The μ-Calculus Alternation Hierarchy Collapses over Structures with Restricted Connectivity

Julian Gutierrez[a], Felix Klaedtke[b], Martin Lange[c]

[a] *University of Oxford, United Kingdom*
[b] *NEC Europe Ltd., Germany*
[c] *University of Kassel, Germany*

Abstract

The alternation hierarchy of least and greatest fixpoint operators in the μ-calculus is strict. However, the strictness of the hierarchy does not necessarily carry over when considering restricted classes of structures. For instance, over the class of infinite words the alternation-free fragment of the μ-calculus is already as expressive as the full logic. Our current understanding of when and why the μ-calculus alternation hierarchy is (and is not) strict is limited. This article makes progress in answering these questions by showing that the alternation hierarchy of the μ-calculus collapses to the alternation-free fragment over some classes of structures, including infinite nested words and finite graphs with feedback vertex sets of a bounded size. Common to these classes is that the connectivity between the components in a structure from such a class is restricted in the sense that the removal of certain vertices from the structure's graph decomposes it into graphs in which all paths are of finite length. The collapse results herein are obtained in an automata-theoretic setting. They subsume, generalize, and strengthen several prior results on the expressivity of the μ-calculus over restricted classes of structures.

Keywords: modal μ-calculus, fixpoints, alternation hierarchy, parity automata

1. Introduction

The μ-calculus [20], hereafter $\mathcal{L}_\mu$, extends modal logic with least and greatest fixpoint operators, which act as monadic second-order (MSO) quantifiers within the logic. The possibility to arbitrarily mix and nest fixpoint operators makes $\mathcal{L}_\mu$ an expressive logic, which subsumes many dynamic, temporal, and description logics such as PDL and CTL*. In fact, $\mathcal{L}_\mu$ is essentially the most expressive logic of that kind because it can express, up to bisimulation equivalence, all MSO-definable properties [18].

An important question about the expressivity of $\mathcal{L}_\mu$ is whether more alternation—the nesting of mutually dependent least and greatest fixpoint operators in formulas—gives more expressive power. Bradfield [8] proved that indeed this is in general the case, that is, there is a hierarchy of properties that require unbounded alternation of least and greatest fixpoint operators. Lenzi [24] independently showed a similar strictness result—for a fragment of $\mathcal{L}_\mu$—with respect to an alternation hierarchy different from the one we consider in this article. In both cases, their strictness results apply to the class of finite directed graphs and therefore to all bigger classes of structures. However, the strictness of the alternation hierarchy need not necessarily carry over when considering classes of structures that are either incomparable to or smaller than the class of finite directed graphs. Trivial examples over which the alternation hierarchy is non-strict are, for example, classes that only consist of a single graph. In that case, each formula is equivalent to either *true* or *false*, depending on whether the graph satisfies the formula or not.

Overall, little is known about the expressivity of $\mathcal{L}_\mu$ over restricted classes of structures. Since $\mathcal{L}_\mu$ is bisimulation-invariant and every finite graph, either directed or undirected, is bisimilar to a possibly infinite tree, the strictness of the hierarchy also holds for the class of trees. In fact, as shown by Arnold [4] and Bradfield [9], the hierarchy is strict even on the class of binary infinite trees. Alberucci and Facchini [1] also strengthened the initial strictness result by showing that the hierarchy remains strict over the class of

reflexive finite directed graphs. D'Agostino and Lenzi [13] showed the strictness of the hierarchy over the class of reflexive and symmetric finite graphs.

On the opposite side, there are a few classes of structures over which it is known that the alternation hierarchy is not strict. For instance, the hierarchy collapses to its alternation-free fragment over the class of finite directed acyclic graphs [27]: for every $\mathcal{L}_\mu$ formula φ, there is an alternation-free $\mathcal{L}_\mu$ formula ψ—that is, a formula in which least and greatest fixpoint operators do not mutually depend on each other—such that φ and ψ are satisfied by exactly the same set of finite acyclic graphs. This collapse result is not too surprising since the denotation of the least and greatest fixpoint operators of $\mathcal{L}_\mu$ differs only when considering models which contain infinite paths—and finite directed acyclic graphs only contain finite paths. Thus, in this case, every greatest fixpoint operator can be replaced by a least one, resulting in an alternation-free formula. It is also known, when restricting $\mathcal{L}_\mu$ to infinite words, that $\mathcal{L}_\mu$'s alternation hierarchy collapses to its alternation-free fragment [19]. Moreover, over infinite nested words, as shown by Arenas et al. [3], the hierarchy collapses to the fragment with at most one alternation between fixpoint operators. It is also known that the $\mathcal{L}_\mu$'s alternation hierarchy collapses over the class of transitive finite directed graphs [1, 12, 14]. If the graphs are transitive and undirected, then the hierarchy even collapses to the modal fragment [1, 14].

This article provides further classes of structures over which the alternation hierarchy of $\mathcal{L}_\mu$ collapses to its alternation-free fragment. Our collapse results subsume, generalize, and strengthen some of the collapse results above mentioned. We show that the alternation hierarchy collapses over classes of finite directed graphs with feedback vertex sets of a bounded size. Recall that removing the vertices in a feedback vertex set decomposes the graph into finite directed acyclic graphs and thus the removal of these vertices eliminates the infinite behavior in the original graph. For finite directed acyclic graphs, the empty set is already a feedback vertex set. We also show that, as for infinite words, all $\mathcal{L}_\mu$ properties of infinite nested words can already be expressed within the alternation-free fragment. Our results are obtained in a uniform way by looking at bounded classes of *bottlenecked directed acyclic graphs*. The vertices of such a kind of graphs are grouped into layers and the infinite paths must visit infinitely often vertices in certain layers, which are bounded in their size. Intuitively speaking, these bounded layers are the bottlenecks and the removal of these vertices disconnects the graph into graphs in which all paths have finite length. Nested words and the unfoldings of finite directed graphs with bounded feedback vertex sets are special instances of such graphs.

Our work is carried out in an automata-theoretic setting. Roughly speaking, the question of whether the alternation hierarchy collapses to the alternation-free fragment over a class of structures $\mathbf{U}$ can be answered positively by showing that alternating parity automata are as expressive as weak alternating automata over $\mathbf{U}$. Translations between automata and $\mathcal{L}_\mu$ formulas are known, for instance, [15, 23, 30, 32]. Yet, the translation from weak alternating automata to alternation-free formulas we provide here is more direct than the known ones in the sense that it avoids the construction of formulas in vectorial form, cf. [5].

Another contribution of this paper is a generalization of the ranking construction by Kupferman and Vardi [22], which can be used to translate alternating coBüchi word automata into language-equivalent weak alternating word automata. We generalize it to the parity acceptance condition and to more complex classes of structures, namely, to bounded bottlenecked graphs. Kupferman and Vardi [21] have already generalized their ranking construction for word automata to solve the non-emptiness problem for nondeterministic parity tree automata. However, our generalization of their ranking construction [22] is conceptually simpler: it eliminates the odd colors of a parity automaton in a single step. An additional step is needed to obtain a weak automaton from the resulting Büchi automaton. In contrast, Kupferman and Vardi's generalization [21] successively eliminates colors, alternating between odd and even ones. The acceptance conditions of the intermediate word automata are a combination of a parity condition and a Büchi or coBüchi condition.

Structure of the paper. Preliminaries on $\mathcal{L}_\mu$ and alternating automata are given in Section 2 and translations between $\mathcal{L}_\mu$ and automata appear in Section 3. Section 4 presents our generalization of the ranking construction. Section 5 contains the collapse results. Finally, in Section 6, we draw conclusions.

2. Preliminaries

In this section, we provide notation and terminology that we use throughout the article.

2.1. The μ-Calculus

Graphs. Let A be a non-empty finite set of *actions* and let Σ be an alphabet. A (Σ, A)-*graph* is a directed, labeled, and pointed graph $\big(V, (E_a)_{a\in A}, v_\mathrm{I}, \lambda\big)$, where V is a set of vertices, $E_a \subseteq V \times V$ is a set of edges labeled by $a \in A$, $v_\mathrm{I} \in V$ is the source, and $\lambda : V \to \Sigma$ a labeling function. We require in the following that V is at most countable.

Syntax. We define the μ-calculus, $\mathcal{L}_\mu$ for short, over $(2^\mathcal{P}, A)$-graphs, where $\mathcal{P}$ is a non-empty set of propositions. Let $\mathcal{V} = \{X, Y, \dots\}$ be a countable set of variables. The syntax of $\mathcal{L}_\mu$ is given by the grammar

$$\varphi ::= X \mid p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid [a]\,\varphi \mid \langle a\rangle\,\varphi \mid \mu X.\,\varphi \mid \nu X.\,\varphi\,,$$

where X ranges over $\mathcal{V}$, p over $\mathcal{P}$, and a over A.

The operators μ and ν act as binders for the respective variables. A *free* occurrence of a variable X in a formula φ is one that is not under the scope of a binding operator for this variable in φ. A formula φ is a *sentence* if no variable occurs free in φ.

The *set of subformulas* of a formula ψ is

$$\mathrm{sub}(\psi) := \begin{cases} \{\psi\} & \text{if } \psi \in \mathcal{V} \text{ or } \psi \in \mathcal{P}, \\ \{p, \neg p\} & \text{if } \psi = \neg p \text{ with } p \in \mathcal{P}, \\ \{\psi\} \cup \mathrm{sub}(\psi') \cup \mathrm{sub}(\psi'') & \text{if } \psi = \psi' \star \psi'' \text{ with } \star \in \{\vee, \wedge\}, \\ \{\psi\} \cup \mathrm{sub}(\psi') & \text{if } \psi = [a]\,\psi' \text{ or } \psi = \langle a\rangle\,\psi' \text{ with } a \in A, \text{ or} \\ & \quad \psi = \kappa X.\,\psi' \text{ with } \kappa \in \{\mu, \nu\} \text{ and } X \in \mathcal{V}. \end{cases}$$

We say that ψ is a *subformula* of ψ' if $\mathrm{sub}(\psi) \subseteq \mathrm{sub}(\psi')$. The *size* of ψ, written $|\psi|$, is its number of syntactically distinct subformulas, that is, the cardinality of $\mathrm{sub}(\psi)$.

We assume in the following, without loss of generality, that the free variables are disjoint from the bound variables in a formula ψ and that each bound variable is uniquely bound by a fixpoint operator. Then we can assume to have a function fp_ψ that associates a bound variable $X \in \mathrm{sub}(\psi)$ to its unique binding fixpoint formula, that is, $fp_\psi(X)$ is ψ's subformula $\kappa X.\,\psi'$ with $\psi' \in \mathrm{sub}(\psi)$ and $\kappa \in \{\mu, \nu\}$.

Semantics. The semantics of $\mathcal{L}_\mu$ is as follows. Let $\mathcal{G} = \big(V, (E_a)_{a\in A}, v_\mathrm{I}, \lambda\big)$ be a $(2^\mathcal{P}, A)$-graph. A valuation σ assigns each variable in $\mathcal{V}$ to a set of vertices. For $X \in \mathcal{V}$ and $U \subseteq V$, we write $\sigma[X \mapsto U]$ if we alter σ at X, that is, $\sigma[X \mapsto U](Y) := U$ if $Y = X$ and $\sigma[X \mapsto U](Y) := \sigma(Y)$, otherwise. The set $[\![\varphi]\!]^\mathcal{G}_\sigma$ of vertices in $\mathcal{G}$ that satisfy φ under σ is defined as follows:

$$\begin{aligned}
[\![X]\!]^\mathcal{G}_\sigma &:= \sigma(X) \\
[\![p]\!]^\mathcal{G}_\sigma &:= \big\{v \in V \mid p \in \lambda(v)\big\} \\
[\![\neg p]\!]^\mathcal{G}_\sigma &:= \big\{v \in V \mid p \notin \lambda(v)\big\} \\
[\![\varphi \wedge \psi]\!]^\mathcal{G}_\sigma &:= [\![\varphi]\!]^\mathcal{G}_\sigma \cap [\![\psi]\!]^\mathcal{G}_\sigma \\
[\![\varphi \vee \psi]\!]^\mathcal{G}_\sigma &:= [\![\varphi]\!]^\mathcal{G}_\sigma \cup [\![\psi]\!]^\mathcal{G}_\sigma \\
[\![[a]\,\varphi]\!]^\mathcal{G}_\sigma &:= \big\{v \in V \mid \text{if } (v, v') \in E_a \text{ then } v' \in [\![\varphi]\!]^\mathcal{G}_\sigma, \text{ for all } v' \in V\big\} \\
[\![\langle a\rangle\,\varphi]\!]^\mathcal{G}_\sigma &:= \big\{v \in V \mid (v, v') \in E_a \text{ and } v' \in [\![\varphi]\!]^\mathcal{G}_\sigma, \text{ for some } v' \in V\big\} \\
[\![\mu X.\,\varphi]\!]^\mathcal{G}_\sigma &:= \bigcap \big\{U \in 2^V \mid [\![\varphi]\!]^\mathcal{G}_{\sigma[X\mapsto U]} \subseteq U\big\} \\
[\![\nu X.\,\varphi]\!]^\mathcal{G}_\sigma &:= \bigcup \big\{U \in 2^V \mid [\![\varphi]\!]^\mathcal{G}_{\sigma[X\mapsto U]} \supseteq U\big\}
\end{aligned}$$

The grammar of $\mathcal{L}_\mu$ guarantees that formulas are in negation normal form, that is, negations only occur directly in front of propositions in $\mathcal{P}$. This syntactic feature ensures monotonicity and thus existence of the least and greatest fixpoints expressed by μ and ν, respectively.

