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

Julian Gutierrez
University of Cambridge, United Kingdom

Felix Klaedtke
ETH Zurich, Switzerland

Martin Lange
University of Kassel, Germany

It is known that the alternation hierarchy of least and greatest fixpoint operators in the μ-calculus is strict. However, the strictness of the alternation hierarchy does not necessarily carry over when considering restricted classes of structures. A prominent instance is the class of infinite words over which the alternation-free fragment is already as expressive as the full μ-calculus. Our current understanding of when and why the μ-calculus alternation hierarchy is not strict is limited. This paper 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. Our collapse results 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.

1 Introduction

The μ-calculus [15], hereafter $\mathscr{L}_\mu$, 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 $\mathscr{L}_\mu$ an expressive logic, which subsumes many dynamic, temporal, and description logics such as PDL and CTL*. In fact, $\mathscr{L}_\mu$ is essentially the most expressive logic of that kind as it can express, up to bisimulation equivalence, all MSO-definable properties [13].

An important question about the expressivity of $\mathscr{L}_\mu$ is whether more alternation—the nesting of mutually dependent least and greatest fixpoint operators in formulas—gives more expressive power. Bradfield [7] proved that indeed this is in general the case, i.e., there is a hierarchy of properties that require unbounded alternation of least and greatest fixpoint operators. Lenzi [19] independently showed a similar strictness result—for a fragment of $\mathscr{L}_\mu$—with respect to an alternation hierarchy different from the one we consider in this paper. 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 classes that only consist of a single graph. Here, 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 $\mathscr{L}_\mu$ over restricted classes of structures. Since $\mathscr{L}_\mu$ 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 [8], 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.

To appear in EPTCS.

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 [20]. That is, for every $\mathcal{L}_\mu$ formula φ, there is an alternation-free $\mathcal{L}_\mu$ formula ψ, i.e., one 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 the $\mathcal{L}_\mu$'s alternation hierarchy collapses to its alternation-free fragment [14]. Moreover, over infinite nested words, as shown by Arenas et al. [3], the alternation hierarchy collapses to the fragment with at most one alternation between least and greatest fixpoint operators. Finally, it is known that $\mathcal{L}_\mu$'s alternation hierarchy collapses over the class of transitive finite directed graphs [1, 10, 9]. If the graphs are transitive and undirected, then the hierarchy even collapses to the modal fragment [1, 10].

This paper provides further classes of structures over which the alternation hierarchy of $\mathcal{L}_\mu$ collapses to its alternation-free fragment. In fact, our collapse results subsume, generalize, and strengthen some of the collapse results mentioned above. In particular, 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. Finite directed acyclic graphs have the empty set as 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 collapse results are obtained in a uniform way by looking at bounded classes of so-called 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, e.g., [23, 11, 18, 25]. 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 technical contribution of this paper is a generalization of the ranking construction developed by Kupferman and Vardi [17], 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 [16] have already generalized their ranking construction for word automata and applied it to solve the nonemptiness problem for nondeterministic parity tree automata. However, our generalization of their ranking construction [17] is conceptually simpler: It eliminates the odd colors of a parity automaton in a single construction step. An additional step is needed to obtain from the resulting Büchi automaton a weak automaton. In contrast, Kupferman and Vardi's generalization [16] successively eliminates the colors, alternating between odd and even colors. The acceptance conditions of the intermediate word automata are a combination of a parity acceptance condition and a Büchi or coBüchi acceptance condition.

We proceed as follows. Preliminaries on $\mathcal{L}_\mu$ and alternating automata are given in Section 2. Transla-

tions between $\mathscr{L}_\mu$ formulas and automata appear in Section 3. Section 4 presents our generalization of the ranking construction. Section 5 contains our collapse results. Finally, in Section 6, we draw conclusions and outline directions for future work. Due to space limitations some of the proof details have been omitted. They can be found in the full version of the paper, which is available from the authors' web pages.

2 Preliminaries

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

2.1 The μ-Calculus

Graphs Let A be a nonempty finite set, whose elements are called *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 and Semantics We define the μ-calculus, $\mathscr{L}_\mu$ for short, over $(2^{\mathscr{P}},A)$-graphs, where $\mathscr{P}$ is a nonempty set of propositions. Let $\mathscr{V}=\{X,Y,\dots\}$ be a countable set of variables. The syntax of $\mathscr{L}_\mu$ 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 $\mathscr{V}$, p over $\mathscr{P}$, and a over A. The semantics of $\mathscr{L}_\mu$ is as follows. Let $\mathcal{G}=\big(V,(E_a)_{a\in A},v_{\mathrm{I}},\lambda\big)$ be a $(2^{\mathscr{P}},A)$-graph. A valuation σ assigns each variable in $\mathscr{V}$ to a set of vertices. For $X\in\mathscr{V}$ and $U\subseteq V$, we write $\sigma[X\mapsto U]$ if we alter σ at X, i.e., $\sigma[X\mapsto U](Y):=U$ 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 $\mathscr{L}_\mu$ guarantees that formulas are in negation normal form, i.e., negations only occur directly in front of propositions in $\mathscr{P}$. This syntactic feature ensures monotonicity and thus existence of the least and greatest fixpoints expressed by μ and ν, respectively.

