Sigma}^0_n$ and $\mathrm{D}_k(\mathbf{\Sigma}^0_n) = \{U \setminus V : U \in \mathbf{\Sigma}^0_n\,,\ V \in \mathrm{D}_{k-1}(\mathbf{\Sigma}^0_n)\}$.

Let $\overline{\mathrm{D}_k(\mathbf{\Sigma}^0_n)}$ denote the dual class. Recall that this is not the same as $\mathrm{D}_k(\mathbf{\Pi}^0_n)$. Indeed, $\mathrm{D}_{2k+1}(\mathbf{\Pi}^0_n) = \overline{\mathrm{D}_{2k+1}(\mathbf{\Sigma}^0_n)}$ and $\mathrm{D}_{2k}(\mathbf{\Pi}^0_n) = \mathrm{D}_{2k}(\mathbf{\Sigma}^0_n)$. We have

$$\begin{aligned}
\mathrm{D}_{2k}(\mathbf{\Sigma}^0_n) &= \{U_1 \cap V_1^{\complement} \cup \cdots \cup U_k \cap V_k^{\complement}\}\,, \\
\mathrm{D}_{2k+1}(\mathbf{\Sigma}^0_n) &= \{U_1 \cap V_1^{\complement} \cup \cdots \cup U_k \cap V_k^{\complement} \cup U\}\,, \\
\overline{\mathrm{D}_{2k}(\mathbf{\Sigma}^0_n)} &= \{U_1 \cap V_1^{\complement} \cup \cdots \cup U_{k-1} \cap V_{k-1}^{\complement} \cup U \cup V^{\complement}\}\,, \\
\overline{\mathrm{D}_{2k+1}(\mathbf{\Sigma}^0_n)} &= \{U_1 \cap V_1^{\complement} \cup \cdots \cup U_k \cap V_k^{\complement} \cup V^{\complement}\}\,,
\end{aligned}$$

where the sets U, V, U_i, V_i range over $\mathbf{\Sigma}^0_n$. From this characterization one easily obtains the following table of the operation $\diamond$. For $n > 0$ let $S_n(k)$ be a $\mathrm{D}_k(\mathbf{\Sigma}^0_n)$-complete set, and let $P_n(k)$ be a $\overline{\mathrm{D}_k(\mathbf{\Sigma}^0_n)}$-complete set.

LEMMA 6.13. *For each $n > 0$, $i > 0$, $j \geq 0$*

- $S_n(2i) \diamond S_n(2j) \equiv_W S_n(2i + 2j)$
- $S_n(2i + 1) \diamond S_n(2j) \equiv_W S_n(2i + 2j + 1)$
- $S_n(2i) \diamond P_n(2j) \equiv_W P_n(2i + 2j)$
- $S_n(2i + 1) \diamond P_n(2j) \equiv_W P_n(2i + 2j)$
- $P_n(2i) \diamond S_n(2j) \equiv_W S_n(2i + 2j)$
- $P_n(2i + 1) \diamond S_n(2j) \equiv_W P_n(2i + 2j + 1)$
- $P_n(2i) \diamond P_n(2j) \equiv_W P_n(2i + 2j - 2)$
- $P_n(2i + 1) \diamond P_n(2j) \equiv_W P_n(2i + 2j)$
- $S_n(2i + 1) \diamond S_n(2j + 1) \equiv_W S_n(2i + 2j + 1)$
- $S_n(2i + 1) \diamond P_n(2j + 1) \equiv_W P_n(2i + 2j + 2)$
- $P_n(2i + 1) \diamond S_n(2j + 1) \equiv_W P_n(2i + 2j + 2)$
- $P_n(2i + 1) \diamond P_n(2j + 1) \equiv_W P_n(2i + 2j + 1)$

The equivalences above, together with closure by $\diamond$, immediately provide complete LGA-recognizable languages for $\mathrm{D}_k(\mathbf{\Sigma}^0_n)$ for each k, n. Building upon this we produce the whole Wadge hierarchy of LGA-recognizable languages.

6.5.2 Bestiarum vocabulum

For an ordinal α let $\exp(\alpha) = \omega_1^\alpha$. Hence,

$$\exp^{k+1}(\alpha) = \exp(\exp^k(\alpha)) = \underbrace{\omega_1^{\omega_1^{\cdot^{\cdot^{\cdot^{\omega_1^\alpha}}}}}}_{k+1 \text{ times } \omega_1}\,.$$

To ease notation, we sometimes assume that $\exp^0(\alpha) = 1$.

Before describing the hierarchy, we recall the Wadge degrees of $\mathrm{D}_k(\mathbf{\Sigma}^0_n)$-complete sets.

Proposition 6.14 ([48]). *For each $k > 0$, $d_W(S_n(k)) = d_W(P_n(k)) = \exp^n(k)$.*

The aim of the remaining part of this chapter is to provide an effective characterization of the Wadge hierarchy for linear game automata (LGA Wadge hierarchy for short). In particular we want to prove that linear game automata recognize only languages of levels $\beta = \sum_{i=n}^{0} \beta_i$, where each β_i is of the form

$$\beta_i = \exp^i(\omega)\eta_i + \sum_{p=j_i}^{1} \exp^i(p)k_p$$

with $\eta_i < \omega^\omega$, $k_{2q} \in \{0,1\}$, and $j_i, k_{2q+1} < \omega$. In order to verify this statement we therefore have to check two points:

(1) the class of ordinals β is closed under the ordinal counterpart of the action of a state (upper bound),

(2) every level of degree β is inhabited by an LGA recognizable language (lower bound).

As a start, we provide for every such ordinal β an automaton A_β, such that $L(A_\beta)$ is non self dual. In Theorem 6.18, we will verify that $d_W(L(A_\beta)) = \beta$.

To make the notation more readable, we use bracketed ordinal $[\beta]$ to denote the automaton A_β. Since LGA are closed under complementation, when we construct an automaton recognizing a non self dual set of degree β, we also immediately get the automaton $[\beta]^{\mathsf{C}}$. We write $[\beta]^{\pm}$ for $[\beta] \vee [\beta]^{\mathsf{C}}$.

Let us start with the basic building bricks of our construction: the automata $[1]$, $[\omega^m]$, $[\exp^i(1)]$, and $[\exp^i(\omega)\omega^p]$. Together with these automata we show how to make a step with those ordinals, i.e., how to define the automaton for $[\alpha+\gamma]$, once we already have the automaton $[\alpha]$ and γ is one of the above. Let

$$[1] = \perp\,, \qquad [\alpha+1] = \langle - \rangle(\perp, [\alpha]^{\pm})\,.$$

Note that $[2] = K_1^{\Sigma}$, and $[2]^{\mathsf{C}} = K_1^{\Pi}$. For $m > 1$ let

$$\begin{aligned} [\omega] &= [-]([3]^{\mathsf{C}}, \top)\,, & [\alpha+\omega] &= [-]([3]^{\mathsf{C}}, [\alpha+1]^{\mathsf{C}})\,,\\ [\omega^m] &= [-]([\omega^{m-1}+1]^{\mathsf{C}}, \top)\,, & [\alpha+\omega^m] &= [-]([\omega^{m-1}+1]^{\mathsf{C}}, [\alpha+1]^{\mathsf{C}})\,. \end{aligned}$$

For $i > 1$ let

$$\begin{aligned} [\exp(1)] &= \langle - \rangle([2]^{\mathsf{C}}, \perp)\,, & [\alpha+\exp(1)] &= \langle - \rangle([2]^{\mathsf{C}}, [\alpha+1])\,,\\ [\exp^i(1)] &= \langle - \rangle([\exp^{i-1}(1)]^{\mathsf{C}}, \perp)\,, & [\alpha+\exp^i(1)] &= \langle - \rangle([\exp^{i-1}(1)]^{\mathsf{C}}, [\alpha+1])\,. \end{aligned}$$

Note that $[\exp^i(1)] = K_{i+1}^{\Sigma}$, $[\exp^i(1)]^{\mathsf{C}} \equiv K_{i+1}^{\Pi}$. For $p > 0$ let

$$\begin{aligned} [\exp^i(\omega)] &= [-]([\exp^i(2)]^{\mathsf{C}}, \top)\,,\\ [\alpha+\exp^i(\omega)] &= [-]([\exp^i(2)]^{\mathsf{C}}, [\alpha+1]^{\mathsf{C}})\,,\\ [\exp^i(\omega)\omega^p] &= [-]([\exp^i(\omega)\omega^{p-1}+1]^{\mathsf{C}}, \top)\,,\\ [\alpha+\exp^i(\omega)\omega^p] &= [-]([\exp^i(\omega)\omega^{p-1}+1]^{\mathsf{C}}, [\alpha+1]^{\mathsf{C}})\,. \end{aligned}$$

Thus, from the player's point of view when involved in Wadge Games, a player in charge of, for example, the language recognized by $[\exp^i(\omega)]$ is given the choice between the languages recognized by the $[\exp^i(n)]^{\mathsf{C}}$'s. The decision is determined by the number of labels a played on the leftmost branch of the tree before the first b. If the player keeps not playing b forever on the leftmost branch, the tree will be rejected.

Using the basic building blocks and basic steps defined above wea can inductively define automata $[\sum_{i=n}^{1} \delta_i]$, such that each δ_i is of the form $\exp^i(\omega)\eta + \exp^i(1)p$ with $\eta < \omega^\omega$ and $p < \omega$.

To define automata for all β described above, we need one more kind of bricks and two more kinds of steps. For $\eta < \omega^\omega$, $1 \leq i < \omega$, we have:

$$[\exp^i(2)] = [\exp^i(1)] \diamond [\exp^i(1)]^{\mathsf{C}}$$

$$[\alpha + \exp^i(\omega)\eta + \sum_{p=m}^{1} \exp^i(p+2)k_p] = [\alpha + \exp^i(\omega)\eta + \sum_{p=m}^{1} \exp^i(p)k_p] \diamond [\exp^i(2)]$$

$$[\alpha + \exp^i(\omega)\eta + \sum_{p=m}^{1} \exp^i(p+2)k_p + \exp^i(2)] = [\alpha + \exp^i(\omega)\eta + \sum_{p=m}^{\ell} \exp^i(p)k_p + 1] \diamond [\exp^i(2)]\,.$$

6.5.3 Computing Wadge degrees

The strategy we use in order to give an effective description of the LGA Wadge hierarchy is as follows. We start by defining the notion of signed ordinal. Intuitively, ordinals with sign "+" denote Wadge degrees of non self-dual sets such that, in Wadge games, the player in charge of one of those sets is accepting at the beginning of any play. Dually, ordinals with sign "−" denote Wadge degrees of non self-dual sets such that, in Wadge games, the player in charge of one of those sets is rejecting at the beginning of any play. Ordinals with sign "±" denote instead Wadge degrees of self dual sets. Then, we construct canonical sets for every signed degree of the form $\exp^i(\omega)\eta_i + \sum_{p=j_i}^{1} \exp^i(p)\eta_i' + \cdots + \exp(\omega)\eta_1 + \sum_{p=j_1}^{1} \exp(p)\eta_1' + \eta_0$. With their help we prove that the class Φ of signed ordinals from the previous subsection is closed under some set-theoretic operations and that the degree of the results of these operations can be computed. Thanks to this closure property, by verifying that every action of a state corresponds to one of those operations, we are able to show that every language recognized by a linear game automaton has a signed degree which belongs to Φ. By checking that every canonical automaton A_β has Wadge degree β, we finally prove that linear game automata recognize exactly languages of levels in Φ. We end the section by showing that the problem of determining the signed degree of any LGA recognizable language is decidable.

As expected, the technical difficulty of the effective description of the Wadge hierarchy of linear game automata lies in the calculation of the Wadge degree of sets of the form $U \diamond V$, $U \vee V$, $\sup_k^\epsilon (U^{\langle k \rangle} \diamond V)$, for $\epsilon \in \{+,-\}$, and $U \to V$, if $d_W(U) = \exp^i(1)$ for some $i < \omega$. The calculations for the diamond operation are used in the proof of Theorem 6.18. All of them are crucial for the proof of Theorem 6.15. Their long description can be found in the Appendix.

A *signed ordinal* is a pair (ϵ, γ), where γ is an ordinal and $\epsilon \in \{+,-,\pm\}$. Sometimes we call ϵ the sign of the ordinal γ. Slighty abusing notation, we write

$[\gamma]^\epsilon$ for (ϵ, γ). For every $\gamma = \sum_{i=n}^{0} \gamma_i$, where each γ_i is of the form

$$\gamma_i = \exp^i(\omega)\eta_i + \sum_{p=j_i}^{1} \exp^i(p)\eta_i'$$

with $\eta_i, \eta_i' < \omega^\omega$ and $j_i < \omega$, for each possible sign ϵ we inductively define canonical sets of trees $C_{[\gamma]^\epsilon}$ verifying the three following properties:

(1) $d_W(C_{[\gamma]^\epsilon}) = \gamma$,

(2) $(C_{[\gamma]^+})^{\complement} \equiv_W C_{[\gamma]^-}$,

(3) $C_{[\gamma]^\pm} \equiv_W C_{[\gamma]^-} \vee C_{[\gamma]^+}$.

We thence say that a language L has signed degree $[\gamma]^\epsilon$ if there is a canonical $C_{[\gamma]^\epsilon}$ such that $L \equiv_W C_{[\gamma]^\epsilon}$.

Let us start with the basic building bricks of our construction: the sets $C_{[1]^-}$, $C_{[\omega^m]^-}$, $C_{[\exp^i(r)\omega^p]^-}$, and $C_{[\exp^i(\omega)\omega^p]^-}$. Together with these sets we define the canonical set $C_{[\alpha+\gamma]^-}$ of signed degree $[\alpha+\gamma]^-$ as the set $C_{[\alpha]^-} + C_{[\gamma]^-}$, once we already have the sets $C_{[\alpha]^-}$ and $C_{[\gamma]^-}$. The canonical $C_{[\alpha+\gamma]^+}$ and $C_{[\alpha+\gamma]^\pm}$ are then defined as $(C_{[\alpha+\gamma]^-})^{\complement}$ and $C_{[\alpha+\gamma]^-} \vee C_{[\alpha+\gamma]^+}$ respectively. Recall that by $T_{\{a,b\}}$ we denote the space of all full binary trees over $\{a, b\}$. Let

$$C_{[1]^-} = \emptyset\,, \qquad C_{[1]^+} = T_{\{a,b\}}\,, \qquad C_{[1]^\pm} = C_{[1]^-} \vee C_{[1]^+}\,.$$

For every $i, p > 0$, we take for $C_{[\exp^i(p)]^-}$ any $\mathrm{D}_p(\mathbf{\Sigma}_i^0)$-complete subset of $T_{\{a,b\}}$. Then, we define $C_{[\exp^i(p)]^+}$ as $(C_{[\exp^i(p)]^-})^{\complement}$, and $C_{[\exp^i(p)]^\pm}$ as $C_{[\exp^i(p)]^-} \vee C_{[\exp^i(p)]^+}$. For $m > 1$ let

- $C_{[\omega]^-} = \sup_k^- C_{[k]^-}$,
- $C_{[\omega]^+} = (C_{[\omega]^-})^{\complement}$,
- $C_{[\omega]^\pm} = C_{[\omega]^-} \vee C_{[\omega]^+}$,
- $C_{[\omega^m]^-} = \sup_k^- (C_{[\omega^{m-1}+1]^+})^{\langle k \rangle}$,
- $C_{[\omega^m]^+} = (C_{[\omega^m]^-})^{\complement}$,
- $C_{[\omega^m]^\pm} = C_{[\omega^m]^-} \vee C_{[\omega^m]^-}$.

Analogously, for $p > 1$ and $i > 0$, let

- $C_{[\exp^i(\omega)]^-} = \sup_k^- C_{[\exp^i(k)]^-}$,
- $C_{[\exp^i(\omega)]^+} = (C_{[\exp^i(\omega)]^-})^{\complement}$,
- $C_{[\exp^i(\omega)]^\pm} = C_{[\exp^i(\omega)]^-} \vee C_{[\exp^i(\omega)]^+}$,
- $C_{[\exp^i(\omega)\omega^p]^-} = \sup_k^- (C_{[\exp^i(\omega)\omega^{p-1}+1]^+})^{\langle k \rangle}$,
- $C_{[\exp^i(\omega)\omega^m]^+} = (C_{[\exp^i(\omega)\omega^m]^-})^{\complement}$,
- $C_{[\exp^i(\omega)\omega^m]^\pm} = C_{[\exp^i(\omega)\omega^m]^-} \vee C_{[\exp^i(\omega)\omega^m]^-}$,

and, for $r > 0$ and $i > 0$,

- $C_{[\exp^i(r)\omega^p]^-} = \sup_k^- (C_{[\exp^i(r)\omega^{p-1}+1]^+})^{\langle k\rangle}$,
- $C_{[\exp^i(r)\omega^m]^+} = (C_{[\exp^i(r)\omega^m]^-})^{\complement}$,
- $C_{[\exp^i(r)\omega^m]^\pm} = C_{[\exp^i(r)\omega^m]^-} \vee C_{[\exp^i(r)\omega^m]^-}$.

