The Descriptive Complexity
of Modal μ Model-checking Games

Karoliina Lehtinen

University of Kiel
Kiel, Germany

kleh@informatik.uni-kiel.de

This paper revisits the well-established relationship between the modal μ calculus L_μ and parity games to show that it is even more robust than previously known. It addresses the question of whether the descriptive complexity of L_μ model-checking games, previously known to depend on the syntactic complexity of a formula, depends in fact on its semantic complexity. It shows that up to formulas of semantic complexity Σ_2^μ, the descriptive complexity of their model-checking games coincides exactly with their semantic complexity. Beyond Σ_2^μ, the descriptive complexity of the model-checking parity games of a formula Ψ is shown to be an upper bound on the semantic complexity of Ψ; whether it is also a lower bound remains an open question.

1 Introduction

The modal μ-calculus [9], written L_μ, is a verification logic consisting of a simple modal logic augmented with a least fixpoint μ and its dual ν. It is a prime example of the intersection between logic and games: In its model-checking games, the antagonism between the verifying player Even and her opponent Odd describes the duality between conjunctions and disjunctions, and between least and greatest fixpoints. The complexity of the winning condition – a parity condition over a set of priorities – corresponds to the complexity of the formula, as measured by its index, *i.e.* the number of alternations between least and greatest fixpoints.

The index of a formula is a robust measure of a formula's complexity: for each fixed index, the model-checking problem is in P, but no fixed index suffices to express all L_μ-expressible properties [1]. The decidability of the *semantic index* of a formula, that is to say the least index of any equivalent formula, is a long-standing open problem. Despite the efforts put towards solving it and its automata-theoretic counterpart, the Mostowski–Rabin index problem for parity automata, only the low levels of the alternation hierarchy are currently known to be decidable. Formulas semantically in *ML*, the fragment without fixpoints, and those in Π_1^μ and Σ_1^μ, the fragments with only one type of fixpoint, were first shown to be decidable by Otto [20] and Küsters and Wilke [12] respectively. Alternative formula-focused decidability proofs for both were derived in [14]. Some results in automata theory [4] can be interpreted within the L_μ context to yield another decidability result: given a formula in the Π_2^μ alternation class, it is decidable whether it is equivalent on ranked trees to a formula in the Σ_2^μ alternation class. An alternative proof with a topological account was given in [21], while in [15] a game-theoretic characterisation of Σ_2^μ extends the same result onto unranked trees: Σ_2^μ is decidable for Π_2^μ.

This paper revisits one of the fundamental concepts of L_μ literature in light of the index-problem: the relationship between L_μ formulas and parity games. Parity games can be used to describe the semantics of L_μ and are therefore a recurring topic in L_μ literature. This relationship is complexity-preserving, in the following sense: On the one hand, the model-checking games for L_μ formulas of index I are parity games with priorities from I; on the other, a L_μ formula of index I suffices to describe the winning

regions of such games. As a result, the descriptive complexity of the model-checking games of a formula Ψ is bounded by the index of Ψ. This paper asks whether this complexity-preserving relationship can be extended to also account for the *semantic* complexity of a formula: can the bounds on the complexity of model-checking games be tightened to the semantic index of Ψ?

In its first part (Section 3), this paper argues that if the winning regions of the model-checking games for Ψ are described by Φ – written Φ interprets Ψ – then Ψ is semantically as simple as Φ, *i.e.* equivalent to a formula of the same index as Φ. In other words, the descriptive complexity of the model-checking games of a formula is an upper bound on the semantic complexity of the formula.

The second part of the paper (Section 4) considers the converse: if a formula is semantically of index I, then is it interpreted by a formula of index I? That is to say, do semantically simple formulas generate equally simple model-checking games? This is shown to be the case for the first semantic levels of the alternation hierarchy. In some cases, the input formula must be transformed into disjunctive form [8] first, a normal form for L_μ:

- If a L_μ formula is equivalent to a modal formula, it is interpreted by a modal formula;

- If a disjunctive L_μ formula is equivalent to a Π_1^μ formula, it is interpreted by a Π_1^μ formula;

- If a disjunctive L_μ formula is equivalent to a Σ_2^μ formula, it is interpreted by a Σ_2^μ formula;

Interestingly, these theorems are based on characterising the formulas semantically in ML, Π_1^μ and Σ_2^μ as those for which the model-checking games coincide with variations of parity games that characterise the class. For these (semantic) alternation-classes at least, the descriptive complexity of the model-checking games of a formula Ψ corresponds exactly to the semantic index of Ψ, rather than its syntactic index.

Section 4.4 considers the general case and shows that if a *co-disjunctive* formula is equivalent to a *disjunctive* formula of index I, then it is also interpreted by a disjunctive formula of index I. I conjecture that the same holds for disjunctive input formulas, but the currently available techniques do not suffice to prove this.

In short, this paper shows that the relation between L_μ and parity games can be extended in a natural way to account for the L_μ index problem: the descriptive complexity of the model-checking games of Ψ is an upper bound on the semantic complexity of Ψ. For several non-trivial cases, and in particular beyond the alternation classes currently known to be decidable, the descriptive complexity of the model-checking games generated by Ψ coincides exactly with the semantic complexity of Ψ. In the general case, the index of an equivalent disjunctive formula bounds the descriptive complexity of a co-disjunctive formula. Overall, the L_μ–parity game relationship is even more robust than previously thought.

One practical consequence is a method for simplifying formulas by analysing their model-checking games (see Example 15). More generally, if one can show that a fragment $F \subset L_\mu$ generates model-checking games of bounded descriptive complexity, then F is embedded in a fixed level of the L_μ alternation hierarchy. Furthermore, the study of the descriptive complexity of the model-checking games is a novel approach to the L_μ-index problem. Although the three decision procedures for ML, Π_1^μ and Σ_2^μ (for Π_2^μ formulas) each uses a completely different strategy, describing the winning regions of formulas semantically in the target class subsumes all three. Extending the interpretation theorems of this paper seems likely to lead to characterisations of higher L_μ alternation classes, providing a stepping stone towards further decidability results. In particular, a generalisation of the Σ_2^μ interpretation theorem would yield a reduction of the decidability of the index problem to a boundedness question, in the style of what Colcombet *et. al.* achieved for non-deterministic automata [5].

The descriptive complexity of parity games has already been considered in [6] where the authors ask which formalisms can describe the winning regions of classes of parity games without a fixed index.

They show that guarded second order logic suffices in the general case while least fixed point logic does not. On finite game arenas, the winning regions are definable in least fixed point logic if and only if solving parity games is in P. In contrast, the winning regions of the classes of parity games considered here are trivially L_μ expressible, and the focus is on exactly which fragment of L_μ is necessary.

The following section fixes the notation and terminology used throughout. Some familiarity with L_μ and parity games is assumed – see for example [3] for an introduction.