The *size* of φ, written $|\varphi|$, is its number of syntactically distinct subformulas. A formula φ is a *sentence* iff φ does not have free variables. In this case, $[\![\varphi]\!]^{\mathcal{G}}_\sigma$ does not depend on σ. For a sentence φ and a set of $(2^{\mathscr{P}},A)$-graphs $\mathbf{U}$, we define

$$L_{\mathbf{U}}(\varphi) := \big\{\mathcal{G}\in\mathbf{U} \mid v_{\mathrm{I}}\in[\![\varphi]\!]^{\mathcal{G}}_\sigma, \text{ with } v_{\mathrm{I}} \text{ the source of } \mathcal{G} \text{ and } \sigma \text{ some valuation}\big\}\,.$$

Alternation Hierarchy $\mathscr{L}_\mu$ 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 [22]:

- $\Sigma_0 = \Pi_0$ is the set of formulas without fixpoint operators, i.e., 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 $X \in \mathscr{V}$ then $\mu X.\,\varphi \in \Sigma_{n+1}$; (iv) if $\varphi, \psi \in \Sigma_{n+1}$ and $X \in \mathscr{V}$ 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* iff $\mathrm{ad}(\varphi) \leq 1$, i.e., it is in Δ_2.

We remark that there is no agreement in the literature how to define the alternation hierarchy of $\mathscr{L}_\mu$. For instance, Emerson and Lei [12] define Σ_n and Π_n slightly differently. The differences are insubstantial for our results. Furthermore, we point out that our definition of the alternation depth of a formula is purely based on the formula's syntax and not on the property it describes.

2.2 Alternating Automata

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

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. For a word w, w_i denotes the symbol of w at position $(i+1)$. 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 the following conditions: (i) T is prefix-closed, i.e., $v \in T$ 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 $\overline{t}(\pi)$ for the word $t(\varepsilon)t(\pi_0)t(\pi_0\pi_1)\ldots t(\pi_0\pi_1\ldots\pi_{n-1}) \in \Sigma^*$ if π is a finite branch of length n and $t(\varepsilon)t(\pi_0)t(\pi_0\pi_1)\ldots \in \Sigma^\omega$ if π is infinite.

Automata In the following, we define alternating automata where the inputs are $(2^{\mathscr{P}}, A)$-graphs, where $\mathscr{P}$ is a nonempty finite set of propositions and A is a nonempty 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* $(\mathscr{P},A)$*-automaton*, $(\mathscr{P},A)$-PA for short, is a tuple $\mathcal{A} = (Q,\delta,q_{\mathrm{I}},\alpha)$, where Q is a finite set of states, $\delta : Q \to \mathbf{B}^+\big(\mathscr{P} \cup \overline{\mathscr{P}} \cup (Q \times \{\Diamond,\Box\} \times A)\big)$ is the transition function with $\overline{\mathscr{P}} := \{\overline{p} \mid p \in \mathscr{P}\}$, $q_{\mathrm{I}} \in Q$ is the initial state, and $\alpha : Q \to \mathbb{N}$ determines the (parity) acceptance condition. Assume that $\mathscr{P} \cap \overline{\mathscr{P}} = \emptyset$. 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, i.e., $\|\mathcal{A}\| := \big|\bigcup_{q\in Q}\{\psi \mid \psi \text{ is a subformula of } \delta(q)\}\big|$. In the following, we assume that $|Q| \in \mathcal{O}(\|\mathcal{A}\|)$, which holds when, e.g., every state occurs in some transition of $\mathcal{A}$.

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

$$\big\{q \in \mathscr{P} \bigm| q \in \lambda(v)\big\} \cup \big\{\overline{q} \in \overline{\mathscr{P}} \bigm| q \notin \lambda(v)\big\} \cup \mathscr{M} \models \delta(p)$$

and the following conditions are satisfied:

(a) If $(q,\Diamond,a) \in \mathscr{M}$, 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 with $\rho(x') = (v',q)$.

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

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 $(q,\Diamond,a) \in \mathscr{M}$, the read-only head must move along some a-labeled edge starting at v. Similarly, for $(q,\Box,a) \in \mathscr{M}$, 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 $\overline{\rho}(\pi) = (v_0,q_0)(v_1,q_1)\ldots$ is *accepting* iff $\max\{\alpha(q) \mid q \in \inf(q_0q_1\ldots)\}$ is even, where $\inf(q_0q_1\ldots)$ denotes the set of states that occur infinitely often in $q_0q_1\ldots$. The run ρ is *accepting* iff every infinite branch in ρ is accepting. The *language* of $\mathcal{A}$ with respect to a set $\mathbf{U}$ of $(2^{\mathscr{P}},A)$-graphs is the set

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

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 $(\mathscr{P},A)$-PA.

– $\mathcal{A}$ is *Büchi* iff $\{\alpha(q) \mid q \in Q\} \subseteq \{1,2\}$.

– $\mathcal{A}$ is *coBüchi* iff $\{\alpha(q) \mid q \in Q\} \subseteq \{0,1\}$.

– $\mathcal{A}$ is *weak* iff there is a partition $Q_0,\ldots,Q_n$ on Q, for some $n \geq 0$ such that for all $i \in \{0,\ldots,n\}$, the following holds: (i) All states in the component Q_i have the same parity, i.e., $\alpha(q) \equiv \alpha(q') \mod 2$, for all $q,q' \in Q_i$. (ii) $\delta(q) \in \mathbf{B}^+\big(\mathscr{P} \cup \overline{\mathscr{P}} \cup \bigcup_{j\in\{i,\ldots,n\}}(Q_j \times \{\Diamond,\Box\} \times A)\big)$, for all $q \in Q_i$. That is, when reading a vertex label the automaton can stay in the current component Q or go to components with higher indices.

