Disjunctive form and the modal μ alternation hierarchy

Karoliina Lehtinen
Laboratory for Foundations of Computer Science
University of Edinburgh
M.K.Lehtinen@sms.ed.ac.uk

This paper studies the relationship between disjunctive form, a syntactic normal form for the modal μ calculus, and the alternation hierarchy. First it shows that all disjunctive formulas which have equivalent tableau have the same syntactic alternation depth. However, tableau equivalence only preserves alternation depth for the disjunctive fragment: there are disjunctive formulas with arbitrarily high alternation depth that are tableau equivalent to alternation-free non-disjunctive formulas. Conversely, there are non-disjunctive formulas of arbitrarily high alternation depth that are tableau equivalent to disjunctive formulas without alternations. This answers negatively the so far open question of whether disjunctive form preserves alternation depth. The classes of formulas studied here illustrate a previously undocumented type of avoidable syntactic complexity which may contribute to our understanding of why deciding the alternation hierarchy is still an open problem.

1 Introduction

The modal μ calculus [2], L_μ, is a modal logic augmented with its namesake least fixpoint operator μ and the dual greatest fixpoint operator, ν. Alternating between these two operators gives the logic its great expressivity [1] while both model checking and satisfiability remain pleasingly decidable. The complexity of model checking is, at least currently, tied to the number of such alternations, called the alternation depth of the formula being checked [10]. The problem of deciding the least number of alternations required to express a property, also known as the Rabin-Mostowski index problem, is a long standing open problem.

Disjunctive normal form is a syntactic restriction on L_μ formulas which first appeared in [9] and was then used as a tool for proving completeness of Kozen's axiomatization [14]. It is based on the tableau decomposition of a formula which forces it to be in many ways well-behaved, making it a useful tool for various manipulations. For instance, satisfiability and synthesis are straight-forward for disjunctive formulas. In [5] it is used to analyse modal L_μ from a logician's perspective. More recently, disjunctive form was found to allow for simple formula optimisation: if a formula is equivalent to a formula without greatest fixpoints, then such a formula is easily produced by simple syntactic manipulation on the disjunctive form of the formula [11].

Each of these results uses the fact that any formula can be effectively transformed into an equivalent disjunctive formula with the same tableau – indeed, disjunctive form is perhaps the closest one gets to a canonical normal form for L_μ. The transformation itself, described in [9], is involved and it has so far been an open question whether it preserves the alternation depth of formulas. If this was the case, it would be sufficient to study the long-standing open problem of the decidability of the alternation hierarchy on this well-behaved fragment.

In this paper, we show that although the disjunctive fragment of L_μ is itself well-behaved with respect to the alternation hierarchy, the transformation into it does not preserve alternation depth.

R. Matthes, M. Mio (Eds.): Fixed Points
in Computer Science 2015 (FICS 2015)
EPTCS 191, 2015, pp. 117–131, doi:10.4204/EPTCS.191.11

The transformation into disjunctive form takes the tableau decomposition of a formula, and produces a disjunctive formula that generates the same tableau. The first contribution of this paper is to show that all disjunctive formulas generating the same tableau have the same alternation depth. This result brings some clarity to the transformation into disjunctive form since one of the more difficult steps of the construction is representing the parity of infinite paths of the tableau with a finite priority assignment. The result presented here means that all valid choices are just as good, as all yield a disjunctive formula of the same alternation depth. As a result, the alternation hierarchy is decidable for the disjunctive fragment of L_μ with respect to tableau equivalence, a stricter notion of equivalence than semantic equivalence, as defined in [14].

The second contribution of this paper is to show that this does not extend to non-disjunctive formulas. Not only does tableau equivalence not preserve alternation depth in general, but the alternation depth of a formula does not guarantee *any* upper bound on the alternation depth of equivalent disjunctive formulas. Indeed, for arbitrarily large n, there are formulas with a single alternation which are tableau equivalent only to disjunctive formulas with at least n alternations.

Conversely, there are formulas of L_μ with arbitrarily large alternation depth which are tableau equivalent to a disjunctive formula without alternations. This shows that the alternation depths of tableau equivalent formulas are only directly related within the disjunctive fragment.

The signficance of these results in twofold. First, they outline the limits of what can be achieved using disjunctive form: disjunctive form does not preserve alternation depth so despite being a useful tool for satisfiability-related problems, it is unlikely to be of much help in contexts where the alternation depth of a formula matters, such as model-checking or formula optimisation beyond the first levels of the alternation hierarchy.

Secondly, and perhaps most significantly, these results impact our understanding of the alternation hierarchy. This paper's results imply that deciding the alternation hierarchy for the disjunctive fragment of L_μ, an open but easier problem, is not sufficient for deciding the alternation hierarchy in the general case. The counterexamples used to show this illustrate a previously undocumented type of accidental complexity which appears to be difficult to identify. These may shed light on why deciding the alternation hierarchy is still an open problem and examplify a category of formulas with unnecessary alternations which need to be tackled with novel methods.

Related work Deciding the modal μ alternation hierarchy is exactly equivalent to deciding the Rabin-Mostowski index of alternating parity automata. The corresponding problem has also been studied for automata operating on words [3] and automata which are deterministic [13, 12], or non-deterministic [4, 7] rather than alternating. As will be highlighted throughout this paper, many of the methods used here are similar to methods applied to different types of automata.

2 Preliminaries

2.1 The modal μ calculus

For clarity and conciseness, the semantics of L_μ are given directly in terms of parity games. As is well documented in the literature, this approach is equivalent to the standard semantics [2]. The following definitions are fairly standard, although we draw the reader's attention to the use of the less typical modality $\rightarrow\mathcal{B}$ in the syntax of L_μ and the unusual but equivalent definition of alternation depth.

Definition 1. (L_μ) Given a set of atomic propositions $Prop = \{P, Q, ...\}$ and a set of fixpoint variables $Var = \{X, Y, ...\}$, the syntax of L_μ is given by:

$$\phi := \top \mid \bot \mid P \mid \neg P \mid X \mid \phi \wedge \phi \mid \phi \vee \phi \mid \rightarrow \mathscr{B} \text{ where } \mathscr{B} \text{ is a set of formulas } \mid \mu X.\phi \mid \nu X.\phi$$

The modality $\rightarrow\mathscr{B}$ replaces the more usual modalities $\Diamond\phi$ and $\Box\phi$. If $\mathscr{B}$ is a set of formulas, $\rightarrow\mathscr{B}$ stands for $(\bigwedge_{\phi\in\mathscr{B}} \Diamond\phi) \wedge \Box \bigvee_{\phi\in\mathscr{B}} \phi$: every formula in $\mathscr{B}$ must be realised in some successor state and each successor state must realise at least one of the formulas in $\mathscr{B}$. The modalities $\Diamond\phi$ and $\Box\phi$ are expressed in this syntax by $\rightarrow\{\phi,\top\}$ and $\rightarrow\{\phi\} \vee \rightarrow\bot$ respectively, where $\bot$ denotes the empty set.

Without loss of expressivity, this syntax only allows for formulas in positive form: negation is only applied to propositions. Furthermore, without loss of expressivity, but perhaps conciseness, we require all formulas to be guarded: all fixpoint variables are within the scope of a modality within their binding formula. For the sake of clarity, we restrict our study to the uni-modal case but expect the multi-modal case to behave broadly speaking similarly. To minimise the use of brackets, the scope of fixpoint bindings should be understood to extend as far as possible.