2   Preliminairies

2.1 L_μ syntax

Fix countably infinite sets $Prop = \{P, Q, ...\}$ of propositional variables, and $Var = \{X, Y, ...\}$ of fixpoint variables. For the sake of clarity and conciseness the scope of this paper is restricted to the unimodal L_μ; however, the results presented here extend easily to the multi-modal case.

Definition 1. *(L_μ)* The syntax of unimodal L_μ is given by:

$$\phi := P \mid X \mid \neg P \mid \phi \wedge \phi \mid \phi \vee \phi \mid \Diamond \phi \mid \Box \phi \mid \mu X.\phi \mid \nu X.\phi \mid \bot \mid \top$$

Conjunctions take precedence over disjunctions. The scope of fixpoint bindings extends as far as possible to the right while the scope of modalities extends as little as possible to the right. For example, $\mu X.\Diamond X \wedge C \vee B$ is parsed as $\mu X.(((\Diamond X) \wedge C) \vee B)$. Note that here negation can only be applied to propositional variables.

A fixpoint variable X which does not appear in the scope of $\mu X.\phi$ or $\nu X.\phi$ is said to be *free*, and the set of free fixpoint variables of a formula ψ is written $Free_\psi$. A sentence is a formula without free fixpoint variables. A fixpoint μ or ν *binds* a variable $X \in Free_\phi$ in $\mu X.\phi$ or $\nu X.\phi$ respectively. The formula ϕ is called the binding formula of X, and will be written ϕ_X. For notational purposes, the binding formula of X will be treated as an immediate subformula of X. The *parse tree* of a sentence is the tree with the subformulas of Ψ for nodes, rooted at the formula itself, and where the children of a node consist of the parse-trees of its immediate subformulas. In other words it consists of the formula, written as a tree, with back-edges from fixpoint variables to their bindings.

A formula is *guarded* if every fixpoint variable is in the scope of a modality within its binding. Without loss of expressivity [18, 11], all L_μ formulas are assumed to be in guarded form.

Disjunctive form is a normal form for L_μ formulas [8] which imposes some additional structure onto the L_μ syntax. The dual definition yields co-disjunctive formulas. Every L_μ can be turned into a disjunctive formula, although the transformation may not preserve complexity [13]. Unlike in general L_μ model-checking games, in the model-checking games of disjunctive formulas, the verifying player has strategies which only agree with one play per branch [14].

2.2   Alternations

The definition of alternation depth, here referred to as index to match the automata-theoretic terminology, is meant to only capture alternations which generate algorithmic complexity, matching the definition given in Niwiński 1986 [19]. The presentation here emphasizes the relationship of a formula's index to the priorities in parity games. A thorough discussion on how to define alternation depth, and comparison of definitions used in the literature can be found in Bradfield and Stirling 2007 [3].

Definition 2. *(Priority assignment and index)* A priority assignment for a sentence Ψ is a mapping $\Omega : Var_\Psi \to \{m,...,q-1,q\}$, where $m \in \{0,1\}$ and q is a positive integer, of priorities to the fixpoint variables Var_Ψ of Ψ such that μ-bound variables receive odd priorities while ν-bound variables receive even priorities. A priority assignment is order preserving if whenever X is free in the formula binding Y, $\Omega(X) \geq \Omega(Y)$ holds.

A formula has index I if it has an order preserving priority assignment with co-domain I. A formula has semantic index I if it is equivalent to a formula of index I.

Definition 3. *(Alternation hierarchy)* The base of the alternation hierarchy is *ML*, the modal fragment of L_μ, consisting of formulas without any fixpoints. Formulas with only ν-bound fixpoints, or only μ-bound fixpoints respectively, have index $\{0\}$, or $\{1\}$, corresponding to the alternation classes Π_1^μ, or Σ_1^μ. The classes Π_i^μ and Σ_i^μ for positive even i correspond to formulas with indices $\{1,...,i\}$ and $\{0,...,i-1\}$ respectively, while for odd i they correspond to formulas with indices $\{0,...,i-1\}$ and $\{1,...,i\}$ respectively.

Example 4. The formula $\mu X.\nu Y.(\Box Y \wedge \mu Z.\Box(X \vee Z))$ accepts the order-preserving assignment $\Omega(X) = 1, \Omega(Y) = 0$ and $\Omega(Z) = 1$, and has therefore index $\{0,1\}$ and is in Σ_2^μ. It is however equivalent to $\mu X.\Box X$ (see Section 2.3) and is therefore semantically in Σ_1^μ.

Theorem 5. *[1, 16] The alternation hierarchy is strict: for each index $I = \{m,...,q\}$, where $m \in \{0,1\}$, there are formulas that are not equivalent to a formula with smaller index.*

Deciding an index I means deciding, given an arbitrary L_μ formula, whether it is equivalent to any formula of index I.

2.3 L_μ semantics and parity games

We define the semantics of L_μ in terms of its model-checking games, known as parity games. The correspondence with the classical semantics is standard [3].

L_μ formulas operate on regular unranked trees. For unimodal L_μ, these trees do not have edge-labels.

Definition 6. *(Trees)* An unranked tree $\mathfrak{T} = (S,E,r,P)$, rooted at $r \in S$ consists of: a set of states S, a successor relation $E \subseteq S \times S$, and a labelling $P : S \to \mathscr{P}(Prop_\mathfrak{T})$ mapping states to subsets of a finite set $Prop_\mathfrak{T} \subset Prop$. For each state $s \in S$, the set $\{w \in S \mid \exists w_1,...,w_k. (w,w_1) \in E, (w_1,w_2) \in E, ... (w_k,s) \in E\}$ of its ancestors, is finite and well-ordered with respect to the transitive closure of E.

The scope of this paper is restricted to regular trees with finite but unbounded branching, which can be finitely represented as trees with back edges. Since L_μ enjoys a finite model property [10], the index problem on trees with infinite branching reduces to the index problem on trees with finite branching.

Definition 7. *(Parity game)* A parity game arena is $A = (V,E,v_\iota,\Omega)$ consisting of: a set V of states partitioned into those belonging to Even, V_e and those belonging to Odd, V_o; an edge relation $E \subseteq V \times V$; an initial node v_ι; and a priority assignment $\Omega : V \to I$. The co-domain I of Ω is said to be the index of A. I is a prefix of the natural numbers, starting at either 0 or 1.

A play in a parity game is a potentially infinite sequence of successive positions starting with v_ι. A play is finite if it ends in a position without successors; then the owner of the position loses. For infinite plays, the winner depends on the highest priority seen infinitely often, called the dominant priority. The winner is the player of the parity of the dominant priority.