Observe that whenever $\epsilon \in \{+,-\}$ all canonical $C_{[\gamma]^\epsilon}$ are non self-dual, and self-dual when $\epsilon = \pm$. Moreover, as expected, their Wadge degree is exactly γ. Roughly, assigning "+" to the signed degree of a canonical set C corresponds to the fact that, from the player point of view in a Wadge Game, when the player in charge of C starts to play he is accepting. Dually for "−".

Let Φ denote the set of signed ordinals of the form $[\beta]^\epsilon = [\sum_{i=n}^{0} \beta_i]^\epsilon$, where each β_i is of the form

$$\beta_i = \exp^i(\omega)\eta_i + \sum_{p=j_i}^{1} \exp^i(p)k_p$$

with $\eta_i < \omega^\omega$, $k_{2q} \in \{0,1\}$, and $j_i, k_{2q+1} < \omega$. Notice that for every $[\beta]^\epsilon \in \Phi$ there is a canonical set $C_{[\beta]^\epsilon}$.

The long combinatorial proof of the following effective closure property can be found in the Appendix.

Theorem 6.15. *For each $\gamma_1, \gamma_2 \in \Phi$ it holds that the signed degree of $C_{\gamma_1} \diamond C_{\gamma_2}$, $C_{\gamma_1} \vee C_{\gamma_2}$, $sup_k^+ (C_{\gamma_1}^{\langle k\rangle} \diamond C_{\gamma_2})$ and $sup_k^- (C_{\gamma_1}^{\langle k\rangle} \diamond C_{\gamma_2})$ belong to Φ and their Wadge degrees can be computed effectively. The same holds for $C_{\gamma_1} \to C_{\gamma_2}$, if $\gamma_1 = [\exp^i(1)]^\epsilon$ for some $i < \omega$ and $\epsilon \in \{+,-\}$.*

In the next proposition we verify that every language recognized by a linear game automaton has a signed degree which belongs to Φ. This is done essentially by verifying that every action of a node corresponds, in set theoretic terms, to one of the operations of Theorem 6.15.

Proposition 6.16. *If L is* LGA *recognizable, then its signed degree belongs to Φ.*

Proof. In order to prove the proposition, we have to verify that if the languages recognized by the automata A, B_0, B_1 have signed degrees in Φ, then for every state q, the signed degree of the automaton resulting from the action of q over A, B_0, B_1 still belongs to Φ. Because of Lemma 6.1, we consider linear game automata in normal form over the alphabet $\{a,b\}$. The proof goes by induction on the number of states. For the base case, assume the considered automaton has only a single all rejecting, resp. all accepting state. Its signed Wadge degree is then $[1]^-$, resp. $[1]^+$. For the induction step, assume that the automaton is obtained by the action of a existential state q. The universal case is obtained by complementation. An existential state q can either be looping or not looping. If it is not looping, than it is easy to see that q, in set-theoretic terms, is acting either as an alternative ($\vee$) or as a disjunctive product ($\diamond$), or as a finite combination of these two operations. Therefore, by Theorem 6.15 and induction hypothesis it has a signed degree in Φ. If the considered node q is looping, two situations may occur depending on the parity of q. If its parity is even, then the node is acting like a supremum. More precisely, suppose that the result of the action of q is the automaton $\langle + \rangle(A, B_0, B_1)$. Then the language recognized by this automaton is Wadge equivalent to the set $\sup_k^+(L(A)^{\langle k \rangle} \diamond (L(B_0) \diamond L(B_1)))$, and by Theorem 6.15 and induction hypothesis its signed degree belongs to Φ. If the parity of q is odd, the result of the action of q is $\langle - \rangle(A, B_0, B_1)$. In this case we have that the language recognized by the resulting automaton is Wadge equivalent to

$$L(\langle - \rangle(A, \perp)) \to \sup_k^-(L(A)^{\langle k \rangle} \diamond (L(B_0) \diamond L(B_1))).$$

Two cases have to be scrutinized: (1) either $L(A)$ is $\mathbf{\Sigma}^0_n$-complete, or (2) $L(A) \in \mathbf{\Delta}^0_{n+1} \setminus \mathbf{\Sigma}^0_n$, for a certain $n > 0$. In case (1), by Lemma 6.8, $L(\langle - \rangle(A, \perp))$ is still $\mathbf{\Sigma}^0_n$-complete and its signed degree is $[\exp^n 1]^-$. In case (2), by Theorem 6.9 and Theorem 6.11, the language recognized by $L(\langle - \rangle(A, \perp))$ is $\mathbf{\Sigma}^0_{n+1}$-complete and its signed degree is $[\exp^{n+1} 1]^-$. In both situations, by applying Theorem 6.15 and the induction hypothesis, the signed degree of the language recognized by $\langle - \rangle(A, B_0, B_1)$ belongs to Φ. Notice in particular that the previous argument shows that, up to Wadge equivalence, unrestrained existential negative states generate sets of the form $C_{[\exp^i(1)]^-} \to C_{[\gamma]^\epsilon}$, with $[\gamma]^\epsilon \in \Phi$. □

From the previous theorem we immediately obtain that:

Corollary 6.17. *The* LGA *Wadge hierarchy has height at most* $(\omega^\omega)^\omega$.

If we verify that the family of LGA-recognizable languages contains languages L with $d_W(L) = \beta$ for every $[\beta]^\epsilon \in \Phi$, we are able to prove that (1) linear game automata recognize only languages of levels β, with $[\beta]^\epsilon \in \Phi$, and that (2) the height of the LGA hierarchy is exactly $(\omega^\omega)^\omega$. This is done by showing that canonical automata $[\beta]$ are aptly named.

Theorem 6.18. *For every* LGA *automaton* $[\beta]$, $d_W([\beta]) = \beta$, *where* $\beta = \sum_{i=n}^0 \beta_i$, *and each* β_i *is of the form*

$$\beta_i = \exp^i(\omega)\eta_i + \sum_{p=j_i}^{1} \exp^i(p)k_p$$

with $\eta_i < \omega^\omega$, $k_{2q} \in \{0, 1\}$, *and* $j_i, k_{2q+1} < \omega$.

Proof. The proof goes by induction on such ordinals. The initial case [1] is trivial. For the induction step, we have to consider all the possible building blocks described in Subsection 6.5.2.

First remark that, by applying the inductive hypothesis and the definition of the set-theoretic operations of sum and of multiplication by ω, the automaton $[\alpha + 1]$ has Wadge degree $\alpha + 1$, and the automata $[\omega^m]$ and $[\exp^i(\omega)\omega^p]$ have respectively Wadge degree ω^m and $\exp^i(\omega)\omega^p$, for $p, m \geq 1$. Since $[\exp^i(1)] = K^\Sigma_{i+1}$, by Propositions 6.2 and 6.14 we obtain that its Wadge degree is exactly $\exp^i(1)$. By applying Lemma 6.13 and Proposition 6.14, we also obtain that the Wadge degree of $[\exp^i(2)]$ is $\exp^i(2)$ and that $d_W(L([\exp^i(\omega)])) = \sup_{n\in\omega}(\exp^i(n)) = \exp^i(\omega)$.

In order to verify the remaining cases, we reason as follows. First observe that in Wadge Games, as soon as the player in charge of the language recognized by the automaton $[\alpha + \exp^i(\omega)]^{\mathsf{C}}$, resp. $[\alpha + \omega]^{\mathsf{C}}$ has played the first b on the leftmost branch,he is like a player in charge of the set $L([\exp^i(p)] \diamond [\alpha+1])$, resp. $L([p] \diamond [\alpha+1])$, for a certain finite p. On the other hand, for a player in charge in a Wadge Game of the language recognized by the automaton $[\alpha + \exp^i(\omega)\omega^p]^{\mathsf{C}}$, resp. $[\alpha + \omega^p]^{\mathsf{C}}$, as soon as he has played the first b on the leftmost branch, he is like a player in charge of the set recognized by $[\exp^i(\omega)\omega^{p-1}n] \diamond [\alpha + 1]$, resp. by $[\omega^{p-1}n] \diamond [\alpha + 1]$, for a certain finite n. Analogously, as long as the player in charge of the language recognized by the automaton $[\alpha+\exp^i(1)]$, resp. $[\alpha + \exp(1)]$, does not play any node labelled by b on the leftmost branch, he is like a player in charge of the set $L([\exp^i(1)])$, resp. $L([\exp(1)])$. But as soon as he plays the first b on the leftmost branch, he is like a player in charge of the set $L([\exp^{i-1}(1)]^{\mathsf{C}} \diamond [\alpha + 1])$, resp. $L([2]^{\mathsf{C}} \diamond [\alpha + 1])$. Therefore, if we prove that the following identities hold:

- $d_W(L([2]^{\mathsf{C}} \diamond [\alpha + 1])) = \alpha + 2$
- $d_W(L([\exp^{i-1}(1)]^{\mathsf{C}} \diamond [\alpha + 1])) = \alpha + \exp^{i-1}(1)$
- $d_W(L([\omega^{p-1}n] \diamond [\alpha + 1])) = \alpha + \omega^{p-1}n$,
- $d_W(L([\exp^i(\omega)\omega^{p-1}n] \diamond [\alpha + 1])) = \alpha + \exp^i(\omega)\omega^{p-1}n$,

we are done for the considered cases. But those identities can be verified by applying the computations for the diamond operation presented in Section A.3 of the Appendix. For the same reason, we obtain that the Wadge degree of the language recognized by the automaton $[\alpha + \exp^i(\omega)\eta + \sum_{p=m}^{1} \exp^i(p + 2)k_p + \exp^i(2)]$ has degree $\alpha + \exp^i(\omega)\eta + \sum_{p=m}^{1} \exp^i(p+2)k_p + \exp^i(2)$, and that the language recognized by $[\alpha + \exp^i(\omega)\eta + \sum_{p=m}^{1} \exp^i(p + 2)k_p]$ has exactly the degree $\alpha + \exp^i(\omega)\eta + \sum_{p=m}^{1} \exp^i(p + 2)k_p$. □

Observe that, by analyzing the proof of the previous theorem, it can be easily verified that the signed Wadge degree of every automaton $[\beta]$ is $[\beta]^-$, the signed Wadge degree of every automaton $[\beta]^{\mathsf{C}}$ is $[\beta]^+$, and that the signed Wadge degree of every automaton $[\beta]^\pm$ is $[\beta]^\pm$.

As a corollary we obtain the expected lower bound on the height of the hierarchy.

Corollary 6.19. *The* LGA *Wadge hierarchy has height at least* $(\omega^\omega)^\omega$.

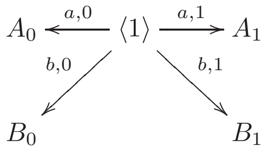
From Corollaries 6.17 and 6.19, we finally have that:

Proposition 6.20. *The Wadge hierarchy of linear game automata has the height* $(\omega^\omega)^\omega = \omega^{\omega^2}$.

We end this section by showing that the problem of determining the signed Wadge degree of a LGA-recognizable tree language is decidable.

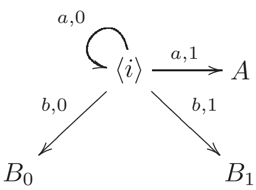
Theorem 6.21. *For each* LGA *we can calculate effectively the signed degree of the recognized language.*

Proof. We proceed by induction on the number of states. Let C be an LGA. If C has only one state, it is either totally accepting or totally rejecting. In the first case the signed degree is $[1]^+$, in the second case it is $[1]^-$. Suppose that C has more states. By duality we may assume that the initial state q_0 is existential: if it is universal, compute the signed degree for the complement of C, and return the degree negated. Suppose that q_0 is not looping. By linearity, C can be represented as in the figure below for some automata A_0, A_1, B_0, B_1, each having less states than C.

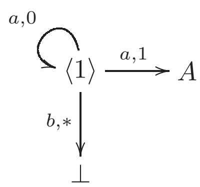


Clearly $L(C) \equiv L(A_0) \diamond L(A_1) \vee L(B_0) \diamond L(B_1)$. Hence, we can use the induction hypothesis to get the degrees of $L(C_{q_i})$, and then Theorem 6.15 to compute $d_W(C) = d_W(C_{q_1}) \diamond d_W(C_{q_2}) \vee d_W(C_{q_3}) \diamond d_W(C_{q_4})$.

If q_0 is looping, we can assume w.l.o.g. that C is of the form shown in the figure below with $i = 0, 1$.



If $i = 1$, there exists $n \in \omega$ such that $L(A)$ is either Σ^0_n-complete, or in $\Delta^0_{n+1} \setminus \Sigma^0_n$. If $L(A)$ is $\mathbf{\Sigma}^0_n$-complete, by Lemma 6.8, the language recognized by C', defined in the figure below is also Σ^0_n-complete.



Since $d_W(A) = d_W(C') = [\exp^n(1)]^-$ and $([\exp^n(1)]^-)^{\langle k\rangle} = [\exp^n(1)]^-$ for each $k > 0$, we have $d_W(C) = [\exp^n(1)]^- \to d_W(B_1) \diamond d_W(B_2) \diamond [\exp^n(1)]^-$. On the other hand, if $L(A) \in \Delta^0_{n+1} \setminus \Sigma^0_n$, by Theorem 6.9 and Theorem 6.11, the language recognized by C' is Σ^0_{n+1}-complete, and we already observed that $d_W(C) = [\exp^{n+1}(1)]^- \to d_W(\sup^-_k(A^{\langle k\rangle} \diamond (B_1 \diamond B_2))$. We conclude by the inductive hypothesis and Theorem 6.15.

If $i = 0$, $d_W(C) = \sup^+_k(d_W(A)^{\langle k\rangle} \diamond d_W(B_1 \diamond B_2))$, and again the claim follows from Theorem 6.15 and the induction hypothesis. □

6.6 Summarizing remarks

While for ω-regular languages the understanding of the corresponding index and topological hierarchies is complete, for trees the situation is not so satisfactory. The only case examined satisfactorily is that of deterministic automata. This is due to the work of Niwinski and Walukiewicz [101, 102], and of Murlak [90, 92, 93]. For non-deterministic or alternating automata the only results obtained are strictness theorems for various classes [32, 33, 89, 96], and lower bounds for the heights of the hierarchies [51, 114].

In this chapter we proposed a novel class of automata, named linear game automata, capturing an interesting aspect of alternation and with all three hierarchies – index, Borel, and Wadge – decidable. Moreover we shown that the weak index and the Borel rank coincide over LGA-recognizable languages.

We saw that, despite their apparent simplicity, LGA yield a class of languages surprisingly complex from the topological point of view: the height of their Wadge hierarchy is $(\omega^\omega)^\omega$. Admittedly, this is much less than the hierarchy for weak alternating automata, which is known to be at least ϵ_0 high [51]. But this was expected, as LGA form a very restricted subclass of weak alternating automata. What is surprising however, is that the height of the Wadge hierarchy for LGA is much larger than that for deterministic automata, which was shown in [92] to be $(\omega^\omega)^3 + 3$, and the same as for deterministic *push-down* automata on infinite words [49].

Conclusion

This thesis focused on the complexity of some fragments of the modal μ-calculus, or equivalently of some subclasses of alternating automata. More precisely, in the first part of this work we have studied the expressive power of the modal μ-calculus over some restricted classes of transition systems, while in the second part we gave a complete effective characterization of the three classical hierarchies (index, Borel and Wadge) for a class of tree automata capturing a very weak form of alternation.

For what concerns the **First Part** of the dissertation, some of the results obtained in **Chapter 2** and **Chapter 3** can be summarized by the following figure:

	KT	K4	KB4	GL
fixpoint alternation hierarchy	strict	collapse to the alternation free fragment	collapse to the modal fragment	collapse to the modal fragment

where KT stands for the class of all reflexive transition systems, K4 for the class of all transitive transition systems, KB4 for the class of all transitive and symmetric transition systems, and GL for the class of all transitive and upward well-founded models. The collapse of the modal μ-calculus into its modal fragment over GL was already proved by van Benthem in [13] and Visser in [123] by using the De Jongh-Sambin fixpoint theorem. Our proof is independent of this important result, and uses results from Chapter 2 and the fact that on upward well-founded models the modal μ-calculus collapses into the first ambiguous class of the fixpoint alternation hierarchy. In extending the language by allowing fixpoints to bound also negative occurrences of free variables and showing that it collapses on the modal fragment, we were then able to provide a new proof of the uniqueness theorem of Bernardi, De Jongh and Sambin and a constructive proof of the existence theorem of De Jongh and Sambin.

Chapter 4, which is the last chapter of the first part, can be seen as a kind of "bridge" between the first and the second parts of the thesis. Indeed, in this chapter we shown that on transitive models modal logic corresponds exactly to the Borel fragment of the modal μ-calculus. This was done by providing a bunch of equivalent effective characterizations for the temporal logic EF on arbitrary trees. More specifically we proved that up to EF-bisimilarity, the property of being definable by an EF formula and the property of being a Borel set coincide. Since we were able to verify that every language definable in weak monadic second order logic with the child relation is Borel, we immediately obtained that the logic EF is the EF-bisimulation invariant fragment of WMSO. By verifying that all these properties are also equivalent with an effective algebraic characterization of EF-definability for *finitely* branching trees given by Bojanczyk and Idziaszek [24], as a corollary we obtained their decidability.