Definition 2. *(Structures)* A structure $\mathscr{M} = (S, s_0, R, P)$ consists of a set of states S, rooted at some initial state $s_0 \in S$, and a successor relation $R \subseteq S \times S$ between the states. Every state s is associated with a set of propositions $P(s) \subseteq Prop$ which it is said to satisfy.

Definition 3. *(Parity games)* A parity game is a potentially infinite two-player game on a finite graph $\mathscr{G} = (V_0, V_1, E, v_I, \Omega)$ of which the vertices $V_0 \cup V_1$ are partitioned between the two players Even and Odd and annotated with positive integer priorities via $\Omega : V_0 \cup V_1 \to \mathbb{N}$. The even player and her opponent, the odd player, move a token along the edges $E \subseteq V_0 \cup V_1 \times V_0 \cup V_1$ of the graph starting from an initial position $v_I \in V_0 \cup V_1$, each choosing the next position when the token is on a vertex in their partition. Some positions p might have no successors in which case they are winning for the player of the parity of $\Omega(p)$. A play consists of the potentially infinite sequence of vertices visited by the token. For finite plays, the last visited parity decides the winner of the play. For infinite play, the parity of the highest priority visited infinitely often decides the winner of the game: Even wins if the highest priority visited infinitely often is even; otherwise Odd wins. Note that since some readers may be used to an equivalent definition using the lowest priority to define the winner, whenever possible, "most significant" will be used to indicate the highest priority.

Definition 4. *(Strategies)* A positional strategy σ for one of the players in $\mathscr{G} = (V_0, V_1, E, v_I, \Omega)$ is a mapping from the player's positions s, in V_0 for Even and in V_1 for Odd, in the game to a successor position s' such that $(s, s') \in E$. A play respects a player's strategy σ if the successor of any position in the play belonging to the player is the one dictated by σ. If σ is Even's strategy and τ is Odd's strategy then there is a unique play $\sigma \times \tau$ respecting both strategies from every position. The winner of the parity game at a position is the player who has a strategy σ , said to be a winning strategy, such that they win $\sigma \times \tau$ from that position for any counter-strategy τ. The following states that such strategies are sufficient: players do not need to take into account the history of a play to play optimally.

Fact 5. *Parity games are positionally determined: for every position either Even or Odd has a winning strategy [6].*

This means that strategies gain nothing from looking at the whole play rather than just the current position. As a consequence, we may take a strategy to be memoryless: it is a mapping from a player's positions to a successor.

For any L_μ formula ϕ and a structure $\mathscr{M}$ we define a parity game $\mathscr{M} \times \phi$, constructed in polynomial time, and say that $\mathscr{M}$ satisfies ϕ, written $\mathscr{M} \models \phi$, if and only if the Even player has a winning strategy in $\mathscr{M} \times \phi$.

Definition 6. *(Model-checking parity game)* For any formula ϕ of modal μ, and a model $\mathscr{M}$, define a parity game $\mathscr{M} \times \phi$ with positions (s, ψ) where s is a state of $\mathscr{M}$ and ψ is either a proper subformula of ϕ, or the formula $\bigvee \mathscr{B}$, or the formula $\Diamond\psi$ for any $\rightarrow\mathscr{B}$ and $\psi \in \mathscr{B}$ in ϕ. The initial position is (s_0, ϕ) where s_0 is the root of $\mathscr{M}$. Positions (s, ψ) where ψ is a disjunction or $\Diamond\psi'$ belong to Even while conjunctions and positions $\rightarrow\mathscr{B}$ belong to Odd. Other positions have at most one successor; let them be Even's although the identity of their owner is irrelevant. There are edges from $(s, \psi \vee \psi')$ and $(s, \psi \wedge \psi')$ to both (s, ψ) and (s, ψ'); from $(s, \mu X.\phi)$ and $(s, \nu X.\phi)$ to (s, ϕ); from (s, X) to $(s, \nu X.\psi)$ if X is bound by ν, or $(s, \mu X.\psi)$ if it is bound by μ; finally, from $(s, \rightarrow\mathscr{B})$ to every $(s', \bigvee \mathscr{B})$ where (s, s') is an edge in $\mathscr{M}$, and also to every $(s, \Diamond\psi)$ where $\psi \in \mathscr{B}$ and from $(s, \Diamond\psi)$ to every (s', ψ) where (s, s') is an edge in the model $\mathscr{M}$. Positions (s, P),$(s, \neg P)$, $(s, \top)$ and $(s, \bot)$ have no successors. The parity function assigns an even priority to $(s, \top)$ and also to (s, P) if s satisfies P in $\mathscr{M}$ and to $(s, \neg P)$ if s does not satisfy P in $\mathscr{M}$; otherwise (s, P) and $(s, \neg P)$ receive odd priorities, along with $(s, \bot)$. Fixpoint variables are given positive integer priorities such that ν-bound variables receive even priorities while μ-bound variables receive odd priorities. Furthermore, whenever X has priority i, Y has priority j and $i < j$, X must not appear free in the formula ψ binding Y in $\mu Y.\psi$ or $\nu Y.\psi$. In other words, inner fixpoints receive lower, less significant priorities while outer fixpoint receive high priorities. Other nodes receive the least priority used, 0 or 1.

We now use parity games to define the semantics of L_μ.

Definition 7. *(Satisfaction relation)* A structure $\mathscr{M}$, rooted at s_0 is said to satisfy a formula Ψ of L_μ, written $\mathscr{M} \models \Psi$ if and only if the Even player has a winning strategy from (s_0, Ψ) in $\mathscr{M} \times \Psi$.

Note that the definition of the model-checking parity game requires a priority assignment to fixpoint variables in a formula that satisfies the conditions that ν-variables receive even priorities, μ-variables receive odd priorities and whenever X has priority i, Y has priority j and $i < j$, X must not appear free in the formula ψ binding Y in $\mu Y.\psi$ or $\nu Y.\psi$. For any formula, there are several valid assignments. For example, one could assign a distinct priority to every fixpoint, with the highest priority going to the outermost bound fixpoint and the priorities decreasing the further into the formula a fixpoint is bound. We further restrict a parity assignment to be surjective into an initial fragment of $\mathbb{N}$: if a priority is unused, all greater priorities can be reduced by 2. We define the alternation depth of a formula to be the minimal valid assignment. Although variations of this definition exists, our motivation is to match closely the alternations required in the model checking parity game.

Definition 8. Let a priority assignment be a function $\Omega : Var \rightarrow \{0...n\}$ for some integer n, which is surjective on at least $\{1,...,n\}$, such that if $\Omega(X) < \Omega(Y)$ then X does not appear free in the formula binding Y and the parity of $\Omega(X)$ is even for ν-bound variables and odd for μ-bound variables. We don't require the priority 0 to be used, but include it in the co-domain for simplicity. In this paper, we take the alternation depth of a formula to be the co-domain of the least priority assignment of a formula. The correspondance with the priorities of the model checking parity game should make it clear that this definition is equivalent to the more typical syntactic ones in the literature, for example in [2]. An alternation free formula is a formula which has both priority assignements with co-domain $\{0,1\}$ and $\{0,1,2\}$ where 0 is not used.

Deciding whether a formula is equivalent to a formula with smaller alternation depth is a long standing open problem.