A positional (or memoryless) strategy σ for Even (and similarly for Odd) consists of a choice $\sigma(v)$ of successor at the nodes v in V_e (or V_o). A play $\pi = v_0, v_1, ...$, where $v_0 = v_\iota$, agrees with a strategy σ if, at every position $v_i \in V_e$ (or V_o) along the play, $v_{i+1} = \sigma(v_i)$.

A pair of strategies, one for each player, induces a unique play. A strategy for a player is winning if every play that agrees with it is winning for this player. Parity games are positionally determined [7, 17]: positional strategies suffice and exactly one of the players has a winning strategy from every position. A parity game $G(A)$ on arena A is said to be winning for the player with a winnning strategy.

We define the semantics of L_μ in terms of winning strategies in parity games: For an unranked tree $\mathfrak{T}$ and formula Ψ, we say that $\mathfrak{T}$ satisfies Ψ, written $\mathfrak{T} \models \Psi$ if and only if Even has a winning strategy in the model-checking parity game $G(\mathfrak{T} \times \Psi)$ defined below.

Definition 8. *(Model-checking parity game)* Let Ψ be a sentence of L_μ and Ω_Ψ a priority assignment with co-domain I on the fixpoint variables of Ψ. Then, for any unranked tree $\mathfrak{T}$, define the model-checking parity game arena $\mathfrak{T} \times \Psi$ as follows: $\mathfrak{T} \times \Psi = (V, E, v_\iota, \Omega)$ where:

- V is the set of states (s, ϕ), where s is a state of $\mathfrak{T}$ and ϕ is a subformula of Ψ.

- V_o consists of positions $(s, \phi \wedge \psi)$ and $(s, \Box\phi)$ while $V_e = V \setminus V_o$;

- Positions (s, C), for C a literal, are terminal and if $C \in P(s)$, belong to Odd; else it belongs to Even;

- There is an edge from $(s, \phi \vee \psi)$ to (s, ϕ) and (s, ψ);
  an edge from $(s, \phi \wedge \psi)$ to (s, ϕ) and (s, ψ);
  an edge from (s, X), $(s, \mu X. \phi_X)$ and $(s, \nu X. \phi_X)$ to (s, ϕ_X);
  an edge from $(s, \Diamond\phi)$ and $(s, \Box\phi)$ to (s', ϕ) for every successor s' of s;

- v_ι the initial position is (r, Ψ), where r is the root of $\mathfrak{T}$;

- Ω assigns $\Omega_\Psi(X)$ to positions $s \times X$; Ω assigns the minimal priority of I to all other positions.

Definition 9. *(Semantics of L_μ via games)* If Ω_Ψ is an order-preserving priority assignment for Ψ, then for all $\mathfrak{T}$, Even has a winning strategy in the parity game $G(\mathfrak{T} \times \Psi)$ if and only if $\mathfrak{T} \models \Psi$. Furthermore, Even has a winning strategy in $G(\mathfrak{T} \times \Psi)$ from each position (s, ϕ) of $\mathfrak{T} \times \Psi$, where s is a state of $\mathfrak{T}$ and ϕ is a subformula of Ψ, if and only if $s \models \phi$.

Definition 10. The model-checking parity games generated by a formula Ψ are encoded as trees by assigning propositional variables E_i to positions belonging to Even with priority i and O_i to positions belonging to Odd of priority i. Encoding more data about the provenance of positions will be useful in the second half of this paper: we will mark modal positions, that is to say positions $s \times \Box\phi$ and $s \times \Diamond\phi$ with a propositional variable M.

The winning regions of parity games can be described by L_μ formulas [7], shown to be complete for their index (*i.e.* not equivalent to any formula of lower index)[2].

For an index $I = \{m, ..., q\}$ with $m \in \{0, 1\}$, define the formula:

$$\mathsf{Parity}_I = \gamma_q X_q.... \gamma_m X_m. \bigvee_{i \in I} (E_i \wedge \Diamond X_i) \vee (O_i \wedge \Box X_i)$$

where γ_i is μ for odd i and ν for even i.

Definition 11. Φ *interpret* Ψ if for all trees $\mathfrak{T}$, it is the case that $\mathfrak{T} \models \Psi$ if and only if $\mathfrak{T} \times \Psi \models \Phi$.

Theorem 12. *[7, 22] For all Ψ of index I, Parity_I interprets Ψ.*

3 The descriptive complexity of parity games

The last section recalled the classic result that the syntactic complexity of Ψ is an upper bound on the descriptive complexity of its model-checking game. Here we show that if a formula of index I interprets Ψ, then the formula Ψ is equivalent to one of index I. In other words, the semantic complexity of a L_μ formula is a lower bound to the descriptive complexity of its model-checking games.

Theorem 13. *Let Ψ be a formula of L_μ. If for some formula* Win *and all structures $\mathfrak{T}$ we have $\mathfrak{T} \times \Psi \models$* Win *if and only if $\mathfrak{T} \models \Psi$, that is to say* Win *interprets Ψ, then Ψ is equivalent to a formula which has the same alternation depth as* Win.

The proof depends on a product-like operation on formulas which, if Win interprets Ψ, yields a formula $\Psi \times$ Win, equivalent to Ψ, with the alternation depth of Win. The formula $\Psi \times$ Win is built with the intention that the parity game arena $(\mathfrak{T} \times \Psi) \times$ Win is the same as $\mathfrak{T} \times (\Psi \times$ Win$)$. The choice of overloading the notation $\times$ is meant to emphasize this associativity. Note however the type of the objects in these statements: $\Psi \times$ Win is a formula if Ψ and Win are both formulas while $\mathfrak{T} \times \Psi$ is a tree if $\mathfrak{T}$ is a tree.

Definition 14. *($\Psi \times$ Win)* Let Ψ be a formula of L_μ with index J and priority assignment Ω_Ψ, and let Win be a formula over propositional variables $\mathscr{P}_{\mathsf{Win}}$ with priority assignment Ω_{Win} with co-domain I with minimal element m. Define $\Psi \times$ Win, using a fresh set of fixpoint variables $W_{\phi \times X}$ where ϕ ranges over subformulas of Ψ and X ranges over the fixpoint variables of Win, as follows:

$$P \times E_m = \neg P \text{ and } P \times O_m = P \text{ where } P \text{ is a literal;}$$

$$\phi \times P = \top \text{ if for each state } s \text{ of any tree, the position } (s, \phi) \text{ satisfies } P \in \mathscr{P}_{\mathsf{Win}}.$$

