

Optimal Bounds in Parametric LTL Games

Martin Zimmermann¹

*Institute of Informatics
University of Warsaw
Warsaw, Poland*

Abstract

Parameterized linear temporal logics are extensions of Linear Temporal Logic (LTL) by temporal operators equipped with variables that bound their scope. In model-checking, such specifications were introduced as “PLTL” by Alur et al. and as “PROMPT-LTL” by Kupferman et al. We show how to determine in doubly-exponential time, whether a player wins a game with PLTL winning condition with respect to some, infinitely many, or all variable valuations. Hence, these problems are not harder than solving LTL games. Furthermore, we present an algorithm with triply-exponential running time to determine optimal variable valuations that allow a player to win a game. Finally, we give doubly-exponential upper and lower bounds on the values of such optimal variable valuations.

Keywords: Infinite Games, Parametric Linear Temporal Logic, PROMPT-LTL, Optimal Winning Strategies

2000 MSC: 68Q60, 03B44

1. Introduction

A crucial aspect of automated verification and synthesis is the choice of a specification formalism; a decision which is subject to several conflicting objectives. On the one hand, the formalism should be expressive enough to specify desirable properties of reactive systems, but at the same time simple

Email address: zimmermann@mimuw.edu.pl (Martin Zimmermann)

¹This work was prepared while the author was affiliated with RWTH Aachen University and was supported by the project *Games for Analysis and Synthesis of Interactive Computational Systems (GASICS)* of the *European Science Foundation*.

enough to be employed by practitioners without formal training in automata theory or logics. Furthermore, the formalism should have *nice* algorithmic properties such as a feasible model-checking problem. In practice, Linear Temporal Logic (LTL) has emerged as a good compromise: it is expressively equivalent to first-order logic (with order-relation) over words [1], its model-checking problem is **PSPACE**-complete [2], and it has a compact, variable-free syntax and intuitive semantics: for example, the specification “every request q is answered by a response p ” is expressed by the formula $\mathbf{G}(q \rightarrow \mathbf{F} p)$.

However, LTL lacks capabilities to express timing constraints, e.g., the request-response condition is satisfied even if the response time doubles with each request. Similarly, when synthesizing a controller for a request-response specification, we prefer an implementation that answers every request as soon as possible, but there is no guarantee that such an optimal controller is computed when solving a game with this winning condition.

The simplest way to enrich LTL with timing constraints is to add the operator $\mathbf{F}_{\leq k}$, where $k \in \mathbb{N}$ is an arbitrary, but fixed constant, with the expected semantics: the formula $\mathbf{F}_{\leq k} \varphi$ is satisfied, if φ holds at least once within the next k steps. Koymans [3] and Alur and Henzinger [4] investigated generalizations of this approach in the form of logics with temporal operators bounded by constant intervals of natural numbers. This allows to infer some quantitative information about a system: the formula $\mathbf{G}(q \rightarrow \mathbf{F}_{\leq k} p)$ is satisfied if every request is answered within k steps. But finding the right bound k is not practicable: it is generally not known beforehand and depends on the granularity of the model of the system. On the other hand, adding $\mathbf{F}_{\leq k}$ does not increase the expressiveness of LTL, as it can be expressed by a disjunction of nested next-operators.

To overcome these shortcomings, several parameterized temporal logics [5, 6, 7] were introduced for the verification of closed systems: here, constant bounds on the temporal operators are replaced by parametric bounds. In this formalism, we can ask whether there *exists* a bound on the response time, as opposed to asking whether some fixed k is a bound. Furthermore, we can ask for optimal bounds.

We are mainly concerned with Parametric Linear Temporal Logic (PLTL) introduced by Alur et al. [5], which adds the operators $\mathbf{F}_{\leq x}$ and $\mathbf{G}_{\leq y}$ to LTL. In PLTL, the request-response specification is expressed by $\mathbf{G}(q \rightarrow \mathbf{F}_{\leq x} p)$, stating that every request is answered within the next x steps, where x is a variable. Hence, satisfaction of a formula is defined with respect to a variable valuation α mapping variables to natural numbers: $\mathbf{F}_{\leq x} \varphi$ holds, if

φ is satisfied at least once within the next $\alpha(x)$ steps, while $\mathbf{G}_{\leq y} \varphi$ holds, if φ is satisfied for the next $\alpha(y)$ steps.

The model-checking problem for a parameterized temporal logic is typically no harder than the model-checking problem for the unparameterized fragment, e.g., Alur et al. showed that deciding whether a transition system satisfies a PLTL formula with respect to some, infinitely many, or all variable valuations is **PSPACE**-complete [5], as is LTL model-checking [2]. Also, for two interesting fragments of PLTL and several notions of optimality, they showed that optimal variable valuations for which a formula is satisfied by a given transition system can be determined in polynomial space as well.

In this work, we consider infinite games with winning conditions in PLTL and lift the results on model-checking parameterized specifications to synthesis from parameterized specifications. We show that determining whether a player wins a PLTL game with respect to some, infinitely many, or all variable valuations is **2EXPTIME**-complete, as is determining the winner of an LTL game [8]. Again, we observe the same phenomenon as in model-checking: the addition of parameterized operators does not increase the computational complexity of the decision problems. Afterwards, we give an algorithm with triply-exponential running time to compute optimal variable valuations (and winning strategies realizing them) for the two fragments of PLTL already considered by Alur et al. for model checking. We complement this with doubly-exponential upper and lower bounds on values of optimal variable valuations for games in these fragments.

This work is an extended and corrected version of [9]: in the conference version, we claimed the optimization problems to be solvable in doubly-exponential time. However, the proof of Lemma 16.1 of [9] contains a mistake. In the present work, we fix this error, but this increases the running time of the algorithm to triply-exponential. The exact complexity of the optimization problems remains open.

2. Definitions

An arena $\mathcal{A} = (V, V_0, V_1, E)$ consists of a finite directed graph (V, E) and a partition $\{V_0, V_1\}$ of V denoting the positions of Player 0 (drawn as ellipses) and Player 1 (drawn as rectangles). The size $|\mathcal{A}|$ of \mathcal{A} is $|V|$. It is assumed that every vertex has at least one outgoing edge. A play is an infinite path $\rho = \rho_0\rho_1\rho_2\cdots$ through \mathcal{A} . A strategy for Player i is a mapping $\sigma: V^*V_i \rightarrow V$ such that $(\rho_n, \sigma(\rho_0\cdots\rho_n)) \in E$ for all $\rho_0\cdots\rho_n \in V^*V_i$. A

play ρ is consistent with σ if $\rho_{n+1} = \sigma(\rho_0 \cdots \rho_n)$ for all n with $\rho_n \in V_i$. A parity game $\mathcal{G} = (\mathcal{A}, v_0, \Omega)$ consists of an arena \mathcal{A} , an initial vertex v_0 and a coloring function $\Omega: V \rightarrow \mathbb{N}$. Player 0 wins a play $\rho_0\rho_1\rho_2 \cdots$ if the maximal color occurring in $\Omega(\rho_0)\Omega(\rho_1)\Omega(\rho_2) \cdots$ infinitely often is even. The number of colors of \mathcal{G} is $|\Omega(V)|$. A strategy σ for Player i is winning for her, if every play that starts in v_0 and is consistent with σ is won by her. Then, we say Player i wins \mathcal{G} .

A memory structure $\mathcal{M} = (M, m_0, \text{upd})$ for an arena (V, V_0, V_1, E) consists of a finite set M of memory states, an initial memory state $m_0 \in M$, and an update function $\text{upd}: M \times V \rightarrow M$, which we extend to $\text{upd}^*: V^+ \rightarrow M$ by $\text{upd}^*(\rho_0) = m_0$ and $\text{upd}^*(\rho_0 \cdots \rho_n \rho_{n+1}) = \text{upd}(\text{upd}^*(\rho_0 \cdots \rho_n), \rho_{n+1})$. A next-move function for Player i is a function $\text{nxt}: V_i \times M \rightarrow V$ that satisfies $(v, \text{nxt}(v, m)) \in E$ for all $v \in V_i$ and all $m \in M$. It induces a strategy σ with memory \mathcal{M} via $\sigma(\rho_0 \cdots \rho_n) = \text{nxt}(\rho_n, \text{upd}^*(\rho_0 \cdots \rho_n))$. A strategy is called finite-state if it can be implemented with a memory structure, and positional if it can be implemented with a single memory state. The size of \mathcal{M} (and, slightly abusive, σ) is $|M|$.