For a sentence φ, $[\![\varphi]\!]^{\mathcal{G}}_\sigma$ does not depend on σ and we omit it if it is clear from the context that φ is a sentence. For a sentence φ and a set of $(2^{\mathcal{P}}, A)$-graphs $\mathbf{U}$, we define

$$L_{\mathbf{U}}(\varphi) := \left\{ \mathcal{G} \in \mathbf{U} \;\middle|\; v_{\mathrm{I}} \in [\![\varphi]\!]^{\mathcal{G}}, \text{ where } v_{\mathrm{I}} \text{ is the source of } \mathcal{G} \right\}.$$

Alternation Hierarchy. $\mathcal{L}_\mu$ formulas determine an infinitely large hierarchy, which relies on the mutual interdependencies between least and greatest fixpoint operators. To define this hierarchy, we follow Niwiński [29]:

- $\Sigma_0 = \Pi_0$ is the set of formulas without fixpoint operators, that is, modal logic formulas.
- For $n \geq 0$, Σ_{n+1} is the smallest set that contains the formulas in $\Sigma_n \cup \Pi_n$ and is closed under the following rules: (i) if $\varphi, \psi \in \Sigma_{n+1}$ then $\varphi \wedge \psi \in \Sigma_{n+1}$ and $\varphi \vee \psi \in \Sigma_{n+1}$; (ii) if $\varphi \in \Sigma_{n+1}$ and $a \in A$ then $[a]\,\varphi \in \Sigma_{n+1}$ and $\langle a \rangle\,\varphi \in \Sigma_{n+1}$; (iii) if $\varphi \in \Sigma_{n+1}$ and $X \in \mathcal{V}$ then $\mu X.\,\varphi \in \Sigma_{n+1}$; (iv) if $\varphi, \psi \in \Sigma_{n+1}$ and $X \in \mathcal{V}$ then $\varphi[\psi/X] \in \Sigma_{n+1}$ provided that no free variable of ψ gets bound by a fixpoint operator in φ and where $\varphi[\psi/X]$ denotes the formula obtained from substituting the free occurrences of X by ψ in φ.
- For $n \geq 0$, Π_{n+1} is analogously defined as Σ_{n+1}: instead of closure under the least fixpoint operator μ, we require closure with respect to the greatest fixpoint operator ν.
- For $n \geq 0$, we also define $\Delta_n := \Sigma_n \cap \Pi_n$.

The *alternation depth* of φ, denoted by $\mathrm{ad}(\varphi)$, is the smallest $n \geq 0$ such that $\varphi \in \Delta_{n+1}$. A formula φ is *alternation-free* if $\mathrm{ad}(\varphi) \leq 1$, that is, it is in Δ_2.

We remark that there are other definitions of alternation for $\mathcal{L}_\mu$ in the literature, for instance, the definition of Emerson and Lei [16]. Niwiński's alternation hierarchy and Emersons and Lei's hierarchy are both based on the formulas' syntax and not on the property a formula describes. The differences between the different hierarchies are with respect to the undesirable sensitivity to vacuous fixpoints. For instance, the $\mathcal{L}_\mu$ formula $\mu X.\,\nu Z.\,\mu U.\,\nu Y.\, Y \wedge X$, which is taken from [8], is in Σ_2 according to Niwiński and in Σ_4 according to Emerson and Lei. Niwiński's hierarchy captures the actual dependency between least and greatest fixpoints in $\mathcal{L}_\mu$ more accurately. The differences of the two hierarchy are, however, inessential to our collapse results as our focus is on the alternation-free fragment where these differences completely disappear.

Example 2.1. Consider the formulas $\varphi = \nu X.\,\big(\mu Y.\, p \vee \langle a \rangle\, Y\big) \wedge [a]\, X$ and $\psi = \mu X.\,\nu Y.\,\big([c]\, Y \vee \langle a \rangle\, Y \vee \langle b \rangle\, X\big)$. The formula φ expresses that along all paths with a actions, there is a path on which p eventually holds. The formula ψ is the $\mathcal{L}_\mu$ representation of a satisfiability formula in μ-arithmetic; see [9] for details.

By the definition of the alternation hierarchy, it is easy to see that the formula φ is alternation-free because it is in both Σ_2 and Π_2, and hence in Δ_2. Since the subformula $\mu Y.\, p \vee \langle a \rangle\, Y$ of φ has no free variables, the subformula's least fixpoint can be computed independently of φ's greatest fixpoint. In contrast, the formula ψ is not alternation-free. It is in Σ_2 but not in Π_2; ψ is also in Π_3 and therefore in Δ_3. In fact, ψ is a strict formula [9] for Δ_3 in the sense that it is not logically equivalent to any formula in Δ_2. The least and greatest fixpoints in ψ cannot be computed independently: The inner fixpoint formula $\nu Y.\,\big([c]\, Y \vee \langle a \rangle\, Y \vee \langle b \rangle\, X\big)$ has the free variable X of a different kind, that is, X is bound by the least fixpoint operator in ψ. Moreover, the computation for the greatest fixpoint depends on the computation of the outer least fixpoint.

In addition to the alternation hierarchy, we measure the dependencies of fixpoint formulas within a sentence ψ by assigning to each variable X occurring in ψ a non-negative integer $ai_\psi(X)$, called its *priority*. For a variable X occurring in ψ, $ai_\psi(X)$ is defined to be the smallest natural number that satisfies:

1. $ai_\psi(X)$ is odd if $fp_\psi(X) = \mu X.\,\psi'$ with $\psi' \in \mathrm{sub}(\psi)$.
2. $ai_\psi(X)$ is even if $fp_\psi(X) = \nu X.\,\psi'$ with $\psi' \in \mathrm{sub}(\psi)$.
3. For all $Y \in \mathrm{sub}(\psi)$, with $X \neq Y$, if there is a free occurrence of Y in $fp_\psi(X)$ then $ai_\psi(X) \geq ai_\psi(Y)$.

For instance, in Example 2.1, we have $ai_\varphi(X) = 0$ and $ai_\varphi(Y) = 1$, and $ai_\psi(X) = 1$ and $ai_\psi(Y) = 2$. Note that the priority of a fixpoint variable Z that appears in a formula φ can be smaller than the index of the set, either Σ_n or Π_n, to which a subformula in $\mathrm{sub}(fp_\varphi(Z))$ belongs. For instance, in Example 2.1, observe that $ai_\varphi(X) = 0$ whereas $fp_\varphi(Y)$ is in Σ_1. Informally, the situation we have in this case is that although $fp_\varphi(Y)$ is a subformula of $fp_\varphi(X)$, the computation of the least fixpoint formula for Y is not dependent on the computation of the outer greatest fixpoint formula for X—since X does not appear free in $fp_\varphi(Y)$. In fact, we could replace $fp_\varphi(Y)$ in φ by any sentence and X would still have priority 0.

2.2. Alternating Automata

Propositional Logic. We denote the set of *positive Boolean formulas* over the proposition set $\mathcal{P}$ by $\mathbf{B}^+(\mathcal{P})$, that is, $\mathbf{B}^+(\mathcal{P})$ consists of the formulas that are inductively built from the Boolean constants tt and ff, the propositions in $\mathcal{P}$, and the Boolean connectives $\vee$ and $\wedge$. For $\mathcal{M} \subseteq \mathcal{P}$ and $\varphi \in \mathbf{B}^+(\mathcal{P})$, write $\mathcal{M} \models \varphi$ if φ holds when assigning true to the propositions in $\mathcal{M}$ and false to those in $\mathcal{P} \setminus \mathcal{M}$.

Words and Trees. We denote the set of finite words over the alphabet Σ by Σ^*, the set of infinite words over Σ by Σ^ω, and the empty word by ε. We denote the symbol at position $i+1$ in a word w by w_i. We write $v \preceq w$ if v is a prefix of w.

A (Σ-labeled) *tree* is a function $t : T \to \Sigma$, where $T \subseteq \mathbb{N}^*$ satisfies that: (i) T is prefix-closed, that is, $v \in T$ and $u \preceq v$ implies $u \in T$, and (ii) if $vi \in T$ and $i > 0$ then $v(i-1) \in T$. The elements in T are called the *nodes* of t and the empty word ε is called the *root* of t. A node $vi \in T$ with $i \in \mathbb{N}$ is called a *child* of the node $v \in T$. A *branch* in t is a word $\pi \in \mathbb{N}^* \cup \mathbb{N}^\omega$ such that either $\pi \in T$ and π does not have any children, or π is infinite and every finite prefix of π is in T. We write $\bar{t}(\pi)$ for the word $t(\varepsilon)t(\pi_0)t(\pi_0\pi_1)\dots t(\pi_0\pi_1\dots\pi_{n-1}) \in \Sigma^*$ if π has finite length n and $t(\varepsilon)t(\pi_0)t(\pi_0\pi_1)\dots \in \Sigma^\omega$ if π is infinite.

Automata. In the following, we define alternating automata where the inputs are $(2^{\mathcal{P}}, A)$-graphs, where $\mathcal{P}$ is a non-empty finite set of propositions and A is a non-empty finite set of actions. Such automata are essentially alternating parity tree automata that operate over the tree unfolding of the given input. The classical automata models for words and trees are special instances when encoding the letters of an alphabet Σ by subsets of propositions and by viewing words and trees in a straightforward way as (Σ, A)-graphs.

A *parity* $(\mathcal{P}, A)$*-automaton*, $(\mathcal{P}, A)$-PA for short, is a tuple $\mathcal{A} = (Q, \delta, q_{\mathrm{I}}, \alpha)$, where Q is a finite set of states, $\delta : Q \to \mathbf{B}^+\big(\mathcal{P} \cup \bar{\mathcal{P}} \cup (Q \times \{\Diamond, \Box\} \times A)\big)$ is the transition function with $\bar{\mathcal{P}} := \{\bar{p} \mid p \in \mathcal{P}\}$, $q_{\mathrm{I}} \in Q$ is the initial state, and $\alpha : Q \to \mathbb{N}$ determines the (parity) acceptance condition. Assume that $\mathcal{P} \cap \bar{\mathcal{P}} = \emptyset$. We refer to $\alpha(q)$ as the *color* of the state $q \in Q$. The *index* of $\mathcal{A}$ is $\mathrm{ind}(\mathcal{A}) := |\{\alpha(q) \mid q \in Q\}|$ and the *size* of $\mathcal{A}$ is the number of syntactically distinct subformulas that occur in the transitions, that is, $\|\mathcal{A}\| := \big|\bigcup_{q \in Q} \{\psi \mid \psi \text{ is a subformula of } \delta(q)\}\big|$. In the following, we assume that $|Q| \in \mathcal{O}(\|\mathcal{A}\|)$, which holds when, for example, every state occurs in some transition of $\mathcal{A}$.

Let $\mathcal{A} = (Q, \delta, q_{\mathrm{I}}, \alpha)$ be a $(\mathcal{P}, A)$-PA and $\mathcal{G} = \big(V, (E_a)_{a \in A}, v_{\mathrm{I}}, \lambda\big)$ a $(2^{\mathcal{P}}, A)$-graph. A *run* of $\mathcal{A}$ on $\mathcal{G}$ is a tree $\varrho : R \to V \times Q$ with some $R \subseteq \mathbb{N}^*$ such that $\varrho(\varepsilon) = (v_{\mathrm{I}}, q_{\mathrm{I}})$ and for each node $x \in R$ with $\varrho(x) = (v, p)$, there is a set $\mathcal{M} \subseteq Q \times \{\Diamond, \Box\} \times A$ such that

$$\big\{q \in \mathcal{P} \;\big|\; q \in \lambda(v)\big\} \cup \big\{\bar{q} \in \bar{\mathcal{P}} \;\big|\; q \notin \lambda(v)\big\} \cup \mathcal{M} \models \delta(p)$$

and the following conditions are satisfied:

(a) If $(q, \Diamond, a) \in \mathcal{M}$, then there is a node $v' \in V$ with $(v, v') \in E_a$ such that there is a child $x' \in R$ of x such that $\varrho(x') = (v', q)$.