In particular:

$$X \times E_{\Omega_\Psi(X)} = \phi \wedge \psi \times O_m = \phi \vee \psi \times E_m = \top;$$

$$\Diamond \phi \times E_m = \Box \phi \times O_m = \top \times O_m = \bot \times E_m = \top;$$

$$\phi \times P = \bot \text{ for } P \in \mathscr{P}_{\mathsf{Win}} \text{ otherwise.}$$

$$\Box \phi \times \Box \psi = \Diamond \phi \times \Box \psi = \Box(\phi \times \psi);$$

$$\phi \times \Box \psi = \bigwedge_{\phi' \in im(\phi)} \phi' \times \psi \text{ where } im(\phi) \text{ is the set of immediate subformulas of } \phi;$$

$$\Box \phi \times \Diamond \psi = \Diamond \phi \times \Diamond \psi = \Diamond(\phi \times \psi);$$

$$\phi \times \Diamond \psi = \bigvee_{\phi' \in im(\phi)} \phi' \times \psi \text{ where } im(\phi) \text{ is the set of immediate subformulas of } \phi;$$

If reached computing a subformula of $\mu W_{\phi \times X}.(\phi \times \psi_X)$ (or $\nu W_{\phi \times X}.(\phi \times \psi_X)$)
but not a subformula $\nu W_{\phi' \times Z}.\phi_{W_{\phi' \times Z}}$ (or $\mu W_{\phi' \times Z}.\phi_{W_{\phi' \times Z}}$) thereof with $\Omega(Z) > \Omega(X)$ then:

$$\phi \times X = W_{\phi \times X}, \text{ else:}$$

$$= \mu W_{\phi \times X}.(\phi \times \phi_X) \text{ if } X \text{ is a } \mu \text{ variable;}$$

$$= \nu W_{\phi \times X}.(\phi \times \phi_X) \text{ if } X \text{ is a } \nu \text{ variable.}$$

$$\phi \times \mu X.\psi_X = \mu W_{\phi \times X}.\phi \times \phi_X;$$

$$\phi \times \nu X.\psi_X = \nu W_{\phi \times X}.\phi \times \phi_X;$$

$$\phi \times \psi \wedge \psi' = (\phi \times \psi) \wedge (\phi \times \psi');$$

$$\phi \times \psi \vee \psi' = (\phi \times \psi) \vee (\phi \times \psi').$$

where $\Omega_{\Psi \times \text{Win}}(W_{X \times \phi}) = \Omega_{\text{Win}}(X)$. Superfluous fixpoint bindings which do not bind any free variables can then be removed. The construction terminates as computations of $\phi \times X$ will eventually result in the variable $W_{\phi \times X}$ being bound.

Proof. (Theorem 13) For the correctness of this construction, it is sufficient to compare rule by rule the parity games on $(\mathfrak{T} \times \Psi) \times \text{Win}$ and $\mathfrak{T} \times (\Psi \times \text{Win})$.

Case $P \times E_m = \neg P$: The position $((s,P),E_m)$ is winning for Even exactly when (s,P) belongs to Even: when $s \models \neg P$. The position $(s,\neg P)$ is also winning for Even exactly when $s \models \neg P$.

Case $P \times O_m = P$: The position $((s,P),O_m)$ is winning for Even exactly when (s,P) belongs to Odd: when $s \models P$. The position (s,P) is also winning for Even exactly when $s \models P$.

Case $\phi \times P$: For $\phi \notin \mathscr{P}_{\text{Win}}$, whether (s,ϕ) satisfies P can only depend on ϕ and not s. If for all states s of any tree, (s,ϕ) satisfies P, then $((s,\phi),P)$ is winning for Even, as is $(s,\top)$. Otherwise $((s,\phi),P)$ is winning for Odd, as is $(s,\bot)$.

Case $\square\phi \times \square\psi = \square(\phi \times \psi)$: $((s,\square\phi),\square\psi)$ and $((s,\Diamond\phi),\square\psi)$ are both positions belonging to Odd with successors $((s',\phi),\psi)$ for s' a successor of s. On the other hand $(s,\square(\phi,\psi))$ has successors $(s',\phi \times \psi)$ for s' a successor of s. The case for $\square\phi \times \Diamond\phi = \Diamond\phi \times \Diamond\psi = \Diamond(\phi \times \psi)$ is similar.

Case $\phi \times \square\psi = \bigwedge_{\phi' \in im(\phi)} \phi' \times \psi$: The position $((s,\phi),\square\psi)$ belongs to Odd and has successors $((s,\phi'),\psi)$ where ϕ' is an immediate subformula of ϕ. On the other hand, $(s,\bigwedge_{\phi' \in im(\phi)} \phi' \times \psi)$ also belongs to Odd and has successors $(s,\phi' \times \psi)$ where ϕ' is an immediate subformula of ϕ. The case for $\phi \times \Diamond\psi = \bigvee_{\phi' \in im(\phi)} \phi' \times \psi$ is similar.

Case for fixpoints: $((s,\phi),X)$ has a unique successor $((s,\phi),\phi_X))$ where ϕ_X is the formula binding X and is of the priority of X. On the other hand $(s,W_{\phi \times X})$ and $(s,\gamma W_{\phi \times X})$ for $\gamma \in \{\mu,\nu\}$ have a unique successor $(s,\phi \times \phi_X)$, and is of the priority of $W_{\phi \times X}$ which is the same as the priority of X.

Case for $\phi \times \psi \wedge \psi'$: The position $((s,\phi),\psi \wedge \psi')$ belongs to Odd and has successors $((s,\phi),\psi)$ and $((s,\phi),\psi')$. The position $(s,(\phi \times \psi) \wedge (\phi \times \psi')$ also belongs to Odd and has successors $(s,\phi \times \psi)$ and $(s,\phi \times \psi')$. The case for $(\phi \times \psi \vee \psi')$ is similar.

The formula $\Psi \times \text{Win}$ inherits its priority assignment from Win, and the condition on the introduction of fixpoint variables guarantees that the priority assignment is order-preserving. Then we have that $\mathfrak{T} \models \Psi \Leftrightarrow \mathfrak{T} \times \Psi \models \text{Win} \Leftrightarrow \mathfrak{T} \models \Psi \times \text{Win}$: the formula $\Psi \times \text{Win}$, of index I, is equivalent to Ψ. $\square$

This concludes the argument that if Ψ can be interpreted by a formula Φ with index I, then Ψ is semantically of index I. Since $\Psi \times \Phi$ is an effective transformation that turns Ψ into a formula which is syntactically of index I, a formula can be simplified whenever a suitable interpreting formula is found. The following example shows how this can be used to argue that a formula is semantically simple without recourse to the formula's semantics.

Example 15. Consider this formula

$$\Psi = \mu X.\nu Y.\mu Z.(A \wedge \Diamond Y) \vee (B \wedge \Diamond(Z \wedge \phi)) \vee (C \wedge \square X)$$

where $\phi \in \Pi_2^\mu$ and does not have variables X,Y,Z. We use Theorem 13 to argue that this formula, with seemingly opaque semantics, is semantically Π_2^μ.