An arena $\mathcal{A} = (V, V_0, V_1, E)$ and a memory structure $\mathcal{M} = (M, m_0, \text{upd})$ for \mathcal{A} induce the expanded arena $\mathcal{A} \times \mathcal{M} = (V \times M, V_0 \times M, V_1 \times M, E')$ where $((s, m), (s', m')) \in E'$ if and only if $(s, s') \in E$ and $\text{upd}(m, s') = m'$. A game \mathcal{G} with arena \mathcal{A} and initial vertex v_0 is reducible to \mathcal{G}' with arena \mathcal{A}' via \mathcal{M} , written $\mathcal{G} \leq_{\mathcal{M}} \mathcal{G}'$, if $\mathcal{A}' = \mathcal{A} \times \mathcal{M}$ and every play $(\rho_0, m_0)(\rho_1, m_1)(\rho_2, m_2) \cdots$ in \mathcal{G}' starting in (v_0, m_0) is won by the player who wins the projected play $\rho_0\rho_1\rho_2 \cdots$ (which starts in v_0) in \mathcal{G} .

Remark 1. If $\mathcal{G} \leq_{\mathcal{M}} \mathcal{G}'$ and Player i has a positional winning strategy for \mathcal{G}' , then she also has a finite-state winning strategy with memory \mathcal{M} for \mathcal{G} .

It is well-known that parity games are determined with positional strategies [10, 11] and the winner can be determined in time $\mathcal{O}(m(n/d)^{\lceil d/2 \rceil})$ [12], where n , m , and d denote the number of vertices, edges, and colors.

Lemma 2. *Let \mathcal{G} be a game with initial vertex v_0 and let \mathfrak{A} be a deterministic parity automaton² recognizing the language of winning plays for Player 0 that*

²A run of a parity automaton $(Q, \Sigma, q_0, \delta, \Omega)$ with coloring function $\Omega: Q \rightarrow \mathbb{N}$ is accepting, if the maximal color that is seen infinitely often by the run is even. See, e.g., [13] for a complete definition.

start in v_0 . Then, \mathcal{G} can be reduced to a parity game via a memory structure of size $|\mathfrak{A}|$.

This can be shown by turning the automaton $\mathfrak{A} = (Q, V, q_0, \delta, \Omega)$ into $\mathcal{M} = (M, m_0, \text{upd})$ with $M = Q$, $m_0 = \delta(q_0, v_0)$ and $\text{upd}(m, v) = \delta(m, v)$ and showing $\mathcal{G} \leq_{\mathcal{M}} \mathcal{G}'$, where $\mathcal{G}' = (\mathcal{A} \times \mathcal{M}, (v_0, m_0), \Omega')$ with $\Omega'(v, m) = \Omega(m)$.

3. PLTL and PLTL Games

Let \mathcal{V} be an infinite set of variables and let us fix a finite³ set P of atomic propositions which we use to build our formulae and to label arenas in which we evaluate them. The formulae of PLTL are given by the grammar

$$\varphi ::= p \mid \neg p \mid \varphi \wedge \varphi \mid \varphi \vee \varphi \mid \mathbf{X}\varphi \mid \varphi \mathbf{U}\varphi \mid \varphi \mathbf{R}\varphi \mid \mathbf{F}_{\leq z}\varphi \mid \mathbf{G}_{\leq z}\varphi ,$$

where $p \in P$ and $z \in \mathcal{V}$. We use the derived operators $\mathbf{tt} := p \vee \neg p$ and $\mathbf{ff} := p \wedge \neg p$ for some fixed $p \in P$, $\mathbf{F}\varphi := \mathbf{tt} \mathbf{U}\varphi$, and $\mathbf{G}\varphi := \mathbf{ff} \mathbf{R}\varphi$. Furthermore, we use $\varphi \rightarrow \psi$ as shorthand for $\neg\varphi \vee \psi$, where we have to require the antecedent φ to be a (negated) atomic proposition and identify $\neg\neg p$ with p . We assume negation to bind stronger than every other connective and operator, which allows us to omit some brackets. In the original work on PLTL [5], the operators $\mathbf{U}_{\leq x}$, $\mathbf{F}_{> y}$, $\mathbf{G}_{> x}$, and $\mathbf{U}_{> y}$ are also allowed. However, since they do not add expressiveness (see Lemma 2.2 of [5]), we treat them as derived operators instead of adding them as primitive operators.

The set of subformulae of a PLTL formula φ is denoted by $\text{cl}(\varphi)$ and we define the size of φ to be the cardinality of $\text{cl}(\varphi)$. Furthermore, we define $\text{var}_{\mathbf{F}}(\varphi) = \{z \in \mathcal{V} \mid \mathbf{F}_{\leq z}\psi \in \text{cl}(\varphi)\}$ to be the set of variables parameterizing eventually operators in φ , $\text{var}_{\mathbf{G}}(\varphi) = \{z \in \mathcal{V} \mid \mathbf{G}_{\leq z}\psi \in \text{cl}(\varphi)\}$ to be the set of variables parameterizing always operators in φ , and set $\text{var}(\varphi) = \text{var}_{\mathbf{F}}(\varphi) \cup \text{var}_{\mathbf{G}}(\varphi)$. From now on, we denote variables in $\text{var}_{\mathbf{F}}(\varphi)$ by x and variables in $\text{var}_{\mathbf{G}}(\varphi)$ by y . A formula φ is variable-free, if $\text{var}(\varphi) = \emptyset$.

In order to evaluate PLTL formulae, we define a variable valuation to be a mapping $\alpha: \mathcal{V} \rightarrow \mathbb{N}$. Now, we can define the model relation between an ω -word $w \in (2^P)^\omega$, a position n of w , a variable valuation α , and a PLTL formula as follows:

³We require P to be finite so that its power set is an alphabet. This greatly simplifies our notation and exposition when we translate formulae into automata, but is not essential.

- $(w, n, \alpha) \models p$ if and only if $p \in w_n$,
- $(w, n, \alpha) \models \neg p$ if and only if $p \notin w_n$,
- $(w, n, \alpha) \models \varphi \wedge \psi$ if and only if $(w, n, \alpha) \models \varphi$ and $(w, n, \alpha) \models \psi$,
- $(w, n, \alpha) \models \varphi \vee \psi$ if and only if $(w, n, \alpha) \models \varphi$ or $(w, n, \alpha) \models \psi$,
- $(w, n, \alpha) \models \mathbf{X} \varphi$ if and only if $(w, n + 1, \alpha) \models \varphi$,
- $(w, n, \alpha) \models \varphi \mathbf{U} \psi$ if and only if there exists a $k \geq 0$ such that $(w, n + k, \alpha) \models \psi$ and $(w, n + j, \alpha) \models \varphi$ for every j in the range $0 \leq j < k$,
- $(w, n, \alpha) \models \varphi \mathbf{R} \psi$ if and only if for every $k \geq 0$: either $(w, n + k, \alpha) \models \psi$ or there exists a j in the range $0 \leq j < k$ such that $(w, n + j, \alpha) \models \varphi$,
- $(w, n, \alpha) \models \mathbf{F}_{\leq x} \varphi$ if and only if there exists a j in the range $0 \leq j \leq \alpha(x)$ such that $(w, n + j, \alpha) \models \varphi$, and
- $(w, n, \alpha) \models \mathbf{G}_{\leq y} \varphi$ if and only if for every j in the range $0 \leq j \leq \alpha(y)$: $(w, n + j, \alpha) \models \varphi$.

For the sake of brevity, we write $(w, \alpha) \models \varphi$ instead of $(w, 0, \alpha) \models \varphi$ and say that w is a model of φ with respect to α .

As usual for parameterized temporal logics, the use of variables has to be restricted: bounding eventually and always operators by the same variable leads to an undecidable satisfiability problem [5]. Unlike the original definition – which uses two disjoint sets of variables to bound eventually and always operators – we prefer to use a single set, which saves us some notational overhead.

Definition 3. A PLTL formula φ is well-formed, if $\text{var}_{\mathbf{F}}(\varphi) \cap \text{var}_{\mathbf{G}}(\varphi) = \emptyset$.