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

Finally, dualizing an alternating automaton corresponds to complementation [21]. In our case, the *dual automaton* of a $(\mathscr{P},A)$-PA $\mathcal{A} = (Q,\delta,q_I,\alpha)$ is defined as the $(\mathscr{P},A)$-PA $\overline{\mathcal{A}} := (Q,\overline{\delta},q_I,\overline{\alpha})$, where for each $q \in Q$, $\overline{\delta}(q) := \overline{\delta(q)}$ with

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

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

3 From the μ-Calculus to Automata and Back

Translations between $\mathscr{L}_\mu$ 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 $\mathscr{P}$ and A be nonempty finite sets of propositions and actions, respectively. Furthermore, throughout this section, let $\mathbf{U}$ be a set of $(2^{\mathscr{P}},A)$-graphs

3.1 From $\mathscr{L}_\mu$ to Parity Automata

The following translation is similar to the one in [25]. However, since our automaton model does not support ε-transitions, we need to require for the translation that formulas are guarded, i.e., variables occur under the scope of a modal operator within their defining fixpoint formulas. For a proof of the following lemma, see, e.g., [24].

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]

From guarded formulas one easily obtains equivalent parity automata.

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

3.2 From Weak Automata to Alternation-free $\mathscr{L}_\mu$

The following translation is similar to the one in [18]. However, our variant avoids a vectorial form for $\mathscr{L}_\mu$ formulas.

Let $\mathcal{A} = (Q,\delta,q_\mathrm{I},\alpha)$ be a $(\mathscr{P},A)$-WA. Without loss of generality, assume $Q = \{q_0,\ldots,q_n\}$, $q_\mathrm{I} = q_0$, and for $i,j \in \{0,\ldots,n\}$, if q_j occurs in the Boolean formula $\delta(q_i)$ then $i < j$ or $\alpha(q_i) \equiv \alpha(q_j) \mod 2$. Also, assume that the Boolean constants $\mathtt{tt}$ and $\mathtt{ff}$ do not occur in $\mathcal{A}$'s transitions.

From $\mathcal{A}$ we define the $\mathscr{L}_\mu$ sentence $\varphi_\mathcal{A}$ with the variables $X_0,\ldots,X_n \in \mathscr{V}$. Intuitively, X_i evaluates to the set of vertices of an input that can be labeled by q_i in an accepting run. We obtain the formula $\varphi_\mathcal{A}$

[1] We are not aware of polynomial translations into the guarded fragment. The claimed polynomial upper bounds of translations found in the literature are flawed. Counterexamples are families of formulas like $\mu X_1 \ldots \mu X_n.\ \bigvee_{i=1}^n X_i \wedge \langle a \rangle \bigwedge_{i=1}^n X_i$, where a is an action. For the given translations, these formulas cause exponential blow-ups.

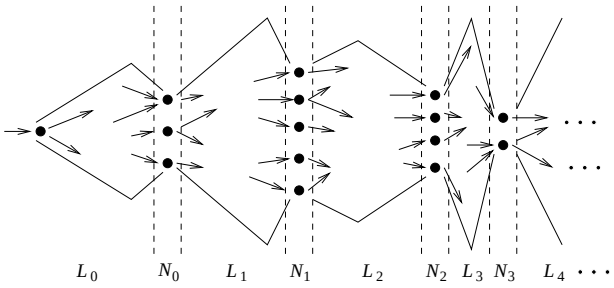

Figure 1: Bottlenecked directed acyclic graph (BDAG)

from the formulas $\psi_n, \ldots, \psi_0$ defined inductively for $i = n, \ldots, 0$: let $\psi_i := \kappa_i X_i.\, tr_i\big(\delta(q_i)\big)$, where $\kappa_i := \mu$ if $\alpha(q_i)$ is odd and $\kappa_i := \nu$ if $\alpha(q_i)$ is even, and the function tr_i is as follows:

$$tr_i(\varphi) := \begin{cases} p & \text{if } \varphi = p \text{ with } p \in \mathscr{P} \\ \neg p & \text{if } \varphi = \bar{p} \text{ with } \bar{p} \in \bar{\mathscr{P}} \\ \langle a \rangle\, \psi_j & \text{if } \varphi = (q_j, \Diamond, a) \text{ and } j > i \\ \langle a \rangle\, X_j & \text{if } \varphi = (q_j, \Diamond, a) \text{ and } j \leq i \\ [a]\, \psi_j & \text{if } \varphi = (q_j, \Box, a) \text{ and } j > i \\ [a]\, X_j & \text{if } \varphi = (q_j, \Box, a) \text{ and } j \leq i \\ tr_i(\psi) \star tr_i(\psi') & \text{if } \varphi = \psi \star \psi' \text{ with } \star \in \{\wedge, \vee\} \end{cases}$$

We point out that at most the variables $X_0, \ldots, X_{i-1}$ occur free in ψ_i. With $\varphi_{\mathcal{A}} := \psi_0$ we obtain the following theorem.