2.2 Tableau decomposition

Definition 9. *(Tableau)* A tableau $\mathscr{T} = (T, L)$ of a formula Ψ consists of a potentially infinite tree T of which each node n has a label $L(n) \subseteq sf(\Psi)$ where $sf(\Psi)$ is the set of proper subformulas of Ψ. The labelling respects the following tableau rules with the restriction that the modal rule is only applied where no other rule is applicable.

$$\frac{\{\Gamma, \phi, \psi\}}{\{\Gamma, \psi \wedge \phi\}}\,(\wedge) \qquad \frac{\{\Gamma, \phi\} \qquad \{\Gamma, \psi\}}{\{\Gamma, \psi \vee \phi\}}\,(\vee) \qquad \frac{\{\Gamma, \phi\}}{\{\Gamma, \sigma X.\phi\}}\,(\sigma) \text{ with } \sigma \in \{\mu, \nu\}$$

$$\frac{\{\Gamma, \phi\}}{\{\Gamma, X\}}\,(X) \text{ where } X \text{ is a fixpoint variable bound by } \sigma X.\phi, \text{ with } \sigma \in \{\mu, \nu\}$$

$$\frac{\{\psi\} \cup \{\bigvee \mathscr{B} | \to\mathscr{B} \in \Gamma, \mathscr{B} \neq \mathscr{B}'\} \text{ for every } \to\mathscr{B}' \in \Gamma, \psi \in \mathscr{B}'}{\{\Gamma\}}\,(\to)$$

Note that each branching node is either a choice node, corresponding to a disjunction, or a modal node. Although the rules only contain a binary disjunctive rule, we may write, for the sake conciseness, a sequence of binary choice nodes as a single step. Also note that when a modal rule is applied, all formulas in a label are either modal formulas or literals, that is to say propositional variables and their negations. The latter form the modal node's set of literal and are a semantically important component of the tableau. An inconsistent set of literals is equivalent to $\perp$ and a node with such a set of literals in its label has no successors.

Sequences of subformulas along a path in the tableau are called traces and correspond to plays in the model checking parity game. A μ-trace is a trace winning for the Odd player.

Definition 10. *(μ-trace)* Given an infinite branch in a tableau, that is to say a sequence $n_0 n_1 \ldots$ of nodes starting at the root, where n_{i+1} is a child of n_i, a trace on it is an infinite sequence $f_0 f_1 \ldots$ of formulas satisfying the following: each formula is taken from the label of the corresponding node, $f_i \in L(n_i)$ for all $i \geq 0$; successive formulas f_i and f_{i+1} are identical if f_i is not the formula that the tableau rule from n_i to n_{i+1} acts on; if the tableau rule from n_i to n_{i+1} is a disjunction, conjunction, or fixpoint binding elimination acting on f_i, then f_{i+1} is an immediate subformula of f_i; if the tableau rule from n_i to n_{i+1} is a modality, then f_i has to be a formula $\to\mathscr{B}$ and f_{i+1} is either $\bigvee \mathscr{B}$ or a formula $\psi \in \mathscr{B}$; if the tableau rule from n_i to n_{i+1} is a fixpoint regeneration acting on the fixpoint variable f_i, then f_{i+1} is the binding formula for f_i. A trace is a μ-trace if the most significant fixpoint variable that regenerates infinitely often on it is a μ-variable.

Since labels are to be thought of as conjuncts, it is sufficient for an infinite path in a tableau to allow one μ-trace for the infinite path to be winning for the Odd player.

Definition 11. *(Parity of a path)* An infinite path in a tableau is said to be even if there are no μ-traces on it, otherwise it is said to be odd.

Note that the order of applications of the tableau rules is non deterministic so a formula may appear to have more than one tableau. However, tableau equivalence, defined next, only looks at the structure of branching, whether branching nodes are modal or disjunctive, the literals at modal nodes and the parity of infinite paths, so a formula has a unique tableau, up to tableau equivalence. We define tableau cores to be the semantic elements of the tableau – node types, literals at modal nodes, branching structure and the parity of infinite paths – which do not depend on the syntax of the generating formula. Finally, we define trees with back edges which are finite representations of tableau cores.

Definition 12. *(Tableau core)* A tableau core is $\mathscr{C} = (C, \Omega)$ where C is a potentially infinite but still finitely branching tree of which the nodes are either modal nodes or disjunctive nodes and modal nodes are decorated with a set of literals. Ω is a parity assignment with a finite prefix of $\mathbb{N}$ as co-domain. An infinite path in $\mathscr{C}$ is of the parity of the most significant priority seen infinitely often. $\mathscr{C} = (C, \Omega)$ is a tableau core for $\mathscr{T} = (T, L)$ if once the sequences of disjunctions in $\mathscr{T}$ are collapsed into one non-binary disjunction there is a bijection b between the branching nodes of T and the nodes of C which respects the following: the successor relation in the sense that $b(i)$ is a child of $b(j)$ in C if and only if i is a child of j in T, whether nodes are modal or disjunctive, the literals at modal nodes, and the parity of infinite paths. That is to say, if a path in $\mathscr{T}$ maps to a path in $\mathscr{C}$ then the highest priority seen infinitely often on the path in $\mathscr{C}$ is even if and only if the path in $\mathscr{T}$ has no μ-trace.

Definition 13. *(Tableau equivalence)* Two tableaus $(\mathscr{T}_0, L_0)$ and $(\mathscr{T}_1, L_0)$ are equivalent if their cores are bisimilar with respect to their branching structure, whether nodes are disjunctive or modal, the literals at modal nodes and the parity of infinite branches. Two formulas are tableau equivalent if they generate equivalent tableaus.

Definition 14. *(Tree with back edges)* Tableaus are potentially infinite but regular, so they allow finite representations. A finite representation of a tableau $\mathscr{A} = (A, \Omega)$ is a finite tree with back edges, A which is bisimilar to the core of the tableau. Every node is either a modal node or a disjunctive node and modal nodes are associated with a set of literals. The tree has a priority assignment Ω which assigns priorities to nodes such that the highest priority on an infinite path is of the parity of that path.

To summarise, a tableau $\mathscr{T}$ is a potentially infinite tree labelled with sets of subformulas – it is specific to the formula which labels its root; a tableau core, $\mathscr{C}$ is a potentially infinite object which carries the same semantics but is not specific to one formula; finally, a tree with back edges, called $\mathscr{A}$ because of its resemblance to alternating parity automata, is a finite representation of a tableau core. The next section will present the one-to-one correspondence between disjunctive formulas and trees with back edges.

Theorem 15. *[9] Tableau equivalent formulas are semantically equivalent.*

Note that tableau equivalence is a stricter notion than semantic equivalence; $\psi \vee \neg\psi$ and $\top$ have different tableau for example.

2.3 Disjunctive normal form

Disjunctive form was introduced in [9] as a syntactic restriction on the use of conjunctions. It forces a formula to follow a simple structure of alternating disjunctions and modalities where modalities are qualified with a conjunction of propositions. Such formulas are in many ways well-behaved and easier to manipulate than arbitrary L_μ formulas.