We consider the following fragments of PLTL. Let φ be a PLTL formula:

- φ is an LTL formula, if φ is variable-free.
- φ is a PROMPT-LTL formula [6], if $\text{var}_{\mathbf{G}}(\varphi) = \emptyset$ and $|\text{var}_{\mathbf{F}}(\varphi)| \leq 1$.
- φ is a PLTL $_{\mathbf{F}}$ formula, if $\text{var}_{\mathbf{G}}(\varphi) = \emptyset$.
- φ is a PLTL $_{\mathbf{G}}$ formula, if $\text{var}_{\mathbf{F}}(\varphi) = \emptyset$.

- φ is a unipolar formula, if it is either a $\text{PLTL}_{\mathbf{F}}$ or a $\text{PLTL}_{\mathbf{G}}$ formula.

Every LTL, PROMPT-LTL, $\text{PLTL}_{\mathbf{F}}$, and every $\text{PLTL}_{\mathbf{G}}$ formula is well-formed by definition.

Note that we defined PLTL formulae to be in negation normal form. Nevertheless, a negation can be pushed to the atomic propositions using the duality of the pairs $(p, \neg p)$, (\wedge, \vee) , (\mathbf{X}, \mathbf{X}) (\mathbf{U}, \mathbf{R}) , and $(\mathbf{F}_{\leq z}, \mathbf{G}_{\leq z})$. Thus, we can define the negation of a PLTL formula.

Lemma 4. *For every PLTL formula φ there exists an efficiently constructible PLTL formula $\neg\varphi$ such that*

1. $(w, n, \alpha) \models \varphi$ if and only if $(w, n, \alpha) \not\models \neg\varphi$,
2. $|\neg\varphi| = |\varphi|$.
3. If φ is well-formed, then so is $\neg\varphi$.
4. If φ is an LTL formula, then so is $\neg\varphi$.
5. If φ is a $\text{PLTL}_{\mathbf{F}}$ formula, then $\neg\varphi$ is a $\text{PLTL}_{\mathbf{G}}$ formula and vice versa.

Now, we define games with PLTL winning conditions. To evaluate them on a play, we have to label the arena with atomic propositions from the set P we use to build our formulae. A labeled arena $\mathcal{A} = (V, V_0, V_1, E, \ell)$ consists of an arena (V, V_0, V_1, E) as defined in Section 2 and a labeling function $\ell: V \rightarrow 2^P$. In figures, we denote the labeling of a vertex v by a set of propositions above or below of v , where we omit empty labels. The trace of a play ρ is $\text{tr}(\rho) = \ell(\rho_0)\ell(\rho_1)\ell(\rho_2)\cdots$. To keep things simple, we refer to labeled arenas as arenas, too, as long as no confusion can arise.

A PLTL game $\mathcal{G} = (\mathcal{A}, v_0, \varphi)$ consists of an arena $\mathcal{A} = (V, V_0, V_1, E, \ell)$, an initial vertex $v_0 \in V$, and a well-formed PLTL formula φ . The size of \mathcal{G} , denoted by $|\mathcal{G}|$, is defined as $|\mathcal{G}| = |\mathcal{A}| + |\varphi|$. LTL, PROMPT-LTL, $\text{PLTL}_{\mathbf{F}}$, $\text{PLTL}_{\mathbf{G}}$, and unipolar games are defined by restricting the winning condition to LTL, PROMPT-LTL, $\text{PLTL}_{\mathbf{F}}$, $\text{PLTL}_{\mathbf{G}}$, and unipolar formulae, respectively. A play in $(\mathcal{A}, v_0, \varphi)$ is an infinite path through \mathcal{A} starting in v_0 . The notions of winning a play and of winning strategies are defined with respect to a variable valuation which allows the evaluation of φ . We say that Player 0 wins a play ρ with respect to a variable valuation α if $(\text{tr}(\rho), \alpha) \models \varphi$, otherwise Player 1 wins ρ with respect to α . A strategy σ for Player i is a winning strategy for her with respect to α if every play that is consistent with σ is won by Player i with respect to α . If Player i has such a winning strategy, then we say that she wins \mathcal{G} with respect to α . Winning an LTL game \mathcal{G} is independent of α , hence we are justified to say that Player i wins \mathcal{G} .

Definition 5. Let \mathcal{G} be a PLTL game. The set $\mathcal{W}_i(\mathcal{G})$ of winning variable valuations for Player i is

$$\mathcal{W}_i(\mathcal{G}) = \{\alpha \mid \text{Player } i \text{ wins } \mathcal{G} \text{ with respect to } \alpha\} .$$

If \mathcal{G} is an LTL game, then $\mathcal{W}_i(\mathcal{G})$ contains either every variable valuation – which is the case if Player i wins \mathcal{G} –, or it is empty – which is the case if Player $1 - i$ wins \mathcal{G} .

Example 6. Consider the arena \mathcal{A} depicted in Figure 1. The propositions q_0 and q_1 represent requests of some resources and p_0 and p_1 represent the corresponding responses. At vertex v_0 , Player 1 can choose to request one or both of q_0 and q_1 , and at vertex v_5 , Player 0 can respond to at most one of the requests.

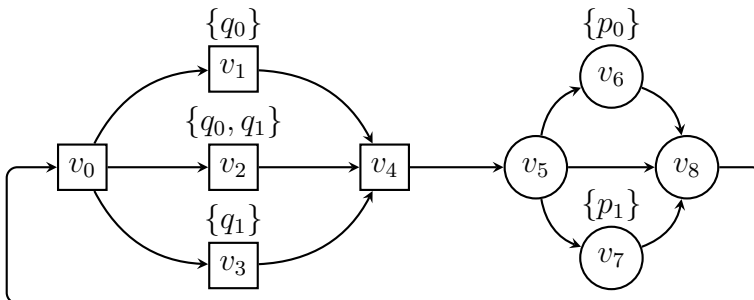


Figure 1: The arena for Example 6

Now, consider the formula $\varphi = \bigwedge_{i \in \{0,1\}} \mathbf{G}(q_i \rightarrow \mathbf{F}_{\leq x_i} p_i)$. To win the $\text{PLTL}_{\mathbf{F}}$ game $\mathcal{G} = (\mathcal{A}, v_0, \varphi)$ with respect to a variable valuation α , Player 0 has to answer every request q_i within $\alpha(x_i)$ positions. Let $\alpha(x_i) = 9$ for both x_i . We have $\alpha \in \mathcal{W}_0(\mathcal{G})$, witnessed by the strategy for Player 0 that alternates at v_5 between moving to v_6 and v_7 .

LTL games were – in a more general framework – investigated by Pnueli and Rosner, who showed them to be **2EXPTIME**-complete. Their results hold in the setting of graph-based games, too, and serve as the yardstick we measure our results on PLTL games against.

Theorem 7 ([14, 8, 15]). *LTL games are determined with finite-state strategies of size $2^{2^{O(|\varphi|)}}$ and determining the winner is **2EXPTIME**-complete.*

Due to Lemma 4, we can dualize PLTL games: let $\mathcal{A} = (V, V_0, V_1, E, \ell)$ be an arena and $\mathcal{G} = (\mathcal{A}, v_0, \varphi)$ be a PLTL game. Then, $\overline{\mathcal{A}} = (V, V_1, V_0, E, \ell)$ is the dual arena of \mathcal{A} , and $\overline{\mathcal{G}} = (\overline{\mathcal{A}}, v_0, \neg\varphi)$ is the dual game of \mathcal{G} . Since the negation of a well-formed formula is well-formed, too, the dual game satisfies the definition of a PLTL game. Furthermore, the dual of a PLTL $_{\mathbf{F}}$ game is a PLTL $_{\mathbf{G}}$ game and vice versa. Hence, we can solve many problems by only considering one type of unipolar game.

Lemma 8. *Let \mathcal{G} be a PLTL game.*

1. $\mathcal{W}_i(\mathcal{G})$ is the complement of $\mathcal{W}_{1-i}(\mathcal{G})$.
2. $\mathcal{W}_i(\mathcal{G}) = \mathcal{W}_{1-i}(\overline{\mathcal{G}})$.

The first statement can be proven by noticing that a PLTL game with respect to a fixed variable valuation is an ω -regular game (see Theorem 22), and therefore determined; the second one is a simple consequence of the definition of the dual game. Hence, every α either is in $\mathcal{W}_0(\mathcal{G})$ or in $\mathcal{W}_1(\mathcal{G})$.

A simple, but very useful property of PLTL is the monotonicity of the parameterized operators. If φ is satisfied at least once within the next k steps, then it is also satisfied at least once within the next k' steps, provided we have $k' > k$. Dually, if φ is satisfied during each of the next k steps, then also during the next k' steps, provided $k' < k$.