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

4 From Parity Automata to Weak Automata

In this section, we show that parity automata and weak automata have the same expressive power over so-called *bottlenecked directed acyclic graphs* (BDAGs) with a bounded width. BDAGs are fundamental to this paper 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) iff it does not contain cycles, i.e., 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 $w \in \mathbb{N}$ iff $\mathcal{G}$ 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 the vertex set to L_i does not contain infinite paths.

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 bottlenecked graphs of width at most w. In the following, let $w \in \mathbb{N}$ and let $\mathbf{BDAG}_{\leq w}$ be the class of $(2^{\mathscr{P}},A)$-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 $(\mathscr{P},A)$-PA and $\rho : R \to V \times Q$ a run of $\mathcal{A}$ on $\mathcal{G} \in \mathbf{BDAG}_{\leq w}$ with $\mathcal{G} = \big(V,(E_a)_{a\in A},v_{\mathrm{I}},\lambda\big)$. Without loss of generality, we assume that in the run ρ equally labeled nodes have isomorphic subtrees and therefore that ρ is *memoryless*; formally, for all $x,y \in R$ if $\rho(x) = \rho(y)$ then for all $z \in \mathbb{N}^*$, whenever $xz \in R$ then $yz \in R$ and $\rho(xz) = \rho(yz)$. For the memoryless run ρ, we define the graph $G^\rho := (V^\rho,E^\rho)$ with $V^\rho := \big\{\rho(x) \;\big|\; x \in R\big\}$ and $E^\rho := \big\{\big(\rho(x),\rho(y)\big) \;\big|\; x,y \in R \text{ and } y \text{ is a child of } x\big\}$. The graph G^ρ is a representation of the memoryless run ρ in which equally labeled nodes are merged. Furthermore, G^ρ is a BDAG of width at most $|Q|w$.

Let $c \geq 0$. A state $q \in Q$ is *c-releasing* iff $\alpha(q) > c$ and $\alpha(q) \not\equiv c \mod 2$. An infinite path of the form $(h_0,q_0)(h_1,q_1)\ldots$ in G^ρ is *c-dominated* iff there is a state $q \in \inf(q_0q_1\ldots)$ with $\alpha(q) = c$ and no c-releasing state in $\inf(q_0q_1\ldots)$. The function $f : V^\rho \to [2|Q|w]$ is a *c-ranking* for G^ρ iff the following two conditions hold:

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

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

The c-ranking f is *safe* iff every infinite path in G^ρ either visits infinitely many vertices with c-releasing states or f gets trapped in an odd rank on the path, i.e., iff for every infinite path $(h_0,q_0)(h_1,q_1)\ldots$ in G^ρ, either there is a state $q \in \inf(q_0q_1\ldots)$ with $\alpha(q) > c$ and $\alpha(q) \not\equiv c \mod 2$, 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 $(h,q) \in V^\rho$ 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. The following theorem reduces 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.

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 proof of Theorem 4.2 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 [17]. 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.

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 we check whether the guessed ranking is safe. Details of the construction are given in Theorem 4.4 below. For proving the correctness of the construction, it does not suffice to only assume the existence of a safe ranking. The ranking must also satisfy additional technical requirements, which are guaranteed by the following lemma.

Lemma 4.3. *Let $c \geq 0$. If G^ρ has a safe c-ranking then there is a safe c-ranking $g : V^\rho \to [2|Q|w]$ that satisfies the following additional properties:*

- $g(v_\mathrm{I}, q_\mathrm{I}) = 2|Q|w$, *and*
- $g(h_1, q) = g(h_2, q)$, *for all vertices $(h_1, q), (h_2, q) \in V^\rho$ for which there exists a vertex $(h', p) \in V^\rho$ with $\big((h', p), (h_1, q)\big) \in E^\rho$ and $\big((h', p), (h_2, q)\big) \in E^\rho$.*

We finally present the construction of the Büchi automaton that checks whether a safe ranking exists.

Theorem 4.4. *Let $c \geq 0$. There is a $(\mathscr{P}, A)$-BA $\mathcal{B}_c$ with $|Q| \cdot (2|Q|w+1)$ states and $L_{\mathbf{BDAG}_{\leq w}}(\mathcal{B}_c)$ equals*

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

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 $\varphi \in \mathbf{B}^+\big(\mathscr{P} \cup \overline{\mathscr{P}} \cup (Q \times \{\Diamond, \Box\} \times A)\big)$, $q \in Q$, 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 $\rho' : R \to V \times \big(Q \times [2|Q|w]\big)$ 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 $\rho : R \to V \times Q$ with $\rho(x) := (h, q)$, for every $x \in R$ with $\rho'(x) = \big(h, (q, r)\big)$, i.e., 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 $\rho'(x) = (h, (q, r))$, $\rho'(y) = (h, (q, r))$, 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 $f : V^\rho \to [2|Q|w]$ be a safe c-ranking on the graph representation $G^\rho = (V^\rho, E^\rho)$ of the run $\rho : R \to V \times Q$ 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 $\rho' : R \to V \times \big(Q \times [2|Q|w]\big)$ of $\mathcal{B}_c$ on $\mathcal{G}$. However, we cannot use f directly, since the following situation might occur. Assume that there are vertices $h, h_1, h_2 \in V$ with $(h,h_1),(h,h_2) \in E_a$, for some $a \in A$. Furthermore, assume $\rho(x) = (h,p)$ and $\delta(p) = (q,\Box,d)$, for some node $x \in R$ and states $p,q \in Q$. Then, the node x must have children $y,y' \in R$ such that $\rho(y) = (h_1,q)$ and $\rho(y') = (h_2,q)$. If ρ' attaches the ranks of (h,p), (h_1,q) and (h_2,q) to the labels of the nodes x, y, and y', respectively, i.e., $\rho'(x) = \big(p, f(h,p)\big)$, $\rho'(y) = \big(q, f(h_1,q)\big)$, $\rho'(y') = \big(q, (f(h_2,q)\big)$, we do not obtain a run when the ranks of (h_1,q) and (h_2,q) differ. However, Lemma 4.3 allows us to assume that $f(h_1,q) = f(h_2,q)$. In the following, let f be a safe c-ranking with the additional properties in Lemma4.3.