In the much shorter **Second Part** of the dissertation we introduced a new subclass of weak alternating tree automata, linear game automata, and provided an effective characterization for all the three corresponding hierarchies: index hierarchy, Borel hierarchy and Wadge hierarchy. Moreover, we verified that for every language recognized by those automata, the Borel rank and the Mostowski-Rabin index coincide, making another step towards a positive answer to Murlak's conjecture, stating that for weakly recognizable tree languages the weak index hierarchy and the Borel hierarchy coincide level by level ([93]).

Future work

Part 1

Concerning the modal μ-calculus hierarchy on restricted class of models, the missing case is the symmetric case. It would be nice to fill in this gap. We conjecture that the fixpoint hierarchy is strict for this class of transition systems.

Another open problem is to decide for any given definable language its position in the fixpoint hierarchy. This is a very difficult and important question. To our knowledge, it is only known how to decide the low levels of the hierarchy ([78, 103, 129]). As we saw, recently, Colcombet and Löding [42] have been able to reduce the analogous problem for tree automata to the uniform universality problem for what are called distance-parity automata.

Also, finding new, possibly effective, characterizations of tree logics is another important problem in the area. As an extension of a result presented in this dissertation, it would be for instance nice to know whether on full trees it holds that any recognizable tree language is Borel iff it is definable in weak monadic second order logic.

Finally note that, although Janin and Lenzi [63] proved[1] that over the class of graphs of bounded degree, every μ-formula can be translated into a MSO formula of the third monadic class Σ_3, the question whether the modal μ-calculus is equivalent to the bisimulation invariant fragment of some fixed level of the quantifier alternation hierarchy of MSO remains open.

Part 2

We believe that the notion of game automata is well suited to take us further. Indeed, the next step is to consider weak and then strong game automata, where weak strong automata are weak alternating automaton where the transition relation is a total function, while strong game automata are (strong) alternating automaton where the transition relation is a total function. This last class is already quite expressive, as it contains inhabitants of every level of the (strong) index hierarchy and subsumes deterministic languages. Extending decidability of the index, Borel/projective and Wadge hierarchies to this class would be an

[1]The authors remark that their result is a consequence of the work of Courcelle [43], who shows that this is true on a quite general class of graphs.

important result, though possibly the last one accessible with the tools we used for instance in Chapter 6.

Because of its difficulty, the general case probably requires new techniques. However for a start, it would be very nice to already solve those hierarchical questions for weak alternating automaton. A first step in this direction would be finding an answer to Murlak's conjecture by showing that for weakly recognizable languages, the weak index and the Borel rank coincide. Another step would be also to prove that the lower bound obtained by Duparc and Murlak for the corresponding Wadge hierarchy is tight.

Nowadays, there is a regain of interest among the automata theory community in the idea of considering *quantitative extensions* of the standard theory of regular languages. This is due to a series of papers of Bojanczyk, Colcombet, Kirsten and Löding, among others. First of all, Kirsten [72] gave a much simpler and self-contained proof of the one proposed by Hashiguchi [58, 59] for the star-height decision problem through a reduction to the limitedness problem for a form of automata called nested distance desert automata[2]. The same strategy was used by Colcombet and Löding. In [41], the star-height problem over trees has been solved by a reduction to the limitedness problem of nested distance desert automata over trees. As we already mentioned, in [42] they tried a similar attempt for solving the problem of deciding the index hierarchy of non-deterministic automata over infinite trees.

At the same time as Kirsten's work, Bojanczyk [19] introduced an extension MSOU of MSO, where a new second-order quantifier $\mathsf{U}X.\psi(X)$ was added meaning "there exists a set X (of infinite trees or words) as big as I want such that $\psi(X)$ holds". He showed that for two fragments of this logic the satisfiability problem is decidable. The word cases were studied in more depth by Bojanczyk and Colcombet [23]. In this paper, by introducing BS-automata, a certain class of counter non-deterministic automata very close to distance desert automata, the authors provided some fragments of MSOU that have decidable satisfiability over infinite words. But non-determinism is important for full MSO, where existential quantication over infinite sets is allowed, and it comes with a cost: BS-automata are *not* closed under complementation, and it is not clear what the correct automaton model for full MSOU is. Moreover, it is still an open problem if full MSOU has decidable satisfiability over infinite words. Interestingly, all these works point to the following argument: there are robust extensions of regular languages, extensions that have descriptions in terms of both automata and logic. This argument was investigated in two recent papers by Bojanczyk [21, 25], the last one joint with Torunczyk, where new classes of languages of infinite words were defined and were showed to have two equivalent descriptions: in terms of a deterministic counter automaton (called a Max-, resp. Min-automaton), and in terms of an extension of weak MSO. Moreover, it was shown that Min- and Max-automata fit in a more general picture, where deterministic automata with prefix-closed acceptance conditions define extensions of weak MSO.

The study of those quantitative extensions of regular languages appears to be a particularly fertile and interesting territory for the use of topological meth-

[2] A distance desert automaton is a non-deterministic automaton running over words which can count the number of occurrences of some "special" states. A nested distance desert automaton is nothing but a distance desert automaton in which multiple counters and reset of the counters are allowed (with a certain constraint of nesting of counters).

ods. Already in [21], Bojanczyk used a topological argument in order to show that non-deterministic Max-automata recognize strictly more languages than deterministic ones. Cabessa, Duparc, Facchini and Murlak [38] compared ω-regular and Max-regular languages in terms of topological complexity. The authors proved that up to Wadge equivalence the classes coincide. Moreover, when restricted to Δ^0_2-languages, they showed that the classes contain virtually the same languages. On the other hand, separating examples of arbitrary complexity exceeding Δ^0_2 were constructed. More recently, Hummel, Skrzypczak and Torunczyk [62] were able to show that MSOU on ω-words can even define non-Borel sets. From this fact, they concluded that there is no model of non-deterministic automata with a Borel acceptance condition which captures all of MSOU. In the same article, the authors also gave an exact topological complexity of the classes of languages recognized by non-deterministic B-, S- and BS-automata on infinite words. Furthermore, they verified that, since they inhabit all finite levels of the Borel hierarchy, the corresponding alternating automata have higher topological complexity than non-deterministic ones.

All the previous papers show that, as future work, it would be very interesting to further study the topological complexity of these quantitative extensions of regular languages. It would be for instance nice to have an effective description of the Wadge hierarchy of deterministic and non-deterministic counter automata, but also to know whether there are non Borel languages recognized by an alternating BS-automaton.

Appendix A

Computations of Chapter 6

A.1 Basic properties of operations

Remember that we say that a non self dual set L is *initializable* if $L \geq_W L \to L$. The following lemmas summarize simple yet useful properties of the operations on languages. They can be proved with standard Wadge game arguments.

LEMMA A.1. *For initializable A, B and arbitrary A', B' and A_n, B_n, $n < \omega$*

$$
\begin{aligned}
(A \to A') \diamond (B \to B') &\equiv_W A \diamond B \to ((A \to A') \diamond B' \vee A' \diamond (B \to B'))\ ,\\
(A \to A') \diamond B &\equiv_W A \diamond B \to A' \diamond B\ ,\\
(A \to A') \diamond \overset{+}{\sup_n} B_n &\equiv_W \overset{+}{\sup_n} (A \to A') \diamond B_n\ ,\\
(A \to A') \diamond \overset{-}{\sup_n} B_n &\equiv_W A \to \left((A' \diamond \overset{-}{\sup_n} B_n) \vee (\overset{-}{\sup_n} (A \to A') \diamond B_n) \vee (\overset{+}{\sup_n} (A \to A') \diamond B_n) \right) \quad \textit{for } A > \bot\ ,\\
(\bot \to A') \diamond \overset{-}{\sup_n} B_n &\equiv_W \bot \to \left((A' \diamond \overset{-}{\sup_n} B_n) \vee (\overset{-}{\sup_n} (\bot \to A') \diamond B_n) \right)\ ,\\
(\top \to A') \diamond \overset{-}{\sup_n} B_n &\equiv_W \top \to \left((A' \diamond \overset{-}{\sup_n} B_n) \vee (\overset{+}{\sup_n} (\top \to A') \diamond B_n) \right)\ ,\\
\overset{+}{\sup_m} A_m \diamond \overset{+}{\sup_n} B_n &\equiv_W \top \to \left(\overset{+}{\sup_m} (A_m \diamond \overset{+}{\sup_n} B_n) \vee (\overset{+}{\sup_n} (\overset{+}{\sup_m} A_m) \diamond B_n) \right)\ ,\\
\overset{+}{\sup_m} A_m \diamond \overset{-}{\sup_n} B_n &\equiv_W \top \to \left(\overset{+}{\sup_m} (A_m \diamond \overset{-}{\sup_n} B_n) \vee (\overset{+}{\sup_n} (\overset{+}{\sup_m} A_m) \diamond B_n) \right)\ ,\\
\overset{-}{\sup_m} A_m \diamond \overset{-}{\sup_n} B_n &\equiv_W \bot \to \left(\overset{-}{\sup_m} (A_m \diamond \overset{-}{\sup_n} B_n) \vee (\overset{-}{\sup_n} (\overset{-}{\sup_m} A_m) \diamond B_n) \right)\ .
\end{aligned}
$$

LEMMA A.2. *For arbitrary sets A, B, C, D it holds that*

- $(A \vee B) \diamond C \equiv_W A \diamond C \vee B \diamond C$,
- *if $A \leq_W C$ and $B \leq_W D$, then $A \diamond B \leq_W C \diamond D$.*

Let us now recall the elegant relation between operations on sets and Wadge degrees.

Proposition A.3 ([48]). *For arbitrary Borel sets L, M, L_n with $n < \omega$ it holds that*

$$d_W(M + L) = d_W(M) + d_W(L)\,,$$
$$d_W(\sup_n^- L_n) = d_W(\sup_n^+ L_n) = \sup_n(d_W(L_n) + 1)\,.$$

A.2 Closure by $\vee$ and $\to$

For $n > 0$, let $\hat{\Phi}_n$ denote the set of signed ordinals of the form $\left[\Sigma_{i=1}^N \exp^n(i)k_i\right]^\mu$ for $\mu \in \{+,-,\pm\}$ and some natural N, k_i, with $k_i \in \{0,1\}$ for even i. For uniformity, $\hat{\Phi}_0 = \{0^+, 0^-, 0^\pm\}$. Let $\Phi_n = \{[\exp^n(\omega)\alpha + \beta]^\mu : \mu \in \{+,-,\pm\}\,,\ \alpha < \omega^\omega\,,\ \beta^\mu \in \hat{\Phi}_n\}$ for $n > 0$ and $\Phi_0 = \{[\alpha]^\mu : \mu \in \{+,-,\pm\}\,,\ \alpha < \omega^\omega\}$. Recall that by Φ we denote the set of signed ordinals of the form $[\alpha_k + \ldots + \alpha_1 + \alpha_0]^\mu$, with $\alpha_i^+ \in \Phi_i$.
To make the notation more readable, in what follows we write $[\alpha]^\mu$ for the canonical set $C_{[\alpha]^\mu}$ of signed degree $[\alpha]^\mu$.
The closure of Φ by $\vee$ is very simple. If $[\alpha]^\mu$ and $[\beta]^\nu$ are comparable, $[\alpha]^\mu \vee [\beta]^\nu$ is simply equal to the larger of the two. If $[\alpha]^\mu$ and $[\beta]^\nu$ are incomparable, then necessarily $[\beta]^\nu = [\alpha]^{\bar{\mu}}$ and the result is $[\alpha]^\pm$.
Let us now concentrate on $\to$. First we state yet another simple observation. By $\bar{\mu}$ we denote the dual sign:

$$\bar{\mu} = \begin{cases} + & \text{if } \mu = -\,, \\ - & \text{if } \mu = +\,, \\ \pm & \text{if } \mu = \pm\,. \end{cases}$$

Lemma A.4. *For arbitrary tree sets A_n,*

$$[1]^\mu \to \sup_n^\nu A_n \equiv_w \begin{cases} \sup_n^\nu A_n & \textit{if } \mu = \nu \in \{-,+\} \\ [1]^\mu \to (\sup_n^+ A_n) \vee (\sup_n^- A_n) & \textit{if } \mu = \bar{\nu} \in \{-,+\}\,. \end{cases}$$

Observe that by Proposition A.3 $[\exp^i(1)]^+ \to [\beta]^\pm \equiv_W [\beta + \exp^i(1)]^+$. Thus, it follows that the result is in Φ. Consider now $[\exp^i(1)]^+ \to [\beta]^\nu$ for $\nu \in \{-,+\}$. If $[\beta]^\nu = B \to [\beta']^\pm$, with B initializable, it follows that $[\exp^i(1)]^\mu \to [\beta]^\nu \equiv_W ([\exp^i(1)]^+ \to B) \to [\beta']^\pm$. It is easy to see that

$$[\exp^i(1)]^+ \to B \equiv_W \begin{cases} [\exp^i(1)]^+ & \text{for } \ B \leq_W [\exp^i(1)]^+ \\ B & \text{for } \ B \geq_w [\exp^i(1)]^+\,, \\ [\exp^i(1)2]^+ & \text{for } \ B \equiv_W [\exp^i(1)]^-\,, \end{cases}$$

and we can conclude from the previous case. The remaining case is that of $[\beta]^\nu = \sup^\nu[\beta' + \beta_n]^+$, with $\beta_n = \exp^i(\omega)\omega^p n$ or $\beta_n = \exp^i(n)$. If $i > 0$, we have $[\exp^i(1)]^\mu \to [\beta]^\nu \equiv_W ([\exp^i(1)]^\mu \to [1]^{\bar{\nu}}) \to [\beta]^\nu \equiv_W [\exp^i(1)]^\mu \to ([1]^{\bar{\nu}} \to [\beta]^\nu)$ By Lemma A.4, this is equal to $[\exp^i(1)]^\mu \to [1]^{\bar{\nu}} \to [\beta]^\pm$. Hence, $[\exp^i(1)]^\mu \to [\beta]^\nu \equiv_W [\exp^i(1)]^\mu \to [\beta]^\pm$ and we conclude by the first case. For $i = 0$ use Lemma A.4.

A.3 Closure by $\diamond$

First of all note that since $\beta^+ \vee \beta^- = \beta^\pm$, by Lemma A.2 $\alpha^\mu \diamond \beta^\pm$ corresponds to $(\alpha^\mu \diamond \beta^+) \vee (\alpha^\mu \diamond \beta^-)$. Therefore, if $\alpha^\mu \diamond \beta^\nu \in \Phi$, for both $\nu = +,-$, $\alpha^\mu \diamond \beta^\pm \in \Phi$ holds because we know that Φ is closed under $\vee$.