Lemma 9. *Let $\mathcal{G} = (\mathcal{A}, v_0, \varphi)$ be a PLTL game and let α and β be variable valuations satisfying $\beta(x) \geq \alpha(x)$ for every $x \in \text{var}_{\mathbf{F}}(\varphi)$ and $\beta(y) \leq \alpha(y)$ for every $y \in \text{var}_{\mathbf{G}}(\varphi)$. If $\alpha \in \mathcal{W}_0(\mathcal{G})$, then $\beta \in \mathcal{W}_0(\mathcal{G})$.*

Hence, $\mathcal{W}_0(\mathcal{G})$ is upwards-closed for variables parameterizing eventually operators and downwards-closed for variables parameterizing always operators. For a unipolar game \mathcal{G} , this implies that $\mathcal{W}_i(\mathcal{G})$ is semilinear. The exact descriptive complexity of the set of winning valuations for a non-unipolar PLTL game is open. This is even the case for PLTL model checking [5]. By applying the downwards-closure of the parameterized always operator, we can show that the projection of $\mathcal{W}_0(\mathcal{G})$ to variables parameterizing eventually operators is semilinear. Obtaining the dual result is more complicated (since we cannot rely on upwards-closure of $\mathcal{W}_0(\mathcal{G})$), but applying the alternating color technique presented in Subsection 4.1 shows the projection of $\mathcal{W}_0(\mathcal{G})$ to variables parameterizing always operators to be semilinear as well.

4. Solving PLTL Games

In this section, we show how to solve PLTL games. Since we consider initialized games, solving them only requires to determine the winner from the initial vertex and a corresponding winning strategy. However, winning a PLTL game (and being a winning strategy) is defined with respect to a variable valuation. Hence, solving a PLTL game \mathcal{G} refers to properties of the sets $\mathcal{W}_i(\mathcal{G})$. We are interested in the following decision problems.

- Membership: given a PLTL game \mathcal{G} , $i \in \{0, 1\}$, and a variable valuation α , does $\alpha \in \mathcal{W}_i(\mathcal{G})$ hold?
- Emptiness: given a PLTL game \mathcal{G} and $i \in \{0, 1\}$, is $\mathcal{W}_i(\mathcal{G})$ empty?
- Finiteness: given a PLTL game \mathcal{G} and $i \in \{0, 1\}$, is $\mathcal{W}_i(\mathcal{G})$ finite?
- Universality: given a PLTL game \mathcal{G} and $i \in \{0, 1\}$, does $\mathcal{W}_i(\mathcal{G})$ contain every variable valuation?

We encode variable valuations in binary (and restrict them to variables occurring in the winning condition). Hence, we measure the running time of algorithms for the membership problem in $|\mathcal{G}| + \sum_{z \in \text{var}(\varphi)} \lceil \log_2(\alpha(z) + 1) \rceil$ and the running time of algorithms for the latter three problems in $|\mathcal{G}|$.

We obtain an algorithm with doubly-exponential running time for the membership problem when we compute optimal winning strategies in Section 5. For the time being, we just remark that the membership problem is trivially **2EXPTIME**-hard. Let \mathcal{G} be an LTL game: then, $\mathcal{W}_0(\mathcal{G})$ contains the empty variable valuation if and only if Player 0 wins \mathcal{G} .

In the remainder of this section, we consider the latter three problems. In Subsection 4.1, we extend the alternating color technique of Kupferman et al. [6] for PROMPT-LTL to PLTL_F. Then, in Subsection 4.2, we apply this technique to solve the emptiness problem for PLTL_F games in doubly-exponential time by a reduction to solving LTL games. Finally, in Subsection 4.3, we prove that this result and the monotonicity of PLTL suffice to solve the latter three problems for full PLTL in doubly-exponential time.

4.1. Digression: The Alternating Color Technique

Kupferman et al. introduced PROMPT-LTL and solved several of its decision problems – among them model-checking, assume-guarantee model-checking, and the realizability problem – by using their alternating color

technique [6]. Intuitively, the alternating color technique allows to replace a parameterized eventually operator by an LTL formula: the positions of a trace are colored – either red or green – and a parameterized eventually $\mathbf{F}_{\leq x} \psi$ is replaced by the requirement that ψ holds within one color change (which is expressible in LTL). If there is an upper bound on the distance between adjacent color changes, then the waiting time for the parameterized eventually is also bounded. In games, the bound on the distance is obtained by applying finite-state determinacy.

Although the technique in its original formulation is only applicable to PROMPT-LTL formulae it is easy to see that the restriction to a single variable is not essential. Furthermore, it turns out to be useful to abandon the restriction when we consider the optimization problems for PLTL games in Section 5. Hence, we state the technique here in a slightly more general version than the one presented in the original work on PROMPT-LTL.

Let $p \notin P$ be a fixed fresh proposition. An ω -word $w' \in (2^{P \cup \{p\}})^\omega$ is a p -coloring of $w \in (2^P)^\omega$ if $w'_n \cap P = w_n$, i.e., w_n and w'_n coincide on all propositions in P . The additional proposition p can be thought of as the color of w'_n : we say that a position n is green if $p \in w'_n$, and say that it is red if $p \notin w'_n$. Furthermore, we say that the color changes at position n , if $n = 0$ or if the colors of w'_{n-1} and w'_n are not equal. In this situation, we say that n is a change point. A p -block is a maximal monochromatic infix $w'_m \cdots w'_n$ of w' , i.e., the color changes at m and $n + 1$ but not in between. Let $k \geq 1$: we say that w' is k -spaced, if the color changes infinitely often and each p -block has length at least k ; we say that w' is k -bounded, if each p -block has length at most k (which implies that the color changes infinitely often).

Given a PLTL $_{\mathbf{F}}$ formula φ and $X \subseteq \text{var}(\varphi)$, let φ_X denote the formula obtained by inductively replacing every subformula $\mathbf{F}_{\leq x} \psi$ with $x \notin X$ by

$$(p \rightarrow (p \mathbf{U} (\neg p \mathbf{U} \psi_X))) \wedge (\neg p \rightarrow (\neg p \mathbf{U} (p \mathbf{U} \psi_X))) .$$

We have $\text{var}(\varphi_X) = X$ (i.e., X denotes the variables that are not replaced) and $|\varphi_X| \in \mathcal{O}(|\varphi|)$. Furthermore, the formula $\text{alt}_p = \mathbf{G} \mathbf{F} p \wedge \mathbf{G} \mathbf{F} \neg p$ is satisfied if the colors change infinitely often. Finally, consider the formula $\varphi_X \wedge \text{alt}_p$, which is satisfied by w with respect to α , if the following holds:

- The color changes infinitely often.
- Every subformula $\mathbf{F}_{\leq x} \psi$ with $x \notin X$ is satisfied within one color change.

- The parameterized eventually operators with variables $x \in X$ – which are not replaced in φ_X – are satisfied within the bounds specified by α .

Next, we show that φ and φ_X are “equivalent” on ω -words which are bounded and spaced. Our correctness lemma (slightly) differs from the original one presented in [6], since we have not just one variable (as in a PROMPT-LTL formula) and allow to replace just some parameterized eventually operators. However, the proof itself is similar to the original one.

Lemma 10 (cf. Lemma 2.1 of [6]). *Let φ be a $\text{PLTL}_{\mathbf{F}}$ formula, let $X \subseteq \text{var}(\varphi)$, and let $w \in (2^P)^\omega$.*

1. *If $(w, \alpha) \models \varphi$, then $(w', \alpha) \models \varphi_X \wedge \text{alt}_p$ for every k -spaced p -coloring w' of w , where $k = \max_{x \in \text{var}(\varphi) \setminus X} \alpha(x)$.*
2. *Let $k \in \mathbb{N}$. If w' is a k -bounded p -coloring of w with $(w', \alpha) \models \varphi_X$, then $(w, \beta) \models \varphi$, where $\beta(z) = \begin{cases} 2k & \text{if } z \in \text{var}(\varphi) \setminus X, \\ \alpha(z) & \text{if } z \in X. \end{cases}$*

Now, we apply this result to $\text{PLTL}_{\mathbf{F}}$ games.

4.2. Solving the Emptiness Problem for $\text{PLTL}_{\mathbf{F}}$ Games

The **2EXPTIME**-membership proof of the PROMPT-LTL realizability problem by Kupferman et al. can be adapted to solve the emptiness problem for graph-based $\text{PLTL}_{\mathbf{F}}$ games.