Definition 16. *(Disjunctive formulas)* The set of disjunctive form formulas of (unimodal) L_μ is the smallest set $\mathscr{F}$ satisfying:

- $\bot, \top$, propositional variables and their negations are in $\mathscr{F}$;
- If $\psi \in \mathscr{F}$ and $\phi \in \mathscr{F}$ then $\psi \vee \phi \in \mathscr{F}$;
- If $\mathscr{A}$ is a set of literals and $\mathscr{B} \subseteq \mathscr{F}$ ($\mathscr{B}$ is finite), then $\bigwedge \mathscr{A} \wedge \rightarrow \mathscr{B} \in \mathscr{F}$;

- $\mu X.\psi$ and $\nu X.\psi$ as long as $\psi \in \mathscr{F}$.

Every formula is known to be equivalent to an effectively computable formula in disjunctive form [9]. The transformation into disjunctive form involves taking the formula's tableau decomposition and compressing the node labels into a single subformula. The tricky part is finding a tree with back edges and its priority assignment to represent the tableau finitely, including the parity of infinite paths. The transformation then turns the tree with back edges into a disjunctive formula with alternation depth dependent on the priority assignment. Conversely, a disjunctive formula and its minimal priority assignment induces a tree with back edges representing its tableau. The minimal priority function required to finitely represent a tableau is therefore equivalent to the minimal alternation depth of a disjunctive formula generating the tableau. The following theorem recalls the construction of disjunctive formulas from trees with back edges labelled with priorities from [9] and shows that the alternation depth of the resulting formula stems from the priority assignment of the tree with back edges.

Theorem 17. *Let $\mathscr{A} = (A, \Omega)$ be a tree with back edges that is bisimilar to a core of the tableau $\mathscr{T}$ with priority assignment Ω with co-domain $\{0...q\}$. Then there is a disjunctive formula with alternation depth $\{0...q\}$ which generates a tableau equivalent to $\mathscr{T}$.*

Proof. First of all, we construct $\mathscr{A}' = (A', \Omega')$, bisimilar to $\mathscr{A}$ but with a priority assignment with the following property: on all paths from root to leaf, the priorities of nodes that are the targets of back edges occur in decreasing order. This is straight-forward by looking at the infinite tableau core $\mathscr{A}$ unfolds into, remembering which nodes stem from the same node in $\mathscr{A}$ and their priority assigned by Ω. First consider all branches that see the highest priority q infinitely often and cut them short by creating back edges at nodes of priority q, pointing to the bisimilar ancestor node (also of priority q) that is closest to the root. Then repeat this for each priority in decreasing order, but for each priority $q-1$ treat the ancestor of priority q that back edges point to (if it exists) as the root, so that nodes that have back edges pointing to them end up in decreasing order of priority. Note that every cycle is now dominated by the priority of the first node from the root seen infinitely often.

The disjunctive formula is then obtained by assigning a subformula $f(n)$ to every node of A as follows. If n is a leaf with literals Q, then $f(n) = \bigwedge Q$; if n is a disjunctive node with children n_0 and n_1, then $f(n) = f(n_0) \vee f(n_1)$; if n is the source of a back edge of which the target is m, then $f(n) = X_m$ where X_m is a fixpoint variable; if n is a modal node, then $f(n) = \bigwedge Q \wedge \rightarrow \mathscr{B}$ where Q is the set of literals at n and $\mathscr{B}$ is the set of $f(n_i)$ for n_i children of n; other nodes inherit the formula assigned to their unique child. If n is the target of a back edge, $f(n)$ is obtained as previously detailed but in addition, it binds the fixpoint variable X_n with a ν-binding if n is of even parity and with a μ-binding otherwise.

If r is the root node of A', then $f(r)$ is a disjunctive formula that generates a tableau that is equivalent to $\mathscr{T}$. This should be clear from the fact that the tableau of $f(n)$ consists of the infinite tree generated by A' and the labelling $L(n) = \{f(n)\}$ for all n. Ω' restricted to the target of back edges is a priority assignment for the disjunctive formula $\Psi = f(n)$ since it respects the parity of paths and on each branch the priorities occur in decreasing order. This guarantees that if $\Omega'(X) < \Omega'(Y)$ then X is not free in the formula binding Y.

Therefore Ψ has a tableau that is equivalent to $\mathscr{T}$ and accepts a priority assignment with co-domain $\{0...q\}$. □

Conversely, a disjunctive formula induces a tree with back edges generating its tableau by taking its tableau until each branch reaches a fixpoint variable which is the source of a back edge to its binding formula. The priority assignment of the formula is also a priority assignment for the tree with back-edges. This yields a one-to-one correspondence between trees with back edges and disjunctive formulas.

3 Tableau equivalence preserves alternation depth for disjunctive L_μ

This section argues that all disjunctive formulas generating the same tableau $\mathscr{T}$ have the same alternation depth. The structures used to identify the alternation depth are similar to ones found in [8] to compute the Rabin-Mostowski index of a parity games and the flowers described in [12] to find the Rabin-Mostowski index of non-deterministic automata. Here I show that tableau equivalence preserves these structures and consequently also the alternation depth of disjunctive formulas.

Definition 18 describes a witness showing that the priority assignment Ω of a tree with back edges $\mathscr{A} = (A, \Omega)$ representing $\mathscr{T}$ requires at least q priorities. This witness is preserved by bisimulation with respect to node type, literals and parity of infinite branches. Since all finite representations of a tableau $\mathscr{T}$ are bisimilar with respect to these criteria, they all have the same maximal witness, indicating the least number of priorities $\mathscr{T}$ can be represented with.

Informally, the witness of strictness is a series of cycles of alternating parity where each cycle is contained within the next.

Definition 18. *(q-witness)* A q-witness in a tree with back edges (A, Ω) representing a tableau $\mathscr{T}$ consists of q cycles $c_1 ... c_q$ such that for each $i \leq q$, the cycle c_i is of the parity of i and for all $0 < i < q$, the cycle c_i is a subcycle of c_{i+1}.

Lemma 19. *If a tree with back edges (A, Ω) has a q-witness, then the co-domain of the priority assignment Ω has at least q elements.*

Proof. Given a q-witness $c_1 ... c_q$, for every pair of cycles c_i and c_{i+1}, since they are of different parity and c_i is contained in c_{i+1}, the dominant priority on c_{i+1} must be strictly larger than the dominant priority on c_i. Therefore there must be at least q priorities in the cycle c_q which contains all the other cycles of the witness. □

Lemma 20. *If a tree with back edges $\mathscr{A}$ representing a tableau $\mathscr{T}$ does not have a q-witness, then there is an tree with back edges $\mathscr{A}'$ which also represents $\mathscr{T}$ but has a priority assignment with fewer priorities.*

Proof. Assume a tree with back edges $\mathscr{A} = (A, \Omega)$ representing $\mathscr{T}$ with a priority assignment with co-domain $\{0 ... q\}$ does not have a q witness. Let S_q be the set of nodes of priority q. Let S_{i-1} for $1 < i \leq q$ be the set of nodes of priority $i-1$ which appear as the second highest priority in a cycle where all the nodes of highest priority are in S_i, and as the nodes of highest priority in some cycle. Note that if S_1 was non-empty, then there would be a q-witness, so S_1 and consequently S_0 must be empty. Then define a new priority function as follows: the new priority function Ω' is as Ω, except for nodes in any S_i – these receive the priority $i-2$ instead of the priority i. Since S_1 and S_0 are empty, this is possible whilst keeping all priorities positive. Ω' with co-domain $\{0 ... q-1\}$ preserves the parity of infinite branches since there are no cycles in which the priority of all dominant nodes is decreased more than the priority of all sub-dominant nodes and each node retains the same parity. Therefore, if a finite representation of $\mathscr{T}$ does not have a q-witness, then there is a finite representation $\mathscr{A}' = (A, \Omega')$ with a smaller priority assignment. □