(b) If $(q, \Box, a) \in \mathcal{M}$, then for all nodes $v' \in V$ with $(v, v') \in E_a$ there is a child $x' \in R$ of x such that $\varrho(x') = (v', q)$.

Roughly speaking, $\mathcal{A}$ starts in its initial state by scanning the input graph from its source. The label (v, p) of the node x in the run is the current configuration of $\mathcal{A}$. That is, $\mathcal{A}$ is currently in the state p and the read-only head is at the vertex v of the input. The transition $\delta(p)$ specifies with respect to the

labeling $\lambda(v)$ a constraint that has to be respected by the automaton's successor states. In particular, for a proposition $(q, \Diamond, a) \in \mathcal{M}$, the read-only head must move along some a-labeled edge starting at v. Similarly, for $(q, \Box, a) \in \mathcal{M}$, a copy of the read-only head must move along every a-labeled edge starting at vertex v.

An infinite branch π in a run ϱ with $\bar{\varrho}(\pi) = (v_0, q_0)(v_1, q_1)\dots$ is *accepting* if $\max\{\alpha(q) \mid q \in \inf(q_0 q_1 \dots)\}$ is even, where $\inf(q_0 q_1 \dots)$ is the set of states that occur infinitely often in $q_0 q_1 \dots$. The run ϱ is *accepting* if every infinite branch in ϱ is accepting. The *language* of $\mathcal{A}$ with respect to a set $\mathbf{U}$ of $(2^{\mathcal{P}}, A)$-graphs is

$$L_{\mathbf{U}}(\mathcal{A}) := \{\mathcal{G} \in \mathbf{U} \mid \text{there is an accepting run of } \mathcal{A} \text{ on } \mathcal{G}\}.$$

By restricting the acceptance condition and the automaton's transitions, we obtain the following automata classes. Let $\mathcal{A} = (Q, \delta, q_{\mathrm{I}}, \alpha)$ be a $(\mathcal{P}, A)$-PA.

- $\mathcal{A}$ is *Büchi* if $\{\alpha(q) \mid q \in Q\} \subseteq \{2n-1, 2n\}$, for some $n \geq 1$.
- $\mathcal{A}$ is *coBüchi* if $\{\alpha(q) \mid q \in Q\} \subseteq \{2n, 2n+1\}$, for some $n \geq 0$.
- $\mathcal{A}$ is *weak* if there is a partition $Q_0, \dots, Q_n$ on Q, for some $n \geq 0$ such that for all $i \in \{0, \dots, n\}$, we have: (i) All states in the component Q_i have the same parity, that is, $\alpha(q) \equiv \alpha(q') \bmod 2$, for all $q, q' \in Q_i$. (ii) $\delta(q) \in \mathbf{B}^+\big(\mathcal{P} \cup \bar{\mathcal{P}} \cup \bigcup_{j \in \{i,\dots,n\}} (Q_j \times \{\Diamond, \Box\} \times A)\big)$, for all $q \in Q_i$. That is, when reading a vertex label the automaton can stay in the current component Q_i or go to components with higher indices.

We also call $\mathcal{A}$ a $(\mathcal{P}, A)$-BA, $(\mathcal{P}, A)$-CA, and $(\mathcal{P}, A)$-WA when it is Büchi, coBüchi, and weak, respectively.

Finally, dualizing an alternating automaton corresponds to complementation [28]. In our case, the *dual automaton* of a $(\mathcal{P}, A)$-PA $\mathcal{A} = (Q, \delta, q_I, \alpha)$ is defined as the $(\mathcal{P}, A)$-PA $\overline{\mathcal{A}} := (Q, \overline{\delta}, q_I, \overline{\alpha})$, where for each $q \in Q$, $\overline{\delta}(q) := \overline{\delta(q)}$ with

$$\begin{aligned}
\overline{\mathsf{tt}} &:= \mathsf{ff} & \overline{\mathsf{ff}} &:= \mathsf{tt} \\
\overline{p} &:= \bar{p}, \text{ for } p \in \mathcal{P} & \overline{\bar{p}} &:= p, \text{ for } \bar{p} \in \bar{\mathcal{P}} \\
\overline{(q, \Diamond, a)} &:= (q, \Box, a) & \overline{(q, \Box, a)} &:= (q, \Diamond, a) \\
\overline{\beta \wedge \gamma} &:= \overline{\beta} \vee \overline{\gamma} & \overline{\beta \vee \gamma} &:= \overline{\beta} \wedge \overline{\gamma}
\end{aligned}$$

and $\overline{\alpha}(q) := \alpha(q) + 1$. It is not too hard to show that the dual automaton accepts the complement language, that is, $L_{\mathbf{U}}(\overline{\mathcal{A}}) = \mathbf{U} \setminus L_{\mathbf{U}}(\mathcal{A})$. Furthermore, note that $\overline{\mathcal{A}}$ is weak if $\mathcal{A}$ is weak.

3. From the μ-Calculus to Automata and Back

Translations between $\mathcal{L}_\mu$ and automata are known for various automaton models. For the sake of completeness, we present in this section such translations with respect to our automaton model from Section 2.2. In the remainder of the text, let $\mathcal{P}$ and A be non-empty finite sets of propositions and actions, respectively. Furthermore, throughout this section, let $\mathbf{U}$ be a set of $(2^{\mathcal{P}}, A)$-graphs.

3.1. From Formulas to Parity Automata

The following translation is similar to the one in [32]. However, since our automaton model does not support ε-transitions, we need to require for the translation that formulas are guarded, that is, variables occur under the scope of a modal operator within their defining fixpoint formulas. For a proof of the following lemma, see, for instance, [31].

Lemma 3.1. *For every sentence φ, there is a guarded sentence ψ of size $2^{\mathcal{O}(|\varphi|)}$ such that $L_{\mathbf{U}}(\psi) = L_{\mathbf{U}}(\varphi)$ and $\mathrm{ad}(\psi) = \mathrm{ad}(\varphi)$.*[1]

[1] The acclaimed polynomial upper bounds of translations into guarded form found in the literature are flawed [10].

From guarded formulas one easily obtains equivalent parity automata. This result is well-known, cf. [15]. We present the construction for the sake of completeness but refer to the literature for a correctness proof.

Theorem 3.2. *For every guarded sentence φ, there is a $(\mathcal{P}, A)$-PA $\mathcal{A}_\varphi$ with $\mathcal{O}(|\varphi|)$ states and $L_{\mathbf{U}}(\mathcal{A}_\varphi) = L_{\mathbf{U}}(\varphi)$. Moreover, $\|\mathcal{A}_\varphi\| \in \mathcal{O}(|\varphi|)$ and $\mathrm{ind}(\mathcal{A}_\varphi) \leq \mathrm{ad}(\varphi) + 1$.*

Proof. The components of the parity automaton $\mathcal{A}_\varphi := (Q, \delta, q_{\mathrm{I}}, \alpha)$ are as follows. The set of states Q consists of states of the form (ψ, d), where $\psi \in \mathrm{sub}(\varphi)$ and $d \in \{0, \ldots, \mathrm{ad}(\varphi)\}$. The initial state q_{I} is $(\varphi, 0)$ and the acceptance condition α is defined as $\alpha(\psi, d) := d$. For defining the $\mathcal{A}_\varphi$'s transition function δ, let trav_d, with $d \in \{0, \ldots, \mathrm{ad}(\varphi)\}$, be the following function defined by induction over ψ's formula structure:

$$\mathit{trav}_d(\psi) := \begin{cases} p & \text{if } \psi = p \text{ with } p \in \mathcal{P} \\ \bar{p} & \text{if } \psi = \neg p \text{ with } p \in \mathcal{P} \\ \mathit{trav}_{\mathit{ai}_\varphi(X)}\big(\mathit{fp}_\varphi(X)\big) & \text{if } \psi = X \text{ with } X \in \mathcal{V} \\ \mathit{trav}_d(\psi') \star \mathit{trav}_d(\psi'') & \text{if } \psi = \psi' \star \psi'' \text{ with } \star \in \{\wedge, \vee\} \\ \big((\psi', d), \Diamond, a\big) & \text{if } \psi = \langle a \rangle\, \psi' \\ \big((\psi', d), \Box, a\big) & \text{if } \psi = [a]\, \psi' \\ \mathit{trav}_d(\psi') & \text{if } \psi = \kappa X.\, \psi' \text{ with } \kappa \in \{\mu, \nu\} \end{cases}$$

Note that $\mathit{trav}_d(\psi)$ is well-founded because of guardedness: no $\mathit{trav}_d(\psi)$ can be defined in terms of itself because the formula parameter decreases in each clause apart from that for variables X. Guardedness then ensures that a clause for a modal operator is eventually met, which terminates the recursive definition. For a state $(\psi, d) \in Q$, we define $\delta(\psi, d) := \mathit{trav}_0(\psi)$. The parameter d signals when we have traversed through a fixpoint variable. It is initialized with 0 and set to $\mathit{ai}_\varphi(X)$ after we have gone through a variable X and replaced it by its defining fixpoint formula.

Since the acceptance condition α maps into the set $\{0, \ldots, \mathrm{ad}(\varphi)\}$ we obviously have $\mathrm{ind}(\mathcal{A}_\varphi) \leq \mathrm{ad}(\varphi)+1$. However, according to the above construction an upper bound on the number of states of $\mathcal{A}_\varphi$ is only $|\varphi| \cdot (\mathrm{ad}(\varphi) + 1)$. To optain the claimed bounds, we can optimize the construction by removing the states in Q that are not reachable from $\mathcal{A}_\varphi$'s initial state $(\varphi, 0)$. Except from the initial state, the reachable states have the form (ψ, d), where $\langle a \rangle\, \psi$ or $[a]\, \psi$ is a subformula of φ. Furthermore, d ranges only over some of the priorities. Namely, d is either 0 or $d = \mathit{ai}_\varphi(X)$, for some variable X that occurs in ψ and for which either $\langle a \rangle\, \psi$ or $[a]\, \psi$ is a subformula of $\mathit{fp}_\varphi(X)$ that is not under the scope of another modal operator. Since each $d > 0$ for ψ is associated to a fixpoint formula in φ, it follows that $\mathcal{A}_\varphi$ has $\mathcal{O}(|\varphi|)$ states and $\|\mathcal{A}_\varphi\| \in \mathcal{O}(|\varphi|)$. □

3.2. From Weak Automata to Alternation-free Formulas

The following translation essentially combines the ideas of a linear translation from weak alternating Büchi automata into alternation-free equation systems [23] with a standard translation from the latter into formulas of the alternation-free μ-calculus. This avoids the explicit introduction of modal equation systems [11, 25] or the μ-calculus with vectorial form [5]. These are expressively equivalent to $\mathcal{L}_\mu$ but provide a less elegant syntax at the benefit of more compact representations. This is also why a translation into plain formulas involves an exponential blow-up.

Theorem 3.3. *For every $(\mathcal{P}, A)$-WA $\mathcal{A}$ with n states, there is an alternation-free sentence $\varphi_{\mathcal{A}}$ with $|\varphi_{\mathcal{A}}| \in \mathcal{O}\big(\|\mathcal{A}\| \cdot 2^n\big)$ and $L_{\mathbf{U}}(\varphi_{\mathcal{A}}) = L_{\mathbf{U}}(\mathcal{A})$.*

Proof. Let $\mathcal{A} = (Q, \delta, q_{\mathrm{I}}, \alpha)$ be a $(\mathcal{P}, A)$-WA. Let $Q_0, \ldots, Q_m$ be a partition of its state set into components according to the definition of being weak. For every $i = m, \ldots, 0$, each state $q \in Q_i$ and each $S \subseteq Q_i \setminus \{q\}$ we define a formula ψ_q^S that, intuitively, describes the behavior of $\mathcal{A}$, parameterized by the states in S, when started in state q. It uses the variables $X_{q'}$, for $q' \in Q$, that describe the set of all nodes in a graph from which $\mathcal{A}$ has an accepting run when started in state q'. Translating the transition function directly using these variables would lead to circular dependencies among them rather than a tree-like structure that is

required for formulas. The parameter S is used to avoid these circular dependencies: it stores those states q', respectively variables $X_{q'}$ for which a fixpoint quantifier has been introduced already. Consequently, ψ_q^S may contain these $X_{q'}$ with $q' \in S$ as free variables.