Theorem 11. *The emptiness problem for $\text{PLTL}_{\mathbf{F}}$ games is in **2EXPTIME**.*

The proof presented in the following is similar to the one of Kupferman et al., but the presentation is more involved, since we consider graph-based games⁴. Most importantly, we have to allow Player 0 to produce p -colorings of plays. Since she has to be able to change the color while it is not her turn, we have to add choice vertices to the arena that allow her to produce change points at any time. However, adding the choice vertices requires to ignore them when evaluating a formula to determine the winner of a play. Thus, we introduce blinking semantics for infinite games: under this semantics, only every other vertex contributes to the trace of a play.

⁴Alternatively, one could transform a graph-based $\text{PLTL}_{\mathbf{F}}$ game into a realizability problem.

We begin by transforming the original arena \mathcal{A} into an arena \mathcal{A}_b in which Player 0 produces p -colorings of the plays of the original arena, i.e., \mathcal{A}_b will consist of two disjoint copies of \mathcal{A} , one labeled by p , the other one not. Assume a play is in vertex v in one component. Then, the player whose turn it is at v chooses a successor v' of v and Player 0 picks a component. The play then continues in this component's vertex v' . We split this into two sequential moves: first, the player whose turn it is chooses a successor and then Player 0 chooses the component. Thus, we have to introduce a new choice vertex for every edge of \mathcal{A} which allows Player 0 to choose the component. Formally, given an arena $\mathcal{A} = (V, V_0, V_1, E, \ell)$, we define the extended arena $\mathcal{A}_b = (V', V'_0, V'_1, E', \ell')$ by

- $V' = V \times \{0, 1\} \cup E$,
- $V'_0 = V_0 \times \{0, 1\} \cup E$ and $V'_1 = V_1 \times \{0, 1\}$,
- $E' = \{((v, 0), e), ((v, 1), e), (e, (v', 0)), (e, (v', 1)) \mid e = (v, v') \in E\}$, and
- $\ell'(e) = \emptyset$ for every $e \in E$ and $\ell'(v, b) = \begin{cases} \ell(v) \cup \{p\} & \text{if } b = 0, \\ \ell(v) & \text{if } b = 1. \end{cases}$

A path through \mathcal{A}_b has the form $(\rho_0, b_0)e_0(\rho_1, b_1)e_1(\rho_2, b_2)\cdots$ for some path $\rho_0\rho_1\rho_2\cdots$ through \mathcal{A} , where $e_n = (\rho_n, \rho_{n+1})$ and $b_n \in \{0, 1\}$. Also, we have $|\mathcal{A}_b| \in \mathcal{O}(|\mathcal{A}|^2)$.

The definition of \mathcal{A}_b necessitates a modification of the game's semantics: only every other vertex is significant when it comes to determining the winner of a play in \mathcal{A}_b , the choice vertices have to be ignored. This motivates blinking semantics for PLTL games. Let $\mathcal{G} = (\mathcal{A}, v_0, \varphi)$ be a PLTL game and let $\rho = \rho_0\rho_1\rho_2\cdots$ be a play. Player 0 wins ρ under blinking semantics with respect to α , if $(\text{tr}(\rho_0\rho_2\rho_4\cdots), \alpha) \models \varphi$. Analogously, Player 1 wins ρ under blinking semantics with respect to α , if $(\text{tr}(\rho_0\rho_2\rho_4\cdots), \alpha) \not\models \varphi$. The notions of winning strategies and winning \mathcal{G} under blinking semantics with respect to α are defined as for games with standard semantics.

Finite-state determinacy of LTL games under blinking semantics can be proven analogously to the case for LTL games under standard semantics.

Lemma 12. *LTL games under blinking semantics are determined with finite-state strategies of size $2^{2^{\mathcal{O}(|\varphi|)}}$ and determining the winner is **2EXPTIME**-complete.*

Corollary 13. *PLTL games under blinking semantics with respect to a fixed variable valuation α are determined with finite-state strategies.*

Now, we can state the “equivalence” of a $\text{PLTL}_{\mathbf{F}}$ game $(\mathcal{A}, v_0, \varphi)$ and its counterpart in \mathcal{A}_b with blinking semantics obtained by replacing (some) parameterized eventually operators.

Lemma 14. *Let $\mathcal{G} = (\mathcal{A}, v_0, \varphi)$ be a $\text{PLTL}_{\mathbf{F}}$ game, let $X \subseteq \text{var}(\varphi)$, and let $\mathcal{G}' = (\mathcal{A}_b, (v_0, 0), \varphi_X \wedge \text{alt}_p)$.*

1. *If Player 0 wins \mathcal{G} with respect to a variable valuation α , then she also wins \mathcal{G}' under blinking semantics with respect to α .*
2. *If Player 0 wins \mathcal{G}' under blinking semantics with respect to a variable valuation α , then there exists a variable valuation β with $\beta(z) = \alpha(z)$ for every $z \in X$ such that she wins \mathcal{G} with respect to β .*

Before we prove the lemma, let us mention that this suffices to prove Theorem 11: let $\mathcal{G} = (\mathcal{A}, v_0, \varphi)$ be a $\text{PLTL}_{\mathbf{F}}$ game and consider the case $X = \emptyset$. Then, φ_X is an LTL formula and Player 0 wins \mathcal{G} with respect to some variable valuation α if and only if she wins the LTL game $\mathcal{G}' = (\mathcal{A}_b, (v_0, 0), \varphi_X \wedge \text{alt}_p)$ under blinking semantics. As \mathcal{G}' is only polynomially larger than \mathcal{G} and as LTL games under blinking semantics can be solved in doubly-exponential time, we have shown **2EXPTIME**-membership of the emptiness problem for $\mathcal{W}_0(\mathcal{G})$, if \mathcal{G} is a $\text{PLTL}_{\mathbf{F}}$ game.

Now, let us turn to the proof of Lemma 14.

Proof. 1. Let σ be a winning strategy for Player 0 for \mathcal{G} with respect to α and define $k = \max_{x \in \text{var}(\varphi) \setminus X} \alpha(x)$. We turn σ into a strategy σ' for \mathcal{G}' that mimics the behavior of σ at vertices (v, b) and colors the play in alternating p -blocks of length k at the choice vertices. Hence, the trace of the resulting play (without choice vertices) in \mathcal{A}_b is a k -spaced p -coloring of the trace of a play that is consistent with σ . Thus, Lemma 10.1 is applicable and shows that σ' is winning under blinking semantics with respect to α . Formally, let

$$\sigma'((\rho_0, b_0)(\rho_0, \rho_1) \cdots (\rho_{n-1}, \rho_n)(\rho_n, b_n)) = (\rho_n, \sigma(\rho_0 \cdots \rho_n))$$

if $(\rho_n, b_n) \in V'_0$, which implies $\rho_n \in V_0$. Thus, at a non-choice vertex, Player 0 mimics the behavior of σ . At choice vertices, she alternates between the two copies of the arena every k steps, i.e.,

$$\sigma'((\rho_0, b_0)(\rho_0, \rho_1) \cdots (\rho_n, b_n)(\rho_n, \rho_{n+1})) = \begin{cases} (\rho_{n+1}, 0) & \text{if } n \bmod 2k < k, \\ (\rho_{n+1}, 1) & \text{if } n \bmod 2k \geq k. \end{cases}$$

Let $\rho = \rho_0\rho_1\rho_2\cdots$ be a play in \mathcal{A}_b that is consistent with σ' and let

$$\rho' = \rho_0\rho_2\rho_4\cdots = (v_0, b_0)(v_1, b_1)(v_2, b_2)\cdots .$$

By definition of σ' , the sequence $v_0v_1v_2\cdots$ is a play in \mathcal{A} that is consistent with σ and thus winning for Player 0 with respect to α , i.e., we have $(\text{tr}(v_0v_1v_2\cdots), \alpha) \models \varphi$. Furthermore, $\text{tr}(\rho')$ is a k -spaced p -coloring of $\text{tr}(v_0v_1v_2\cdots)$. Hence, $(\text{tr}(\rho'), \alpha) \models \varphi_X \wedge \text{alt}_p$ due to Lemma 10.1. Thus, σ' is a winning strategy for $(\mathcal{A}_b, (v_0, 0), \varphi_X \wedge \text{alt}_p)$ under blinking semantics with respect to α .