Lemma 21. *All tableau equivalent trees with back edges have the same q-witnesses: for all q, either all or none of the trees with back edges representing a same tableau $\mathscr{T}$ have a q-witness.*

Proof. First we recall that if $\mathscr{A}$ is the finite representation of $\mathscr{T}$ induced by a disjunctive formula Ψ then the tableau of $\mathscr{T}$ is an infinite tree bisimilar to $\mathscr{A}$ with respect to node type, literals and parity of infinite branches. Hence any finite representation of $\mathscr{T}$ is bisimilar to $\mathscr{A}$. It then suffices to show that q-witnesses are preserved under bisimulation. This is straight-forward: let $\mathscr{A}'$ be bisimilar to a finite tree with back edges $\mathscr{A}$ with respect to node type, literals at modal nodes and the parity of infinite paths. Then infinite paths in $\mathscr{A}$ are bisimilar to infinite paths in $\mathscr{A}'$. Since both $\mathscr{A}$ and $\mathscr{A}'$ are finite, an infinite path stemming from a cycle in $\mathscr{A}$ is bisimilar to a cycle in $\mathscr{A}'$. A q-witness contains at least one node which lies on all the cycles of the witness. If $\mathscr{A}$ has q cycles, call the node on all of its cycles n and consider (one of) the deepest node(s) n' in $\mathscr{A}'$ bisimilar to n. That is to say, choose n' such that if another node bisimilar to n' is reachable from n', it must be an ancestor of n'. Since n' is bisimilar to n, there must be a cycle c'_i bisimilar to each c_i reachable from n'. Since n' is maximally deep, it is contained in each of these cycles c'_i. Then, a q-witness can be reconstructed in $\mathscr{A}'$ by taking the cycle c'_1, and then for each $i > 0$ the cycle consisting of all $c'_j, j \leq i$. Since all c'_i cycles have n' in common, there is a cycle combining $c'_j, j \leq i$ for any i. Since bisimulation respects the parity of cycles, this yields a q-witness in $\mathscr{A}'$. □

Theorem 22. *All disjunctive formulas with tableau $\mathscr{T}$ have the same alternation depth.*

Proof. All trees with back edges representing the same tableau $\mathscr{T}$ have the same maximal witness, from the previous lemma, so from Lemma 20 they accept a minimal priority function with domain $\{0...q\}$. Since a disjunctive formula induces a tree with back edges with a minimal priority function corresponding to the formula's alternation depth, any two disjunctive formulas that are tableau equivalent must have the same alternation depth. □

This concludes the proof that tableau equivalence preserves alternation depth on disjunctive formulas. The restriction to disjunctive formulas is crucial: as the next section shows, in the general case tableau equivalent formulas may have vastly different alternation depths.

4 Disjunctive form does not preserve alternation depth

Every formula has a tableau which allows it to be turned into a semantically equivalent disjunctive formula. This section studies the relationship between a formula's alternation depth and the alternation depth of its tableau equivalent disjunctive form. As the previous section shows, any two disjunctive formulas with the same tableau have the same alternation depth; therefore comparing a non-disjunctive formula to any tableau equivalent disjunctive formula will do.

The first subsection demonstrates that not only does disjunctive form not preserve alternation depth, but also that there is no hope for bounding the alternation depth of disjunctive formulas with respect to their semantic alternation depth: for any n there are one alternation formulas which are tableau equivalent to n alternation disjunctive formulas. In other words, the alternation depth of a L_μ formula, when transformed into disjunctive form, can be arbitrarily large. Conversely, as shown in the second subsection, formulas of arbitrarily large alternation depth can be tableau equivalent to a disjunctive formula without alternations. Hence the alternation depth of tableau equivalent formulas are only related within the disjunctive fragment.

4.1 Disjunctive formulas with large alternation depth

While the main theorem is proved by Example 27, the Examples 23 and 25 leading up to it should give the interested reader some intuition about the mechanics which lead the tableau of a formula to have higher alternation depth than one might expect.

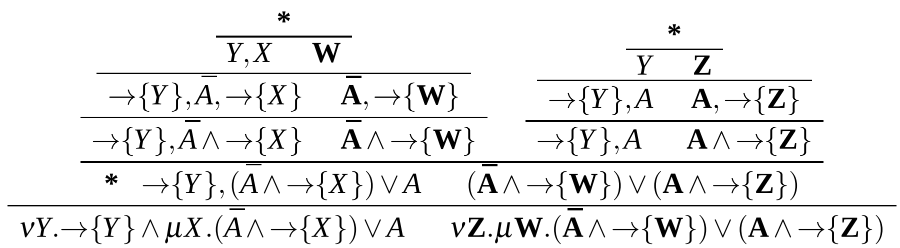

Figure 4.1: Tableaus for $\nu Y.\!\rightarrow\!\{Y\} \wedge \mu X.(\bar{A} \wedge \rightarrow\!\{X\}) \vee A$ and $\nu \mathbf{Z}.\mu \mathbf{W}.(\bar{\mathbf{A}} \wedge \rightarrow\!\{\mathbf{W}\}) \vee (\mathbf{A} \wedge \rightarrow\!\{\mathbf{Z}\})$

Example 23. The first example is a rather simple one: a disjunctive formula with one alternation that can be expressed in non-disjunctive form without any alternations. The disjunctive formula $\nu X.\mu Y.(A \wedge \rightarrow\!\{X\}) \vee (\bar{A} \wedge \rightarrow\!\{Y\})$ signifies that all paths are infinite and A occurs infinitely often on all paths. Compare it to the formula $\nu X.\!\rightarrow\!\{X\} \wedge \mu Y.(\bar{A} \wedge \rightarrow\!\{Y\}) \vee A$ which is alternation free.

The tableaus of both these formulas are shown side by side in Figure 4.1. Both branches regenerate into either exactly the ancestral node marked * or a node that reaches a node identical to the one marked * in a single non branching step.

The cores of the two tableaus, that is to say their branching nodes, are clearly isomorphic with respect to the node type and branching structure. Furthermore, for both formulas, there is μ-trace on any path that only goes through the left hand branch infinitely often. There is no μ trace on *any* path that goes through the right hand path infinitely often, for either formula. As a result, both tableaus agree on the parity of infinite branches. The two formulas are tableau equivalent and therefore also semantically equivalent.

Remark 24. Observe that there is nothing obviously inefficient about how the disjunctive formula handles alternations. Indeed, simply inverting the order of the fixpoints yields a formulas which can not be expressed without an alternation: $\mu X.\nu Y.A \wedge \rightarrow\!\{X\} \vee \bar{A} \wedge \rightarrow\!\{Y\}$.

While the above example proves that disjunctive form does not preserve alternation, it must be noted that the alternating parity automata corresponding to these formulas require in both cases two priorities, although only one requires an alternation. The next example shows formulas in which the number of priorities is not preserved either.