Lemma A.5. *Fix a natural number $n > 0$. For each $[\alpha]^\mu, [\beta]^\nu \in \hat{\Phi}_n$, one can effectively find $[\gamma_1]^{\lambda_1}, [\gamma_2]^{\lambda_2}, [\gamma_3]^{\lambda_3}$ such that $[\gamma_i]^{\lambda_i} \in \hat{\Phi}_n$ or $[\gamma_i]^{\lambda_i} = [\gamma_i' + 1]^{\lambda_1}$ and*

$$[\alpha]^\mu \diamond [\beta]^\nu = [\gamma_1]^{\lambda_1}\,, \quad [\alpha + 1]^\mu \diamond [\beta]^\nu = [\gamma_2]^{\lambda_2}\,, \quad [\alpha + 1]^\mu \diamond [\beta + 1]^\nu = [\gamma_3]^{\lambda_3}\,.$$

Proof. We only give a proof of the first assertion. The remaining two are very similar.
We proceed by induction on the sum of the coefficients of α and β. The basic step is $\alpha = \exp^n(p)$, $\beta = \exp^n(q)$. By Lemma 6.13, we find r and λ such that $\alpha^\mu \diamond \beta^\nu = \exp^n(r)^\lambda$.
In the inductive step we give argument for the case $\alpha^\mu = \exp^n(p)^\mu \to (\gamma^- \vee \gamma^+)$ and $\beta^\nu = \exp^n(q)^\nu \to (\delta^- \vee \delta^+)$; the case when $\alpha = \exp^n(p)$ or $\beta = \exp^n(q)$ is very similar, only uses the second assertion of Lemma A.1.
Since $\exp^n(p)^\mu$ and $\exp^n(q)^\nu$ are initializable, by Lemma A.1 and Lemma A.2, we get

$$\alpha^\mu \diamond \beta^\nu = \exp^n(p)^\mu \diamond \exp^n(q)^\nu \to \left(\alpha^\mu \diamond \delta^- \vee \alpha^\mu \diamond \delta^+ \vee \gamma^- \diamond \beta^\nu \vee \gamma^+ \diamond \beta^\nu\right) .$$

By induction hypothesis we compute $\alpha^\mu \diamond \delta^-, \alpha^\mu \diamond \delta^+, \gamma^- \diamond \beta^\nu, \gamma^+ \diamond \beta^\nu \in \hat{\Phi}_n$ and by the properties of $\vee$ we get $\eta^\kappa = \alpha^\mu \diamond \delta^- \vee \alpha^\mu \diamond \delta^+ \vee \gamma^- \diamond \beta^\nu \vee \gamma^+ \diamond \beta^\nu \in \hat{\Phi}_n$. Again by Lemma 6.13 we get an expression of the form $\exp^n(r)^\lambda \to \eta^\kappa$, which can be presented as $\left(\sum_{i=N}^{1} \exp^n(i) m_i\right)^{\lambda'}$. Indeed, by an argument analogous to the one used after Lemma A.4, one shows that for $\eta = \sum_{i=M}^{L} \exp^n(i)\ell_i$ with $\ell_L > 0$ and for $\lambda \in \{+,-\}$,

$$[\exp^n(r)]^\lambda \to \eta^\kappa \equiv_W \begin{cases} \eta^\kappa & \text{if } L > n \text{ and } \kappa \in \{+,-\}\,, \text{ or} \\ & \quad L = n \text{ and } \kappa = \lambda\,, \\ [\eta + \exp^n(r)]^\lambda & \text{if } L \geq n \text{ and } \kappa = \pm\,, \text{ or} \\ & \quad L = n \text{ and } \kappa = \bar{\lambda}\,, \\ [\sum_{i=M}^{r} \exp^n(i)\ell_i + \exp^n(r)]^\lambda & \text{if } L < n\,. \end{cases}$$

It remains to show that $m_i \in \{0,1\}$ for even i.
Suppose first that r is even, and $\lambda = -$. Then by Lemma 6.13 p and q are even, $r = p+q$ and $\mu = \nu = -$. For each of $\alpha^\mu \diamond \delta^-, \alpha^\mu \diamond \delta^+, \gamma^- \diamond \beta^\nu, \gamma^+ \diamond \beta^\nu$ we can apply Lemma A.1, and get an expression $\exp^n(s)^\kappa \to \iota^{\kappa_1}$, where ι^{κ_1} can be again obtained from the inductive hypothesis. The possible values for $\exp^n(s)^\kappa$ are $\exp^n(p)^- \diamond \exp^n(t)^- = \exp^n(p+t)^-$ or $\exp^n(p)^- \diamond \exp^n(t)^+ = \exp^n(p+t)^+$ with $t > q$ (since q is even and $\beta^\nu \in \Phi$), or $\exp^n(q)^- \diamond \exp^n(u)^- = \exp^n(q+u)^-$ or $\exp^n(q)^- \diamond \exp^n(u)^+ = \exp^n(q+u)^+$ with $u > p$ (since p is even and $\alpha^\mu \in \Phi$). Hence, $s > p+q = r$. In consequence, by the properties of $\to$ and $\vee$, $\eta^\kappa = (\Sigma_{i=L}^{M} \exp^n(i)\ell_i)^\kappa$ with $L > r$, $\ell_L > 0$. Hence,

$$\alpha^\mu \diamond \beta^\nu = \exp^n(r)^- \to \left(\sum_{i=L}^{M} \exp^n(i)\ell_i\right)^\kappa = \begin{cases} \left(\sum_{i=L}^{M} \exp^n(i)\ell_i\right) + \exp^n(r)^- & \kappa = \pm \\ \left(\sum_{i=L}^{M} \exp^n(i)\ell_i\right)^\kappa & \kappa \in \{-,+\} \end{cases} .$$

In either case the result is in $\hat{\Phi}_n$.
Next, suppose that r is even, and $\lambda = +$. Then,

$$r = \begin{cases} p+q & p \equiv q \,(\mathrm{mod}\, 2), \mu = \bar{\nu} \\ p+q-1 & p \text{ odd}, q \text{ even}, \nu = + \text{ (or symmetrically)} \\ p+q-2 & p, q \text{ even}, \mu = \nu = + \end{cases}$$

Again, we can present each of $\alpha^\mu \diamond \delta^-, \alpha^\mu \diamond \delta^+, \gamma^- \diamond \beta^\nu, \gamma^+ \diamond \beta^\nu$ as $\exp^n(t)^\kappa \to \iota^{\kappa_1}$ with $\iota^{\kappa_1} \in \hat{\Phi}_n$. In order to carry on like before it is enough to show that in no case $\exp^n(t)^\kappa = \exp^n(r)^-$. Indeed, if we exclude this possibility, by inductive hypothesis and by the properties of $\to$ and $\vee$, we can conclude that

$$\eta^\kappa = \begin{cases} \left(\sum_{i=L}^{M} \exp^n(i)\ell_i\right)^\kappa \\ \left(\sum_{i=L}^{M} \exp^n(i)\ell_i\right) + \exp^n(r)^+ \end{cases}$$

for $L > r$, $\ell_L > 0$, and so

$$\alpha^\mu \diamond \beta^\nu = \begin{cases} \left(\sum_{i=L}^{M} \exp^n(i)\ell_i\right)^\kappa \\ \left(\sum_{i=L}^{M} \exp^n(i)\ell_i\right) + \exp^n(r)^+ \end{cases}.$$

These two are both in $\hat{\Phi}_n$, since η^κ is. So it remains to see that $\exp^n(t)^\kappa \neq \exp^n(r)^-$. Observe that only an expression of the form $\exp^n(2i)^- \diamond \exp^n(2j)^-$ can give $\exp^n(r)^-$ for even r. The only case giving a chance of such an expression is when $r = p + q$. But then $2i = p$ and, since $\beta^\nu \in \Phi$, $2j > q$ (or symmetrically), so $2i + 2j > p + q = r$, and we are safe.
If r is odd, the claim follows easily by induction hypothesis, and by properties of $\to$. □

Lemma A.6. *Let $N > 0$ and $[\alpha_0]^\mu, [\beta_0]^\nu \in \hat{\Phi}_N$, with $\mu, \nu \in \{+, -\}$. Define $\hat{\alpha}_0 = \alpha_0 + \varepsilon_\alpha$, and $\hat{\beta}_0 = \beta_0 + \varepsilon_\beta$ with $\varepsilon_\alpha, \varepsilon_\beta \in \{0, 1\}$. Let $\alpha = \exp^N(\omega)(\omega^k a_k + \ldots + \omega a_1 + a_0) + \hat{\alpha}_0$ and $\beta = \exp^N(\omega)(\omega^k b_k + \ldots + \omega b_1 + b_0) + \hat{\beta}_0$. Let ℓ_α be the least i for which $a_i > 0$, and similarly for ℓ_β.*

- *If $\alpha_0 > 0$, $\beta_0 > 0$, then*

$$[\alpha]^\mu \diamond [\beta]^\nu = [\hat{\alpha}_0]^\mu \diamond [\hat{\beta}_0]^\nu \to \left[\exp^N(\omega)(\omega^k(a_k + b_k) + \ldots + \omega(a_1 + b_1) + (a_0 + b_0))\right]^\pm.$$

- *If $\alpha_0 > 0$, $\hat{\beta}_0 = 0$, then*

$$[\alpha]^\mu \diamond [\beta]^+ = \left[\exp^N(\omega)(\omega^k(a_k + b_k) + \ldots + \omega^{\ell_\beta}(a_{\ell_\beta} + b_{\ell_\beta}))\right]^+,$$
$$[\alpha]^\mu \diamond [\beta]^- = [\hat{\alpha}_0]^\mu \to \left[\exp^N(\omega)(\omega^k(a_k + b_k) + \ldots + \omega(a_1 + b_1) + (a_0 + b_0))\right]^\pm.$$

- *If $\alpha_0 = 0$, $\beta_0 = 0$, then*

$$\begin{aligned}
[\alpha]^- \diamond [\beta]^- &= \left[\exp^N(\omega)(\omega^k(a_k + b_k) + \ldots + \omega(a_1 + b_1) + (a_0 + b_0))\right]^- && \textit{for } \varepsilon_\alpha = \varepsilon_\beta = 0,\\
[\alpha]^- \diamond [\beta]^- &= \left[\exp^N(\omega)(\omega^k(a_k + b_k) + \ldots + \omega(a_1 + b_1) + (a_0 + b_0)) + 1\right]^- && \textit{for } \varepsilon_\alpha = \varepsilon_\beta = 1,\\
[\alpha]^- \diamond [\beta]^- &= \left[\exp^N(\omega)(\omega^k(a_k + b_k) + \ldots + \omega(a_1 + b_1) + (a_0 + b_0)) + 1\right]^- && \textit{for } \varepsilon_\alpha = 0, \varepsilon_\beta = 1, \ell_\beta \geq \ell_\alpha\\
[\alpha]^- \diamond [\beta]^- &= \left[\exp^N(\omega)(\omega^k(a_k + b_k) + \ldots + \omega(a_1 + b_1) + (a_0 + b_0))\right]^- && \textit{for } \varepsilon_\alpha = 0, \varepsilon_\beta = 1, \ell_\beta < \ell_\alpha\\
[\alpha]^- \diamond [\beta]^+ &= \left[\exp^N(\omega)(\omega^k(a_k + b_k) + \ldots + \omega^{\ell_\beta}(a_{\ell_\beta} + b_{\ell_\beta}))\right]^+ && \textit{for } \varepsilon_\beta = 0,\\
[\alpha]^- \diamond [\beta]^+ &= \left[\exp^N(\omega)(\omega^k(a_k + b_k) + \ldots + \omega(a_1 + b_1) + (a_0 + b_0)) + 1\right]^+ && \textit{for } \varepsilon_\alpha = 0, \varepsilon_\beta = 1,\\
[\alpha]^+ \diamond [\beta]^+ &= \left[\exp^N(\omega)(\omega^k(a_k + b_k) + \ldots + \omega^\ell(a_\ell + b_\ell - \varepsilon))\right]^+ && \textit{for } \varepsilon_\alpha = \varepsilon_\beta = 0,\\
[\alpha]^+ \diamond [\beta]^+ &= \left[\exp^N(\omega)(\omega^k(a_k + b_k) + \ldots + \omega^{\ell_\beta}(a_{\ell_\beta} + b_{\ell_\beta}))\right]^+ && \textit{for } \varepsilon_\alpha = 0, \varepsilon_\beta = 1,
\end{aligned}$$

where $\ell = \max(\ell_\alpha, \ell_\beta)$ and ε equals 1 if $\ell_\alpha = \ell_\beta$ and 0 otherwise.

Proof. In the proof we assume that $\varepsilon_\alpha = \varepsilon_\beta = 0$, the remaining cases being very similar. We prove all the equations simultaneously by induction on (α, β). Let $\theta = \exp^N(\omega) \sum_{i=k}^{0} \omega^i(a_i + b_i)$.

- **Case $\alpha_0 > 0$, $\beta_0 > 0$.** Suppose $\alpha = \alpha' + \exp^N(p)$, $\beta = \beta' + \exp^N(q)$. We have

$$[\alpha]^\mu \diamond [\beta]^\nu = [\exp^N(p)]^\mu \diamond [\exp^N(q)]^\nu \to \left([\alpha]^\mu \diamond [\beta']^\pm \vee [\alpha']^\pm \diamond [\beta]^\nu\right)$$

Let $\alpha_0 = \gamma + \exp^N(p)$, $\beta_0 = \delta + \exp^N(p)$. We get

$$\begin{aligned}
[\alpha]^\mu \diamond [\beta]^\nu &= [\exp^N(p)]^\mu \diamond [\exp^N(q)]^\nu \to \Big(A \to [\theta]^\pm \vee\ B \to [\theta]^\pm\Big)\\
&= [\exp^N(p)]^\mu \diamond [\exp^N(q)]^\nu \to (A \vee B) \to [\theta]^\pm\\
&= [\alpha_0]^\mu \diamond [\beta_0]^\nu \to [\theta]^\pm.
\end{aligned}$$

with $A = [\alpha_0]^\mu \diamond [\delta]^\pm$ (or $A = [\alpha_0]^\mu$ if $\delta = 0$) and $B = [\gamma]^\pm \diamond [\beta_0]^\nu$ (or $B = [\beta_0]^\nu$ if $\gamma = 0$).

• **Case** $\alpha_0 > 0$**,** $\beta_0 = 0$**.** Suppose $\ell_\beta > 0$. Then by Proposition A.3 we get that $[\exp^N(\omega)\omega^{\ell_\beta}]^+ = \sup_i^+[\exp^N(\omega)\omega^{\ell_\beta-1}i]^+$. By the induction hypothesis for

$$b'_i = \begin{cases} b_i & i \neq \ell_\beta \\ b_i - 1 & i = \ell_\beta \end{cases}$$

we have

$$[\alpha]^\mu \diamond [\beta]^+ = \sup_i^+ \alpha^\mu \diamond \left[\exp^N(\omega)\left(\left(\sum_{j=k}^{\ell_\beta}\omega^j b'_j\right) + \omega^{\ell_\beta-1}i\right)\right]^+$$
$$= \sup_i^+ \left[\exp^N(\omega)\left(\left(\sum_{j=k}^{\ell_\beta}\omega^j(a_j + b'_j)\right) + \omega^{\ell_\beta-1}(a_{\ell_\beta-1}+i)\right)\right]^+$$
$$= \left[\exp^N(\omega)\sum_{j=k}^{\ell_\beta}\omega^j(a_j+b_j)\right]^+.$$

Now, suppose that $\ell_\beta = 0$ (this also covers the induction basis). Then $[\exp^N(\omega)]^+ = \sup_i^+[\exp^N(i)]^+$, and like before, using the case $\alpha_0 > 0$, $\beta_0 > 0$,

$$[\alpha]^\mu \diamond [\beta]^+ = \sup_i^+ \alpha^\mu \diamond \left[\exp^N(\omega)\left(\sum_{j=k}^{\ell_\beta}\omega^j b'_j\right) + \exp^N(i)\right]^+$$
$$= \sup_i^+ \left(\alpha_0^\mu \diamond \left[\exp^N(i)\right]^+ \to \left[\exp^N(\omega)\sum_{j=k}^{0}\omega^j(a_j+b'_j)\right]^\pm\right)$$
$$= \left[\exp^N(\omega)\right]^+ \to \left[\exp^N(\omega)\left(\sum_{j=k}^{0}\omega^j(a_j+b'_j)+\right)\right]^\pm$$
$$= \left[\exp^N(\omega)\sum_{j=k}^{\ell_\beta}\omega^j(a_j+b_j)\right]^+.$$

Let us move to the second equation. Let $\alpha = \alpha' + \exp^N(p)$, $\beta = \beta' + \exp^N(\omega)\omega^{\ell_\beta}$. If $\ell_\beta > 0$, we have

$$[\alpha]^\mu \diamond [\beta]^- = [\exp^N(p)]^\mu \to \Big(([\alpha']^\pm \diamond [\beta]^-) \vee (\sup_n^- [\alpha]^\mu \diamond [\beta' + \exp^N(\omega)\omega^{\ell_\beta-1}n]^+) \vee$$
$$\vee (\sup_n^+ [\alpha]^\mu \diamond [\beta' + \exp^N(\omega)\omega^{\ell_\beta-1}n]^+)\Big).$$

As $[\alpha]^\mu \diamond [\beta' + \exp^N(\omega)\omega^{\ell_\beta-1}n]^+ = \left[\exp^N(\omega)\left(\left(\sum_{i=k}^{\ell_\beta}\omega^i(a_i+b'_i)\right) + \omega^{\ell_\beta-1}(a_{\ell_\beta-1}+n)\right)\right]^+$, we obtain

$$[\alpha]^\mu \diamond [\beta]^- = [\exp^N(p)]^\mu \to \left(([\alpha']^\pm \diamond [\beta]^-) \vee \left[\exp^N(\omega)\sum_{i=k}^{\ell_\beta}\omega^i(a_i+b_i)\right]^\pm\right).$$

If $\alpha' = \exp^N(\omega)\gamma + \delta$, with $[\delta]^\pm \in \hat{\Phi}_N$, $\delta > 0$, then we conclude from the induction hypothesis that

$$\begin{aligned}[][\alpha]^\mu \diamond [\beta]^- &= [\exp^N(p)]^\mu \to \left(\left([\delta]^\pm \to [\theta]^\pm\right) \vee \left[\exp^N(\omega)\sum_{i=k}^{\ell_\beta}\omega^i(a_i+b_i)\right]^\pm\right)\\ &= [\exp^N(p)]^\mu \to \left([\delta]^\pm \to [\theta]^\pm\right)\\ &= \left([\exp^N(p)]^\mu \to [\delta]^\pm\right) \to [\theta]^\pm\\ &= [\alpha_0]^\mu \to [\theta]^\pm .\end{aligned}$$