2. Assume that Player 0 wins $(\mathcal{A}_b, (v_0, 0), \varphi_X \wedge \text{alt}_p)$ under blinking semantics with respect to α . Then, due to Corollary 13, she also has a finite-state winning strategy σ' implemented by some memory structure $\mathcal{M}' = (M', m'_0, \text{upd}')$ and some next-move function nxt' . We construct a strategy σ for \mathcal{G} by simulating σ' such that each play in \mathcal{A} that is consistent with σ has a k -bounded p -coloring in \mathcal{A}_b that is consistent with σ' . This suffices to show that σ is winning with respect to a variable valuation β as required.

Since σ' is implemented by \mathcal{M}' , it suffices to keep track of the last vertex of the simulated play – which is never a choice vertex – and the memory state for the simulated play. Hence, we transform \mathcal{M}' into a memory structure $\mathcal{M} = (M, m_0, \text{upd})$ for \mathcal{A} with $M = (V \times \{0, 1\}) \times M'$, $m_0 = ((v, 0), m'_0)$, and

$$\text{upd}(((v, b), m), v') = (\text{nxt}'(e, m'), \text{upd}'(m', \text{nxt}'(e, m')))$$

where $e = (v, v')$ and $m' = \text{upd}'(m, e)$.

Let w be a play prefix of a play in \mathcal{A} . The memory state $\text{upd}^*(w) = ((v, b), m)$ encodes the following information: the simulated play w' in \mathcal{A}_b ends in (v, b) , where v is the last vertex of w , and we have $\text{upd}'^*(w') = m$. Hence, it contains all information necessary to apply the next-move function nxt' to mimic σ' . Finally, we define a next-move function $\text{nxt}: V_0 \times M \rightarrow V$ for Player 0 in \mathcal{A} by

$$\text{nxt}(v, ((v', b), m)) = \begin{cases} v'' & \text{if } v = v' \text{ and } \text{nxt}'((v', b), m) = (v', v''), \\ \bar{v} & \text{otherwise, for some } \bar{v} \in V \text{ with } (v, \bar{v}) \in E. \end{cases}$$

By definition of \mathcal{M} , the second case of the definition is never invoked, since $\text{upd}^*(wv) = ((v', b), m)$ always satisfies $v = v'$.

It remains to show that the strategy σ implemented by \mathcal{M} and nxt is indeed a winning strategy for Player 0 for $(\mathcal{A}, v_0, \varphi)$ with respect to a variable valuation β that coincides with α on all variables in X .

Let $\rho_0\rho_1\rho_2\cdots$ be a play in \mathcal{A} that is consistent with σ . A straightforward induction shows that there exist bits b_0, b_1, b_2, \dots such that the play $(\rho_0, b_0)(\rho_0, \rho_1)(\rho_1, b_1)(\rho_1, \rho_2)(\rho_2, b_2)\cdots$ in \mathcal{A}_b is consistent with σ' .

Hence, the trace of $\rho'' = (\rho_0, b_0)(\rho_1, b_1)(\rho_2, b_2)\cdots$ satisfies $\varphi_X \wedge alt_p$ with respect to α . We show that $\text{tr}(\rho'')$ is k -bounded, where $k = |V| \cdot |M| + 1$. This suffices to finish the proof: let $\beta(x) = 2k$ for $x \in \text{var}(\varphi) \setminus X$ and $\beta(z) = \alpha(z)$ for $z \in X$. Then, we can apply Lemma 10.2 and obtain $(\text{tr}(\rho), 0, \beta) \models \varphi$, as $\text{tr}(\rho'')$ is a k -bounded p -coloring of $\text{tr}(\rho)$. Hence, σ is indeed a winning strategy for Player 0 for $(\mathcal{A}, v_0, \varphi)$ with respect to β .

Towards a contradiction assume that $\text{tr}(\rho'')$ is not k -bounded. Then, there exist consecutive change points i and j such that $j - i \geq k + 1$. Then, there also exist i' and j' with $i \leq i' < j' < j$ such that $\rho_{i'} = \rho_{j'}$ and

$$\text{upd}^{l^*}((\rho_0, b_0) \cdots (\rho_{i'}, b_{i'})) = \text{upd}^{l^*}((\rho_0, b_0) \cdots (\rho_{j'}, b_{j'})) \quad ,$$

i.e., the last vertices of both play prefixes are equal and the memory states after both play prefixes are equal, too. Hence, the play

$$\rho^* = (\rho_0, b_0) \cdots (\rho_{i'-1}, b_{i'-1}) [(\rho_{i'}, b_{i'}) \cdots (\rho_{j'-1}, b_{j'-1})(\rho_{j'-1}, \rho_{j'})]^\omega \quad ,$$

obtained by traversing the cycle between $(\rho_{i'}, b_{i'})$ and $(\rho_{j'}, b_{j'})$ infinitely often, is consistent with σ' , since the memory states reached at the beginning and the end of the loop are the same. Remember that the bits do not change between i and j . Thus, $\text{tr}(\rho^*)$ has only finitely many change points and does not satisfy alt_p under blinking semantics. This contradicts the fact that σ' is a winning strategy for $(\mathcal{A}_b, (v_0, 0), \varphi_X \wedge alt_p)$ under blinking semantics with respect to α . \square

Let $\mathcal{G} = (\mathcal{A}, v_0, \varphi)$ be a PLTL_F game. Lemma 12 allows us to bound the size of a finite-state winning strategy for $(\mathcal{A}_b, (v_0, 0), \varphi_\emptyset \wedge alt_p)$, which in turn bounds the values of a variable valuation that is winning for Player 0 for the game \mathcal{G} .

Corollary 15. *Let \mathcal{G} be a PLTL_F game. If $\mathcal{W}_0(\mathcal{G}) \neq \emptyset$, then there exists a $k \in 2^{2^{\mathcal{O}(|\mathcal{G}|)}}$ such that Player 0 wins \mathcal{G} with respect to the variable valuation that maps every variable to k .*

To conclude let us note that a further generalization of the alternating color technique to full PLTL can be applied to construct the projection of

$\mathcal{W}_0(\mathcal{G})$ to the variables in $\text{var}_{\mathbf{G}}(\varphi)$ while still ensuring bounds on the (replaced) parameterized eventually operators. This substantiates our claim made in Section 3: the projection of $\mathcal{W}_0(\mathcal{G})$ to variables parameterizing always operators is a semilinear set.

4.3. Solving PLTL Games

The doubly-exponential time algorithm for the emptiness problem for $\text{PLTL}_{\mathbf{F}}$ games allows us to solve the emptiness, finiteness, and the universality problem for games with winning conditions in full PLTL as well.

Theorem 16. *The emptiness, the finiteness, and the universality problem for PLTL games are $\mathbf{2EXPTIME}$ -complete.*

Proof. We begin by showing $\mathbf{2EXPTIME}$ -membership for all three problems. Let $\mathcal{G} = (\mathcal{A}, v_0, \varphi)$ be a PLTL game. Due to duality (see Lemma 8. 2), it suffices to consider $i = 0$.

Emptiness of $\mathcal{W}_0(\mathcal{G})$: Let $\varphi_{\mathbf{F}}$ be the formula obtained from φ by inductively replacing every subformula $\mathbf{G}_{\leq y} \psi$ by ψ , and let $\mathcal{G}_{\mathbf{F}} = (\mathcal{A}, v_0, \varphi_{\mathbf{F}})$, which is a $\text{PLTL}_{\mathbf{F}}$ game. Applying the monotonicity of $\mathbf{G}_{\leq y}$, we obtain that $\mathcal{W}_0(\mathcal{G})$ is empty if and only if $\mathcal{W}_0(\mathcal{G}_{\mathbf{F}})$ is empty.

The latter problem can be decided in doubly-exponential time by Theorem 11. Hence, the emptiness of $\mathcal{W}_0(\mathcal{G})$ can be decided in doubly-exponential time as well, since we have $|\varphi_{\mathbf{F}}| \leq |\varphi|$.

Universality of $\mathcal{W}_0(\mathcal{G})$: Applying first complementarity and then duality as stated in Lemma 8, we have that $\mathcal{W}_0(\mathcal{G})$ is universal if and only if $\mathcal{W}_1(\mathcal{G})$ is empty, which is the case if and only if $\mathcal{W}_0(\overline{\mathcal{G}})$ is empty. The latter problem is decidable in doubly-exponential time as shown above.