Example 25. This example and the following ones will be built on one-alternation formulas consisting of single μ/ν alternations embedded in one another without interfering with each other, i.e. all free variables within the inner formula ϕ_1 are bound by the inner fixpoint bindings. This means that the formula accepts a priority assignment with co-domain $\{0,1\}$. Without further ado, consider the formula in question:

$$\alpha = \mu X_0.\nu Y_0.(A \wedge \rightarrow\!\{X_0\}) \vee (B \wedge \rightarrow\!\{Y_0) \wedge \mu X_1.\nu Y_1.(C \wedge \rightarrow\!\{X_1\}) \vee (D \wedge \rightarrow\!\{Y_1\}) \vee E$$

The following Lemma shows it to be equivalent to a formula which requires a priority assignment with co-domain $\{0...3\}$.

Lemma 26. *The formula α is tableau equivalent to a disjunctive formula which requires a parity assignment with co-domain $\{0...3\}$:*

$$\begin{aligned}\beta = \mu X_0.\nu Y_0.\mu X_1.\nu Y_1.&(A \wedge C \wedge \rightarrow\!\{X_0\}) \vee (A \wedge D \wedge \rightarrow\!\{X_0\}) \vee (A \wedge E \wedge \rightarrow\!\{X_0\}) \\ &\vee (B \wedge E \wedge \rightarrow\!\{Y_0\}) \vee (B \wedge C \wedge \rightarrow\!\{X_1\}) \vee (B \wedge D \wedge \rightarrow\!\{Y_1\})\end{aligned} \tag{4.1}$$

$$
\dfrac{
\dfrac{
\dfrac{\dfrac{\dfrac{*}{Y_0,Y_1}}{(B,\to\{Y_0\},D,\to\{Y_1\})}}{(B\wedge\to\{Y_0\}),(D\wedge\to\{Y_1\})}
\quad
\dfrac{\dfrac{\dfrac{*}{Y_0,X_1}}{(B,\to\{Y_0\},C,\to\{X_1\})}}{(B\wedge\to\{Y_0\}),(C\wedge\to\{X_1\})}
\quad
\dfrac{\dfrac{\dfrac{*}{Y_0}}{(B,\to\{Y_0\},E)}}{(B\wedge\to\{Y_0\},E)}
}{(B\wedge\to\{Y_0\}),(C\wedge\to\{X_1\})\vee(D\wedge\to\{Y_1\})\vee E}
\quad
\dfrac{
\dfrac{\dfrac{\dfrac{*}{X_0,Y_1}}{(A,\to\{X_0\},D,\to\{Y_1\})}}{(A\wedge\to\{X_0\}),(D\wedge\to\{Y_1\})}
\quad
\dfrac{\dfrac{\dfrac{*}{X_0,X_1}}{(A,\to\{X_0\},C,\to\{X_1\})}}{(A\wedge\to\{X_0\}),(C\wedge\to\{X_1\})}
\quad
\dfrac{\dfrac{\dfrac{*}{X_0}}{(A,\to\{X_0\},E)}}{(A\wedge\to\{X_0\},E)}
}{(A\wedge\to\{X_0\}),(C\wedge\to\{X_1\})\vee(D\wedge\to\{Y_1\})\vee E}
}{\dfrac{*(A\wedge\to\{X_0\})\vee(B\wedge\to\{Y_0\}),(C\wedge\to\{X_1\})\vee(D\wedge\to\{Y_1\})\vee E}{\mu X_0.\nu Y_0.(A\wedge\to\{X_0\})\vee(B\wedge\to\{Y_0\})\wedge\mu X_1\nu Y_1(C\wedge\to\{X_1\})\vee(D\wedge\to\{Y_1\})\vee E}}
$$

Figure 4.2: Tableau for α

$$
\dfrac{
\dfrac{
\dfrac{\dfrac{\dfrac{*}{Y_1}}{(B,D,\to\{Y_1\})}}{(B\wedge D\wedge\to\{Y_1\})}
\quad
\dfrac{\dfrac{\dfrac{*}{X_1}}{(B,C,\to\{X_1\})}}{(B\wedge C\wedge\to\{X_1\})}
\quad
\dfrac{\dfrac{\dfrac{*}{Y_0}}{(B,E,\to\{Y_0\})}}{(B\wedge E\wedge\to\{Y_0\})}
}{(B\wedge E\wedge\to\{Y_0\})\vee(B\wedge C\wedge\to\{X_1\})\vee(B\wedge D\wedge\to\{Y_1\})}
\quad
\dfrac{
\dfrac{\dfrac{\dfrac{*}{X_0}}{(A,D,\to\{X_0\})}}{(A\wedge D\wedge\to\{X_0\})}
\quad
\dfrac{\dfrac{\dfrac{*}{X_0}}{(A,C,\to\{X_0\})}}{(A\wedge C\wedge\to\{X_0\})}
\quad
\dfrac{\dfrac{\dfrac{*}{X_0}}{(A,E,\to\{X_0\})}}{(A\wedge E\wedge\to\{X_0\})}
}{(A\wedge E\wedge\to\{X_0\})\vee(A\wedge D\wedge\to\{X_0\})\vee(A\wedge C\wedge\to\{X_0\})}
}{\dfrac{*\,(A\wedge E\wedge\to\{X_0\})\vee(A\wedge D\wedge\to\{X_0\})\vee(A\wedge C\wedge\to\{X_0\})\vee(B\wedge E\wedge\to\{Y_0\})\vee(B\wedge C\wedge\to\{X_1\})\vee(B\wedge D\wedge\to\{Y_1\})}{\mu X_0.\nu Y_0.\mu X_1.\nu Y_1.(A\wedge E\wedge\to\{X_0\})\vee(A\wedge D\wedge\to\{X_0\})\vee(A\wedge C\wedge\to\{X_0\})\vee(B\wedge E\wedge\to\{Y_0\})\vee(B\wedge C\wedge\to\{X_1\})\vee(B\wedge D\wedge\to\{Y_1\})}}
$$

Figure 4.3: Tableau for β

Proof. The tableaus for both formulas are written out in Figures 4.2 and 4.3. The two tableaus are isomorphic with respect to branching structure, node type and the literals at modal nodes. To prove their equivalence, it is therefore sufficient to argue that this isomorphism also preserves the parity of infinite branches, that is to say that there is a μ-trace in an infinite path of one if and only if there is a μ-trace in the corresponding infinite path of the other.

To do so, we look, case by case, at the combinations of branches that a path can see infinitely often and check which have a μ trace in each tableau. First argue that the three right-most branches in both tableaus are such that any path that sees them infinitely often has a μ-trace. This is witnessed in both cases by the least fixpoint variable X_0 which will dominate any trace it appears on and appears on a trace on all paths going through one of these branches infinitely often. So, in both tableaus, any path going through one of the right-most branches infinitely often is of odd parity. Now consider the branch that ends in Y_0 before regenerating to the node marked * in both tableaus. All traces on paths that go infinitely often through this branch will see Y_0 regenerate infinitely often. Therefore in both tableaus, a path going through this branch infinitely has a μ trace if and only if it also goes through one of the three rightmost branches infinitely often. Now consider the fifth branch from the right, the branch that regenerates Y_0, X_1 in one case and just X_1 in the other. In both tableaus, a path that goes through this branch infinitely often will have a μ trace unless it goes through the Y_0 branch infinitely often and doesn't go through one of the three right-most branches infinitely often. Finally, in both tableaus, a branch that only sees the left-most branch infinitely often is of even parity since such a path does not admit any μ-traces. However, if a path sees this branch and some other branches infinitely often, its parity is determined by one of the previously analysed cases. Since we have analysed all the infinite paths on these tableaus and concluded that in each case the parity of a path is the same in both tableaus, this concludes the proof that the two tableaus are equivalent. □

The above example yields a disjunctive formula of alternation depth $\{0...3\}$ which semantically only requires alternation depth $\{0,1\}$. This proves that disjunctive form does not preserve the number of priorities the model checking game of a formula requires.