If $\alpha' = \exp^N(\omega)\gamma$, by induction hypothesis $[\alpha']^\pm \diamond [\beta]^- = [\theta]^- \vee \left[\exp^N(\omega)\sum_{i=k}^{\ell_\alpha}\omega^i(a_i+b_i)\right]^+$, so

$$\begin{aligned}[][\alpha]^\mu \diamond [\beta]^- &= [\exp^N(p)]^\mu \to \left([\theta]^- \vee \left[\exp^N(\omega)\sum_{i=k}^{\ell_\alpha}\omega^i(a_i+b_i)\right]^+ \vee \left[\exp^N(\omega)\sum_{i=k}^{\ell_\beta}\omega^i(a_i+b_i)\right]^\pm\right)\\ &= [\exp^N(p)]^\mu \to [\theta]^\pm\end{aligned}$$

as one of the last two disjuncts must be at least $[\theta]^+$. This concludes the case of $\ell_\beta > 0$.
If $\ell_\beta = 0$ we have

$$\begin{aligned}[][\alpha]^\mu \diamond [\beta]^- = [\exp^N(p)]^\mu \to \Big(&([\alpha']^\pm \diamond [\beta]^-) \vee (\overset{-}{\sup_n}\,[\alpha]^\mu \diamond [\beta' + \exp^N(n)]^+) \vee\\ &\vee(\overset{+}{\sup_n}\,[\alpha]^\mu \diamond [\beta' + \exp^N(n)]^+)\Big).\end{aligned}$$

Since $[\alpha]^\mu \diamond [\beta' + \exp^N(n)]^+ = [\alpha_0]^\mu \diamond [\exp^N(n)]^+ \to \left[\exp^N(\omega)\sum_{i=k}^{0}\omega^i(a_i+b'_i)\right]^\pm$, by Lemma A.5 we have

$$[\alpha]^\mu \diamond [\beta]^- = [\exp^N(p)]^\mu \to \left(([\alpha']^\pm \diamond [\beta]^-) \vee [\theta]^\pm\right)$$

If $\alpha' = \exp^N(\omega)\gamma + \delta$, then $\alpha_0 = \delta + \exp^N(p)$ and by induction hypothesis for $[\alpha']^\pm \diamond [\beta]^-$ we get

$$\begin{aligned}[][\alpha]^\mu \diamond [\beta]^- &= [\exp^N(p)]^\mu \to (([\delta]^\pm \to [\theta]^\pm) \vee [\theta]^\pm)\\ &= [\exp^N(p)]^\mu \to [\delta]^\pm \to [\theta]^\pm)\\ &= [\alpha_0]^\mu \to [\theta]^\pm .\end{aligned}$$

If $\alpha' = \exp^N(\omega)\gamma$, then $\alpha_0 = \exp^N(p)$ and by induction hypothesis $[\alpha']^\pm \diamond [\beta]^-$ is at most $[\theta]^\pm$. Hence,

$$\begin{aligned}[][\alpha]^\mu \diamond [\beta]^- &= [\exp^N(p)]^\mu \to [\theta]^\pm\\ &= [\alpha_0]^\mu \to [\theta]^\pm .\end{aligned}$$

- **Case** $\alpha_0 = 0$**,** $\beta_0 = 0$**.** Let $\alpha = \alpha' + \exp^N(\omega)\omega^{\ell_\alpha}$, $\beta = \beta' + \exp^N(\omega)\omega^{\ell_\beta}$. Suppose that $\ell_\alpha > 0$, $\ell_\beta > 0$. Set $a_{\ell_\alpha}^{(n)} = a_{\ell_\alpha} - 1$, $a_{\ell_\alpha-1}^{(n)} = n$, and $a_i^{(n)} = a_i$ for other i's. Analogously define $b_i^{(n)}$. Applying Lemma A.1

and the inductive hypothesis we get the claim

$$
\begin{aligned}
&[\alpha]^- \diamond [\beta]^-) = \\
&= \bot \to \left(\overset{-}{\sup_n} \left([\alpha]^- \diamond [\beta' + \exp^N(\omega)\omega^{\ell_\beta - 1} n]^-\right) \vee \overset{-}{\sup_n} \left([\alpha' + \exp^N(\omega)\omega^{\ell_\alpha - 1} n]^- \diamond [\beta]^-\right)\right) \\
&= \bot \to \left(\overset{-}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i + b_i^{(n)}\right)\right]^- \vee \overset{-}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i^{(n)} + b_i\right)\right]^- \right) \\
&= \bot \to \left(\left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i (a_i + b_i)\right]^- \vee \left[\exp^N(\omega) \sum_{i=k}^{\ell_\alpha} \omega^i (a_i + b_i)\right]^- \right) \\
&= \bot \to \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i (a_i + b_i)\right]^- = \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i (a_i + b_i)\right]^-, \\
&[\alpha]^- \diamond [\beta]^+ = \\
&= \top \to \left(\overset{+}{\sup_n} \left([\alpha]^- \diamond [\beta' + \exp^N(\omega)\omega^{\ell_\beta - 1} n]^-\right) \vee \overset{+}{\sup_n} \left([\alpha' + \exp^N(\omega)\omega^{\ell_\alpha - 1} n]^- \diamond [\beta]^+\right)\right) \\
&= \top \to \left(\overset{+}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i + b_i^{(n)}\right)\right]^- \vee \overset{+}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i^{(n)} + b_i\right)\right]^- \right) \\
&= \top \to \left(\left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i (a_i + b_i)\right]^+ \vee \left[\exp^N(\omega) \sum_{i=k}^{\max(\ell_\alpha, \ell_\beta)} \omega^i (a_i + b_i)\right]^+ \right) \\
&= \top \to \left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i (a_i + b_i)\right]^+ = \left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i (a_i + b_i)\right]^+, \\
&[\alpha]^+ \diamond [\beta]^+ = \\
&= \top \to \left(\overset{+}{\sup_n} \left([\alpha]^+ \diamond [\beta' + \exp^N(\omega)\omega^{\ell_\beta - 1} n]^-\right) \vee \overset{+}{\sup_n} \left([\alpha' + \exp^N(\omega)\omega^{\ell_\alpha - 1} n]^- \diamond [\beta]^+\right)\right) \\
&= \top \to \left(\overset{+}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{\ell_\alpha} \omega^i \left(a_i + b_i^{(n)}\right)\right]^- \vee \overset{+}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i^{(n)} + b_i\right)\right]^- \right) \\
&= \top \to \left(\left[\exp^N(\omega) \sum_{i=k}^{\max(\ell_\alpha, \ell_\beta)} \omega^i (a_i + b_i)\right]^+ \vee \left[\exp^N(\omega) \sum_{i=k}^{\max(\ell_\alpha, \ell_\beta)} \omega^i (a_i + b_i)\right]^+ \right) \\
&= \top \to \left[\exp^N(\omega) \sum_{i=k}^{\max(\ell_\alpha, \ell_\beta)} \omega^i (a_i + b_i)\right]^+ = \left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i (a_i + b_i)\right]^+.
\end{aligned}
$$

Next suppose that $\ell_\alpha = 0$, $\ell_\beta > 0$. Similarly,

$$
\begin{aligned}
&[\alpha]^- \diamond [\beta]^- = \\
&= \bot \to \left(\overset{-}{\sup_n} \left([\alpha]^- \diamond [\beta' + \exp^N(\omega)\omega^{\ell_\beta - 1} n]^- \right) \vee \overset{-}{\sup_n} \left([\alpha' + \exp^N(n)]^- \diamond [\beta]^- \right) \right) \\
&= \bot \to \left(\overset{-}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i + b_i^{(n)} \right) \right]^- \vee \overset{-}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i' + b_i \right) + \exp^N(n) \right]^- \right) \\
&= \bot \to \left(\left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i + b_i \right) \right]^- \vee \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i + b_i \right) \right]^- \right) \\
&= \bot \to \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i + b_i \right) \right]^- = \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i + b_i \right) \right]^-, \\
&[\alpha]^- \diamond [\beta]^+ = \\
&= \top \to \left(\overset{+}{\sup_n} \left([\alpha]^- \diamond [\beta' + \exp^N(\omega)\omega^{\ell_\beta - 1} n]^- \right) \vee \overset{+}{\sup_n} \left([\alpha' + \exp^N(n)]^- \diamond [\beta]^+ \right) \right) \\
&= \top \to \left(\overset{+}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i + b_i^{(n)} \right) \right]^- \vee \overset{+}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i' + b_i \right) \right]^- \right) \\
&= \top \to \left(\left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i + b_i \right) \right]^+ \vee \left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i' + b_i \right) \right]^+ \right) \\
&= \top \to \left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i + b_i \right) \right]^+ = \left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i + b_i \right) \right]^+, \\
&[\alpha]^+ \diamond [\beta]^- = \\
&= \top \to \left(\overset{+}{\sup_n} \left([\alpha]^+ \diamond [\beta' + \exp^N(\omega)\omega^{\ell_\beta - 1} n]^- \right) \vee \overset{+}{\sup_n} \left([\alpha' + \exp^N(n)]^- \diamond [\beta]^- \right) \right) \\
&= \top \to \left(\overset{+}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{\ell_\alpha} \omega^i \left(a_i + b_i^{(n)} \right) \right]^- \vee \overset{+}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i' + b_i \right) + \exp^N(n) \right]^- \right) \\
&= \top \to \left(\left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i + b_i \right) \right]^+ \vee \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i + b_i \right) \right]^+ \right) \\
&= \top \to \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i + b_i \right) \right]^+ = \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i + b_i \right) \right]^+ = \left[\exp^N(\omega) \sum_{i=k}^{\ell_\alpha} \omega^i \left(a_i + b_i \right) \right]^+,
\end{aligned}
$$

$$\begin{aligned}
&[\alpha]^+ \diamond [\beta]^+ = \\
&= \top \to \left(\overset{+}{\sup_n} \left([\alpha]^+ \diamond [\beta' + \exp^N(\omega)\omega^{\ell_\beta - 1} n]^- \right) \vee \overset{+}{\sup_n} \left([\alpha' + \exp^N(n)]^- \diamond [\beta]^+ \right) \right) \\
&= \top \to \left(\overset{+}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{\ell_\alpha} \omega^i \left(a_i + b_i^{(n)} \right) \right]^- \vee \overset{+}{\sup_n} \left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i' + b_i \right) \right]^- \right) \\
&= \top \to \left(\left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i + b_i \right) \right]^+ \vee \left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i' + b_i \right) \right]^+ \right) \\
&= \top \to \left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i + b_i \right) \right]^+ = \left[\exp^N(\omega) \sum_{i=k}^{\ell_\beta} \omega^i \left(a_i + b_i \right) \right]^+ .
\end{aligned}$$

Finally, for $\ell_\alpha = 0$, $\ell_\beta = 0$

$$\begin{aligned}
&[\alpha]^- \diamond [\beta]^- = \\
&= \bot \to \left(\overset{-}{\sup_n} \left([\alpha]^- \diamond [\beta' + \exp^N(n)]^- \right) \vee \overset{-}{\sup_n} \left([\alpha' + \exp^N(n)]^- \diamond [\beta]^- \right) \right) \\
&= \bot \to \left([\theta]^- \vee [\theta]^- \right) = \bot \to [\theta]^- = [\theta]^-,
\end{aligned}$$

$$\begin{aligned}
&[\alpha]^- \diamond [\beta]^+ = \\
&= \top \to \left(\overset{+}{\sup_n} \left([\alpha]^- \diamond [\beta' + \exp^N(n)]^- \right) \vee \overset{+}{\sup_n} \left([\alpha' + \exp^N(n)]^- \diamond [\beta]^+ \right) \right) \\
&= \top \to \left([\theta]^+ \vee \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i' + b_i \right) \right]^+ \right) = \top \to [\theta]^+ = [\theta]^+,
\end{aligned}$$

$$\begin{aligned}
&[\alpha]^+ \diamond [\beta]^+ = \\
&= \top \to \left(\overset{+}{\sup_n} \left([\alpha]^+ \diamond [\beta' + \exp^N(n)]^- \right) \vee \overset{+}{\sup_n} \left([\alpha' + \exp^N(n)]^- \diamond [\beta]^+ \right) \right) \\
&= \top \to \left(\left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i + b_i' \right) \right]^+ \vee \left[\exp^N(\omega) \sum_{i=k}^{0} \omega^i \left(a_i' + b_i \right) \right]^+ \right),
\end{aligned}$$

and since $\sum_{i=k}^{0} \omega^i (a_i + b_i') = \sum_{i=k}^{0} \omega^i (a_i' + b_i) = \left(\sum_{i=k}^{1} \omega^i (a_i + b_i) \right) + (a_0 + b_0 - 1)$, the claim follows. ☐
Note that $[0]^\mu$ is not a valid set. Nevertheless, to simplify notation we adopt a convention $[\alpha]^\mu \diamond [0]^\nu = [\alpha]^\mu$.

Lemma A.7. *Let* $[\alpha]^\mu, [\beta]^\nu \in \Phi$ *and* $\alpha = \sum_{i=N}^{K} \alpha_i$, $\beta = \sum_{j=N}^{L} \beta_j$, $\alpha_i \in \Phi_i$, $\beta_j \in \Phi_j$, $\alpha_K, \beta_L > 0$, $\mu, \nu \in \{+, -\}$. *For* $J < \omega$ *let* $\alpha_{J<} = \sum_{i=N}^{J+1} \alpha_i$, $\beta_{J<} = \sum_{i=N}^{J+1} \beta_i$.

- *If $K < L$ and $[\beta_L]^\nu = [\exp^L(\omega)\gamma]^-$*

$$[\alpha]^\mu \diamond [\beta]^\nu = \begin{cases} \left[\sum_{i=L-1}^K \alpha_i\right]^\mu \to [\alpha_{L-1<}+1]^- \diamond [\beta]^\pm & \textit{for } \sum_{i=L-1}^K \alpha_i \geq \omega\,, \\ \left[\sum_{i=L-1}^K \alpha_i - 1\right]^\mu \to [\alpha_{L-1<}+1]^\pm \diamond [\beta]^\pm & \textit{for } \omega > \sum_{i=L-1}^K \alpha_i > 1\,, \\ [1]^+ \to [\alpha_{L-1<}]^\pm \diamond [\beta]^\nu & \textit{for } \left[\sum_{i=L-1}^K \alpha_i\right]^\mu = [1]^+\,, \\ [\alpha_L+1]^- \diamond [\beta_L]^\nu \to [\alpha_{L<}+1]^- \diamond [\beta_{L<}+1]^- & \textit{for } \left[\sum_{i=L-1}^K \alpha_i\right]^\mu = [1]^-\,. \end{cases}$$

- *If $K < L$ and $[\beta_L]^\nu \neq [\exp^L(\omega)\gamma]^-$*

$$[\alpha]^\mu \diamond [\beta]^\nu = \begin{cases} [\alpha_{L-1<}+1]^- \diamond [\beta]^\nu & \textit{for } \sum_{i=L-1}^K \alpha_i > 1\,, \\ [1]^+ \to [\alpha_{L-1<}]^\pm \diamond [\beta]^\nu & \textit{for } \left[\sum_{i=L-1}^K \alpha_i\right]^\mu = [1]^+\,, \\ [\alpha_L+1]^- \diamond [\beta_L]^\nu \to [\alpha_{L<}+1]^- \diamond [\beta_{L<}+1]^- & \textit{for } \left[\sum_{i=L-1}^K \alpha_i\right]^\mu = [1]^-\,. \end{cases}$$

- *If $K = L$*

$$[\alpha]^\mu \diamond [\beta]^\nu = \begin{cases} [\alpha_L]^\mu \diamond [\beta_L]^\nu \to [\alpha_{L<}+1]^- \diamond [\beta_{L<}+1]^- & \textit{for } \alpha_L, \beta_L \geq \omega\,, \\ [\alpha_L]^\mu \diamond [\beta_L-1]^\nu \to [\alpha_{L<}+1]^- \diamond [\beta_{L<}+1]^\pm & \textit{for } \alpha_L \geq \omega > \beta_L > 1\,, \\ [\alpha_L-1]^\mu \diamond [\beta_L-1]^\nu \to [\alpha_{L<}+1]^\pm \diamond [\beta_{L<}+1]^\pm & \textit{for } \omega > \alpha_L \cdot \beta_L > 1\,, \\ [\alpha_M+1]^\mu \diamond [\beta_M+1]^\nu \to [\alpha_{M<}+1]^- \diamond [\beta_{M<}+1]^- & \textit{for } \alpha_L \cdot \beta_L = 1, \end{cases}$$

where M is the least $i > 1$ such that α_i or β_i is nonzero.