Finiteness of $\mathcal{W}_0(\mathcal{G})$: If $\text{var}_{\mathbf{F}}(\varphi) \neq \emptyset$, then $\mathcal{W}_0(\mathcal{G})$ is infinite, if and only if it is non-empty, due to monotonicity of $\mathbf{F}_{\leq x}$. The emptiness of $\mathcal{W}_0(\mathcal{G})$ can be decided in doubly-exponential time as discussed above.

If $\text{var}_{\mathbf{F}}(\varphi) = \emptyset$, then \mathcal{G} is a $\text{PLTL}_{\mathbf{G}}$ game. We assume that φ has at least one parameterized temporal operator, since the problem is trivial otherwise. Then, the set $\mathcal{W}_0(\mathcal{G})$ is infinite if and only if there is a variable $y \in \text{var}_{\mathbf{G}}(\varphi)$ that is mapped to infinitely many values by the valuations in $\mathcal{W}_0(\mathcal{G})$. By downwards-closure, we can assume that all other variables are mapped to zero. Furthermore, y is mapped to infinitely many values if and only if it is mapped to all possible values, again by downwards-closure. To combine this, we define φ_y to be the formula obtained from φ by inductively replacing

every subformula $\mathbf{G}_{\leq y'} \psi$ for $y' \neq y$ by ψ and define $\mathcal{G}_y = (\mathcal{A}, v_0, \varphi_y)$. Then, $\mathcal{W}_0(\mathcal{G})$ is infinite, if and only if there exists some variable $y \in \text{var}(\varphi)$ such that $\mathcal{W}_0(\mathcal{G}_y)$ is universal. So, deciding whether $\mathcal{W}_0(\mathcal{G})$ is infinite can be done in doubly-exponential time by solving $|\text{var}(\varphi)|$ many universality problems for $\text{PLTL}_{\mathbf{G}}$ games, which were discussed above.

It remains to show **2EXPTIME**-hardness of all three problems. Let \mathcal{G} be an LTL game. The following statements are equivalent. (1) Player 0 wins \mathcal{G} ; (2) $\mathcal{W}_0(\mathcal{G})$ is non-empty; (3) $\mathcal{W}_0(\mathcal{G})$ is universal; (4) $\mathcal{W}_1(\mathcal{G})$ is finite. Hence, all three problems are indeed **2EXPTIME**-hard. \square

All three problems but the finiteness problem for $\text{PLTL}_{\mathbf{G}}$ games only require the solution of a single LTL game under blinking semantics. Furthermore, all these games are only polynomially larger than the original game.

5. Optimal Strategies for Games with PLTL Winning conditions

It is natural to view synthesis of PLTL specifications as optimization problem: which is the *best* variable valuation α such that Player 0 can win \mathcal{G} with respect to α ? For unipolar games, we consider two quality measures for a valuation α in a game with winning condition φ : the maximal parameter $\max_{z \in \text{var}(\varphi)} \alpha(z)$ and the minimal parameter $\min_{z \in \text{var}(\varphi)} \alpha(z)$. For a $\text{PLTL}_{\mathbf{F}}$ game, Player 0 tries to minimize the waiting times. Hence, we are interested in minimizing the minimal or maximal parameter. Dually, for $\text{PLTL}_{\mathbf{G}}$ games, we are interested in maximizing the quality measures. The dual problems (e.g., maximizing the waiting times) are trivial due to upwards- respectively downwards-closure of the set of winning valuations.

The main result of this section states that all optimization problems for unipolar games can be solved in triply-exponential time. In Section 6, we complement this with doubly-exponential lower bounds on the value of an optimal variable valuation in a unipolar PLTL game, thereby showing that the doubly-exponential upper bounds obtained in Subsection 4.2 are (almost) tight.

In the following, we assume all winning conditions to contain at least one variable, since the optimization problems are trivial otherwise.

Theorem 17. *Let $\mathcal{G}_{\mathbf{F}}$ be a $\text{PLTL}_{\mathbf{F}}$ game with winning condition $\varphi_{\mathbf{F}}$ and let $\mathcal{G}_{\mathbf{G}}$ be a $\text{PLTL}_{\mathbf{G}}$ game with winning condition $\varphi_{\mathbf{G}}$. The following values (and winning strategies realizing them) can be computed in triply-exponential time.*

1. $\min_{\alpha \in \mathcal{W}_0(\mathcal{G}_{\mathbf{F}})} \min_{x \in \text{var}(\varphi_{\mathbf{F}})} \alpha(x)$.
2. $\min_{\alpha \in \mathcal{W}_0(\mathcal{G}_{\mathbf{F}})} \max_{x \in \text{var}(\varphi_{\mathbf{F}})} \alpha(x)$.
3. $\max_{\alpha \in \mathcal{W}_0(\mathcal{G}_{\mathbf{G}})} \max_{y \in \text{var}(\varphi_{\mathbf{G}})} \alpha(y)$.
4. $\max_{\alpha \in \mathcal{W}_0(\mathcal{G}_{\mathbf{G}})} \min_{y \in \text{var}(\varphi_{\mathbf{G}})} \alpha(y)$.

A special case of the PLTL $_{\mathbf{F}}$ optimization problems is the PROMPT–LTL optimization problem. Due to our non-triviality requirement, the winning condition in a PROMPT–LTL game has exactly one variable x . Hence, the inner maximization or minimization becomes trivial and the problem asks to determine $\min_{\alpha \in \mathcal{W}_0(\mathcal{G})} \alpha(x)$ and a winning strategy for Player 0 realizing this value. Dually, a PLTL $_{\mathbf{G}}$ optimization problem with a single variable y asks to determine $\max_{\alpha \in \mathcal{W}_0(\mathcal{G})} \alpha(y)$ and for a winning strategy realizing this value.

Due to duality, there is a tight connection between PROMPT–LTL optimization problems and PLTL $_{\mathbf{G}}$ optimization problems with a single variable: let \mathcal{G} be a PLTL $_{\mathbf{G}}$ game with winning condition φ with $\text{var}(\varphi) = \{y\}$. Then,

$$\max_{\alpha \in \mathcal{W}_0(\mathcal{G})} \alpha(y) = \max_{\alpha \in \mathcal{W}_1(\bar{\mathcal{G}})} \alpha(y) = \min_{\alpha \in \mathcal{W}_0(\bar{\mathcal{G}})} \alpha(y) + 1, \quad (1)$$

due to the closure properties and Lemma 8. Here, $\bar{\mathcal{G}}$ is a PROMPT–LTL game. Thus, to compute the optimal variable valuation in a PLTL $_{\mathbf{G}}$ game with a single variable valuation, it suffices to solve a PROMPT–LTL optimization problem. We defer the computation of strategies realizing the optimal values in both types of games to the end of this subsection.

We begin by showing that all four problems mentioned in Theorem 17 can be reduced to optimization problems with a single variable. As we have shown above how to translate an optimization problem for a PLTL $_{\mathbf{G}}$ game with a single variable into a PROMPT–LTL optimization problem, this implies that it suffices to solve PROMPT–LTL optimization problems.

min min: For each $x \in \text{var}(\varphi_{\mathbf{F}})$ we apply the alternating color technique to construct the projection of $\mathcal{W}_0(\mathcal{G}_{\mathbf{F}})$ to the values of x : let $\mathcal{G}_{\mathbf{F}} = (\mathcal{A}, v_0, \varphi_{\mathbf{F}})$ and define $\mathcal{G}_x = (\mathcal{A}_b, (v_0, 0), (\varphi_{\mathbf{F}})_{\{x\}} \wedge \text{alt}_p)$ where \mathcal{A}_b and $(\varphi_{\mathbf{F}})_{\{x\}} \wedge \text{alt}_p$ are defined as in Subsection 4.1. Applying Lemma 14 yields

$$\min_{\alpha \in \mathcal{W}_0(\mathcal{G}_{\mathbf{F}})} \min_{x \in \text{var}(\varphi_{\mathbf{F}})} \alpha(x) = \min_{x \in \text{var}(\varphi)} \min\{\alpha(x) \mid \text{Player 0 wins } \mathcal{G}_x \text{ under blinking semantics w.r.t. } \alpha\}.$$

Since $\text{var}((\varphi_{\mathbf{F}})_{\{x\}}) = \{x\}$, we have reduced the minimization problem to $|\text{var}(\varphi_{\mathbf{F}})|$ many PROMPT–LTL optimization problems, albeit under blinking

semantics. We discuss the necessary adaptations to our proof, which is for the non-blinking case, below. Furthermore, a strategy realizing the optimum on the right-hand side can be turned into a strategy realizing the optimum on the left-hand side using the construction presented in the proof of Lemma 14.2 which turns a strategy for \mathcal{A}_b into a strategy for \mathcal{A} .