The first thing to note is that a winning strategy σ for Even in a model-checking game of Ψ has the following property: a play that agrees with σ eventually reaches a point from which no play that agrees with σ sees a position $s \times X$. We can describe the winning regions of such games with the following

formula:

$$\text{Interpretor} = \mu X. \bigvee_{i \in \{1,2,3\}} (E_i \wedge \Diamond X \vee O_i \wedge \Box X) \vee \nu X_2. \mu X_1. \bigvee_{i \in \{2,1\}} (E_i \wedge \Diamond X_i \vee O_i \wedge \Box X_i)$$

The intuition here is that the first part of the Interpretor formula describes Even's strategy until she can guarantee that a position of priority 3 will not be seen again; the second part of the formula is just the usual $\text{Parity}_{2,1}$ formula. A strategy in the model-checking game for this formula translates directly into the model-checking game of $\text{Parity}_{3,2,1}$. On the other hand, if Even has a winning strategy in a model-checking game $\mathfrak{T} \times \text{Parity}_{3,2,1}$ with the above property, then she also has a winning strategy in $\mathfrak{T} \times \text{Interpretor}$: once a play reaches the point from where her strategy no longer sees priority 3, she moves to the second part of the formula.

The Π_2^μ formula Interpretor therefore interprets Ψ. From Theorem 13, Ψ is semantically Π_2^μ. The mechanics of the transformation can come in the way of clarity, but for the curious reader the tidied-up version of the transformed formula follows. Note how the structure of Interpretor informs the structure of this new formula.

$$\mu X.(A \wedge \Diamond X) \vee (B \wedge \Diamond(X \wedge \phi)) \vee (C \wedge \Box X) \vee \nu Y. \mu Z.(A \wedge \Diamond Y) \vee (B \wedge \Diamond(Z \wedge \phi)) \vee (C \wedge \Box \bot)$$

4 Interpretation theorems

So far, we have seen that if a formula Ψ is interpreted by a formula of index I, then Ψ is itself semantically of index I. This extension of the L_μ-parity game relationship is natural and not too surprising. This section considers the converse: when is it the case that a formula semantically of index I can be interpreted by a formula of syntactic index I? The formula in Example 15 shows that this may happen for syntactic reasons. Here we discuss whether it holds in general. This conjecture can be studied with respect to any index I:

Conjecture 16. *If Ψ is semantically of index I, then there is a formula Φ of syntactic index I such that for all structures $\mathfrak{T}$, $\mathfrak{T} \times \Psi \models \Phi$ if and only if $\mathfrak{T} \models \Psi$.*

If this conjecture is found to be true in general, then the descriptive complexity of the model-checking games of a L_μ formula Ψ is exactly the complexity of the simplest formula equivalent to Ψ.

Say that the class of formulas of index I admits an *interpretation theorem* if the above conjecture holds for I. This section considers this conjecture for ML, Π_1^μ and Σ_2^μ. It shows that these classes admit interpretation theorems:

- *ML* has an interpretation theorem;
- Π_1^μ has an interpretation theorem for disjunctive L_μ;
- Σ_2^μ has an interpretation theorem for disjunctive L_μ;

It then considers what can be said in the general case, and shows that disjunctive L_μ alternation classes all have an interpretation theorem for input in co-disjunctive form.

In each case, the proof strategy relies on defining a characteristic game for each index I, with descriptive complexity I, and characterising formulas with semantic index I as those whose model-checking parity games corresponds to the characteristic game.

4.1   Interpretation theorem for *ML*

This section shows that conjecture 16 holds for semantically modal formulas:

Theorem 17. (Interpretation Theorem for *ML*)

Let Ψ be a semantically modal formula, then there exists (effectively) a modal formula Φ such that Φ interprets Ψ.

The proof relies on the idea that modal formulas are local: a modal formula holds in a tree if and only if it holds in its initial fragment of depth m, the modal depth of the formula given by the maximal nesting depth of modalities. This is also true for formulas equivalent to modal formulas of modal depth m. The characteristic game for semantically modal formulas simulates truncating the tree at depth m.

Lemma 18. *If $\Psi \in L_\mu$ is equivalent to a modal formula Φ of modal depth m, then for all $\mathcal{T}$, it is the case that $\mathcal{T} \models \Psi$ if and only if $\mathcal{T}|^m \models \Psi$ where $\mathcal{T}|^m$ is the initial fragment of $\mathcal{T}$, up to depth m.*

Proof. Φ only has m nested modalities, so the model-checking game of Φ can not reach positions further than m steps from the root. If two trees agree up to depth m, they therefore agree on Φ. Since Ψ is equivalent to Φ, they also have to agree on Ψ. □

We now define a variation of a parity game which will allow us to simulate playing the model-checking game on a truncated structure $\mathcal{T}|^n$ rather than on $\mathcal{T}$.

Definition 19. A n-bounded parity game is a parity game, augmented with a counter, of which some positions $M \in V$ are marked. M must be such that every infinite path goes through M. Then, a play of this game is a play of the parity game, except that the counter, initially at n, is decremented whenever a position in M is seen. If the play reaches M at counter value 0, then the owner of the position loses the game.

If the longest path between two positions in M is no longer than p, the winning regions of these games are described by the modal formula $\text{Bounded}_{p,m}$ defined inductively as follows:

$$
\begin{aligned}
\text{Bounded}_{0,b} =& \perp \\
\text{Bounded}_{a,0} =& (E_i \wedge \neg M \wedge \Diamond \text{Bounded}_{a-1,0}) \vee \\
& (E_i \wedge M \wedge \perp) \vee \\
& (O_i \wedge \neg M \wedge \Box \text{Bounded}_{a-1,0}) \vee \\
& (O_i \wedge M \wedge \top) \\
\text{Bounded}_{a,b} =& (E_i \wedge \neg M \wedge \Diamond \text{Bounded}_{a-1,b}) \vee \\
& (E_i \wedge M \wedge \Diamond \text{Bounded}_{p,b-1}) \vee \\
& (O_i \wedge \neg M \wedge \Box \text{Bounded}_{a-1,b}) \vee \\
& (O_i \wedge M \wedge \Box \text{Bounded}_{p,b-1})
\end{aligned}
$$

Proof. (of Theorem 17) Let Ψ be equivalent to a modal formula of modal depth m. Let p be the longest path in the parse-tree of Ψ (where ϕ_X is taken to be the child of X) without modalities. Since Ψ is taken to be guarded, p is finite. Then the model-checking games of Ψ are m-bounded parity games where M is the set of positions $s \times \phi$ where ϕ is a modal formula: this game simulates the parity game $\mathcal{T}|^m \times \Psi$. To see this, it suffices to observe that the two games are identical until m modalities are seen, after which in the m-bounded game positions $\Diamond \phi$ are winning for Odd, simulating reaching a position without successors at Even's turn, while $\Box \phi$ is winning for Even, simulating reaching a position without successors at Odd's turn. Then, noting that the longest path between two positions in M in any model-checking parity game of Ψ is of length at most p, $\text{Bounded}_{p,m}$ interprets Ψ. □

Hence the modal fragment of L_μ has an effective interpretation theorem: any semantically modal formula can be interpreted by a syntactically modal formula. This means that all semantically modal formulas, no matter how high their syntactic index, generate a class of model-checking games with modal descriptive complexity. The interpreting formula depends on the size of Ψ as well as its semantic modal depth, but is computable from the formula [14].