Proof. Again, we prove all the equations simultaneously by induction on (α, β). It is easy to observe that the inequality "$\geq$" is in each case very simple. We concentrate on "$\leq$". Further more, since in all three cases the arguments are very similar we only give the calculations for the first case: $K < L$ and $[\beta_L]^\nu = [\exp^L(\omega)\gamma]^-$. Suppose first that $\alpha_K = \alpha'_K + \exp^K(\omega)\omega^p$, $K > 0$. Let $\alpha' = \alpha_{K<} + \alpha'_K$, $\beta' = \beta_{K<} + \beta'_K$. We only give the calculation for $p, q > 0$ (if one or both are zero the calulation is entirely analogous):

$$\begin{aligned} [\alpha]^\mu \diamond [\beta]^\nu &= 1^\mu \to (\overset{\mu}{\sup_m}[\alpha' + \exp^K(\omega)\omega^{p-1}m]^- \diamond [\beta]^-) \vee (\overset{\mu}{\sup_n}\,[\alpha]^\mu \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^+) \\ &= 1^\mu \to \left(\overset{\mu}{\sup_m}\left[\left(\sum_{i=L-1}^{K+1} \alpha_i\right) + \alpha'_K + \exp^K(\omega)\omega^{p-1}m\right]^- \to [\alpha_{L-1<}+1]^- \diamond [\beta]^\pm\right) \vee \\ &\quad \vee (\overset{\mu}{\sup_n}\,[\alpha_{L-1<}+1]^- \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^+) \\ &\leq 1^\mu \to \left(\left[\sum_{i=L-1}^K \alpha_i\right]^\mu \to [\alpha_{L-1<}+1]^- \diamond [\beta]^\pm\right) \vee ([\alpha_{L-1<}+1]^- \diamond [\beta]^\mu) \\ &= \left[\sum_{i=L-1}^K \alpha_i\right]^\mu \to [\alpha_{L-1<}+1]^- \diamond [\beta]^\pm. \end{aligned}$$

Suppose now that $\alpha_K = \alpha'_K + \exp^K(p)$, $K > 0$. For $q > 0$

$$\begin{aligned}
[\alpha]^\mu \diamond [\beta]^\nu &= [\exp^K(p)]^\mu \to [\alpha']^\pm \diamond [\beta]^- \vee \sup_n^+ [\alpha]^\mu \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^+) \vee \\
&\quad \vee \sup_n^- [\alpha]^\mu \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^+) \\
&\le [\exp^K(p)]^\mu \to \left(\left[\left(\sum_{i=L-1}^{K+1} \alpha_i\right) + \alpha'_K\right]^\pm \to [\alpha_{L-1<} + 1]^- \diamond [\beta]^\pm\right) \vee \\
&\quad \vee ([\alpha_{L-1<} + 1]^- \diamond [\beta]^+) \vee ([\alpha_{L-1<} + 1]^- \diamond [\beta]^-) \\
&\le \left[\sum_{i=L-1}^{K} \alpha_i\right]^\mu \to [\alpha_{L-1<} + 1]^- \diamond [\beta]^\pm ,
\end{aligned}$$

and for $q = 0$ the computation is analogous.
Finally suppose that $K = 0$. For $\alpha_K = \alpha'_K + \omega^p$ and $\alpha_K = \alpha'_K + 1$ the calculations are entirely analogous to the ones above save for three cases we consider below. For $\sum_{i=L-1}^K \alpha_i = \omega$

$$\begin{aligned}
[\alpha]^\mu \diamond [\beta]^\nu &= 1^\mu \to (\sup_m^\mu [\alpha' + m]^- \diamond [\beta]^-) \vee (\sup_n^\mu [\alpha]^\mu \diamond [\beta'_L + \exp^L(\omega)\omega^{q-1}n]^+) \\
&\le 1^\mu \to \left(\sup_m^\mu [m-1]^- \to [\alpha_{L-1<} + 1]^\pm \diamond [\beta]^\pm\right) \vee ([\alpha_{L-1<} + 1]^- \diamond [\beta]^\mu) \\
&\le 1^\mu \to \left(\sup_m^\mu [m-1]^- \to 1^+ \to [\alpha_{L-1<} + 1]^- \diamond [\beta]^\pm\right) \vee ([\alpha_{L-1<} + 1]^- \diamond [\beta]^\mu) \\
&\le [\omega]^\mu \to [\alpha_{L-1<} + 1]^- \diamond [\beta]^\pm,
\end{aligned}$$

since $[\alpha_{L-1<} + 1]^+ \diamond [\beta]^\pm \le \top \to [\alpha_{L-1<} + 1]^- \diamond [\beta]^- \le \top \to [\alpha_{L-1<} + 1]^- \diamond [\beta]^\pm$.
For $\sum_{i=L-1}^K \alpha_i = 2$

$$\begin{aligned}
[\alpha]^\mu \diamond [\beta]^\nu &\le [1]^\mu \to [\alpha']^\pm \diamond [\beta]^- \vee \sup_n^\mu [\alpha]^\mu \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^+) \\
&\le [1]^\mu \to ([\alpha_{L-1<} + 1]^\pm \diamond [\beta]^\mu) \vee \\
&\quad \vee \sup_n^\mu \big([1]^\mu \to [\alpha_{L-1<} + 1]^\pm \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^\pm\big) \\
&\le [1]^\mu \to ([\alpha_{L-1<} + 1]^\pm \diamond [\beta]^\mu) \vee \sup_n^\mu \big([1]^\mu \to [\alpha_{L-1<} + 1]^\pm \diamond [\beta]^\nu\big) \\
&\le [1]^\mu \to ([\alpha_{L-1<} + 1]^\pm \diamond [\beta]^\mu) \vee \big([1]^\mu \to [1]^\mu \to [\alpha_{L-1<} + 1]^\pm \diamond [\beta]^\nu\big) \\
&= [1]^\mu \to [\alpha_{L-1<} + 1]^\pm \diamond [\beta]^\pm .
\end{aligned}$$

For $\left[\sum_{i=L-1}^K \alpha_i\right]^\mu = [1]^+$

$$\begin{aligned}
[\alpha]^\mu \diamond [\beta]^\nu &= [1]^+ \to [\alpha_{L-1<}]^\pm \diamond [\beta]^- \vee (\sup_n^+ [\alpha_{L-1<} + 1]^+ \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^-) \\
&\le [1]^\mu \to [\alpha_{L-1<}]^\pm \diamond [\beta]^- \vee (\sup_n^+ [1]^+ \to [\alpha_{L-1<}]^\pm \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^-) \\
&\le [1]^\mu \to [\alpha_{L-1<}]^\pm \diamond [\beta]^- \vee (\sup_n^+ [1]^+ \to [\alpha_{L-1<}]^\pm \diamond [\beta]^-) \\
&= [1]^\mu \to [\alpha_{L-1<}]^\pm \diamond [\beta]^- \vee ([1]^+ \to [1]^+ \to [\alpha_{L-1<}]^\pm \diamond [\beta]^-) \\
&= [1]^\mu \to [\alpha_{L-1<}]^\pm \diamond [\beta]^-
\end{aligned}$$

For $\left[\sum_{i=L-1}^{K} \alpha_i\right]^\mu = [1]^-$

$$\begin{aligned}[\alpha]^\mu \diamond [\beta]^\nu &= [1]^- \to [\alpha_{L-1<}]^\pm \diamond [\beta]^- \vee (\overset{-}{\sup_n} [\alpha_{L-1<}+1]^- \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^-) \\ &\leq [1]^- \to [\alpha_{L-1<}]^\pm \diamond [\beta]^- \vee \\ &\quad \vee (\overset{-}{\sup_n} [\alpha_L+1]^- \diamond [\beta'_L + \exp^L(\omega)\omega^{q-1}n]^- \to [\alpha_{L<}+1]^- \diamond [\beta_{L<}+1]^-) \\ &\leq [1]^- \to [\alpha_{L-1<}]^\pm \diamond [\beta]^- \vee ([\alpha_L+1]^- \diamond [\beta_L]^- \to [\alpha_{L<}+1]^- \diamond [\beta_{L<}+1]^-)\end{aligned}$$

If $\alpha_L > 0$ we get $[\alpha_{L-1<}]^\pm \diamond [\beta]^- = [\alpha_L]^\pm \diamond [\beta_L]^- \to [\alpha_{L<}+1]^- \diamond [\beta_{L<}+1]^-$. If $\alpha_L = 0$, let N be the least $i > L$ such that $\alpha_i > 0$. Then we get $[\alpha_{L-1<}]^- \diamond [\beta]^- = [\alpha_{N-1<}]^- \diamond [\beta]^- \leq \left[\sum_{i=N-1}^{L} \beta_i\right]^- \to [\alpha_{N-1<}]^\pm \diamond [\beta_{N-1<}+1]^- = [\beta_L]^- \to [\alpha_{N-1<}]^\pm \diamond [\beta_{L<}]^- \leq [\beta_L]^- \to [\alpha_{L<}+1]^- \diamond [\beta_{L<}+1]^-$. In either case we have

$$\begin{aligned}[\alpha]^\mu \diamond [\beta]^\nu &\leq [1]^- \to ([\alpha_L+1]^- \diamond [\beta_L]^- \to [\alpha_{L<}+1]^- \diamond [\beta_{L<}+1]^-) \\ &\leq [\alpha_L+1]^- \diamond [\beta_L]^- \to [\alpha_{L<}+1]^- \diamond [\beta_{L<}+1]^- .\end{aligned}$$

Case $K < L$ and $[\beta_L]^\nu = [\beta'_L + \exp^L(\omega)\omega^q]^+$. Suppose first that $\alpha_K = \alpha'_K + \exp^K(\omega)\omega^p$. Let $\alpha' = \alpha_{K<} + \alpha'_K$, $\beta' = \beta_{K<} + \beta'_K$. We can treat $K > 0$ and $K = 0$ uniformly. We only give the calculation for $p, q > 0$ (if one or both are zero its entirely analogous):

$$\begin{aligned}[\alpha]^\mu \diamond [\beta]^\nu &= [1]^+ \to (\overset{+}{\sup_m}[\alpha' + \exp^K(\omega)\omega^{p-1}m]^+ \diamond [\beta]^+) \vee (\overset{+}{\sup_n} [\alpha]^\mu \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^+) \\ &= [1]^+ \to \left(\overset{+}{\sup_m}[\alpha_{L-1<}+1]^- \diamond [\beta]^+\right) \vee (\overset{+}{\sup_n} [\alpha_{L-1<}+1]^- \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^+) \\ &= [1]^+ \to \left([1]^+ \to [\alpha_{L-1<}+1]^- \diamond [\beta]^+\right) \vee ([\alpha_{L-1<}+1]^- \diamond [\beta]^+) \\ &= [1]^+ \to [\alpha_{L-1<}+1]^- \diamond [\beta]^+ = [\alpha_{L-1<}+1]^- \diamond [\beta]^+ .\end{aligned}$$

Suppose now that $\alpha_K = \alpha'_K + \exp^K(p)$ or $\alpha_K = \alpha'_K + p$ with $\sum_{i=L-1}^{K} \alpha_i > 1$. For $q > 0$

$$\begin{aligned}[\alpha]^\mu \diamond [\beta]^\nu &= \overset{+}{\sup_n} [\alpha]^\mu \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^+ \\ &= \overset{+}{\sup_n} [\alpha'+1]^- \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^+ \\ &= [\alpha_{L-1<}+1]^- \diamond [\beta]^+\end{aligned}$$

and for $q = 0$ the computation is analogous.
For $\left[\sum_{i=L-1}^{K} \alpha_i\right]^\mu = [1]^+$ we have

$$\begin{aligned}[\alpha]^\mu \diamond [\beta]^\nu &= \overset{+}{\sup_n} [\alpha]^\mu \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^+ \\ &= \overset{+}{\sup_n} [1]^+ \to [\alpha_{L-1<}]^\pm \diamond [\beta' + \exp^L(\omega)\omega^{q-1}n]^+ \\ &\leq \overset{+}{\sup_n} [1]^+ \to [\alpha_{L-1<}]^\pm \diamond [\beta]^+ \\ &= [1]^+ \to [1]^+ \to [\alpha_{L-1<}]^\pm \diamond [\beta]^+ \\ &\leq [1]^+ \to [\alpha_{L-1<}]^\pm \diamond [\beta]^+ .\end{aligned}$$

For $\left[\sum_{i=L-1}^{K}\alpha_i\right]^\mu=[1]^-$

$$\begin{aligned}[\alpha]^\mu\diamond[\beta]^\nu &= \overset{+}{\sup_n}\,[\alpha]^\mu\diamond[\beta'+\exp^L(\omega)\omega^{q-1}n]^+\\ &= \overset{+}{\sup_n}\,[\alpha_L+1]^-\diamond[\beta'_L+\exp^L(\omega)\omega^{q-1}n]^+\to[\alpha_{L<}+1]^-\diamond[\beta_{L<}+1]^-\\ &= [\alpha_L+1]^-\diamond[\beta_L]^+\to[\alpha_{L<}+1]^-\diamond[\beta_{L<}+1]^-\,.\end{aligned}$$

Case $K<L$ and $[\beta_L]^\nu=[\beta'_L+\exp^L(q)]^\nu$. Suppose first that $\alpha_K=\alpha'_K+\exp^K(\omega)\omega^p$. Let $\alpha'=\alpha_{K<}+\alpha'_K$, $\beta'=\beta_{K<}+\beta'_K$. We can treat $K>0$ and $K=0$ uniformly. We only give the calculation for $p>0$ (if $p=0$ it is entirely analogous). For $\mu=-$ we have

$$\begin{aligned}[\alpha]^\mu\diamond[\beta]^\nu &= [\exp^L(q)]^\nu\to[\alpha]^-\diamond[\beta']^\pm\vee(\overset{+}{\sup_m}[\alpha'+\exp^K(\omega)\omega^{p-1}m]^+\diamond[\beta]^\nu)\vee(\overset{-}{\sup_n}\,[\alpha'+\exp^K(\omega)\omega^{p-1}m]^+\diamond[\beta]^\nu)\\ &= [\exp^L(q)]^\nu\to[\alpha_{L-1<}+1]^-\diamond[\beta']^\pm\vee(\overset{+}{\sup_m}[\alpha_{L-1<}+1]^-\diamond[\beta]^\nu)\vee(\overset{-}{\sup_n}\,[\alpha_{L-1<}+1]^-\diamond[\beta]^\nu)\\ &= [\exp^L(q)]^\nu\to[\alpha_{L-1<}+1]^-\diamond[\beta']^\pm\vee([1]^+\to[\alpha_{L-1<}+1]^-\diamond[\beta]^\nu)\vee([1]^-\to[\alpha_{L-1<}+1]^-\diamond[\beta]^\nu)\\ &= [\exp^L(q)]^\nu\to[1]^\pm\to[\alpha_{L-1<}+1]^-\diamond[\beta]^\nu\\ &= [\alpha_{L-1<}+1]^-\diamond[\beta]^\nu\,,\end{aligned}$$

and for $\mu=+$

$$\begin{aligned}[\alpha]^\mu\diamond[\beta]^\nu &= \overset{+}{\sup_m}[\alpha'+\exp^K(\omega)\omega^{p-1}m]^+\diamond[\beta]^\nu\\ &= \overset{+}{\sup_m}[\alpha_{L-1<}+1]^-\diamond[\beta]^\nu\\ &= [1]^+\to[\alpha_{L-1<}+1]^-\diamond[\beta]^\nu\\ &= [\alpha_{L-1<}+1]^-\diamond[\beta]^\nu\,.\end{aligned}$$

Suppose now that $\alpha_K=\alpha'_K+\exp^K(p)$. We have

$$\begin{aligned}[\alpha]^\mu\diamond[\beta]^\nu &= [\exp^K(p)]^\mu\diamond[\exp^L(q)]^\nu\to[\alpha]^\mu\diamond[\beta']^\pm\vee[\alpha']^\pm\diamond[\beta]^\nu\\ &= [\exp^L(q)]^\nu\to[\alpha_{L-1<}+1]^-\diamond[\beta']^\pm\vee[\alpha_{L-1<}+1]^-\diamond[\beta]^\nu\\ &= [\exp^L(q)]^\nu\to[\alpha_{L-1<}+1]^-\diamond[\beta]^\nu\\ &= [\alpha_{L-1<}+1]^-\diamond[\beta]^\nu\,.\end{aligned}$$

For $\alpha_K=\alpha'_K+p$ with $\sum_{i=L-1}^{K}\alpha_i>1$ the computation is just like above. For $\left[\sum_{i=L-1}^{K}\alpha_i\right]^\mu=[1]^+$ we have

$$\begin{aligned}[\alpha]^\mu\diamond[\beta]^\nu &= [1]^+\diamond[\exp^L(q)]^\nu\to[\alpha]^\mu\diamond[\beta']^\pm\vee[\alpha_{L-1<}]^\pm\diamond[\beta]^\nu\\ &= [1]^+\to[\alpha_{L-1<}+1]^+\diamond[\beta']^\pm\vee[\alpha_{L-1<}]^\pm\diamond[\beta]^\nu\end{aligned}$$

If $\beta'_L>0$, we can transform this further as

$$\begin{aligned}&= [1]^+\to(([1]^+\to[\alpha_{L-1<}]^\pm\diamond[\beta']^\pm)\vee[\alpha_{L-1<}]^\pm\diamond[\beta]^\nu)\\ &= [1]^+\to(([\alpha_{L-1<}]^\pm\diamond[\beta']^\pm)\vee[\alpha_{L-1<}]^\pm\diamond[\beta]^\nu)\\ &= [1]^+\to([\alpha_{L-1<}]^\pm\diamond[\beta]^\nu)\end{aligned}$$

and we are done. Suppose $\beta'_L = 0$. Let M be the least $i > L$ such that $\beta_i > 0$. If $\beta_M = \beta'_M + \exp^M(\omega)\omega^r$, then we have

$$\begin{aligned}
&= [1]^+ \to ((\left[\sum_{i=M-1}^{L_1} \alpha_i + 1 \right]^+ \to [\alpha_{M-1<}]^\pm \diamond [\beta']^\pm) \vee [\alpha_{L-1<}]^\pm \diamond [\beta]^\nu) \\
&= [1]^+ \to (([1]^+ \to [\alpha_{L-1<}]^\pm \diamond [\beta']^\pm) \vee [\alpha_{L-1<}]^\pm \diamond [\beta]^\nu) \\
&= [1]^+ \to (([\alpha_{L-1<}]^\pm \diamond [\beta']^\pm) \vee [\alpha_{L-1<}]^\pm \diamond [\beta]^\nu) \\
&= [1]^+ \to ([\alpha_{L-1<}]^\pm \diamond [\beta]^\nu)\,.
\end{aligned}$$