Let $\psi_q^S := \kappa X_q.tr_q^{S\cup\{q\}}(\delta(q))$, with $\kappa = \mu$ if $\alpha(q)$ is odd and $\kappa = \nu$ otherwise, and where tr_q^S is defined as follows.

$$tr_q^S(\gamma \wedge \chi) := tr_q^S(\gamma) \wedge tr_q^S(\chi)$$
$$tr_q^S(\gamma \vee \chi) := tr_q^S(\gamma) \vee tr_q^S(\chi)$$
$$tr_q^S(p) := p \qquad \text{for } p \in \mathcal{P}$$
$$tr_q^S(\bar{p}) := \neg p \qquad \text{for } \bar{p} \in \bar{\mathcal{P}}$$

$$tr_q^S(q', \Diamond, a) := \begin{cases} \langle a \rangle\, X_{q'} & \text{if } q' \in S \\ \langle a \rangle\, \psi_{q'}^S & \text{if } q' \in Q_i \setminus S \\ \langle a \rangle\, \psi_{q'}^\emptyset & \text{if } q' \notin Q_i \end{cases}$$

$$tr_q^S(q', \Box, a) := \begin{cases} [a]\, X_{q'} & \text{if } q' \in S \\ [a]\, \psi_{q'}^S & \text{if } q' \in Q_i \setminus S \\ [a]\, \psi_{q'}^\emptyset & \text{if } q' \notin Q_i \end{cases}$$

Finally, we define $\varphi_{\mathcal{A}} := \psi_{q_I}^\emptyset$.

It should be clear that this scheme defines at most $n \cdot \|\mathcal{A}\| \cdot 2^{k-1}$ many different subformulas which bounds the size of $\varphi_{\mathcal{A}}$ accordingly, where k is the maximum of the cardinalities of the components. It is also not hard to see that $\varphi_{\mathcal{A}}$ is alternation-free because, by assumption, $\mathcal{A}$ is weak and therefore two variables X_q and $X_{q'}$ are bound by the same kind of fixpoint quantifier whenever q and q' belong to the same component Q_i. A close inspection shows that, whenever some $X_{q'}$ has a free occurrence in some ψ_q^S then we must have $q' \in S$ and therefore q and q' belong to the same component.

Finally, we give an invariant, which can be used to prove the correctness of the constructed formula $\varphi_{\mathcal{A}}$. For every $(2^{\mathcal{P}}, A)$-graph $\mathcal{G}$ with vertex set V, every vertex v, every component Q_i, every $q \in Q_i$, every $S \subseteq Q_i \setminus \{q\}$, and every variable interpretation $\sigma : \{X_{q'} \mid q' \in S\} \to 2^V$, we have $v \in [\![\psi_q^S]\!]_\sigma^{\mathcal{G}}$ iff there is a run of $\mathcal{A}$ on $\mathcal{G}$ starting at vertex v that is accepting according to the usual parity condition and the additional provision that any vertex of the form (u, q') in the run is immediately accepting if $u \in \sigma(X_{q'})$. It is possible to prove this by a combined induction on the topological structure of the strongly connected components in $\mathcal{A}$ and the size of S. The claim about the correctness of $\varphi_{\mathcal{A}}$ follows from the invariant. Note that $\varphi_{\mathcal{A}}$ contains no free variables and is thus satisfied by exactly those vertices from which $\mathcal{A}$ accepts $\mathcal{G}$. □

4. From Parity Automata to Weak Automata

In this section, we show that parity automata and weak automata have the same expressive power over *bottlenecked directed acyclic graphs* (BDAGs) with a bounded width. BDAGs are fundamental to this article as our collapse results rely on reductions of different structures—such as various classes of graphs and words—to BDAGs with a bounded width; see Section 5. The schematic form of BDAGs is illustrated in Figure 1. Their definition is as follows.

Definition 4.1. Let $\mathcal{G} = \big(V, (E_a)_{a\in A}, v_{\mathrm{I}}, \lambda\big)$ be a (Σ, A)-graph.

– $\mathcal{G}$ is a *directed acyclic graph* (DAG) if it does not contain cycles, that is, there are no vertices $v_0, \ldots, v_n \in V$ with $n \geq 1$ such that $v_0 = v_n$ and $(v_i, v_{i+1}) \in \bigcup_{a\in A} E_a$, for all $i \in \mathbb{N}$ with $0 \leq i < n$.

– $\mathcal{G}$ is a *bottlenecked DAG* (BDAG) of width $w \in \mathbb{N}$ if $\mathcal{G}$ is a DAG and V can be split into the pairwise disjoint sets $L_0, N_0, L_1, N_1, \ldots$ such that

(i) $\bigcup_{a\in A} E_a \subseteq \bigcup_{i\in\mathbb{N}} \big((L_i \times L_i) \cup (L_i \times N_i) \cup (N_i \times L_{i+1})\big)$,

(ii) $w = \sup\big\{|N_i| \;\big|\; i \in \mathbb{N}\big\}$, and

(iii) each L_i is well-founded, i.e., the graph obtained from $\mathcal{G}$ by restricting V to L_i has no infinite paths.

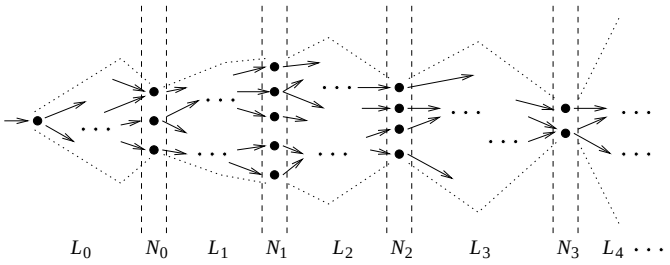

Figure 1: Bottlenecked directed acyclic graph (BDAG)

Note that BDAGs naturally define a *connectivity* measure, which is given by their widths: removing the vertices in the N_is disconnects the structure into DAGs in which all paths are finite and thus the infinite behavior described by the original structure is eliminated.

Before presenting our collapse results in Section 5, we need the following construction, parametric in $w \in \mathbb{N}$, that translates parity automata into language-equivalent weak automata with respect to the class of BDAGs of width at most w. In the following, let $w \in \mathbb{N}$ and let $\mathbf{BDAG}_{\leq w}$ be the class of $(2^{\mathcal{P}}, A)$-graphs that are BDAGs of width at most w. Moreover, for $n \in \mathbb{N}$, we abbreviate the set $\{0, 1, \ldots, n\}$ by $[n]$.

4.1. Rankings

Let $\mathcal{A} = (Q, \delta, q_{\mathrm{I}}, \alpha)$ be a $(\mathcal{P}, A)$-PA and $\varrho : R \to V \times Q$ a run of $\mathcal{A}$ on $\mathcal{G} \in \mathbf{BDAG}_{\leq w}$ with $\mathcal{G} = \big(V, (E_a)_{a \in A}, v_{\mathrm{I}}, \lambda\big)$. We assume that the run ϱ is *memoryless*, that is, equally labeled nodes in ϱ have isomorphic subtrees; formally, for all $x, y \in R$ if $\varrho(x) = \varrho(y)$ then for all $z \in \mathbb{N}^*$, whenever $xz \in R$ then $yz \in R$ and $\varrho(xz) = \varrho(yz)$. This assumption is without loss of generality for the parity condition [15, 28]. In particular, we have that $\mathcal{G} \in L_{\mathbf{BDAG}_{\leq w}}(\mathcal{A})$ iff there is an accepting memoryless run of $\mathcal{A}$ on $\mathcal{G}$. For the memoryless run ϱ, we define the graph $G^{\varrho} := (V^{\varrho}, E^{\varrho})$ with $V^{\varrho} := \big\{\varrho(x) \;\big|\; x \in R\big\}$ and $E^{\varrho} := \big\{\big(\varrho(x), \varrho(y)\big) \;\big|\; x, y \in R \text{ and } y \text{ is a child of } x\big\}$. The graph G^{ϱ} is a representation of the memoryless run ϱ in which equally labeled nodes are merged. Furthermore, since $\mathcal{G} \in \mathbf{BDAG}_{\leq w}$, we have that G^{ϱ} is a BDAG of width at most $|Q|w$. Note that G^{ϱ} contains at most $|Q|$ copies of each vertex $v \in V$.

Let $c \geq 0$. A state $q \in Q$ is *c-releasing* if $\alpha(q) > c$ and $\alpha(q) \not\equiv c \bmod 2$. An infinite path of the form $(h_0, q_0)(h_1, q_1)\ldots$ in G^{ϱ} is *c-dominated* if there is a state $q \in \inf(q_0 q_1 \ldots)$ with $\alpha(q) = c$ and no c-releasing state in $\inf(q_0 q_1 \ldots)$. The function $f : V^{\varrho} \to [2|Q|w]$ is a *c-ranking* for G^{ϱ} if the following conditions hold:

(i) For all $(h, q) \in V^{\varrho}$, if $f(h, q)$ is odd then $\alpha(q) \neq c$.

(ii) For all $v, v' \in V^{\varrho}$ with $v = (h, q)$, if $(v, v') \in E^{\varrho}$ and $f(v) < f(v')$ then q is c-releasing.

Thus, a ranking associates with each vertex in G^{ϱ} a number in $[2|Q|w]$, which we call also *rank*. The ranks along paths may only increase on vertices with c-releasing states and whenever a vertex's state has color c then a vertex's rank is odd. Note that each infinite path in G^{ϱ} either vists vertices with c-releasing states infinitely often or gets trapped in some rank. If all of these ranks are odd we say that a c-ranking is safe. Formally, a c-ranking f is *safe* if for every infinite path $(h_0, q_0)(h_1, q_1)\ldots$ in G^{ϱ}, either there is a state $q \in \inf(q_0 q_1 \ldots)$ that is c-releasing or there is an integer $n \in \mathbb{N}$ such that $f(h_n, q_n)$ is odd and $f(h_j, q_j) = f(h_n, q_n)$, for all $j \geq n$. We point out that the color $\alpha(q)$ of a state $q \in Q$ and the rank $f(h, q)$ of a vertex $(h, q) \in V^{\varrho}$ have different meanings. In particular, the parities of $\alpha(q)$ and $f(h, q)$ can differ.

It holds that the run ϱ is accepting iff for all odd $c \geq 1$, all infinite paths in G^{ϱ} are not c-dominated. In the following subsection, we reduce the problem of checking whether every infinite path in G^{ϱ} is not dominated by one specific color to the problem of checking the existence of a safe ranking for G^{ϱ}.

4.2. Reduction

This subsection is dedicated to the proof of the following theorem. The proof is based on ingredients that appear in the Kupferman and Vardi's correctness proof of the construction that translates alternating coBüchi word automata into weak alternating word automata [22]. Since our automata are parity automata that operate over BDAGs instead of words, some arguments are more subtle than in the coBüchi-word-automata case.

Theorem 4.2. *Let $c \geq 0$. Every infinite path in G^ϱ is not c-dominated iff there is a safe c-ranking for G^ϱ.*

The *if* direction of Theorem 4.2 is easy to see. Assume that there is a safe c-ranking for G^ϱ. It follows that every infinite path visits either infinitely often vertices with a color greater than c and different parity, or eventually avoids visiting vertices with color c. Hence, every infinite path in G^ϱ is not c-dominated.

For the *only-if* direction, one needs to show the existence of a safe c-ranking $f : V^\varrho \to [2|Q|w]$ for G^ϱ. We show its existence by constructing one. For assigning a rank to a vertex, we iteratively delete from G^ϱ vertices by alternatingly deleting vertices from which we cannot reach a vertex with the color c and from which only finitely many vertices are reachable. The assigned rank is the iteration in which the vertex is deleted. Since all infinite paths of the run G^ϱ go through infinitely many bottlenecks, it turns out that every vertex is eventually deleted. We only need finitely many iterations here until all vertices of G^ϱ are deleted since the width of the BDAG G^ϱ is at most $|Q|w$. We also remark that the construction does not work when the runs are trees since the number of iterations for deleting infinite paths might be unbounded. So, one would obtain infinitely many ranks, which in turn would lead to having infinitely many states in the resulting automaton.