4.2 Interpretation theorem for Π_1^μ

The conjecture 16 also holds for *disjunctive formulas* that are semantically Π_1^μ.

Theorem 20. (Interpretation theorem for Π_1^μ)

Let Ψ be a disjunctive formula which is semantically in Π_1^μ. Then there exists (effectively) a Π_1^μ formula Φ such that Φ interprets Ψ.

The decision procedure for Π_1^μ in [14] shows that in disjunctive semantically Π_1^μ formulas, every satisfiable μ-subformula can be replaced with the same subformula bound by ν, while unsatisfiable ones can be replaced by $\bot$. This can be translated into the formula describing the winning regions of the model-checking games of the formula, simply by substituting the least fixpoint in the parity game formula corresponding to a subformula $\mu X.\phi$ with either $\bot$ or a greatest fixpoint, depending on whether the subformula is unsatisfiable.

For example, the appropriate formula describing the winning regions of a disjunctive formula in which all μ-bound subformulas are satisfiable would simply be:

$$\nu Y.(E_e \wedge \Diamond Y) \vee (E_o \wedge \Diamond Y) \vee (O_e \wedge \Box Y) \vee (O_o \wedge \Box Y)$$

Where E_e (E_o) is the disjunction of E_i for even (odd) i; O_e (O_o) is the disjunction of O_i for even (odd) i. For μ-bound subformulas which are unsatisfiable, it is enough to turn the clause $E_i \wedge \Diamond X_i$ corresponding to the μ-variable X_i in question into $E_i \wedge \bot$.

Remark 21. Note that the encoding of the model-checking game arena can incorporate data about the provenance of a node, rather than just its priority, for instance by using additional propositional variables E_X to indicate positions which stem from the fixpoint variable X. To describe the winning regions of games encoded in this manner, we use $E_X \wedge \Diamond Y_{\Omega(X)}$ instead of $E_i \wedge \Diamond Y_i$.

Recall that for modal formulas, the interpreting formula depends only on the modal rank of the formula and the length of the longest path without fixpoints in the formula. Since both of these can be bound in relation to the size of the formula, all semantically modal L_μ formulas of the same size can be interpreted by the same formula. This cannot be said for semantically Π_1^μ formulas: the interpreting formulas depend on which fixpoint subformulas are unsatisfiable and which are interchangeable with ν. It remains an open question whether a *uniform* interpreting formula could be devised for all semantically Π_1^μ formulas in disjunctive form. It is also open whether the restriction to disjunctive formulas can be lifted.

4.3 Interpretation theorem for Σ_2^μ

The previous sections establish effective interpretation theorems for Π_1^μ, restricted to disjunctive formulas, and for the modal fragment of L_μ. This section shifts the focus onto an alternation class not known to be decidable: Σ_2^μ.

Theorem 22. (Interpretation Theorem for Σ_2^μ)

If a disjunctive formula Ψ is semantically in Σ_2^μ, then it is interpreted by a Σ_2^μ formula.

The proof argues that first part of the decidability proof of Σ_2^μ for Π_2^μ presented in [15] translates into an interpretation theorem for Σ_2^μ. Similarly to the modal case, a variation on parity games yields the model-checking games for semantically Σ_2^μ formulas. Their winning regions are described by a Σ_2^μ formula. Here we briefly recall the *n*-challenge game, and the result that its winning regions are described by a Σ_2^μ formula.

Definition 23. Fix a formula Ψ in disjunctive form, of index $\{q, ..., 0\}$. Let $I = \{q, ..., 0\}$ if q is even and $\{q+1, q, ..., 0\}$ otherwise. Write I_e for the even priorities in I.

The *n*-challenge game consists of a normal parity game augmented with a set of challenges, one for each even priority i. A challenge can either be *open* or *met* and has a counter c_i attached to it. Each counter is initialised to n, and decremented when the corresponding challenge is opened. The Odd player can at any point open challenges of which the counter is non-zero, but he must do so in decreasing order: an i-challenge can only be opened if every j-challenge for $j > i$ is opened. When a play encounters the priority j while the j-challenge is open, the challenge is said to be met. All i-challenges for $i < j$ are then *reset*. This means that the counters c_i are set back to n and marked *met*.

A play of this game is a play in a parity game, augmented with the challenge and counter configuration at each step. A play with dominant priority d is winning for Even if either d is even or if every opened $d + 1$ challenge is eventually met or reset.

Proof. (Theorem 22) It is known from [15] that disjunctive Ψ is equivalent to a Σ_2^μ formula if an only if there is an n such that for all trees $\mathfrak{T}$, Even wins the *n*-challenge game on $\mathfrak{T} \times \Psi$ if and only if she wins the parity game. The winning regions of the *n*-challenge games on parity game arenas of index I are described by a Σ_2^μ formula $\mathsf{Challenge}_I^n$. Hence, if Ψ is a disjunctive formula semantically in Σ_2^μ, then there is an n such that for all $\mathfrak{T}$, $\mathfrak{T} \models \Psi$ if and only if $\mathfrak{T} \times \Psi \models \mathsf{Challenge}_I^n$. That is to say, $\mathsf{Challenge}_I^n$ interprets Ψ.   □

This concludes the argument that for formulas up to semantic index $\{1, 0\}$, the descriptive complexity of the model-checking games coincides exactly with the semantic complexity of the formula. The dependence of these theorems on disjunctive form highlights the importance of this normal form for the index problem.

4.4   General interpretation theorem for disjunctive L_μ

This section generalises the argument used for Σ_2^μ. With some concessions over the structure of target formulas, that is to say only considering disjunctive alternation classes, this yields a general interpretation theorem for co-disjunctive formulas.

We generalise the *n*-challenge game by considering a set of challenges (one per target priority) per input priority rather than a single challenge. Fix J to be the *input* index, *i.e.* the index of the input formula, while I is the *target* index, *i.e.* the index of the target alternation class. Then, for the pair J, I, we defines a parameterised challenge game such that an input formula Ψ in co-disjunctive form of index J is interpreted by the disjunctive formula (of index I) describing the winning regions of these games if and only if Ψ is equivalent to a disjunctive formula of index I.