We define the tree $\rho' : R \to V \times \big(Q \times [2|Q|w]\big)$ now by $\rho'(x) := \big(h, (q, f(h,q))\big)$, for $x \in R$ with $\rho(x) = (h,q)$. We first show that ρ' is a run of $\mathcal{B}_c$ on $\mathcal{G}$. By definition of ρ', we have $\rho'(\varepsilon) = \big(v_\mathrm{I}, (q_\mathrm{I}, 2|Q|w)\big)$. Hence, the root of ρ' is well labeled with respect to the initial state of $\mathcal{B}_c$. Consider a node $x \in R$ with $\rho(x) = (h,q)$ and assume that $r \in [2|Q|w]$ is the rank of (h,q), i.e., $r = f(h,q)$. Let S be the set of labels of the successors of node x in ρ. By condition (ii) of a ranking, we have $f(h',q') \leq r$, for each $(h',q) \in S$ if q is not c-releasing. Furthermore, $\alpha(q) = c$ and r odd cannot hold at the same time because of condition (i) of a ranking. Moreover, by Lemma 4.3, we have $f(h',q') = f(h'',q'')$ whenever $q' = q''$, for all $(h',q'),(h'',q'') \in S$. Thus, the set S' of the labels of the successor nodes of x in ρ' is $\big\{(h',(q',r')) \;\big|\; (h',q') \in S \text{ and } r' = f(h',q')\big\}$. It is now easy to obtain from this set S' of labels a model of $\eta(q,r)$ which witnesses that the labeling corresponds to a valid transition of $\mathcal{B}_c$ with respect to the vertex label $\lambda(h)$.

The run ρ' is accepting: since 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' is accepting. □

4.2 Applications

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

Lemma 4.5. *Let $\mathcal{A}$ be a $(\mathscr{P},A)$-BA with n states. There is a $(\mathscr{P},A)$-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. First construct from $\mathcal{A}$ the coBüchi automaton $\mathcal{C}$ by dualizing the transition function of $\mathcal{A}$ and its acceptance condition. $\mathcal{C}$ accepts the complement of $\mathcal{A}$. Let $\mathcal{B}_1$ be the Büchi automaton obtained from Theorem 4.4 for the only odd color 1. This automaton is weak as $\mathcal{C}$ does not have 1-releasing states. It 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 $L_{\mathbf{BDAG}_{\leq w}}(\mathcal{B}_1) = L_{\mathbf{BDAG}_{\leq w}}(\mathcal{C})$. The dual automaton of $\mathcal{B}_1$ accepts $L_{\mathbf{BDAG}_{\leq w}}(\mathcal{A})$. □

We now show how to combine Büchi automata for different odd colors from Theorem 4.4 so that they simultaneously check the existence of safe rankings.

Lemma 4.6. *Let $\mathcal{A}$ be a $(\mathscr{P},A)$-PA with n states and index k. There is a $(\mathscr{P},A)$-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 the odd colors of $\mathcal{A}$ are $c_1,\ldots,c_\ell \in \mathbb{N}$, for some $\ell \leq \lceil k/2\rceil$. For $i \in \{1,\ldots,\ell\}$, let $\mathcal{B}_{c_i}$ be the Büchi automaton from Theorem 4.4. From the automata $\mathcal{B}_{c_1},\ldots,\mathcal{B}_{c_\ell}$, we construct a so-called

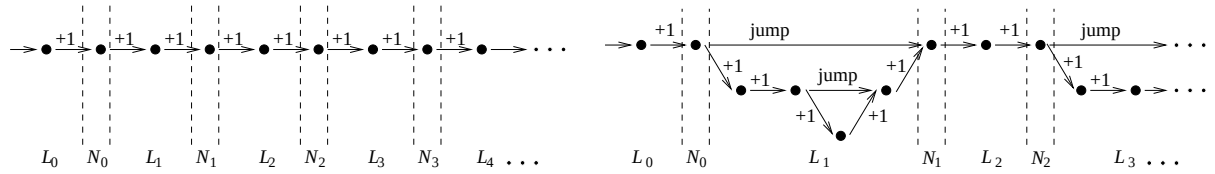

Figure 2: BDAG representation of infinite words (left) and infinite nested words (right)

generalized Büchi automaton $\mathcal{C}$, i.e., one where the acceptance condition is the finite conjunction of finitely many Büchi acceptance conditions. Since the transition functions of $\mathcal{B}_{c_1}, \ldots, \mathcal{B}_{c_\ell}$ agree on the state space of $\mathcal{A}$, 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 of the $\mathcal{B}_i$s. Thus, $\mathcal{C}$ has $n(2nw+1)^\ell$ states. An upper bound on $\|\mathcal{C}\|$ is $\mathcal{O}\big(\|\mathcal{A}\|(nw+1) + n(2nw+1)^\ell\big) \subseteq \mathcal{O}(\|\mathcal{A}\|(2nw+1)^\ell)$. 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 $\mathscr{L}_\mu$ sentence, we obtain that $\mathscr{L}_\mu$'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 $w^{\mathcal{O}(|\varphi| \cdot \mathrm{ad}(\varphi))}$ such that $L_{\mathbf{U}}(\psi) = L_{\mathbf{U}}(\varphi)$. If φ is guarded, then the size of ψ is $\big(|\varphi| \cdot w\big)^{\mathcal{O}(\mathrm{ad}(\varphi))}$.*