The next step is to generalise the construction of Example 25 to arbitrarily many alternations to prove that there is no bound on the number of alternations of a disjunctive formula tableau equivalent to a non-disjunctive formula of n alternations. To do so, we will first define the one-alternation formulas α_n inductively, based on the formula of Example 25. We then argue that the tableau of α_n admits a $(2n+1)$-witness, proving that α_n is not tableau equivalent to any disjunctive formula of less than $2n+1$ alternations. Due to the argument pertaining to traces in increasingly large tableaus, its details are, inevitably, quite involved. However, the mechanics of the tableaus of α_n are not difficult; writing down the tableau of α_2 and working out its disjunctive form should suffice to gain an intuition of the proof to follow.

Example 27. In order to define α_n for any n define:

$$\begin{aligned} a_1 &= \mu X_1.\nu Y_1.((A_1 \wedge \rightarrow\{X_1\}) \vee (B_1 \wedge \rightarrow\{Y_1\}) \vee E_1) \wedge \\ &\quad \mu X_0.\nu Y_0.(A_0 \wedge \rightarrow\{X_0\}) \vee (B_0 \wedge \rightarrow\{Y_0\}) \vee E_0 \\ a_{i+1} &= \mu X_{i+1}.\nu Y_{i+1}.((A_{i+1} \wedge \rightarrow\{X_{i+1}\}) \vee (B_{i+1} \wedge \rightarrow\{Y_{i+1}\}) \vee E_{i+1}) \wedge a_i \end{aligned} \tag{4.2}$$

Then, define:

$$\alpha_n = \mu X_n.\nu Y_n.((A_n \wedge \rightarrow\{X_n\}) \vee (B_n \wedge \rightarrow\{Y_n\})) \wedge a_{n-1}$$

In other words, the formula consists of nested clauses $\mu X_i.\nu Y_i.((A_i \wedge \rightarrow\{X_i\}) \vee (B_i \wedge \rightarrow\{Y_i\}) \vee E_i)$ connected by conjunctions where the outmost clause does not have a $\vee E$.

As the formula grows, its tableau becomes unwieldy, but its structure remains constant: it is just as the tableau of α with more branches. Figure 4.2 can be used as reference.

The tableau of any α_n follows this structure:

- The first choice node $\{(A_n \wedge \rightarrow\{X_n\} \vee B_n \wedge \rightarrow\{Y_n\}), \ldots, (A_0 \wedge \rightarrow\{X_0\} \vee B_0 \wedge \rightarrow\{Y_0\} \vee E_0)\}$ branches into 2×3^n modal nodes – ignoring the modalities attached to each literals for a moment, this is the decomposition of $(A_n \vee B_n) \wedge (A_{n-1} \vee B_{n-1} \vee E_{n-1}) \ldots \wedge (A_0 \vee B_0 \vee E_0)$ into one large disjunction.
- Each choice leads to a modal node with some choice of propositional variables consisting of one of A_n and B_n and then for every $i < n$ one of A_i, B_i or E_i.
- These modal nodes have a single successor each, consisting of a set of fixpoint variables. In every case, one of these is Y_n or X_n and there is only ever at most one fixpoint variable out of $\{X_i, Y_i\}$ for each i. These nodes will be referred to as regeneration nodes. When a regeneration node does not contain X_i nor Y_i for some i, this corresponds to E_i having been chosen rather than A_i or B_i.
- Nodes consisting of a set of fixpoint variables all regenerate, give or take a couple of non-branching steps, into the same choice node, identical to the ancestral choice node labelled:

$$\{(A_n \wedge \rightarrow\{X_n\} \vee B_n \wedge \rightarrow\{Y_n\}), \ldots, (A_0 \wedge \rightarrow\{X_0\} \vee B_0 \wedge \rightarrow\{Y_0\} \vee E_0)\}$$

- An infinite trace in this tableau sees infinitely often only fixpoint variables Y_i and/or X_i for some i. As a consequence if a path goes infinitely often through a regeneration node which does not contain X_i or Y_i, then there is no trace that sees X_i infinitely often on that path.

Lemma 28. *The formula α_n is tableau equivalent only to disjunctive formulas which require a priority assignment with $2n+1$ priorities.*

Proof. Using the above observations, we will show that the tableau for this formula requires at least $2n+1$ alternating fixpoints. We describe a priority assignment to a subset of the nodes of the tableau of α_n such that on the paths within this subset, a path is even if and only if the most significant priority seen infinitely often is even. We then argue that this subset constitutes a $2n+1$-witness.

Consider the paths of the tableau which only contain the following regeneration nodes:

- For all i, the nodes regenerating exactly $Y_nY_{n-1}...Y_i$, and
- For all i the nodes regenerating exactly $Y_n...Y_{i+1}X_iY_{i-1}...Y_0$.

For each i, assign priority $2i$ to the node regenerating $Y_n...Y_i$ and $2i+1$ to the node regenerating $Y_n...Y_{i+1},X_i,Y_{i-1},...Y_0$. We now prove that this priority assignment is such that a path within this sub-tableau is even if and only if the highest priority seen infinitely often is even.

First consider the nodes $Y_n...Y_i$, which have been assigned even priority. A path that sees such a node infinitely often can only have a μ-trace if it sees a node regerating some X_j, $j>i$ infinitely often. Such a node would have an odd priority greater than $Y_n...Y_i$. Therefore, if the most significant priority seen infinitely often is even, the path has no μ trace. Conversely, if a path sees $Y_n...X_i...Y_0$ infinitely often and no $Y_n...Y_j$ where $j>i$ infinitely often, then there is a trace which only regenerated X_i and Y_i infinitely often. This is a μ trace since X_i is more significant than Y_i. This priority assignment therefore describes the parity of infinite paths on this subset of paths of $\mathscr{T}$.

Any assignment of priorities onto $\mathscr{T}$ should, on this subset of paths, agree in parity with the above priority assignment. However, in any tree with back edges generating this tableau, this subset of paths constitutes a $2n+1$ witness: c_0 is a cycle that only sees $Y_n...Y_0$, c_1 contains c_0 and also sees $Y_n...X_1Y_0$ infinitely often and for all $i>1$, the cycle c_{2i} is one containing c_{2i-1} and $Y_n...Y_i$ while c_{2i+1} is one containing c_{2i} and $Y_n...X_i...Y_0$. Each cycle c_j is dominated by the priority j, making $c_0,...,c_{2i+1}$ a $2i+1$-witness. Thus, using Theorem 22 any disjunctive formula with tableau $\mathscr{T}$ must require at least $2n+1$ priorities. □

This concludes the proof that for arbitrary n, there are one-alternation L_μ formulas which are tableau equivalent to disjunctive formulas with n alternations.

4.2 Disjunctive formulas with small alternation depth

The previous section showed that transforming a formula into disjunctive form can increase its alternation depth. The converse is much easier to show: there are very simple formulas for which the transformation into disjunctive form eliminates all alternations.