For defining the c-ranking $f : V^\varrho \to [2|Q|w]$, we introduce the following definitions. Consider a subgraph $G = (V, E)$ of G^ϱ. We call a vertex $v \in V$ *well-founded* in G if every path in G starting from v has finite length. We call that $v \in V$ is *c-free* in G if no vertex with color c is reachable from v in G. Moreover, define an infinite sequence of subgraphs $G_0, G_1, G_2, \ldots$ of G^ϱ, where the vertices V_i and the edges E_i of a graph G_i are inductively defined as follows, for $i \in \mathbb{N}$:

- $V_0 := V^\varrho$ and $E_0 := E^\varrho \setminus \big\{(v, v') \in E^\varrho \;\big|\; v\text{'s state is } c\text{-releasing}\big\}$
- $V_{2i+1} := V_{2i} \setminus \{v \mid v \text{ is well-founded in } G_{2i}\}$ and $E_{2i+1} := E_{2i} \cap (V_{2i+1} \times V_{2i+1})$
- $V_{2i+2} := V_{2i+1} \setminus \{v \mid v \text{ is } c\text{-free in } G_{2i+1}\}$ and $E_{2i+2} := E_{2i+1} \cap (V_{2i+2} \times V_{2i+2})$

Note that G_0 is obtained from G^ϱ be removing edges that start from vertices with c-releasing states. By removing these edges, we have that every infinite path in G_0 does not visit vertices with c-releasing states. Furthermore, since every infinite path in G^ϱ is not c-dominated, every path in G_0 eventually avoids visiting vertices with color c.

Before defining the function $f : V^\varrho \to [2|Q|w]$ and showing that it is a safe c-ranking, we prove some properties about the graphs $G_0, G_1, G_2, \ldots$. Recall that by assumption, every infinite path in G^ϱ is not c-dominated. Thus, all the infinite paths in the G_is are also not c-dominated.

Lemma 4.3. *For every $i \in \mathbb{N}$, the graph G_{2i+1} is empty or has a vertex from which an infinite path starts that only visits c-free vertices.*

Proof. Assume that G_{2i+1} is not empty. By definition, every vertex in G_{2i+1} is not well-founded. Hence, every vertex in G_{2i+1} has at least one successor.

For the sake of contradiction, assume that there is no c-free vertex in G_{2i+1}. Consider some vertex (h_1, q_1) in G_{2i+1}. Let (h'_1, q'_1) be a successor of (h_1, q_1). Since (h'_1, q'_1) is not c-free, there is a vertex (h_2, q_2) reachable from (h'_1, q'_1) with $\alpha(q_2) = c$. Let (h'_2, q'_2) be a successor of (h_2, q_2). Since (h'_2, q'_2) is not c-free, there is a vertex (h_3, q_3) reachable from (h'_2, q'_2) with $\alpha(q_3) = c$. Let (h'_3, q'_3) be a successor of (h_3, q_3). If we continue this way, we construct an infinite path that does not visit any vertex with a c-releasing state but visits infinitely many vertices (h, q) with $\alpha(q) = c$. This path corresponds to a c-dominated path in G^ϱ, which contradicts the assumption that every infinite path in G^ϱ is not c-dominated. □

With the help of the next lemma, we show that one obtains the empty graph from G^ϱ in $2|Q|w$ steps by alternately removing infinite paths according to Lemma 4.3 and well-founded vertices from G^ϱ.

Lemma 4.4. *For every $i \in \mathbb{N}$, either the graph G_{2i+1} is empty or there is an integer $k \in \mathbb{N}$ such that for every $\ell \in \mathbb{N}$ with $\ell \geq k$, we have*

$$\left|V_{2i+2} \cap (N_\ell \times Q)\right| < \left|V_{2i+1} \cap (N_\ell \times Q)\right|,$$

where the vertices of the BDAG $\mathcal{G}$ are partitioned into the sets $L_0, N_0, L_1, N_1, \ldots$ according to Definition 4.1.

Proof. If G_{2i+1} is empty, we are done. Otherwise, consider an infinite path $\pi = (h_0, q_0)(h_1, q_1)\ldots \in (V \times Q)^\omega$ in G_{2i+1} that consists of only c-free vertices. This path exists by Lemma 4.3. Assume that $(h_{k'}, q_{k'}) \in (N_k \cup L_k) \times Q$, for some $k, k' \in \mathbb{N}$. For each $\ell \in \mathbb{N}$ with $\ell \geq k$, there is vertex $(h_{\ell'}, q_{\ell'}) \in N_\ell \times Q$ with $\ell' \in \mathbb{N}$ occurring in π. Note that these vertices $(h_{k'}, q_{k'})$ and $(h_{\ell'}, q_{\ell'})$ exist since π is infinite and the L_is are well-founded. By definition, we remove the vertex $(h_{\ell'}, q_{\ell'})$ from G_{2i+1}. □

Corollary 4.5. *Every vertex in $G_{2|Q|w}$ is well-founded and the graph $G_{2|Q|w+1}$ is empty.*

Proof. Observe that there are infinitely many N_is in the BDAG $\mathcal{G}$ and these N_is contain at most w vertices. Also, note that there is some $k \in \mathbb{N}$ such that $G_{2|Q|w}$ does not contain any vertex $(h, q) \in N_\ell \times Q$, for every $\ell > k$. This follows from Lemma 4.4 and the fact that G^ϱ contains at most $|Q|$ vertices (h, q) with $h \in V$.

Assume a vertex (h, q) in $G_{2|Q|w}$. Every infinite path from (h, q) in G_0 eventually visits a vertex (h', q') with $h' \in N_\ell$, for some $\ell > k$. The vertex (h', q') no longer exists in $G_{2|Q|w}$. It follows that (h, q) is well-founded in $G_{2|Q|w}$. By definition, we remove (h, q) in $G_{2|Q|w+1}$. We conclude that the graph $G_{2|Q|w+1}$ is empty. □

We define the c-ranking $f : V^\varrho \to [2|Q|w]$ as

$$f(v) := \begin{cases} 2i & \text{if } v \text{ is well-founded in } G_{2i}, \\ 2i+1 & \text{if } v \text{ is } c\text{-free in } G_{2i+1}. \end{cases}$$

By Corollary 4.5, the function f assigns to every vertex in G^ϱ a number in $[2|Q|w]$. Obviously, f fulfills the first condition (i) of the definition of a c-ranking. The second condition (ii) follows from the next lemma.

Lemma 4.6. *For all vertices $v, v' \in V^\varrho$, if v' is reachable in G^ϱ from v without visiting a vertex with a c-releasing state then $f(v') \leq f(v)$.*

Proof. Assume that $f(v) = i$. If $v' \notin V_i$ then there is some $j \in \mathbb{N}$ with $j < i$ such that $v' \in V_j$ and $v' \notin V_{j+1}$. By the definition of f, we have then $f(v') < i$. If $v' \in V_i$, we must show that $f(v') = i$. (a) Assume that i is even, that is, v is well-founded in G_i. Since v' is reachable from v without visiting a c-releasing vertex, v' is also well-founded in G_i. Note that if a vertex $v'' \neq v$ on the path from v to v' is removed in some previous iteration then v' would also have been removed in that iteration and we would have $f(v') < f(v)$. (b) Assume that i is odd, that is, v is c-free in G_i. Again, since v' is reachable from v without visiting a c-releasing vertex, v' is also c-free in G_i. In both cases, it follows that $f(v') = f(v)$. □

Finally, we show that the c-ranking f is safe, which completes the proof of Theorem 4.2.

Lemma 4.7. *Every infinite path in G^ϱ that does not visit infinitely many vertices with c-releasing states, gets trapped in an odd rank.*

Proof. Let $\pi = v_0 v_1 \ldots$ be an infinite path in G^ϱ that only visits finitely many vertices with c-releasing states. Due to Lemma 4.6, there is an integer $k \in \mathbb{N}$ such that $f(v_m) = f(v_k)$, for all $m \geq k$. The suffix $v_k v_{k+1} \ldots$ of the path π is an infinite path in $G_{f(v_k)}$. Assume that $f(v_k)$ is even. Every vertex v_m with $m \geq k$ is well-founded in $G_{f(v_k)}$. However, since π's suffix $v_k v_{k+1} \ldots$ is infinite the vertex v_k cannot be well-founded in $G_{f(v_k)}$. We conclude that $f(v_k)$ must be odd. □

4.3. Automaton Construction

In the following, we show that the existence of a safe ranking can be checked by a Büchi automaton. The ranks are guessed during a run with the states of the Büchi automaton. The conditions (i) and (ii) of a ranking are locally checked by the transition function of the automaton. With the acceptance condition of the automaton one checks whether the guessed ranking is safe. The construction details are given in the proof of the following theorem.

Theorem 4.8. *Let $c \geq 0$. There is a $(\mathcal{P}, A)$-BA $\mathcal{B}_c$ with $|Q| \cdot (2|Q|w+1)$ states and $L_{\mathbf{BDAG}_{\leq w}}(\mathcal{B}_c)$ equals*

$$\left\{\mathcal{G} \in \mathbf{BDAG}_{\leq w} \;\middle|\; \text{there is a memoryless run } \varrho \text{ of } \mathcal{A} \text{ on } \mathcal{G} \text{ such that } G^{\varrho} \text{ has a safe } c\text{-ranking}\right\}.$$

Furthermore, $\|\mathcal{B}_c\| \in \mathcal{O}\big(\|\mathcal{A}\| \cdot (|Q|w+1)\big)$.

Proof. We define $\mathcal{B}_c$ as $\big(Q \times [2|Q|w], \eta, p_{\mathrm{I}}, \beta\big)$, where p_{I}, η, and β are as follows:

- The initial state p_{I} is the tuple $(q_{\mathrm{I}}, 2|Q|w)$.
- To define the transition function η, we need the following two definitions. (1) For $q \in Q$ and $r, r' \in [2|Q|w]$, we write $r' \preceq_q r$ if either $r' \leq r$ or q is c-releasing. (2) For $\varphi \in \mathbf{B}^+\big(\mathcal{P} \cup \bar{\mathcal{P}} \cup (Q \times \{\Diamond, \Box\} \times A)\big)$, $q \in Q$, and $r \in [2|Q|w]$, we define $\mathrm{release}_q(\varphi, r)$ as the positive Boolean formula that we obtain by replacing each proposition $(p, \star, a)$ in φ by the disjunction $\bigvee_{r' \preceq_q r} \big((p, r'), \star, a\big)$. For $q \in Q$ and $r \in [2|Q|w]$, we define

$$\eta(q,r) := \begin{cases} \mathrm{release}_q\big(\delta(q), r\big) & \text{if } \alpha(q) \neq c \text{ or } r \text{ is even,} \\ \mathsf{ff} & \text{otherwise.} \end{cases}$$

- The acceptance condition is determined by $\beta : Q \times [2|Q|w] \to \{1, 2\}$ where

$$\beta(q,r) := \begin{cases} 2 & \text{if } q \text{ is } c\text{-releasing or } r \text{ is odd,} \\ 1 & \text{otherwise.} \end{cases}$$

Obviously, $\mathcal{B}_c$ has $|Q| \cdot (2|Q|w+1)$ states. An upper bound on the number of distinct subformulas in the positive Boolean formula $\eta(q,r)$ for $q \in Q$ and $r \in [2|Q|w]$ is $\mathcal{O}\big(m + |Q| \cdot (|Q|w+1)\big)$, where m is the number of distinct formulas in $\delta(q)$. Note that the disjunction $\bigvee_{r' \preceq_q r} \big((p, r'), \star, a\big)$ in $\eta(q,r)$, which replaces a proposition of the form $(p, \star, a)$ in $\delta(q)$, is a subformula of $\bigvee_{0 \leq r' \leq 2|Q|w} \big((p, r'), \star, a\big)\big)$. The disjunction $\bigvee_{0 \leq r' \leq 2|Q|w} \big((p, r'), \star, a\big)\big)$ has $\mathcal{O}(|Q|w+1)$ subformulas. Since we count multiple occurrences of the same subformula in the transitions of an automaton only once, we obtain that $\|\mathcal{B}_c\| \in \mathcal{O}\big(\|\mathcal{A}\| \cdot (|Q|w+1) + |Q| \cdot (|Q|w+1)\big) \subseteq \mathcal{O}\big(\|\mathcal{A}\| \cdot (|Q|w+1)\big)$. It remains to prove that $\mathcal{G} \in L_{\mathbf{BDAG}_{\leq w}}(\mathcal{B}_c)$ iff there is a run ϱ of $\mathcal{A}$ on $\mathcal{G}$ such that the graph G^{ϱ} has a safe c-ranking.

($\Rightarrow$) Let $\varrho' : R \to V \times \big(Q \times [2|Q|w]\big)$ be an accepting, memoryless run of $\mathcal{B}_c$ on $\mathcal{G} = \big(V, (E_a)_{a \in A}, v_{\mathrm{I}}, \lambda\big)$. We define the tree $\varrho : R \to V \times Q$ with $\varrho(x) := (h, q)$, for every $x \in R$ with $\varrho'(x) = \big(h, (q, r)\big)$, that is, the labels of the nodes in ϱ are the projections of the labels of ϱ' on $V \times Q$. The tree ϱ is a run of $\mathcal{A}$ on $\mathcal{G}$ since the transition function of $\mathcal{B}_c$ just annotates state of $\mathcal{A}$ by ranks. We can assume that there are no $x, y \in R$ with $\varrho'(x) = (h, (q, r))$, $\varrho'(y) = (h, (q, r))$, and $r \neq r'$. That is, the rank $r \in [2|Q|w]$ assigned by the run ϱ' to a vertex (h, q) in the graph G^{ϱ} representing ϱ is unique. We define $f(h, q) := r$.