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 4.6). 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 4.5). Finally, we construct the alternation-free sentence ψ with $|\psi| \in (nw)^{\mathcal{O}(k)}$ (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), which results in an exponential blow-up. □

In the following, we derive from Theorem 5.1 further classes of structures over which the alternation hierarchy of $\mathscr{L}_\mu$ collapses to the alternation-free fragment.

Infinite Nested Words Nested words [2] extend words with a hierarchical structure. Infinite words and infinite nested words when represented as graphs are BDAGs of width 1. We omit the details; instead, see Figure 2 for illustrations, where the set of actions is $\{+1\}$ and $\{+1, \mathrm{jump}\}$, respectively.

Then, by Theorem 5.1, the $\mathscr{L}_\mu$ alternation hierarchy over these structures collapses to the alternation-free fragment. This improves prior results in [3, 6] on the expressivity of $\mathscr{L}_\mu$ over infinite nested words.

Graphs with Bounded Feedback Sets In the following, we consider classes of finite graphs that can be unfolded to bisimilar BDAGs with bounded width. The width of these BDAGs is characterized by a minimal feedback vertex set of the original folded graph. A set $F \subseteq V$ is a *feedback vertex set* (FVS) of $\mathcal{G} = \big(V, (E_a)_{a \in A}, v_\mathrm{I}, \lambda\big)$ iff the removal of the vertices in F separates $\mathcal{G}$ into a set of finite DAGs. Finite DAGs have the empty set as a feedback vertex set. We say that a finite graph $\mathcal{G}$ is *k-DAG-decomposable*

iff the minimal cardinality of a FVS of $\mathcal{G}$ is $k \in \mathbb{N}$. Recall that the (Σ, A)-graphs $\mathcal{G} = \big(V, (E_a)_{a\in A}, v_{\mathrm{I}}, \lambda\big)$ and $\mathcal{G}' = \big(V', (E'_a)_{a\in A}, v'_{\mathrm{I}}, \lambda'\big)$ are *bisimilar* iff there is an equivalence 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 $v \in V$.

Lemma 5.2. *For every k-DAG-decomposable (Σ, A)-graph $\mathcal{G}$, with $k \in \mathbb{N}$, there is a bisimilar BDAG $\mathcal{D}$ of width k.*

Proof. Let $\mathcal{G}$ be a (Σ, A)-graph $\big(V, (E_a)_{a\in A}, v_{\mathrm{I}}, \lambda\big)$ with a minimal FVS $F \subseteq V$ of cardinality k. Furthermore, let $\mathcal{T}$ be the tree unfolding of $\mathcal{G}$ and let $\mathtt{unf}$ be the relation between the vertices of $\mathcal{G}$ and $\mathcal{T}$ that witnesses that $\mathcal{G}$ and $\mathcal{T}$ are bisimilar. The construction of a bisimilar BDAG $\mathcal{D}$ of width k is as follows, which is done in a layer-wise manner.

Construct β, a (partial and surjective) function between the vertices of $\mathcal{T}$ and $\mathcal{D}$ (i.e., from the elements of the tree $\mathcal{T}$ to the elements of the acyclic graph $\mathcal{D}$), as follows: Assume that for each layer i, there is a set S_i of *seeds* for such a layer. Using $\mathtt{unf}$ collect all vertices in $\mathcal{T}$ that can be reached from the vertices in S_i until, in every branch, (i) a vertex with no successors is reached or (ii) two occurrences in the unfolding $\mathcal{T}$ of a vertex $v \in F \subseteq V$ in $\mathcal{G}$ are found. Such vertices belong to layer i in $\mathcal{T}$. Then, β maps such a subset of vertices in $\mathcal{T}$, denoted by R_i, to vertices in layer i of $\mathcal{D}$ as follows—and let $\max(R_i)$ be the set of maximal or terminal elements in the forest R_i:

$$\forall u', u'' \in \max(R_i).\ \forall v \in F.\ (v,u') \in \mathtt{unf} \text{ and } (v,u'') \in \mathtt{unf} \Longrightarrow \beta(u') = \beta(u''). \tag{1}$$

Edges in layer i of $\mathcal{D}$ are edges in layer i of $\mathcal{T}$ which respect β, i.e., if (w', w'') in layer i of $\mathcal{T}$ via action $a \in A$ then $(\beta(w'), \beta(w''))$ in layer i of $\mathcal{D}$ via action a. As in $\mathcal{D}$ every vertex belongs to L_i or N_i, then the following holds:

$$\forall u \in \max(R_i).\ \forall v \in F.\ (v,u) \in \mathtt{unf} \Longrightarrow \beta(u) \in N_i\,, \tag{2}$$

otherwise $\beta(u) \in L_i$. In order to define the set of seeds S_{i+1} for the $(i+1)$th layer of $\mathcal{T}$, firstly one needs to define a subset F'_i of $\max(R_i)$ whose successors in $\mathcal{T}$ will be the seeds S_{i+1} of the layer i of $\mathcal{T}$. The set F'_i is a subset of $\max(R_i)$ that satisfies two conditions:

(a) if $\exists (v,u) \in \mathtt{unf}$ with $v \in F$ and $u \in \max(R_i)$ then $\exists! u' \in F'_i$ such that $(v,u') \in \mathtt{unf}$, and

(b) if $u' \in F'_i$ then $\exists (v,u') \in \mathtt{unf}$ such that $v \in F$ and $u' \in \max(R_i)$.

Condition (a) ensures that F'_i contains *no more than one* vertex in the unfolding under $\mathtt{unf}$ of a vertex in F as well as that there is one vertex in F'_i for each vertex in $\max(R_i)$ which is associated under $\mathtt{unf}$ with a vertex in F. Condition (b) ensures that every vertex in F'_i is the occurrence in the unfolding under $\mathtt{unf}$ of a vertex in F. It is because of condition (a) that the function β is partial rather than total.

Edges between vertices in consecutive layers are defined as expected: if (u,s) in $\mathcal{T}$ via action $a \in A$, with $u \in \max(R_i)$ and $s \in S_{i+1}$, then $(\beta(u), \beta(s))$ in $\mathcal{D}$ via action a. The labeling function in $\mathcal{D}$ is as in $\mathcal{T}$ (and obviously as in $\mathcal{G}$): for any vertex $\beta(u)$ in $\mathcal{D}$, $\lambda_{\mathcal{T}}(u) = \lambda_{\mathcal{D}}(\beta(u))$. Finally, $\mathcal{D}$ is constructed recursively using $\mathtt{unf}$ and β by letting the set S_0 be the singleton set that only contains the root of $\mathcal{T}$.

Clearly, $\mathcal{D}$ is a DAG. Within as well as between layers β always respects the acyclic structure produced by $\mathtt{unf}$, even when different occurrences of vertices in $F \subseteq V$ are unified as they are always terminal elements of a given R_i and thus edges in $N_i \times L_{i+1}$, i.e. the source of edges to the next layer.

Finally, $\mathcal{G}$ and $\mathcal{D}$ are bisimilar because either (i) a vertex in $\mathcal{D}$ is obtained by a tree unfolding, and every graph is bisimilar to its own tree unfolding or (ii) a vertex in $\mathcal{D}$ is obtained by unifying occurrences of the same vertex in $\mathcal{G}$, and of course every vertex of a graph is bisimilar to itself. Then, bisimilarity is

preserved when constructing $\mathcal{D}$. To see that $\mathcal{D}$ is a BDAG of width at most k observe that $\sup\{|N_i| \mid i \in \mathbb{N}\} \leq k$ because every N_i defined by rule (2) cannot contain more than one occurrence of a vertex in F due to rule (1). Then, in fact, there must exist some $i \in \mathbb{N}$ such that $|F| = |N_i|$ since F is minimal. □

Then, we obtain the following result.

Theorem 5.3. *Let $k \in \mathbb{N}$ and* **U** *be a class of k-DAG-decomposable $(2^{\mathscr{P}}, A)$-graphs. 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 5.2 that each graph in **U** can be unfolded into a BDAG of width k. We then apply Theorem 5.1 to this obtained class of unfolded BDAGs. □

Since collapse results carry over to smaller classes of structures, Theorem5.3 implies the collapse of the alternation hierarchy over the smaller class of undirected k-DAG-decomposable graphs.

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 $\mathcal{G}$ iff 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 5.4. *Let $k \in \mathbb{N}$ and* **U** *be a class of finite $(2^{\mathscr{P}}, A)$-graphs with minimal FESs of size k. For every sentence φ, there is an alternation-free sentence ψ such that $L_{\mathbf{U}}(\psi) = L_{\mathbf{U}}(\varphi)$.*

6 Conclusion and Future Work

The results in this paper focus on $\mathscr{L}_\mu$'s expressivity. By generalizing and utilizing automata-theoretic methods, we have unified, generalized, and strengthened prior collapse results of $\mathscr{L}_\mu$'s alternation hierarchy, namely, the results on finite acyclic directed graphs [20], infinite words [14], and infinite nested words [3]. Future work includes to investigate whether our automata construction for eliminating odd colors in parity automata can be generalized and to explore over which other classes of structures such generalizations apply. 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 paper, 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 $\mathscr{L}_\mu$ with respect to classes of structures over which its alternation hierarchy collapses.

Acknowledgments The authors thank Christian Dax for initial discussions on the topic of this paper and Julian Bradfield for advice on the alternation hierarchy. Julian Gutierrez acknowledges with gratitude the support of EPSRC grant 'Solving Parity Games and Mu-Calculi' and ERC Advanced grant ECSYM.

References

[1] Luca Alberucci & Alessandro Facchini (2009): *The modal μ-calculus over restricted classes of transition systems*. *J. Symb. Log.* 74(4), pp. 1367–1400, doi: 10.2178/jsl/1254748696.

[2] Rajeev Alur & P. Madhusudan (2009): *Adding Nesting Structure to Words*. *J. ACM* 56(3), pp. 1–43, doi:10.1145/1516512.1516518.