If $\beta_M = \beta'_M + \exp^M(r)$, then we have

$$\begin{aligned}
&= [1]^+ \to (([\alpha_{M-1<}]^\pm \diamond [\beta']^\pm) \vee [\alpha_{L-1<}]^\pm \diamond [\beta]^\nu) \\
&= [1]^+ \to (([\alpha_{L-1<}]^\pm \diamond [\beta']^\pm) \vee [\alpha_{L-1<}]^\pm \diamond [\beta]^\nu) \\
&= [1]^+ \to ([\alpha_{L-1<}]^\pm \diamond [\beta]^\nu)\,.
\end{aligned}$$

For $\left[\sum_{i=L-1}^{K} \alpha_i\right]^\mu = [1]^-$

$$\begin{aligned}
[\alpha]^\mu \diamond [\beta]^\nu &= [1]^- \diamond [\exp^L(q)]^\nu \to [\alpha]^\mu \diamond [\beta']^\pm \vee [\alpha_{L-1<}]^\pm \diamond [\beta]^\nu \\
&= [\exp^L(q)]^\nu \to [\alpha_{L-1<} + 1]^- \diamond [\beta']^\pm \vee [\alpha_{L-1<}]^\pm \diamond [\beta]^\nu \\
&= [\beta_L]^\nu \to [\alpha_{L<} + 1]^- \diamond [\beta_{L<} + 1]^- \\
&= [\alpha_L + 1]^- \diamond [\beta_L]^\nu \to [\alpha_{L<} + 1]^- \diamond [\beta_{L<} + 1]^-
\end{aligned}$$

Case $K = L$. Using similar computations as above, follow the subcases of Lemma A.6. □

Lemma A.7 gives a recursive procedure to compute the result of $[\alpha]^\mu \diamond [\beta]^\nu$. Observe that in each case we reduce the problem to at most two instances: one on elements of (roughly speaking) the same Φ_N, the other on sets corresponding to ordinals such that either their respective K and L are smaller, or are obtained by cutting off a nontrivial (bigger than 1) tail part of the sum defining the ordinal, and replacing it with 1. The first subproblem is solved by Lemma A.6, and the second problem solved recursively.

What remains to be done, is put the results of the two subcomputations together. For this we need to prove that the class Φ is closed by operations of the form $[\sum_{i=L}^{0} \alpha_i]^\mu \to [\sum_{i=M}^{L+1} \alpha_i + j]^\nu$ with $j < \omega$, and $\alpha_i \in \Phi_i$.
Clearly, $[\sum_{i=L}^{0} \alpha_i]^\mu \to [\sum_{i=M}^{L+1} \alpha_i + j]^\nu = [\sum_{i=L}^{0} \alpha_i]^\mu \to [\sum_{i=M}^{L+1} \alpha_i]^\pm$, if $j > 0$.
If $\nu = \pm$, $[\sum_{i=L}^{0} \alpha_i]^\mu \to [\sum_{i=M}^{L+1} \alpha_i]^\nu = [\sum_{i=M}^{0} \alpha_i]^\mu$. If $\nu = +$ or $\nu = -$, the result depends on the form of α_L (w.l.o.g we may suppose that it is nonzero).
If $\alpha_L = \alpha'_L + \exp^L(\omega)\omega^p$, then the result is $[\sum_{i=M}^{0} \alpha_i]^\mu$ as well. Otherwise, it is $[\sum_{i=M}^{L} \alpha_i]^\nu$.
This concludes the proof of the effective closure by $\diamond$.

A.4 Closure by $\sup^+$ and $\sup^-$

We show that for $[\alpha]^\mu, [\beta]^\nu \in \Phi$, we can compute $\sup_i^+(([\alpha]^\mu)^{\langle i\rangle} \diamond [\beta]^\nu)$ and that the result is in Φ. The case $\sup_i^-(([\alpha]^\mu)^{\langle i\rangle} \diamond [\beta]^\nu)$ is solved in the same way.

Case $\alpha = \exp^N(\omega)\omega^p + \alpha'$. If $[\alpha]^\mu = [\exp^N(\omega)\omega^p]^+$, then $([\alpha]^\mu)^{\langle i\rangle} = [\alpha]^\mu$ and $\sup_i^+(([\alpha]^\mu)^{\langle i\rangle} \diamond [\beta]^\nu) = \sup_i^+([\alpha]^\mu \diamond [\beta]^\nu) = [1]^+ \to [\alpha]^\mu \diamond [\beta]^\nu$. We conclude by previously proved closure properties.
Otherwise, $[\alpha]^\mu \geq [\exp^N(\omega)\omega^p]^-$. But then, by the lemmas of the previous section, it follows easily that $([\alpha]^\mu)^{\langle i\rangle} \leq ([\exp^N(\omega)\omega^p]^-)^{\langle i+1\rangle} \leq [\exp^N(\omega)\omega^p(i+2)]^+$. Thus,

$$\sup_i^+(([\alpha]^\mu)^{\langle i\rangle} \diamond [\beta]^\nu) = \sup_i^+([\exp^N(\omega)\omega^p i]^- \diamond [\beta]^\nu) = \sup_i^+([\exp^N(\omega)\omega^p i]^+ \diamond [\beta]^\nu).$$

Let $\sum_{n=M}^{L} \beta_n$ be the usual presentation of β with $\beta_L > 0$. There are only three cases to consider, all the others being verified just by combining those three.

(1) Let $L > N$. If $\nu = +$, then $[\exp^N(\omega)\omega^p i]^+ \diamond [\beta]^\nu = [\beta]^\nu$ and $\sup_i^+(([\alpha]^\mu)^{\langle i \rangle} \diamond [\beta]^\nu) = [1]^+ \to [\beta]^\nu$. We conclude by previously proved closure property for $\to$.
For $\nu \neq +$, let $\beta_L = \exp^L(\omega)\eta + \sum_{j=1}^{k} \exp^L(j)k_j$, with $\eta < \omega^\omega$ and $k_j < \omega$. If $\sum_{j=1}^{k} \exp^L(j)k_j = 0$, then by the properties of $\diamond$ we obtain that $[\exp^N(\omega)\omega^p i]^+ \diamond [\beta]^\nu = [\beta + \exp^N(\omega)\omega^p i]^+$ and therefore
$$\sup_i^+(([\alpha]^\mu)^{\langle i \rangle} \diamond [\beta]^\nu) = \sup_i^+([\beta + \exp^N(\omega)\omega^p i]^+) = [\beta + \exp^N(\omega)\omega^{p+1}]^+.$$
Otherwise, since $[\exp^N(\omega)\omega^p i]^+ \diamond [\exp^L(p)]^\nu = [\exp^L(p)]^\nu$, $[\exp^N(\omega)\omega^p i]^+ \diamond [\beta]^\nu = [\beta]^\nu$ and $\sup_i^+(([\alpha]^\mu)^{\langle i \rangle} \diamond [\beta]^\nu) = [1]^+ \to [\beta]^\nu$. As before, we conclude by previously proved closure properties.

(2) If $M < N$, $[\exp^N(\omega)\omega^p i]^+ \diamond [\beta]^\nu = [\exp^N(\omega)\omega^p i]^+$, and $\sup_i^+(([\alpha]^\mu)^{\langle i \rangle} \diamond [\beta]^\nu) = [\exp^N(\omega)\omega^{p+1}]^+$.

(3) Suppose that $L = N$. Let $\beta_N = \exp^N(\omega)(\omega^{p+1}\eta + \omega^p j) + \beta'_N$ with $\eta < \omega^\omega$, $j < \omega$, and $\beta'_N < \exp^N(\omega)\omega^p$. Then, by applying the properties of the operation $\diamond$ given by Lemma A.6, we reason as follows.
If $\nu = +$, $\beta'_N = 0$, and $j = 0$, we get $[\exp^N(\omega)\omega^p i]^+ \diamond [\beta]^\nu = [\sum_{n=M}^{N+1} \beta_n + \exp^N(\omega)\eta\omega^p]^+$, and since i does not occur in the previous formula, we obtain that
$$\sup_i^+(([\alpha]^\mu)^{\langle i \rangle} \diamond [\beta]^+) = [\sum_{n=M}^{N+1} \beta_n + \exp^N(\omega)\omega^{p+1}\eta]^+.$$
Suppose $\nu \neq +$ or $\beta'_N > 0$. Under each of the two conditions we have
$$[\exp^N(\omega)\omega^p i]^+ \diamond [\beta]^\nu = [\sum_{n=M}^{N+1} \beta_n + \exp^N(\omega)(\omega^{p+1}\eta + \omega^p(i + j))]^+$$
and so
$$\sup_i^+(([\alpha]^\mu)^{\langle i \rangle} \diamond [\beta]^\nu) = [\sum_{n=M}^{N+1} \beta_n + \exp^N(\omega)\omega^{p+1}(\eta + 1)]^+.$$
Finally, if $\nu = +$, $\beta'_N = 0$, and $j > 0$, it holds that
$$[\exp^N(\omega)\omega^p i]^+ \diamond [\beta]^\nu = [\sum_{n=M}^{N+1} \beta_n + \exp^N(\omega)(\omega^{p+1}\eta + \omega^p(i + j - 1))]^+$$
and we conclude like before.

Case $\alpha = \exp^N(p) + \alpha'$ or $\alpha < \omega$. The case $\alpha < \omega$ is solved by similar arguments as the previous case. So, let $\alpha = \exp^N(p) + \alpha'$. If $\alpha = \exp^N(1)$ and $\mu \in \{+, -\}$, we get $([\alpha]^\mu)^{\langle i \rangle} = [\alpha]^\mu$, and the claim follows. Otherwise, $[\alpha] \geq_W [\exp^N(1)]^\pm$. Like before, by the properties of $\diamond$, we have
$$\sup_i^+(([\alpha]^\mu)^{\langle i \rangle} \diamond [\beta]^\nu) = \sup_i^+(([\exp^N(p)]^\mu)^{\langle i \rangle} \diamond [\beta]^\nu) = \sup_i^+([\exp^N(i)]^+ \diamond [\beta]^\nu).$$
If $[\beta]^\nu = [1]^+$, then $\sup_i^+(([\alpha]^\mu)^{\langle i \rangle} \diamond [\beta]^\nu) = [1]^+$. Let $\sum_{n=M}^{L} \beta_n$ be the usual presentation of β with $[\beta_L]^\nu \geq_W [1]^-$. Again, there are only three cases to consider, all the others being verified just by combining those three.

(1) If $M < N$, $[\exp^N(i)]^+ \diamond [\beta]^\nu = [\exp^N(i)]^+$, and $\sup_i^+(([\alpha]^\mu)^{\langle i \rangle} \diamond [\beta]^\nu) = [\exp^N(\omega)]^+$.

(2) Let $L > N$. If $\nu = +$, then $[\exp^N(i)]^+ \diamond [\beta]^\nu = [\beta]^\nu$ and $\sup_i^+(([\alpha]^\mu)^{\langle i\rangle} \diamond [\beta]^\nu) = [1]^+ \to [\beta]^\nu$. We conclude by previously proved closure property for $\to$.
For $\nu \neq +$, let $\beta_L = \exp^L(\omega)\eta + \sum_{j=1}^k \exp^L(j)k_j$, with $\eta < \omega^\omega$ and $k_j < \omega$. If $\sum_{j=1}^k \exp^L(j)k_j = 0$, then $[\exp^N(i)]^+ \diamond [\beta]^\nu = [\beta + \exp^N(i)]^+$ and therefore

$$\overset{+}{\underset{i}{\sup}}(([\alpha]^\mu)^{\langle i\rangle} \diamond [\beta]^\nu) = \overset{+}{\underset{i}{\sup}}([\beta + \exp^N(i)]^+) = [\beta + \exp^N(\omega)]^+.$$

Otherwise, since $[\exp^N(i)]^+ \diamond [\exp^L(p)]^\nu = [\exp^L(p)]^\nu$, $[\exp^N(i)]^+ \diamond [\beta]^\nu = [\beta]^\nu$ and $\sup_i^+(([\alpha]^\mu)^{\langle i\rangle} \diamond [\beta]^\nu) = [1]^+ \to [\beta]^\nu$. As before, we conclude by previously proved closure properties.

(3) Suppose that $L = N$. Let $\beta_N = \exp^N(\omega)\eta + \sum_{j=1}^k \exp^N(j)k_j$ with $\eta < \omega^\omega$, $k_j < \omega$. Then, by applying the properties of the operation $\diamond$, we reason as follows.
If $\nu = +$ and $\sum_{j=1}^k \exp^N(j)k_j = 0$, we get $[\exp^N(i)]^+ \diamond [\beta]^\nu = [\beta]^\nu$, and we obtain that $\sup_i^+(([\alpha]^\mu)^{\langle i\rangle} \diamond [\beta]^\nu) = [1]^+ \to [\beta]^\nu$.
Suppose $\nu \neq +$ or $\sum_{j=1}^k \exp^N(j)k_j > 0$. Under each of the two conditions we have

$$[\exp^N(i)]^+ \diamond [\beta]^\nu \leq_W [\exp^N(\omega)\eta + \exp^N(p)]^+,$$

for a certain $0 < p < \omega$ and so

$$\overset{+}{\underset{i}{\sup}}(([\alpha]^\mu)^{\langle i\rangle} \diamond [\beta]^\nu) = [\exp^N(\omega)\eta + \exp^N(\omega)]^+ = [\exp^N(\omega)(\eta + 1)]^+.$$

Bibliography

[1] ALBERUCCI L.: Strictness of the Modal μ-Calculus Hierarchy. In [57]: 185–201 (2002)

[2] ALBERUCCI L.: A Syntactical Treatment of Simultaneous Fixpoints in the Modal μ-Calculus. Technical Report, University of Bern (2008)

[3] ALBERUCCI L., FACCHINI A.: The Modal μ-Calculus Hierarchy on Restricted Classes of Transition Systems. *The Journal of Symbolic Logic* 74(4): 1367–1400 (2009)

[4] ALBERUCCI L., FACCHINI A.: On Modal μ-Calculus and Gödel-Löb Logic. *Studia Logica* 91: 145–169 (2009)

[5] ALBERUCCI L., KRÄHENBÜHL J.: Justifying Induction on Modal μ-Formulae. Technical Report, University of Bern (2009)

[6] ANDRETTA A.: Equivalence between Wadge and Lipschitz Determinacy. *Ann. Pure Appl. Logic* 123(1-3):163–192 (2003)

[7] ARNOLD A.: The μ-Calculus Alternation-Depth Hierarchy is Strict on Binary Trees. *ITA* 33(4/5): 329-340 (1999)

[8] ARNOLD A., NIWINSKI D. : Fixed Point Characterization of Weak Monadic Logic Definable Sets of Trees. In *Tree Automata and Languages*, Elsevier: 159–188 (1992)

[9] ARNOLD A., NIWINSKI D. : *Rudiments of μ-Calculus*. Studies in Logic vol. 146, Elsevier (2001)

[10] ARNOLD A., NIWINSKI D. : Continuous Separation of Game Languages. *Fundamenta Informaticae* 81(1-3): 19–28 (2008)

[11] ARNOLD A., SANTOCANALE L. : Ambiguous Classes in μ-Calculi Hierarchies. *Theoretical Computer Science* 333 (1-2): 265–296 (2005)

[12] VAN BENTHEM, J.: *Modal Correspondence Theory*. PhD thesis, Mathematisch Instituut & Instituut voor Grondslagenonderzoek, University of Amsterdam (1976)

[13] VAN BENTHEM, J.: Modal Frame Correspondences and Fixed-Points. *Studia Logica* 83: 133–155 (2006)

[14] Bernardi, C.: The Uniqueness of the Fixed Point in Every Diagonalizable Algebra. *Studia Logica* 35: 335–343 (1976)

[15] Berwanger D.: Game Logic is Strong Enough for Parity Games. *Studia Logica* 75: 205–219 (2003)

[16] Berwanger D., Lenzi G.: The Variable Hierarchy of the μ-Calculus is Strict. *STACS '05*: 97–109 (2005)

[17] Berwanger D., Grädel E., Lenzi G.: On the Variable Hierarchy of the Modal μ-Calculus. *CSL '02*: 352–366 (2002)

[18] Berwanger D., Grädel E., Lenzi G.: The Variable Hierarchy of the μ-Calculus is Strict. *Theory of Computing Systems* 40(4): 437–466 (2007)

[19] Bojańczyk M.: A Bounding Quantifier. *CSL '04*: 41–55 (2004)