It follows from the definition of η that f is a c-ranking for G^{ϱ}. Since ϱ' is accepting, on every branch π in ϱ' there are either c-releasing states or odd ranks which, in both cases, occur infinitely often. The case where π visits infinitely many vertices with c-releasing states is obvious. Assume that π visits only finitely many vertices with c-releasing states. Then, the ranks do not increase from some point onwards. Thus, they must eventually stabilize. We conclude that f is safe.

($\Leftarrow$) Let $f : V^{\varrho} \to [2|Q|w]$ be a safe c-ranking on the graph representation $G^{\varrho} = (V^{\varrho}, E^{\varrho})$ of the run $\varrho : R \to V \times Q$ of $\mathcal{A}$ on $\mathcal{G} = \big(V, (E_a)_{a \in A}, v_{\mathrm{I}}, \lambda\big)$. The idea is to attach the ranks given by f to the labels

of the nodes in ϱ to obtain an accepting run of $\mathcal{B}_c$ on $\mathcal{G}$. We define the tree $\varrho' : R \to V \times \big(Q \times [2|Q|w]\big)$ as $\varrho'(x) := \big(h, (q, f(h,q))\big)$, for $x \in R$ with $\varrho(x) = (h,q)$. However, ϱ' is not necessarily a run of $\mathcal{B}_c$ on $\mathcal{G}$ because of the following two cases.

1. The rank of (v_I, q_I) might be smaller than $2|Q|w$. In this case, ϱ' does not start in $\mathcal{B}_c$'s initial state, which is $(q_I, 2|Q|w)$.

2. Assume that there are vertices $h, h', h'' \in V$ with $(h,h'), (h,h'') \in E_a$, for some $a \in A$. Furthermore, assume $\varrho(x) = (h,p)$ and $\delta(p) = (q, \Box, a)$, for some node $x \in R$ and states $p, q \in Q$. The node x has children $y_1, y_2 \in R$ with $\varrho(y_1) = (h', q)$ and $\varrho(y_2) = (h'', q)$. According to the definition of ϱ', we have that $\varrho'(x) = \big(h, (p, f(h,p))\big)$, $\varrho'(y_1) = \big(h', (q, f(h',q))\big)$, $\varrho'(y_2) = \big(h'', (q, (f(h'',q)))\big)$. The rank of (h',q) might be different from the rank of (h'',q). However, for ϱ' being a run of $\mathcal{B}_c$, it is necessary to go to the same state, either $(q, f(h',q))$ or $(q, f(h'',q))$, when moving $\mathcal{B}_c$'s read-only head from the vertex h to the vertices h' and h''.

For both cases, we can easily adjust ϱ' so that we obtain a run of $\mathcal{B}_c$ on $\mathcal{G}$. We define the tree $\varrho'' : R \to V \times \big(Q \times [2|Q|w]\big)$ as follows. We addresse the first case by defining $\varrho''(\varepsilon) := \big(v_I, (q_I, 2|Q|w)\big)$. Whenever the second case arises, we attach the maximum of the ranks to the labels of the y_i nodes. More formally, let $y \in R$, with $\varrho(y) = (h', q)$. In case y is the child of $x \in R$, with $\varrho(x) = (h,p)$, and the chosen set $\mathcal{M}$ satisfying $\delta(p)$ in the run ϱ at the node x contains $(q, \Box, a)$, for some $a \in A$, we define $\varrho(y) := (h', q)$, and $(h,h') \in E_a$ as $\varrho''(y) := \big(h', (q, \max\{f(h'',q) \mid (h,h'') \in E_a\})\big)$. For all other nodes, ϱ'' is as ϱ'.

It is easy to see that ϱ'' is a run of $\mathcal{B}_c$ on $\mathcal{G}$. Observe that if p in the second case is not c-releasing then $\max\{f(h'',q) \mid (h,h'') \in E_a\} \leq f(h,q)$. Furthermore, if one of the ranks in $\{f(h'',q) \mid (h,h'') \in E_a\}$ is odd then $\alpha(q) \neq c$. The run ϱ'' is accepting. This follows from the assumption that f is safe. Every infinite path in G^{ϱ} that does not visit c-releasing vertices infinitely often gets trapped in an odd rank. Then, by the definition of β, every infinite branch in ϱ', and thus also in ϱ'', is accepting. □

4.4. Applications

The first application is to obtain weak automata from Büchi automata.

Lemma 4.9. *Let $\mathcal{A}$ be a $(\mathcal{P}, A)$-BA with n states. There is a $(\mathcal{P}, A)$-WA $\mathcal{B}$ with $n(2nw+1)$ states and $L_{\mathbf{BDAG}_{\leq w}}(\mathcal{B}) = L_{\mathbf{BDAG}_{\leq w}}(\mathcal{A})$. Furthermore, $\|\mathcal{B}\| \in \mathcal{O}\big(\|\mathcal{A}\| \cdot (nw+1)\big)$.*

Proof. We first construct from $\mathcal{A}$ the coBüchi automaton $\mathcal{C}$ by dualizing the transition function of $\mathcal{A}$ and its acceptance condition. The automaton $\mathcal{C}$ accepts the complement of $\mathcal{A}$. Recall that a coBüchi automaton only assigns the colors $\{0,1\}$ to its states. Second, let $\mathcal{B}_1$ be the Büchi automaton for $\mathcal{C}$ obtained from Theorem 4.8 for the only odd color 1. This automaton is weak. This can be seen by the definition of $\mathcal{B}_1$'s transition function and acceptance condition. Note that $\mathcal{C}$ does not have 1-releasing states. From $\mathcal{B}_1$'s definition it also follows easily that $\mathcal{B}_1$ has $n(2nw+1)$ states and $\|\mathcal{B}_1\| \in \mathcal{O}\big(\|\mathcal{A}\| \cdot (nw+1)\big)$. It follows from Theorem 4.2 that $L_{\mathbf{BDAG}_{\leq w}}(\mathcal{B}_1) = L_{\mathbf{BDAG}_{\leq w}}(\mathcal{C})$. Finally, we dualize $\mathcal{B}_1$. The resulting automaton is again weak, has the same size as $\mathcal{B}_1$, and accepts the language $L_{\mathbf{BDAG}_{\leq w}}(\mathcal{A})$. □

We now show how to combine Büchi automata for different odd colors from Theorem 4.8 so that they simultaneously check the existence of safe rankings.

Lemma 4.10. *Let $\mathcal{A}$ be a $(\mathcal{P}, A)$-PA with n states and index k. There is a $(\mathcal{P}, A)$-BA $\mathcal{B}$ with $\mathcal{O}\big(kn(2nw+1)^{\lceil k/2 \rceil}\big)$ states and $L_{\mathbf{BDAG}_{\leq w}}(\mathcal{B}) = L_{\mathbf{BDAG}_{\leq w}}(\mathcal{A})$. Moreover, $\|\mathcal{B}\| \in \mathcal{O}\big(k\|\mathcal{A}\|(2nw+1)^{\lceil k/2 \rceil}\big)$.*

Proof. Assume that $\mathcal{A} = (Q, \delta, q_{\mathrm{I}}, \alpha)$ and that the odd colors of $\mathcal{A}$ are $c_1, \ldots, c_\ell \in \mathbb{N}$, for some $\ell \leq \lceil k/2 \rceil$.

We extend the automaton construction from Theorem 4.8 so that the constructed automaton $\mathcal{C}$ simultaneously checks the existence of a run ϱ of $\mathcal{A}$ on the given input such that G^{ϱ} has c_i-safe rankings, with $i \in \{1, \ldots, \ell\}$. The states of $\mathcal{C}$ have the form $(q, r_1, \ldots, r_\ell)$, where q is a state of $\mathcal{A}$ and the r_is are ranks as

in the Büchi automata $\mathcal{B}_{c_i}$ from Theorem 4.8, that is, $r_i \in [2|Q|w]$. $\mathcal{C}$'s initial state is $(q_\mathrm{I}, 2|Q|w, \ldots, 2|Q|w)$. Its transition function η is defined as

$$\eta(q, r_1, \ldots, r_\ell) := \begin{cases} \mathrm{release}_q\big(\delta(q), r_1, \ldots, r_\ell\big) & \text{if } \alpha(q) \neq c_i \text{ or } r_i \text{ is even, for all } i \in \{1, \ldots, \ell\}, \\ \mathsf{ff} & \text{otherwise,} \end{cases}$$

where $\mathrm{release}_q$ is the extension from the construction of a Büchi automaton $\mathcal{B}_{c_i}$ in Theorem 4.8 to multiple ranks. Namely, $\mathrm{release}_q(\varphi, r_1, \ldots, r_\ell)$ replaces every proposition of the form $(p, \star, a)$ in the positive Boolean formula φ by the disjunction $\bigvee_{(r'_1,\ldots,r'_\ell) \preceq_q (r_1,\ldots,r_\ell)} \big((p, r'_1, \ldots, r'_\ell), \star, a\big)$, where $(r'_1, \ldots, r'_\ell) \preceq_q (r_1, \ldots, r_\ell)$ iff $r'_i \leq r_i$ or q is c_i-releasing, for all $i \in \{1, \ldots, \ell\}$.

The automaton $\mathcal{C}$ is a so-called generalized Büchi automaton, that is, its acceptance condition is the conjunction of the Büchi acceptance conditions $\beta_1, \ldots, \beta_\ell : Q \times [2|Q|w]^\ell \to \{1, 2\}$. The ith acceptance condition is $\beta_i(q, r_1, \ldots, r_\ell) := 2$ if q is c_i-releasing or r_i is odd and $\beta_i(q, r_1, \ldots, r_\ell) := 1$, otherwise.

The automaton $\mathcal{C}$ has $n(2nw+1)^\ell$ states. An upper bound on $\|\mathcal{C}\|$ is $\mathcal{O}\big(\|\mathcal{A}\|(nw+1) + n(2nw+1)^\ell\big) \subseteq \mathcal{O}(\|\mathcal{A}\|(2nw+1)^\ell\big)$. With Theorem 4.2 we conclude that $\mathcal{C}$ accepts the language $L_{\mathbf{BDAG}_{\leq w}}(\mathcal{A})$. It is standard to obtain from $\mathcal{C}$ an equivalent Büchi automaton $\mathcal{B}$ with $\mathcal{O}\big(kn(2nw+1)^{\lceil k/2 \rceil}\big)$ states and $\|\mathcal{B}\| \in \mathcal{O}\big(k\|\mathcal{A}\|(2nw+1)^{\lceil k/2 \rceil}\big)$. □

5. Collapse Results

By consecutively applying the previously presented translations to a $\mathcal{L}_\mu$ sentence, we obtain that $\mathcal{L}_\mu$'s alternation hierarchy over any class only containing BDAGs of width at most w collapses to its alternation-free fragment, for a fixed $w \in \mathbb{N}$.

Theorem 5.1. *Let $w \geq 2$ and $\mathbf{U} \subseteq \mathbf{BDAG}_{\leq w}$. For every sentence φ, there is an alternation-free sentence ψ of size $2^{w^{\mathcal{O}(|\varphi| \cdot \mathrm{ad}(\varphi))}}$ such that $L_\mathbf{U}(\psi) = L_\mathbf{U}(\varphi)$. If φ is guarded, then the size of ψ is $2^{(|\varphi| \cdot w)^{\mathcal{O}(\mathrm{ad}(\varphi))}}$.*

Proof. Suppose φ is guarded and let $n := |\varphi|$ and $k := \mathrm{ad}(\varphi)$. We construct the parity automaton $\mathcal{A}_\varphi$ with $\|\mathcal{A}_\varphi\| \in \mathcal{O}(n)$ and $\mathrm{ind}(\mathcal{A}_\varphi) = k + 1$ (Theorem 3.2). Then, we construct from $\mathcal{A}_\varphi$ the Büchi automaton $\mathcal{B}_\varphi$ with $\|\mathcal{B}_\varphi\| \in (nw)^{\mathcal{O}(k)}$ (Lemma 4.10). From $\mathcal{B}_\varphi$, we obtain the weak automaton $\mathcal{C}_\varphi$ with $\|\mathcal{C}_\varphi\| \in (nw)^{\mathcal{O}(k)} \cdot \big(2(nw)^{\mathcal{O}(k)}w + 1\big) \subseteq (nw)^{\mathcal{O}(k)}$ (Lemma 4.9). Finally, we construct the alternation-free sentence ψ with $|\psi| \in (nw)^{\mathcal{O}(k)} \cdot 2^{(nw)^{\mathcal{O}(k)}} \subseteq 2^{(nw)^{\mathcal{O}(k)}}$ (Theorem 3.3). By construction, $L_{\mathbf{BDAG}_{\leq w}}(\varphi) = L_{\mathbf{BDAG}_{\leq w}}(\psi)$. Since $\mathbf{U} \subseteq \mathbf{BDAG}_{\leq w}$, we have that $L_\mathbf{U}(\varphi) = L_\mathbf{U}(\psi)$. When φ is not guarded we first transform it into guarded form (Lemma 3.1) under an at most exponential blow-up, and replace n by $2^{\mathcal{O}(n)}$ which yields an upper bound of $2^{w^{\mathcal{O}(nk)}}$. □

In the following, we derive from Theorem 5.1 further classes of structures over which the alternation hierarchy of $\mathcal{L}_\mu$ collapses to the alternation-free fragment. We rely here also on the well known fact that $\mathcal{L}_\mu$ cannot distinguish between structures that are bisimilar.