As usual, this game is played on a parity game arena with priorities from J. Unlike for Σ_2^μ, the challenging player is now Even, to allow the challenger to use the fact that in disjunctive form she has strategies which see one play per branch. The input formula on the other hand is in co-disjunctive form, to yield such strategies for Odd.

Definition 24. *(Generalised challenge games)* A *challenge configuration* $(\bar{a}, \bar{c})$ consists of $|I_o| \times |J_o|$ challenges and counters, where I_o and J_o are the odd priorities in I and J respectively. Write $a_{i,j} = met$ or *open* for $i \in I$ and $j \in J$ to indicate whether the *i*-level challenge on j is open. Each challenge $a_{i,j}$ is attached to a positive integer counter value $c_{i,j}$, bounded by n.

Given a configuration $(\bar{a}, \bar{c})$, the least j for which $a_{i,j}$ is open (for any i) is the *priority* of the challenge configuration, and the highest level i at which $a_{i,j}$ is open is its *level*. A valid challenge configuration respects the following constraint: if $a_{i,j} = open$ then $a_{i,k} = open$ for all $k > j$. That is to say, challenges are opened in decreasing order.

The *game configuration* consists of a position in the parity game and a valid challenge configuration. The progress of the game can be divided into two rounds: in the first round Even opens or resets challenges, while the second round is a step in the parity game. In the first round her possible actions are:

- To *k-reset* for any odd $k \in I_o$, setting $a_{i,j} := met$ and $c_{i,j} := n$ for all $i \leq k$ and all $j \in J$;

- To open at any level i, challenges up to any p, as long as the counters allow it: for all $j \geq p$ such that $a_{i,j} = met$, set $a_{i,j} := open$ and $c_{i,j} := c_{i,j} - 1$ if $c_{i,j} > 0$.

Then, in the second round, the player whose turn it is in the parity game picks a successor position. If the underlying parity game ends in a terminal state, then the winner of the underlying parity game immediately wins the challenge game, too. The challenge configuration is updated according to the priority p of this new position:

- $a_{i,j} := met$ for all $j \leq p$, all $i \in I_o$;

- $c_{i,j} := n$ for all $j < p$, all $i \in I_o$.

If $c_{i,p} = 0$ for some i, then the game ends immediately with a win for Odd.

A play is a potentially infinite sequence of game configurations: an underlying parity game play augmented with challenge configurations. The initial challenge-configuration is $a_{i,j} = met$ and $c_{i,j} = n$ for all i, j. The dominant priority of the parity game play is also the dominant priority of the challenge game play. An infinite play with dominant priority d is winning for Odd if: d is odd, or all $a_{i,d+1}$ challenges are in the *met* state infinitely often.

Lemma 25. *The winning regions of a generalised n-challenge game for J, I are described by a disjunctive L_μ formula with index I.*

The construction of the formula $\mathsf{GChallenge}_{J,I}^n$ is similar in spirit to the one described in [15], although slightly more involved, given the additional input priorities to account for.

Lemma 26. *A co-disjunctive L_μ formula Ψ of index J is interpreted by the I-index formula $\mathsf{GChallenge}_{J,I}^n$ if it is equivalent to a disjunctive formula of index I.*

Again, the proof follows the same structure as the proof of the Σ_2^μ case. In brief, it assumes that Ψ of index J is equivalent to some disjunctive Φ of index I and that for all m, there is a structure $\mathfrak{T}$, such that Even wins the parity game $\mathfrak{T} \times \Psi$ but Odd wins the generalised m-challenge game on $\mathfrak{T} \times \Psi$. It then takes a sufficiently large m, and uses a winning strategy σ for Even in $\mathfrak{T} \times \Phi$ to define a challenging strategy γ for her in the generalised m-challenge game on $\mathfrak{T} \times \Psi$. In the Σ_2^μ case the challenges were issues when the higher priority had been seen on all plays since the last challenge. This time, different even priorities cause Even to issue challenges at different levels while different odd priorities cause Even to reset at different levels. Then Odd's winning strategy τ in the m-challenge game on $\mathfrak{T} \times \Psi$ is used to add back edges to $\mathfrak{T}$, turning it into a new structure $\mathfrak{T}'$ which preserves Even's winning strategy σ in

$\mathfrak{T}' \times \Phi$ while turning τ_γ into a winning strategy in $\mathfrak{T}' \times \Psi$. This contradicts the equivalence of Φ and Ψ. The main technical difference is that we no longer invoke König's lemma, and instead uses the fact that target formulas are in disjunctive form to find a suitable challenging-strategy for Even.

This gives us a general interpretation theorem for all disjunctive Σ^μ alternation classes, for co-disjunctive input formulas: the complexity of an equivalent disjunctive formula is an upper bound on the descriptive complexity of the model-checking games generated by the co-disjunctive form of Ψ.

Comparison with automata-theoretic results

The readers familiar with the automata-theoretic efforts to tackle the Rabin–Mostowski index problem may find some parallels between this last interpretation theorem and the reduction of the decidability of the Rabin–Mostowski hierarchy of non-deterministic automata to a boundedness question [5]. In both cases, the core construction gives a way of mapping input priorities to the target priorities. In [5], it is a question of guessing mappings between input and target priorities a bounded number of times. Here it is achieved through the parameterised challenges. The proof in [5] relies on *guidable* automata – a concept for which the L_μ-theoretic conterpart is conspicuously absent.

I believe this comparison to be a cause for cautious optimism when looking for further interpretation theorems: indeed, the automata-theoretic result is not burdened by the same constraints to disjunctive and co-disjunctive form as Theorem 26; perhaps these constraints can be lifted in the L_μ framework and the mapping construction in [5] could yield a family of interpreting formulas, for which the interpretation theorem does not place restrictions on the syntax of the input formula.

The transferability of the techniques is however not trivial. Disjunctive L_μ was designed to be the L_μ counter-part of non-deterministic automata. However, although the model-checking problems for non-deterministic automata and disjunctive L_μ reduce to each other by encoding unranked trees as ranked trees, this reduction is not necessarily index-preserving. Furthermore, only a small fragment of L_μ, namely disjunctive L_μ *restricted to existential modalities* seems to allow a construction akin to guidable automata. Although on this restricted fragment the index problem seems similar to the non-deterministic index problem, it seems unlikely that it would suffice to decide the index problem for disjunctive L_μ. Overall, how the disjunctive L_μ index problem and the non-deterministic parity automata index problem relate to each other is not yet settled.

5   Discussion

We have shown that for formulas equivalent to formulas of index $I \in \{\{\}, \{0\}, \{0,1\}\}$, the descriptive complexity of their model-checking games coincides exactly with I. Beyond these alternation classes, the descriptive complexity of the model-checking games of Ψ is an upper bound on the semantic complexity of Ψ. Furthermore, for a co-disjunctive formula equivalent to a disjunctive formula of index I, the descriptive complexity of its model-checking games is bounded by I.