[20] Bojańczyk M.: Two-Way Unary Temporal Logic over Trees. *LICS '07*: 121–130 (2007)

[21] Bojańczyk M.: Weak MSO with the Unbounding Quantifier. *STACS '09*: 159–170 (2009)

[22] Bojańczyk M.: Beyond ω-Regular Languages. *STACS '10*: 11–16 (2010)

[23] Bojańczyk M., Colcombet T. : Bounds in ω-Regularity. *LICS '06*: 285–296 (2006)

[24] Bojańczyk M., Idziaszek T. : Algebra for Infinite Forests with an Application to the Temporal Logic EF. *CONCUR '09*: 131–145 (2009)

[25] Bojańczyk M., Toruńczyk S. : Deterministic Automata and Extensions of Weak MSO. *FSTTCS '09*: 73–84 (2009)

[26] Bojańczyk M., Segoufin L., Straubing H.: Piecewise Testable Tree Languages. *LICS '08*: 442–451 (2008)

[27] Bojańczyk M., Segoufin L.: Tree Languages Defined in First-Order Logic With One Quantifier Alternation. *ICALP '08*: 233–245 (2008)

[28] Bojańczyk M., Walukiewicz I. : Characterizing EF and EX Tree Logics. *Theor. Comp. Sci.* 358 (2-3): 255–272 (2006)

[29] Bojańczyk M., Walukiewicz I. : Forest Algebras. In: *Automata and Logic: History and Perspectives*: 107–132, Amsterdam University Press (2007)

[30] Boolos G.: *The Logic of Provability.* Cambridge University Press (1993)

[31] Blackburn P., de Rijke M., Venema Y.: *Modal Logic.* Cambridge University Press (2001)

[32] Bradfield, J.: The Modal μ-Calculus Alternation Hierarchy is Strict. *Theoretical Computer Science* 195(2): 133–153 (1998)

[33] Bradfield, J.: Simplifying the Modal μ-Calculus Alternation Hierarchy. *STACS '98*: 39–49 (1998)

[34] Bradfield, J.: Fipoint Alternation: Arithmetic, Transition Systems, and the Binary Tree. *RAIRO Theoretical Informatics and Applications* 33: 341–356 (1999)

[35] Bradfield, J., Stirling C.: Modal Logic and Mu-Calculi. In: J. Bergstra, et al.: *Handbook of Process Algebra* Elsevier, North-Holland: 293–332 (2001)

[36] Büchi J.R.: Weak Second Order Arithmetic and Finite Automata. *Z. Math. Logic Grundlagen Math.* 6: 66–92 (1960)

[37] Cabessa, J.: *A Game Theoretical Approach to the Algebraic Counterpart of the Wagner Hierarchy.* PhD Thesis, University of Paris 7 and University of Lausanne (2007)

[38] Cabessa, J., Duparc J., Facchini A., Murlak F. : The Wadge Hierarchy of Max-Regular Languages. *FSTTCS '09*: 121–132 (2009)

[39] ten Cate B., Fontaine G., Litak T.: Some Modal Aspects of XPath. *Methods for Modalities* (2007)

[40] ten Cate B., Facchini A.: Characterizations of EF on Infinite Trees. Unpublished (2010)

[41] Colcombet T., Löding C.: The Nesting-Depth of Disjunctive μ-calculus for Tree Languages and the Limitedness Problem. *CSL '08*: 416–430 (2008)

[42] Colcombet T., Löding C.: The Non-Deterministic Mostowski Hierarchy and Distance-Parity Automata. *ICALP '08*: 398–409 (2008)

[43] Courcelle B.: The Monadic Second-Order Logic of Graphs XIV: Uniformly Sparse Graphs and Edge Set Quantification. *Theor. Comp. Sci.* 149 (1995)

[44] D'Agostino G., Lenzi G.: On the μ-Calculus over Transitive and Finite Transitive Frames. Submitted (2009)

[45] D'Agostino G., Lenzi G.: On the μ-Calculus over Finite Graphs with Bounded Strongly Connected Components. *GANDALF '10*: 55–71 (2010)

[46] Dawar A., Otto M.: Modal Characterisation Theorems over Special Classes of Frames. *Ann. Pure Appl. Logic* 161(1): 1–42 (2009)

[47] Duparc J. : *The Normal Form of Borel Sets of Finite Rank.* PhD Thesis (in french), University of Paris 7 (1995)

[48] Duparc J. : Wadge Hierarchy and Veblen Hierarchy Part 1: Borel Sets of Finite Rank. *Journal of Symbolic Logic* 66(1): 56-86 (2001)

[49] Duparc, J.: A Hierarchy of Deterministic Context-Free ω-languages. *Theoretical Computer Science* 290: 1253–1300 (2003)

[50] Duparc J., Facchini A., Murlak F. : Linear Game Automata, Decidable Hierarchy Problems for Stripped-Down Alternating Tree Automata. *CSL '09*: 186–195 (2009)

[51] Duparc J., Murlak F. : On the Topological Complexity of Weakly Recognizable Tree Languages. *FCT '07*: 261-273 (2007)

[52] Ebbinghaus H.-D., Flum J.: *Finite Model Theory*. 2nd Edition, Springer (2005)

[53] Emerson E.A., Jutla C.S.: Tree Automata, μ-Calculus and Determinacy (Extended Abstract). *FOCS '91*: 368–377 (1991)

[54] Emerson E.A., Streett R.S.: An Automata Theoretic Decision Procedure for the Propositional Mu-Calculus. *Information and Computation* 81(3): 249–264 (1989)

[55] Finkel O.: Borel Ranks and Wadge Degrees of ω-Context Free Languages. *Mathematical Structures in Computer Science* 16: 813–840 (2006)

[56] Finkel O., Simonnet P.: On Recognizable Tree Languages Beyond the Borel Hierarchy. *Fundamenta Informaticae* 95(2-3): 287–303 (2009)

[57] Grädel E., Thomas W., Wilke T. (Eds.): *Automata, Logics, and Infinite Games*. LNCS 2500, Springer-Verlag Berlin Heidelberg (2002)

[58] Hashiguchi K.: Limitedness Theorem on Finite Automata with Distance Functions. *Journal of Computer and System Sciences*, 24: 33–244 (1982)

[59] Hashiguchi K.: Regular Languages of Star Height One. *Inform. and Control* 53: 199–210 (1982)

[60] Hughes G.E., Cresswell M.J.: *A New Introduction to Modal Logic*. Routledge, London (1996)

[61] Hummel S., Michalewski H., Niwiński D.: On the Borel Inseparability of Game Tree Languages. *STACS '09*: 565–576 (2009).

[62] Hummel S., Skrzypczak M., Toruńczyk S.: On the Topological Complexity of $\mathsf{MSO} + \mathsf{U}$ and Related Automata Models. Accepted at *MFCS '10* (2010)

[63] Janin D., Lenzi G.: Relating Levels of the Mu-Calculus Hierarchy and Levels of the Monadic Hierarchy. *LICS '01*: 347–356 (2001)

[64] Janin D., Lenzi G.: On the Logical Definability of Topologically Closed Recognizable Languages of Infinite Trees. *Computing and Informatics* 21: 185–203 (2002)

[65] Janin D., Lenzi G.: On the Relationship Between Weak Monadic Second Order Logic on Arbitrary Trees, with Application to the μ-Calculus. *Fundamenta Informaticae* 61(3-4): 247–265 (2004)

[66] Janin D., Walukiewicz I.: Automata for the μ-Calculus and Related Results. *MFCS '95*: 552-562 (1996)

[67] Janin D., Walukiewicz I.: On the Expressive Completeness of the Propositional μ-Calculus with Respect to the Monadic Second Order Logic. *CONCUR '96*: 263-277 (1996)

[68] Jurdziński M.: Deciding the Winner in Parity Game is in UP $\cap$ co-UP. *Information Processing Letters*, 68(3): 119–124 (1998)

[69] Jurdziński M.: Small Progress Measures for Solving Parity Games. *STACS '00*: 290–301 (2000)

[70] Kechris A.: *Classical Descriptive Set Theory.* Springer (1995)

[71] Kirsten D.: Alternating Tree Automata and Parity Games. In [57]: 157–167 (2002)

[72] Kirsten D.: Distance Desert Automata and the Star Height One Problem. *FoSSaCS 2004*: 257–272 (2004)

[73] Kozen D.: Results on the Propositional μ-Calculus. *Theoretical Computer Science* 27: 333–354 (1983)

[74] Kozen D.: A Finite Model Theorem for the Propositional μ-Calculus. *Studia Logica* 47(3):233-241 (1988)

[75] Kupferman O., Safra S., Vardi M.: Relating Word and Tree Automata. *LICS '96*: 322–332 (1996)

[76] Kupferman O., Vardi M., Wolper P.: An Automata-Theoretic Approach to Branching-Time Model Checking. *Journal of the ACM* 47(2): 142–155 (1994)

[77] Kupferman O., Vardi M.: $\Pi_2 \cap \Sigma_2 =$ AFMC. *ICALP '03*: 697–713 (2003)

[78] Küsters R., Wilke T.: Deciding the First Level of the μ-Calculus Alternation Hierarchy. *FSTTCS '02*: 241–252 (2002)

[79] Landweber L. H.: Decision Problems for ω-Automata. *Math. Systems Theory* 3: 376–384 (1969)

[80] Lenzi G.: A Hierarchy Theorem for the μ-Calculus. *ICALP '96*: 87-97 (1996)

[81] Lenzi G.: The Transitive μ-Calculus is Büchi-definable. *WSEAS Transactions on Mathematics* 5(9): 1021–1026 (2006)

[82] Lenzi G.: Personal Communication, February 26th 2008.

[83] Libkin L.: *Elements of Finite Model Theory.* Springer (2004)

[84] McNaughton R., Papert S.: *Counter-Free Automata*. MIT Press, Cambridge (1971)

[85] Martin D. A.: Borel Determinacy. *Annals of Mathematics. Second Series* 102(2): 363–371 (1975)

[86] Mazala R.: Infinite Games. In [57]: 23–38 (2002)

[87] Miller A.W.: *Descriptive Set Theory and Forcing*. Springer (1995)

[88] Mostowski A.W.: Hierarchies of Weak Automata and Weak Monadic Formulas. *Theoret. Comput. Sci.* 83: 323–335 (1991)

[89] Mostowski A.W.: Games with forbidden position. Technical Report 78, University of Gdansk (1991)

[90] Murlak F.: On Deciding Topological Classes of Deterministic Tree Languages. *CSL '05*: 573–584 (2005)

[91] Murlak F.: *Effective Topological Hierarchies of Recognizable Tree Languages*. PhD thesis, University of Warsaw (2008)

[92] Murlak F.: The Wadge Hierarchy of Deterministic Tree Languages. *Logical Methods in Comput. Sci.*, 4(4), Paper 15 (2008)

[93] Murlak F.: Weak Index vs Borel Rank. *STACS '08*: 573–584 (2008)

[94] Muller D., Saoudi A., Schupp P.: Alternating Automata, the Weak Monadic Theory of the Tree and its Complexity. *ICALP '86*: 275–283 (1986)

[95] Muller D. R., Schupp P. E.: Alternating Automata on Infinite Trees. *Theoretical Computer Science* 54: 267–276 (1987)

[96] Niwiński D.: On Fixed Point Clones. *ICALP '86*: 464–473 (1986)

[97] Niwiński D.: Fixed Points vs. Infinite Generation. *LICS '88*: 402–409 (1988)

[98] Niwiński D.: Fixed Point Characterization of Infinite Behavior of Finite-State Systems. *Theoretical Computer Science* 189: 1–69 (1997)

[99] Niwiński D., Seidl H.: On Distributive Fixed-Point Expressions. *RAIRO Theoretical Informatics and Applications* 33(4-5): 427–446 (1999)

[100] Niwiński D., Walukiewicz I.: Relating Hierarchies of Word and Tree Automata. *STACS '98*: 320–331 (1998)

[101] Niwiński D., Walukiewicz I.: A Gap Property of Deterministic Tree Languages. *Theor. Comput. Sci.* 303: 215–231 (2003)

[102] Niwiński D., Walukiewicz I.: Deciding Nondeterministic Hierarchy of Deterministic Tree Automata. *Electr. Notes Theor. Comput. Sci.* 123: 195–208 (2005)

[103] Otto M.: Eliminating Recursion in the μ-Calculus. *STACS '99*: 531–540 (1999)

[104] Perrin D., Pin J.-E.: *Infinite Words. Automata, Semigroups, Logic and Games*. Pure and Applied Mathematics Vol 141, Elsevier (2004)

[105] Pin J.-E.: Logic, Semigroups and Automata on Words. *Ann. Math. Artif. Intell.* 16: 343–384 (1996)

[106] Place T.: Characterization of Logics Over Ranked Tree Languages. *CSL '08*: 401–415 (2008)

[107] Rabin V.M.O.: Decidability of Second-Order Theories and Automata on Infinite Trees. *Trans. Amer. Soc.* 141: 1–35 (1969)

[108] Rabin V.M.O.: Weakly Definable Relations and Special Automata. In: *Mathematical Logic and Foundations of Set Theory*, North-Holland: 1–70 (1970)

[109] Schuützenberger M.P.: On Finite Monoids Having Only Trivial Subgroups. *Information and Control* 8: 190–194 (1965)

[110] Scott D., De Bakker, J.: A Theory of Programs. Unpublished manuscript, IBM, Vienna (1969)

[111] Selivanov V.: Wadge Degrees of ω-Languages of Deterministic Turing Machines. *ITA* 37(1): 67–83 (2003).

[112] Simonnet P.: *Automates et théorie descriptive*. PhD thesis, Université Paris VII (1992)

[113] Smorynski, C.: *Self-Reference and Modal Logic*. Springer-Verlag (1985)

[114] Skurczyński J.: The Borel Hierarchy is Infinite in the Class of Regular Sets of Trees. *Theoret. Comput. Sci.* 112: 413–418 (1993)

[115] Stirling C.: *Modal and Temporal Properties of Processes*. Springer-Verlag Berlin Heidelberg (2001)

[116] Streett R.S., Emerson E.A.: An Automata Theoretic Decision Procedure for the Propositional μ-Calculus. *Information and Computation* 81(3): 249–264 (1989)

[117] Thomas W.: A Hierarchy of Sets of Infinite Trees. In *Theoretical Computer Science Proc.*: 335–342 (1983)

[118] Thomas W.: Languages, Automata, and Logic. In G. Rozenberg and A. Salomaa (Eds.): *Handbook of Formal Languages*, Volume III, Springer, New York: 389–455 (1997)

[119] Tarski A.: A Lattice-Theoretical Fixpoint Theorem and its Applications. *Pacific Journal of Mathematics* 5(2): 285–309 (1955)

[120] Urbański T. F.: On Deciding if Deterministic Rabin Language is in Büchi Class. *ICALP '00*: 663–674 (2000)

[121] Vardi M., Wolter P.: An Automata-Theoretic Approach to Automatic Program Verication. *LICS '88*: 322–331 (1988)

[122] Visser A.: Uniform Interpolation and Layered Bisimulation. In: *Gödel '96 (Brno, 1996)*. Lectures Notes in Logic, 6 Berlin, Springer: 139–164 (1996)

[123] Visser A.: Löb's Logic Meets the μ-Calculus. In: A. Middeldorp et al. (Eds.): *Processes, Terms and Cycles, Steps on the Road to Infinity, Essays Dedicated to Jan Willem Klop on the Occasion of His 60th Birthday.* Berlin, Springer: 14–25 (2005)

[124] Wadge W.W.: Degrees of Complexity of Subsets of the Baire Space. *Notice A.M.S.*: A714–A715 (1972)

[125] Wadge W.W.: *Reducibility and Determinateness on the Baire Space.* Ph.D. Thesis, Berkeley (1984)

[126] Wagner K.: Eine topologische Charackterisierung einiger Klassen regulärer Folgenmengen. *J. Inf. Process. Cybern.* EIK 13: 473–487 (1977)

[127] Wagner K.: On ω-Regular Sets. *Inform. and Control* 43: 123–177 (1979)

[128] Walukiewicz I.: Completeness of Kozen's Axiomatisation of the Propositional Mu-Calculus. *Information and Computation* 157: 142–182 (2000)

[129] Walukiewicz I.: Deciding Low Levels of Tree-Automata Hierarchy. *Electr. Notes Theor. Comput. Sci.* 67: 61–75 (2002)

[130] Walukiewicz I.: Monadic Second-Order Logic on Tree-Like Structures. *Theoretical Computer Science* 275(1-2): 311–346 (2002)

[131] Wilke T.: Classifying Discrete Temporal Properties. *STACS '99*: 2–46 (1999)

[132] Wilke T.: Alternating Tree Automata, Parity Games, and Modal μ-Calculus. *Bul. Belg. Math. Soc.* 8(2): 359–391 (2001)

[133] Wilke T., Vardi M.: Automata - From Logic to Algorithms. In J. Flum et al. (Eds.) *Logic and Automata - History and Perspectives*, Texts in Logic and Games, Amsterdam University Press: 629–736 (2007)