Recall that the (Σ, A)-graphs $\mathcal{G} = \big(V, (E_a)_{a \in A}, v_\mathrm{I}, \lambda\big)$ is *bisimilar* to $\mathcal{G}' = \big(V', (E'_a)_{a \in A}, v'_\mathrm{I}, \lambda'\big)$ if there is a relation $R \subseteq V \times V'$ with the following properties. (i) $\lambda(v) = \lambda'(v')$, for all $(v, v') \in R$, (ii) $(v_\mathrm{I}, v'_\mathrm{I}) \in R$, (iii) for all $u, v \in V$, $u' \in V'$, and $a \in A$, if $(u, v) \in E_a$ and $(u, u') \in R$ then $(u', v') \in E'_a$ and $(v, v') \in R$, for some $v' \in V'$, and (iv) for all $u', v' \in V'$, $u \in V$, and $a \in A$, if $(u', v') \in E'_a$ and $(u, u') \in R$ then $(u, v) \in E_a$ and $(v, v') \in R$, for some $v \in V$. We call R a *bisimulation relation*. It is easy to see that (i) every graph is bisimilar to itself, (ii) if a graph $\mathcal{G}$ is bisimilar to a graph $\mathcal{G}'$, then $\mathcal{G}'$ is bisimilar to $\mathcal{G}$, and (iii) if a graph $\mathcal{G}$ is bisimilar to a graph $\mathcal{G}'$ and $\mathcal{G}'$ is bisimilar to a graph $\mathcal{G}''$, then $\mathcal{G}$ is bisimilar to $\mathcal{G}''$.

Theorem 5.2. *Let R be a bisimulation relation between the (Σ, A)-graphs $\mathcal{G}$ and $\mathcal{G}'$. For vertices v and v' of $\mathcal{G}$ and $\mathcal{G}'$, respectively, if $(v, v') \in R$ then $v \in [\![\varphi]\!]^\mathcal{G}$ iff $v' \in [\![\varphi]\!]^{\mathcal{G}'}$, for every sentence φ.*

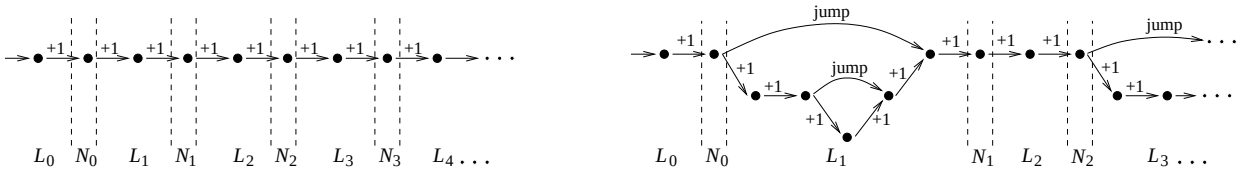

Figure 2: BDAG representation of infinite words (left) and infinite nested words (right)

5.1. Infinite Nested Words

We view infinite words over the alphabet Σ as $(\Sigma, \{+1\})$-graphs with the vertex set $\mathbb{N}$, the source 0, and where the edges of the action $+1$ are given by the successor relation. Let $\mathbf{W}$ be the set of these $(\Sigma, \{+1\})$-graphs. Nested words [2] extend words with a hierarchical structure. Such a hierarchical structure over an infinite word can be given through a *matching relation* $\curvearrowright \subseteq \mathbb{N} \times \mathbb{N}$, which must satisfy the following conditions for all $i, j \in \mathbb{N}$: (1) if $i \curvearrowright j$ then $i < j$, (2) $|\{k \mid i \curvearrowright k\}| \leq 1$ and $|\{k \mid k \curvearrowright j\}| \leq 1$, and (3) if $i \curvearrowright j$ then there are no $i', j' \in \mathbb{N}$ with $i' \curvearrowright j'$ and $i < i' \leq j < j'$. Let $\mathbf{NW}$ be the class of infinite nested words. Thus, $\mathbf{NW}$ consists of $(\Sigma, \{+1, \text{jump}\})$-graphs $\mathcal{G}$, where the vertex set is $\mathbb{N}$, the source is 0, and the edges for the actions $+1$ and jump are given by the successor relation and a matching relation, respectively.

It is easy to see that every graph in $\mathbf{W}$ is a BDAG of width 1. In the following we show that even every graph in $\mathbf{NW}$ is a BDAG of width 1. See Figure 2 for an illustration. By Theorem 5.1, it follows then that the $\mathcal{L}_\mu$ alternation hierarchy over these structures collapses to the alternation-free fragment. This improves prior results in [3, 7] on the expressivity of $\mathcal{L}_\mu$ over infinite nested words. In fact, we establish a more general result. The collapse results for words and nested words are special instances.

Let A be a non-empty finite set of actions, Σ be an alphabet, and $\mathbf{L}$ be the class of (Σ, A)-graphs with vertex set $\mathbb{N}$ and source 0. Furthermore, we require that edges only connect vertices with larger vertices, that is, if (v, v') is an edge then $v < v'$, and that every vertex has a finite out-degree, that is, for each vertex v, there are only finitely many edges of the form (v, v'). For a vertex $u \in \mathbb{N}$ of a graph $\mathcal{G} \in \mathbf{L}$, we define $J(u) := \big\{(v, v') \in \bigcup_{a \in A} E_a \;\big|\; v < u < v'\big\}$. Intuitively, $J(u)$ contains the edges that jump over u.

Lemma 5.3. *Let $\mathcal{G}$ be a (Σ, A)-graph in $\mathbf{L}$. If there is an infinite set $U \subseteq \mathbb{N}$ and $w \in \mathbb{N}$ such that $|J(u)| \leq w$, for all $u \in U$, then $\mathcal{G}$ is bisimilar to a BDAG of width at most $w + 1$. Furthermore, $\mathcal{G}$ is a BDAG of width at most 1 if $|J(u)| = 0$, for all $u \in U$.*

Proof. Let $\mathcal{G}$ be the (Σ, A)-graph $(\mathbb{N}, (E_a)_{a \in A}, 0, \lambda)$. Without loss of generality, we assume that (1) $0 \notin U$, (2) if $u \in U$ then $u + 1 \notin U$, and (3) vertices in U do not jump over vertices in U, that is, for every $v \in U$ there are no $u, v' \in U$ with $(v, v') \in J(u)$. We inductively define an infinite sequence of vertices $u_0, u_1, \ldots$ in U by $u_0 := \min U$ and

$$u_i := \min \big\{u \in U \;\big|\; u > \max\{u_{i-1}\} \cup \{v'' \in \mathbb{N} \,|\, (v, v') \in J(u_{i-1}) \text{ and either } v' = v'' \text{ or } (v', v'') \in \textstyle\bigcup_{a \in A} E_a\}\big\},$$

for $i > 0$.

For each $i \in \mathbb{N}$, the set $J(u_i)$ contains at most w edges. Assume $J(u_i) = \big\{(v_{i,1}, v'_{i,1}), \ldots, (v_{i,n_i}, v'_{i,n_i})\big\}$, for some $n_i \in \mathbb{N}$ with $n_i \leq w$. We obtain the (Σ, A)-graph $\mathcal{G}'$ from $\mathcal{G}$ as follows. For each vertex $v'_{i,j}$, with $i \in \mathbb{N}$ and $j \leq n_i$, we add a new vertex $\hat{v}'_{i,j}$. Furthermore, we redirect the edge from $v_{i,j}$ to $v'_{i,j}$, which jumps over u_i, to its copy $\hat{v}'_{i,j}$, that is, we delete the edge $(v_{i,j}, v'_{i,j})$ and add the edge $(v_{i,j}, \hat{v}'_{i,j})$. For every edge $(v'_{i,j}, v'')$, where v'' is some vertex, we also add the new edge $(\hat{v}'_{i,j}, v'')$. Note that by the construction of the sequence $u_0, u_1, \ldots$, we have that $v'' \leq u_{i+1}$.

It is easy to see that $\mathcal{G}$ and $\mathcal{G}'$ are bisimilar. Furthermore, $\mathcal{G}'$ is a BDAG of width at most $w+1$. Its vertex set can be partitioned into the layers $L_0, N_0, L_1, N_1, \ldots$ as follows: for $i \in \mathbb{N}$, let $N_i := \{u_i, \hat{v}'_{i,1}, \ldots, \hat{v}'_{i,n_i}\}$ and $L_i := \{v \in \mathbb{N} \mid u_{i-1} < v < u_i\}$, with $u_{-1} := -1$. If $|J(u)| = 0$, for all $u \in U$, it follows that $|N_i| = 1$, for all $i \in \mathbb{N}$. Since $\mathcal{G}'$ is isomorphic to $\mathcal{G}$ in this case, we conclude that $\mathcal{G}$ is a BDAG of width at most 1. ☐

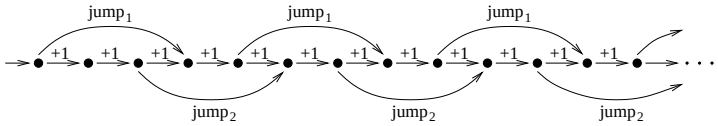

Figure 3: Multiple matching relations

Theorem 5.4. *For every sentence φ, there is an alternation-free sentence ψ such that $L_{\mathbf{NW}}(\varphi) = L_{\mathbf{NW}}(\psi)$.*

Proof. By Lemma 5.3 and the Theorems 5.2 and 5.1, it suffices to show that every $(\Sigma, \{+1, \text{jump}\})$-graph $\mathcal{G} \in \mathbf{NW}$ has an infinite vertex set $U \subseteq \mathbb{N}$ with $|J(u)| = 0$, for every $u \in U$.

Let $\curvearrowright \subseteq \mathbb{N} \times \mathbb{N}$ be a matching relation. We define the vertices $u_0, u_1, \ldots$ of the set U inductively. Let $u_0 := 0$. For $k > 0$, we define $u_k := j$ if $u_{k-1} \curvearrowright j$, for some $j \in \mathbb{N}$. Note that this j is uniquely determined and $j > u_{k-1}$. Otherwise, if there is no $j \in \mathbb{N}$ with $u_{k-1} \curvearrowright j$, we define $u_k := u_{k-1} + 1$. Obviously, there are no edges that jump over the vertices in U, that is, $|J(u_k)| = 0$, for every $k \in \mathbb{N}$. □

Remark 5.5. It also follows from Lemma 5.3 that $\mathcal{L}_\mu$'s alternation hierarchy collapses over graphs that are, for example, of the form as depicted in Figure 3. Note that we have here $|J(u)| \leq 1$, for each vertex $u \in \mathbb{N}$. However, it remains open whether the hierarchy collapses in general over the class of infinite multiply-nested words [6, 26], that is, infinite words that can have more than one matching relation.

5.2. Finite Graphs with Bounded Feedback Sets

We now consider classes of finite graphs that can be unfolded to bisimilar BDAGs with bounded width. The width of these BDAGs is characterized by the cardinality of a feedback vertex set of the original folded graph. A set $F \subseteq V$ is a *feedback vertex set* (FVS) of $\mathcal{G} = \big(V, (E_a)_{a \in A}, v_\text{I}, \lambda\big)$ if the removal of the vertices in F separates $\mathcal{G}$ into a set of finite DAGs. Finite DAGs have the empty set as an FVS.

Lemma 5.6. *Every finite (Σ, A)-graph with an FVS of cardinality $k \in \mathbb{N}$ has a bisimilar BDAG of width at most k.*

Proof. Let $\mathcal{G}$ be a (Σ, A)-graph $\big(V, (E_a)_{a \in A}, v_\text{I}, \lambda\big)$ with the FVS $F \subseteq V$ of cardinality k. Furthermore, let $\mathcal{T}$ be the tree unfolding of $\mathcal{G}$ and let R be the canonical bisimulation relation between the vertices of $\mathcal{G}$ and $\mathcal{T}$.