Lemma 29. *For any formula ψ, the formula $(\mu X.{\rightarrow}\{X\} \vee {\rightarrow}\bot) \wedge \psi$ is tableau equivalent to a disjunctive formula without ν-operators.*

Proof. The semantics of $(\mu X.{\rightarrow}\{X\} \vee {\rightarrow}\bot) \wedge \psi$ are that a structure must not have infinite paths and ψ must hold. Consider $\mathscr{T}$, the tableau for $(\mu X.{\rightarrow}\{X\} \vee {\rightarrow}\bot) \wedge \psi$. It is easy to see that every modal node will either contain ${\rightarrow}\{X\}$ or ${\rightarrow}\bot$. The latter case terminates that branch of the tableau, while the former will populate every successor node with X which will then regenerate into $({\rightarrow}\{X\} \vee {\rightarrow}\bot)$. As a result, all infinite paths have a μ trace; there are no even infinite paths. Any disjunctive formula generating $\mathscr{T}$ will therefore only require the μ operator. □

Taking ψ to be a formula of arbitrarily high alternation depth, $(\mu X.{\rightarrow}\{X\} \vee {\rightarrow}\bot) \wedge \psi$ shows that the transformation into disjunctive form can reduce the alternation depth an arbitrarily large amount. Together with the previous section, this concludes the argument that there are no bounds on the difference in alternation depth of tableau equivalent formulas.

5 Discussion

To summarise, we have studied how tableau decomposition and the transformation into disjunctive form affects the alternation depth of a formula. The first observation is that within the confines of the disjunctive fragment of L_μ, alternation depth is very well-behaved with respect to tableau equivalence: any two tableau equivalent disjunctive formulas have the same alternation depth. However, the story is quite different for L_μ without the restriction to disjunctive form: the alternation depth of a L_μ formula can not be used to predict any bounds on the alternation depth of tableau equivalent disjunctive formulas and vice versa.

Part of the significance of this result are the implications for our understanding of the alternation hierarchy.

The formulas in Section 4 illutrate some of the different types of accidental complexity which any procedure for deciding the alternation hierarchy would need to somehow overcome. The formula $(\mu X.\!\to\!\{X\} \vee \bot) \wedge \psi$, from Lemma 29 which is semantically a ν-free formula for any ψ is an example of a type of accidental complexity which the tableau decomposition eliminates. However, the formula in Example 23 illustrate a more subtle form of accidental complexity that is immune to disjunctive form: $\nu X.\mu Y.(A \wedge \to\{X\}) \vee (\overline{A} \wedge \to\{Y\})$ is semantically alternation free while the syntactically almost identical formula $\mu X.\nu Y.(A \wedge \to\{X\}) \vee (\overline{A} \wedge \to\{Y\})$ is not. These formulas pinpoint a very specific challenge facing algorithms that try to reduce the alternation depth of formulas; as such, they are valuable case studies for those seeking to understand the L_μ alternation hierarchy.

Finally, we showed that the following is decidable: for any L_μ formula, the least alternation depth of a tableau equivalent disjunctive formula is decidable. This raises the question of whether the same is true if we lift the restriction to disjunctive form, but keep the restriction to tableau equivalence: for a L_μ formula, is the least alternation depth of any tableau equivalent formula decidable? Tableau equivalence is a stricter equivalence to semantic equivalence, so this problem is likely to be easier than deciding the alternation hierarchy with respect to semantic equivalence but it would still be a considerable step towards understanding accidental complexity in L_μ.

Acknowledgements I thank the anonymous reviewers for their thoughtful comments, which have helped improve the presentation of this paper and relate this work to similar results for other automata.

6 Bibliography

References

[1] J.C. Bradfield (1996): *The modal mu-calculus alternation hierarchy is strict*. In U. Montanari & V. Sassone, editors: *CONCUR '96: Concurrency Theory*, *Lecture Notes in Computer Science* 1119, Springer Berlin Heidelberg, pp. 233–246, doi:10.1007/3-540-61604-7_58.

[2] J.C. Bradfield & C. Stirling (2007): *Modal mu-calculi*. *Handbook of modal logic* 3, pp. 721–756, doi:10.1016/S1570-2464(07)80015-2.

[3] O. Carton & R. Maceiras (1999): *Computing the Rabin index of a parity automaton*. *RAIRO - Theoretical Informatics and Applications - Informatique Thorique et Applications* 33(6), pp. 495–505, doi:10.1051/ita:1999129. Available at http://eudml.org/doc/92617.

[4] T. Colcombet & C. Löding (2008): *The non-deterministic Mostowski hierarchy and distance-parity automata*. In: *Automata, languages and programming*, Springer, pp. 398–409, doi:10.1007/978-3-540-70583-3_33.

[5] G. D'Agostino & M. Hollenberg (2000): *Logical Questions Concerning The mu-Calculus: Interpolation, Lyndon and Los-Tarski*. *J. Symb. Log.* 65(1), pp. 310–332, doi:10.2307/2586539.

[6] E. A. Emerson & C. S. Jutla (1991): *Tree automata, mu-calculus and determinacy*. In: *Foundations of Computer Science, 1991. Proceedings., 32nd Annual Symposium on*, IEEE, pp. 368–377, doi:10.1109/SFCS.1991.185392.

[7] A. Facchini, F. Murlak & M. Skrzypczak (2013): *Rabin-Mostowski index problem: a step beyond deterministic automata*. In: *Proceedings of the 2013 28th Annual ACM/IEEE Symposium on Logic in Computer Science*, IEEE Computer Society, pp. 499–508, doi:10.1109/LICS.2013.56.

[8] M. Huth, J. Kuo & N. Piterman (2012): *The Rabin Index of Parity Games*. In K. Eder, J. Loureno & O. Shehory, editors: *Hardware and Software: Verification and Testing*, *Lecture Notes in Computer Science* 7261, Springer Berlin Heidelberg, pp. 259–260, doi:10.1007/978-3-642-34188-5_25.

[9] D. Janin & I. Walukiewicz (1995): *Automata for the modal μ-calculus and related results*. In: *Proc. MFCS '95 LNCS 969*, pp. 552–562, doi:10.1007/3-540-60246-1_160.

[10] M. Jurdziński (2000): *Small progress measures for solving parity games*. In: *STACS 2000*, Springer, pp. 290–301, doi:10.1007/3-540-46541-3_24.

[11] M.K. Lehtinen & S. Quickert (2015): *Deciding the first levels of the modal μ alternation hierarchy by formula construction*. In: *(forthcoming) Proc. CSL '15*.

[12] A. Niwiski & I. Walukiewicz (2005): *Deciding Nondeterministic Hierarchy of Deterministic Tree Automata*. *Electronic Notes in Theoretical Computer Science* 123, pp. 195 – 208, doi:10.1016/j.entcs.2004.05.015. Proceedings of the 11th Workshop on Logic, Language, Information and Computation (WoLLIC 2004).

[13] D. Niwiski & I. Walukiewicz (1998): *Relating hierarchies of word and tree automata*. In M. Morvan, C. Meinel & D. Krob, editors: *STACS 98*, *Lecture Notes in Computer Science* 1373, Springer Berlin Heidelberg, pp. 320–331, doi:10.1007/BFb0028571.

[14] I. Walukiewicz (2000): *Completeness of Kozen's axiomatisation of the propositional μ-calculus*. *Information and Computation* 157(1), pp. 142–182, doi:10.1006/inco.1999.2836.