[3] Marcelo Arenas, Pablo Barceló & Leonid Libkin (2011): *Regular Languages of Nested Words: Fixed Points, Automata, and Synchronization*. *Theor. Comput. Syst.* 49(3), pp. 639–670, doi: 10.1007/s00224-010-9292-5.

[4] André Arnold (1999): *The modal μ-calculus alternation-depth is strict on binary trees*. *Theor. Inform. Appl.* 33(4–5), pp. 329–339, doi:10.1051/ita:1999121.

[5] André Arnold & Damian Niwiński (2001): *Rudiments of μ-calculus*. *Studies in Logic and the Foundations of Mathematics* 146, North-Holland.

[6] Laura Bozzelli (2007): *Alternating Automata and a Temporal Fixpoint Calculus for Visibly Pushdown Languages*. In: *CONCUR'07, Lect. Notes Comput. Sci.* 4703, pp. 476–491, doi:10.1007/978-3-540-74407-8_32.

[7] Julian C. Bradfield (1998): *The modal mu-calculus alternation hierarchy is strict*. *Theoret. Comput. Sci.* 195(2), pp. 133–153, doi:10.1016/S0304-3975(97)00217-X.

[8] Julian C. Bradfield (1999): *Fixpoint alternation: arithmetic, transition systems, and the binary tree*. *Theor. Inform. Appl.* 33(4–5), pp. 341–356, doi:10.1051/ita:1999122.

[9] Giovanna D'Agostino & Giacomo Lenzi (2010): *On the μ-calculus over transitive and finite transitive frames*. *Theoret. Comput. Sci.* 411(50), pp. 4273–4290, doi:10.1016/j.tcs.2010.09.002.

[10] Anuj Dawar & Martin Otto (2009): *Modal characterisation theorems over special classes of frames*. *Ann. Pure Appl. Logic* 161(1), pp. 1–42, doi:10.1016/j.apal.2009.04.002.

[11] E. Allen Emerson & Charanjit S. Jutla (1991): *Tree Automata, Mu-Calculus and Determinacy*. In: *FOCS'91*, pp. 368–377, doi:10.1109/SFCS.1991.185392.

[12] E. Allen Emerson & Chin-Laung Lei (1986): *Efficient Model Checking in Fragments of the Propositional Mu-Calculus*. In: *LICS'86*, pp. 267–278.

[13] David Janin & Igor Walukiewicz (1996): *On the Expressive Completeness of the Propositional mu-Calculus with Respect to Monadic Second Order Logic*. In: *CONCUR'96, Lect. Notes Comput. Sci.* 1119, pp. 263–277, doi:10.1007/3-540-61604-7_60.

[14] Roope Kaivola (1995): *Axiomatising Linear Time Mu-calculus*. In: *CONCUR'95, Lect. Notes Comput. Sci.* 962, pp. 423–437, doi:10.1007/3-540-60218-6_32.

[15] Dexter Kozen (1983): *Results on the Propositional μ-Calculus*. *Theoret. Comput. Sci.* 27(3), pp. 333–354, doi:10.1016/0304-3975(82)90125-6.

[16] Orna Kupferman & Moshe Y. Vardi (1998): *Weak Alternating Automata and Tree Automata Emptiness*. In: *STOC'98*, pp. 224–233, doi:10.1145/276698.276748.

[17] Orna Kupferman & Moshe Y. Vardi (2001): *Weak Alternating Automata Are Not that Weak*. *ACM Trans. Comput. Log.* 2(3), pp. 408–429, doi:10.1145/377978.377993.

[18] Orna Kupferman & Moshe Y. Vardi (2005): *From Linear Time to Branching Time*. *ACM Trans. Comput. Log.* 6(2), pp. 273–294, doi:10.1145/1055686.1055689.

[19] Giacomo Lenzi (1996): *A Hierarchy Theorem for the μ-Calculus*. In: *ICALP'96, Lect. Notes Comput. Sci.* 1099, pp. 87–97, doi:10.1007/3-540-61440-0_119.

[20] Radu Mateescu (2002): *Local Model-Checking of Modal Mu-Calculus on Acyclic Labeled Transition Systems*. In: *TACAS'02, Lect. Notes Comput. Sci.* 2280, pp. 281–295, doi:10.1007/3-540-46002-0_20.

[21] David E. Muller & Paul E. Schupp (1987): *Alternating Automata on Infinite Trees*. *Theoret. Comput. Sci.* 54(2–3), pp. 267–276, doi:10.1016/0304-3975(87)90133-2.

[22] Damian Niwiński (1986): *On fixed-point clones*. In: *ICALP'86, Lect. Notes Comput. Sci.* 226, pp. 464–473, doi:10.1007/3-540-16761-7_96.

[23] Damian Niwiński (1988): *Fixed points vs. infinite generation*. In: *LICS'88*, pp. 402–409, doi:10.1109/LICS.1988.5137.

[24] Igor Walukiewicz (1995): *Completeness of Kozen's Axiomatization of the Propositional μ-Calculus*. In: *LICS'95*, pp. 14–24, doi:10.1109/LICS.1995.523240.

[25] Thomas Wilke (2001): *Alternating Tree Automata, Parity Games, and Modal μ-Calculus*. *Bull. Soc. Math. Belg.* 8(2), pp. 359–391.