To build a BDAG $\mathcal{D}$ of width at most k, we identify some bisimilar subtrees in $\mathcal{T}$ and merge their roots. In more detail, we obtain $\mathcal{D}$ as follows. Visit the vertices on every branch of $\mathcal{T}$ by starting from the root of the tree and going to the successor vertices until (i) two vertices in the unfolding of a vertex in $v \in F$ occur in the branch or until (ii) a vertex with no successor is reached. Because F is a FVS of $\mathcal{G}$ either (i) or (ii) are true in a branch after visiting finitely many of the branch's vertices. Let T_0 be this initial finite subtree of $\mathcal{T}$. Based on T_0, construct a DAG by merging some vertices of T_0. Namely, merge the leaves t and t' of T_0 if they have successors in $\mathcal{T}$ and stem from the same vertex $v \in F$ in $\mathcal{G}$, that is, $(v, t) \in R$ and $(v, t') \in R$. We call the resulting DAG D_0 the layer 0 of $\mathcal{D}$. Furthermore, let N_0 be the set of D_0's terminal vertices that stem from vertices in $\mathcal{T}$ with successors and let L_0 be the set of the remaining vertices of D_0. Now, the steps described above for constructing D_0 are carried out for the $|N_0|$ vertices in $\mathcal{T}$ to construct the DAG D_1, the layer 1 of $\mathcal{D}$. Instead of a single tree T_0 we obtain a forest of finitely many finite trees. For this forest, we merge the trees' leaves analogously, obtaining D_1 and the vertex sets N_1 and L_1. In general, those steps are repeated with respect to D_i, with $i \in \mathbb{N}$, to construct D_{i+1}, so that a DAG $\mathcal{D}$ with the disjoint sets $L_0, N_0, L_1, N_1, L_2, N_2, \ldots$ of vertices is built.

To see that $\mathcal{G}$ and $\mathcal{D}$ are bisimilar note that either (i) a vertex in $\mathcal{D}$ stems directly from a vertex of $\mathcal{G}$'s tree unfolding and every graph is bisimilar to its own tree unfolding, or (ii) a vertex in $\mathcal{D}$ is obtained by merging different occurrences of the same vertex in $\mathcal{G}$ and every vertex of a graph is bisimilar to itself. To see that $\mathcal{D}$ is a BDAG of width at most k observe that $\sup\{|N_i| \mid i \in \mathbb{N}\} \leq k$. This is the case because, due to merging, every N_i does not contain more than one occurrence of a vertex in F. □

Using the previous lemma, we can, then, obtain the following result.

Theorem 5.7. *Let $k \in \mathbb{N}$ and $\mathbf{U}$ be a class finite $(2^{\mathcal{P}}, A)$-graphs with FVS of cardinality at most k. For every sentence φ, there is an alternation-free sentence ψ such that $L_{\mathbf{U}}(\psi) = L_{\mathbf{U}}(\varphi)$.*

Proof. It follows from Lemma 5.6 that each graph in $\mathbf{U}$ can be unfolded into a BDAG of width at most k. We then apply Theorem 5.1 to this obtained class of unfolded BDAGs. □

Since collapse results carry over to smaller classes of structures, Theorem 5.7 implies the collapse of the alternation hierarchy over the smaller class of undirected graphs with bounded feedback vertex sets.

Finally, we consider classes of graphs that can be decomposed by removing a bounded number of edges. Let $\mathcal{G} = \big(V, (E_a)_{a\in A}, v_{\mathrm{I}}, \lambda\big)$ be a (Σ, A)-graph. A set $F \subseteq \bigcup_{a\in A} E_a$ is a *feedback edge set* (FES) of $\mathcal{G}$ if the removal of the edges in F separates $\mathcal{G}$ into a set of finite DAGs. Since every graph with FES F has also a FVS of cardinality at most $|F|$, we obtain the following corollary.

Corollary 5.8. *Let $k \in \mathbb{N}$ and $\mathbf{U}$ be a class of finite $(2^{\mathcal{P}}, A)$-graphs with FESs of cordinality at most k. For every sentence φ, there is an alternation-free sentence ψ such that $L_{\mathbf{U}}(\psi) = L_{\mathbf{U}}(\varphi)$.*

6. Conclusion

The results in this article focus on $\mathcal{L}_\mu$'s expressivity. By generalizing and utilizing automata-theoretic methods, we have unified, generalized, and strengthened prior collapse results of $\mathcal{L}_\mu$'s alternation hierarchy, namely, the results on finite acyclic directed graphs [27], infinite words [19], and infinite nested words [3]. Future work includes to investigate whether the presented automaton construction for eliminating odd colors in parity automata can be generalized and to explore over which other classes of structures such generalizations apply. It would also be interesting to find other classes of structures where for each structure in such a class one can always find a (bisimilar) bottlenecked directed acyclic graph of bounded width. The ultimate goal is to characterize the classes of graphs over which the alternation-free fragment has already the same expressivity as the full μ-calculus.

We mainly ignore complexity issues in this article, except the established upper bounds on the sizes of the resulting alternation-free formulas. It remains as future work to provide lower bounds and to investigate the computational complexity of the satisfiability problem for $\mathcal{L}_\mu$ with respect to classes of structures over which its alternation hierarchy collapses.

Acknowledgments. This article is an extended version of the conference paper [17]. The authors thank Christian Dax for initial discussions on the topic of this article, Julian Bradfield for advice on the alternation hierarchy and Florian Bruse for helping to correct the translation from weak automata into the alternation-free μ-calculus. We also thank the anonymous reviewers for pointing us to some flaws in earlier versions of the article. This work was done while Felix Klaedtke was at ETH Zurich. Finally, Julian Gutierrez acknowledges the support of EPSRC grant EP/G012962/1 (at Edinburgh) and ERC grants ECSYM (at Cambridge) and RACE (at Oxford). Martin Lange acknowledges the support of the ERC grant MCUNLEASH.

References

[1] L. Alberucci and A. Facchini. The modal μ-calculus over restricted classes of transition systems. *J. Symb. Log.*, 74(4):1367–1400, 2009.

[2] R. Alur and P. Madhusudan. Adding nesting structure to words. *J. ACM*, 56(3):1–43, 2009.

[3] M. Arenas, P. Barceló, and L. Libkin. Regular languages of nested words: Fixed points, automata, and synchronization. *Theor. Comput. Syst.*, 49(3):639–670, 2011.

[4] A. Arnold. The modal μ-calculus alternation-depth is strict on binary trees. *Theor. Inform. Appl.*, 33(4–5):329–339, 1999.

[5] A. Arnold and D. Niwiński. *Rudiments of μ-calculus*, volume 146 of *Studies in Logic and the Foundations of Mathematics*. North-Holland, 2001.

[6] A. Blass and Y. Gurevich. A note on nested words. Technical Report MSR-TR-2006-139, Microsoft Research, 2006.

[7] L. Bozzelli. Alternating automata and a temporal fixpoint calculus for visibly pushdown languages. In *Proceedings of 18th International Conference on Concurrency Theory (CONCUR'07)*, volume 4703 of *Lect. Notes Comput. Sci.*, pages 476–491, 2007.

[8] J. C. Bradfield. The modal mu-calculus alternation hierarchy is strict. *Theoret. Comput. Sci.*, 195(2):133–153, 1998.

[9] J. C. Bradfield. Fixpoint alternation: arithmetic, transition systems, and the binary tree. *Theor. Inform. Appl.*, 33(4–5):341–356, 1999.

[10] F. Bruse, O. Friedmann, and M. Lange. Guarded transformation for the modal mu-calculus. *CoRR*, abs/1305.0648, 2013.

[11] R. Cleaveland and B. Steffen. A linear-time model-checking algorithm for the alternation-free modal μ-calculus. In *Proceedings of the 3rd International Conference on Computer Aided Verification (CAV'91)*, volume 575 of *Lect. Notes Comput. Sci.*, pages 48–58. Springer, 1992.

[12] G. D'Agostino and G. Lenzi. On the μ-calculus over transitive and finite transitive frames. *Theoret. Comput. Sci.*, 411(50):4273–4290, 2010.

[13] G. D'Agostino and G. Lenzi. On modal μ-calculus over reflexive symmetric graphs. *J. Logic Comput.*, 23(3):445–455, 2013.

[14] A. Dawar and M. Otto. Modal characterisation theorems over special classes of frames. *Ann. Pure Appl. Logic*, 161(1):1–42, 2009.

[15] E. A. Emerson and C. S. Jutla. Tree automata, mu-calculus and determinacy. In *Proceedings of the 32nd Annual Symposium on Foundations of Computer Science (FOCS'91)*, pages 368–377. IEEE Computer Society, 1991.

[16] E. A. Emerson and C.-L. Lei. Efficient model checking in fragments of the propositional mu-calculus. In *Proceedings of the 1st Symposium on Logic in Computer Science (LICS'86)*, pages 267–278. IEEE Computer Society, 1986.

[17] J. Gutierrez, F. Klaedtke, and M. Lange. The μ-calculus alternation hierarchy collapses over structures with restricted connectivity. In *Proceedings of the 3rd International Symposium on Games, Automata, Logics and Formal Verification (GandALF'12)*, volume 96 of *Elec. Proc. Theo. Comput. Sci.*, pages 113–126. `eptcs.org`, 2012.

[18] D. Janin and I. Walukiewicz. On the expressive completeness of the propositional mu-calculus with respect to monadic second order logic. In *Proceedings of the 7th International Conference on Concurrency Theory (CONCUR'96)*, volume 1119 of *Lect. Notes Comput. Sci.*, pages 263–277. Springer, 1996.

[19] R. Kaivola. Axiomatising linear time mu-calculus. In *Proceedings of the 6th International Conference on Concurrency Theory (CONCUR'95)*, volume 962 of *Lect. Notes Comput. Sci.*, pages 423–437. Springer, 1995.

[20] D. Kozen. Results on the propositional μ-calculus. *Theoret. Comput. Sci.*, 27(3):333–354, 1983.

[21] O. Kupferman and M. Y. Vardi. Weak alternating automata and tree automata emptiness. In *Proceedings of the 30th Annual ACM Symposium on the Theory of Computing (STOC'98)*, pages 224–233. ACM Press, 1998.

[22] O. Kupferman and M. Y. Vardi. Weak alternating automata are not that weak. *ACM Trans. Comput. Log.*, 2(3):408–429, 2001.

[23] O. Kupferman and M. Y. Vardi. From linear time to branching time. *ACM Trans. Comput. Log.*, 6(2):273–294, 2005.

[24] G. Lenzi. A hierarchy theorem for the μ-calculus. In *Proceedings of the 23rd International Colloquium on Automata, Languages and Programming (ICALP'96)*, volume 1099 of *Lect. Notes Comput. Sci.*, pages 87–97. Springer, 1996.

[25] A. Mader. *Verification of Modal Properties Using Boolean Equation Systems*. PhD thesis, Munich, University of Technology, 1997.

[26] P. Madhusudan and G. Parlato. The tree width of auxiliary storage. In *Proceedings of the 38th ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages (POPL'11)*, pages 283–294. ACM Press, 2011.

[27] R. Mateescu. Local model-checking of modal mu-calculus on acyclic labeled transition systems. In *Proceedings of the 8th International Conference on Tools and Algorithms for the Construction and Analysis of Systems (TACAS'02)*, volume 2280 of *Lect. Notes Comput. Sci.*, pages 281–295. Springer, 2002.

[28] D. E. Muller and P. E. Schupp. Alternating automata on infinite trees. *Theoret. Comput. Sci.*, 54(2–3):267–276, 1987.

[29] D. Niwiński. On fixed-point clones. In *Proceedings of the 13th International Colloquium on Automata, Languages and Programming (ICALP'86)*, volume 226 of *Lect. Notes Comput. Sci.*, pages 464–473. Springer, 1986.

[30] D. Niwiński. Fixed points vs. infinite generation. In *Proceedings of the 3rd Annual Symposium on Logic in Computer Science (LICS'88)*, pages 402–409. IEEE Computer Society, 1988.

[31] I. Walukiewicz. Completeness of Kozen's axiomatization of the propositional μ-calculus. In *Proceedings of the 10th Annual IEEE Symposium on Logic in Computer Science (LICS'95)*, pages 14–24. IEEE Computer Society, 1995.

[32] T. Wilke. Alternating tree automata, parity games, and modal μ-calculus. *Bull. Soc. Math. Belg.*, 8(2):359–391, 2001.