This links the index-problem to the parity game–L_μ relationship, which is shown to be even more robust than previously accounted for. As a result, the index-problem can be approached by studying the descriptive complexity of model-checking games. On one hand, for practical purposes formulas can be simplified by looking at the model-checking games they generate. On the other hand, interpretation theorems are a stepping stone on the pursuit of further decision procedures.

Besides finding a truly general interpretation theorem which would show that the descriptive complexity of model-checking games coincides exactly with the semantic complexity of formulas, I leave the reader with a couple of related open questions.

For low alternation classes ML, Π_1^μ and Σ_2^μ, some of the existing decision procedures can be recast as interpretation theorems. For both Π_1^μ and Σ_2^μ, the interpretation theorem only applies for disjunctive input formula. This once again highlights the importance of disjunctive form for the index problem: the descriptive complexity of the model-checking games of disjunctive formulas appears to be lower than for general L_μ formulas. Whether this is truly the case, remains an open question.

The generality of the last interpretation theorem comes at a cost. In particular, it is not constructive, in the sense that if a co-disjunctive formula is indeed equivalent to a disjunctive formula of index I, even if we find the interpreting disjunctive formula of index I, this only allows us to construct a formula of index I, which may not itself be disjunctive. I conjecture that a similar theorem could be shown for disjunctive input formulas. As discussed in the previous section, some automata-theoretic techniques [5] could perhaps be used to address some of the limits of the current techniques.

References

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

[2] Julian C Bradfield (1999): *Fixpoint alternation: Arithmetic, transition systems, and the binary tree*. RAIRO-Theoretical Informatics and Applications 33(4-5), pp. 341–356, doi:10.1051/ita:1999122.

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

[4] Thomas Colcombet, Denis Kuperberg, Christof Löding & Michael Vanden Boom (2013): *Deciding the weak definability of Büchi definable tree languages*. In: LIPIcs-Leibniz International Proceedings in Informatics, 23, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, doi:10.4230/LIPIcs.CSL.2013.215.

[5] Thomas Colcombet & Christof Löding (2008): *The non-deterministic Mostowski hierarchy and distance-parity automata*. In: International Colloquium on Automata, Languages, and Programming, Springer, pp. 398–409, doi:10.1007/978-3-540-70583-3_33.

[6] Anuj Dawar & Erich Grädel (2008): *The descriptive complexity of parity games*. In: International Workshop on Computer Science Logic, Springer, pp. 354–368, doi:10.1007/978-3-540-87531-4_26.

[7] E Allen Emerson & Charanjit 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.

[8] David Janin & Igor Walukiewicz (1995): *Automata for the modal μ-calculus and related results*. doi:10.1007/3-540-60246-1_160.

[9] Dexter Kozen (1983): *Results on the propositional μ-calculus*. Theoretical Computer Science 27(3), pp. 333–354, doi:10.1016/0304-3975(82)90125-6. Special Issue Ninth International Colloquium on Automata, Languages and Programming (ICALP) Aarhus, Summer 1982.

[10] Dexter Kozen (1988): *A finite model theorem for the propositional μ-calculus*. Studia Logica 47(3), pp. 233–241, doi:10.1007/BF00370554.

[11] Orna Kupferman, Moshe Y Vardi & Pierre Wolper (2000): *An automata-theoretic approach to branching-time model checking*. Journal of the ACM (JACM) 47(2), pp. 312–360, doi:10.1145/333979.333987.

[12] Ralf Küsters & Thomas Wilke (2002): *Deciding the First Level of the μ-calculus Alternation Hierarchy* (0209). doi:10.1007/3-540-36206-1_22.

[13] Karoliina Lehtinen (2015): *Disjunctive form and the modal μ alternation hierarchy*. FICS15, The 10th International Workshop on Fixed Points in Computer Science, EPTCS 191, p. 117, doi:`10.4204/EPTCS.191.11`.

[14] Karoliina Lehtinen & Sandra Quickert (2015): *Deciding the first levels of the modal μ alternation hierarchy by formula construction*. In: *LIPIcs-Leibniz International Proceedings in Informatics*, 41, Schloss Dagstuhl-Leibniz-Zentrum fuer Informatik, doi:`10.4230/LIPIcs.CSL.2015.457`.

[15] Karoliina Lehtinen & Sandra Quickert (2017): Σ_2^μ *is decidable for* Π_2^μ. In: *Unveiling Dynamics and Complexity - 13th Conference on Computability in Europe, CiE 2017, Turku, Finland, June 12-16, 2017, Proceedings*, pp. 292–303, doi:`10.1007/978-3-319-58741-7_28`.

[16] Giacomo Lenzi (1996): *A hierarchy theorem for the μ-calculus*. In Friedhelm Meyer & Burkhard Monien, editors: *Automata, Languages and Programming*, Lecture Notes in Computer Science 1099, Springer Berlin Heidelberg, pp. 87–97, doi:`10.1007/3-540-61440-0_119`.

[17] Donald A Martin (1975): *Borel determinacy*. Annals of Mathematics, pp. 363–371, doi:`10.2307/1971035`.

[18] Radu Mateescu (2002): *Local Model-Checking of Modal Mu-Calculus on Acyclic Labeled Transition Systems*. In Joost-Pieter Katoen & Perdita Stevens, editors: *Tools and Algorithms for the Construction and Analysis of Systems*, Lecture Notes in Computer Science 2280, Springer Berlin Heidelberg, pp. 281–295, doi:`10.1007/3-540-46002-0_20`.

[19] Damian Niwiński (1986): *On fixed-point clones*, pp. 464–473. Springer Berlin Heidelberg, Berlin, Heidelberg, doi:`10.1007/3-540-16761-7_96`.

[20] Martin Otto (1999): *Eliminating Recursion in the μ-Calculus*, pp. 531–540. Springer Berlin Heidelberg, Berlin, Heidelberg, doi:`10.1007/3-540-49116-3_50`.

[21] Michal Skrzypczak & Igor Walukiewicz (2016): *Deciding the Topological Complexity of Büchi Languages*. In Yuval Rabani Ioannis Chatzigiannakis, Michael Mitzenmacher & Davide Sangiorgi, editors: *43rd International Colloquium on Automata, Languages, and Programming (ICALP 2016)*, Leibniz International Proceedings in Informatics (LIPIcs) 55, Schloss Dagstuhl–Leibniz-Zentrum für Informatik, Dagstuhl, Germany, pp. 99:1–99:13, doi:`10.4230/LIPIcs.ICALP.2016.99`.

[22] Igor Walukiewicz (2002): *Monadic second-order logic on tree-like structures*. Theoretical Computer Science 275(1), pp. 311–346, doi:`10.1016/S0304-3975(01)00185-2`.