min max: This problem can directly be reduced to a PROMPT-LTL optimization problem: let $\varphi'_{\mathbf{F}}$ be the PROMPT-LTL formula obtained from $\varphi_{\mathbf{F}}$ by renaming each $x \in \text{var}(\varphi_{\mathbf{F}})$ to z and let $\mathcal{G}' = (\mathcal{A}, v_0, \varphi'_{\mathbf{F}})$, where \mathcal{A} and v_0 are the arena and the initial vertex of $\mathcal{G}_{\mathbf{F}}$. Then,

$$\min_{\alpha \in \mathcal{W}_0(\mathcal{G}_{\mathbf{F}})} \max_{x \in \text{var}(\varphi_{\mathbf{F}})} \alpha(x) = \min_{\alpha \in \mathcal{W}_0(\mathcal{G}')} \alpha(z) ,$$

due to upwards-closure of $\mathcal{W}_0(\mathcal{G}_{\mathbf{F}})$, and a strategy realizing the optimum on the right-hand side also realizes the optimum on the left-hand side.

max max: For every $y \in \text{var}(\varphi_{\mathbf{G}})$ let φ_y be obtained from $\varphi_{\mathbf{G}}$ by replacing every subformula $\mathbf{G}_{\leq y'} \psi$ with $y' \neq y$ by ψ and let $\mathcal{G}_y = (\mathcal{A}, v_0, \varphi_y)$, where \mathcal{A} and v_0 are the arena and the initial vertex of $\mathcal{G}_{\mathbf{G}}$. Then, we have

$$\max_{\alpha \in \mathcal{W}_0(\mathcal{G}_{\mathbf{G}})} \max_{y \in \text{var}(\varphi_{\mathbf{G}})} \alpha(y) = \max_{y \in \text{var}(\varphi_{\mathbf{G}})} \max_{\alpha \in \mathcal{W}_0(\mathcal{G}_y)} \alpha(y) ,$$

due to downwards-closure of $\mathcal{W}_0(\mathcal{G}_{\mathbf{G}})$, and a strategy realizing the optimum on the right-hand side also realizes the optimum on the left-hand side.

max min: Let $\varphi'_{\mathbf{G}}$ be obtained from $\varphi_{\mathbf{G}}$ by renaming every variable in $\varphi_{\mathbf{G}}$ to z and let $\mathcal{G}' = (\mathcal{A}, v_0, \varphi'_{\mathbf{G}})$, where \mathcal{A} and v_0 are the arena and the initial vertex of $\mathcal{G}_{\mathbf{G}}$. Then,

$$\max_{\alpha \in \mathcal{W}_0(\mathcal{G}_{\mathbf{G}})} \min_{y \in \text{var}(\varphi_{\mathbf{G}})} \alpha(y) = \max_{\alpha \in \mathcal{W}_0(\mathcal{G}')} \alpha(z) ,$$

due to downwards-closure of $\mathcal{W}_0(\mathcal{G}_{\mathbf{G}})$ and a strategy realizing the optimum on the right-hand side also realizes the optimum on the left-hand side.

All reductions increase the size of the arena at most quadratically and the size of the winning condition at most linearly. Furthermore, to minimize the minimal parameter value in a PLTL $_{\mathbf{F}}$ game and to maximize the maximal parameter value in a PLTL $_{\mathbf{G}}$ game, we have to solve $|\text{var}(\varphi)|$ many PROMPT-LTL optimization problems (for the other two problems just one) to solve the original unipolar optimization problem with winning condition φ . Thus, due to the duality of unipolar optimization problems with a single

variable shown in Equation (1), it remains to show that a PROMPT-LTL optimization problem can be solved in triply-exponential time.

So, let $\mathcal{G} = (\mathcal{A}, v_0, \varphi)$ be a PROMPT-LTL game with $\text{var}(\varphi) = \{x\}$. If $\mathcal{W}_0(\mathcal{G}) \neq \emptyset$ (which can be checked in doubly-exponential time), then Corollary 15 yields a $k \in 2^{2^{\mathcal{O}(|\mathcal{G}|)}}$ such that $\min_{\alpha \in \mathcal{W}_0(\mathcal{G})} \alpha(x) \leq k$. Hence, we have a doubly-exponential upper bound on an optimal variable valuation.

In the following, we denote by α_n the variable valuation mapping x to n and every other variable to zero. Since φ only contains the variable x , the smallest $n < k$ such that $\alpha_n \in \mathcal{W}_0(\mathcal{G})$ is equal to $\min_{\alpha \in \mathcal{W}_0(\mathcal{G})} \alpha(x)$. As the number of such valuations α_n is doubly-exponential in $|\mathcal{G}|$, it suffices to show that $\alpha_n \in \mathcal{W}_0(\mathcal{G})$ can be decided in triply-exponential time in the size of \mathcal{G} , provided that $n < k$. This is achieved by a game reduction to a parity game.

Fix a variable valuation α_n and assume we have a deterministic parity automaton $\mathfrak{P}_{\varphi, \alpha_n} = (Q, 2^P, q_0, \delta, \Omega)$ recognizing the language

$$L(\mathfrak{P}_{\varphi, \alpha_n}) = \{w \in (2^P)^\omega \mid (w, \alpha_n) \models \varphi\} .$$

Note that the language is uniquely determined by φ and the value $n = \alpha_n(x)$, where x is the variable appearing in φ . Consider the parity game $\mathcal{G}_n = (\mathcal{A} \times \mathcal{M}, (v_0, m_0), \Omega')$ where $\mathcal{M} = (Q, m_0, \text{upd})$ with $m_0 = \delta(q_0, \ell(v_0))$, $\text{upd}(m, v) = \delta(m, \ell(v))$, and $\Omega'(v, m) = \Omega(m)$. We have $\alpha_n \in \mathcal{W}_0(\mathcal{G})$ if and only if Player 0 wins \mathcal{G}_n as explained below Lemma 2. However, to meet our time bounds, $\mathfrak{P}_{\varphi, \alpha_n}$ has to be of (at most) triply-exponential size (in $|\mathcal{G}|$) with (at most) doubly-exponentially many priorities, provided that we have $\alpha_n(x) < k$. If this is the case, then we can solve the parity game \mathcal{G}_n in triply-exponential time. Furthermore, a winning strategy for the parity game associated to the minimal n can be turned in triply-exponential time into a finite-state winning strategy for the PROMPT-LTL game \mathcal{G} which realizes the optimal value. This strategy is implemented by a memory structure induced by the automaton $\mathfrak{P}_{\varphi, \alpha_n}$ as explained below Lemma 18.

If the PROMPT-LTL game \mathcal{G} has blinking semantics, then we have to adapt the construction slightly: instead of using an automaton $\mathfrak{P}_{\varphi, \alpha_n}$ for the language $\{w \in (2^P)^\omega \mid (w, \alpha_n) \models \varphi\}$ in the reduction, we turn $\mathfrak{P}_{\varphi, \alpha_n}$ into a deterministic parity automaton $\mathfrak{P}'_{\varphi, \alpha_n}$ that recognizes the language $\{w \in (2^P)^\omega \mid (w_0 w_2 w_4 \cdots, \alpha_n) \models \varphi\}$, which doubles the size, but does not change the number of priorities. Again, we denote by \mathcal{G}'_n the parity game obtained in the reduction via memory induced by $\mathfrak{P}'_{\varphi, \alpha_n}$. Then, Player 0 wins \mathcal{G} with respect to α_n under blinking semantics if and only if she wins \mathcal{G}'_n , which can again be determined in triply-exponential time.

Thus, the main step in the proof of Theorem 17 is to construct an automaton that has the following properties.

Lemma 18. *Let $n \leq k$. We can construct in triply-exponential time a deterministic parity automaton $\mathfrak{P}_{\varphi, \alpha_n}$ recognizing $\{w \in (2^P)^\omega \mid (w, \alpha_n) \models \varphi\}$ such that $|\mathfrak{P}_{\varphi, \alpha_n}| \in 2^{2^{2^{\mathcal{O}(|\mathcal{G}|)}}}$ and $\mathfrak{P}_{\varphi, \alpha_n}$ has at most $2^{2^{\mathcal{O}(|\mathcal{G}|)}}$ many priorities.*

Before we spend the next subsection proving the existence of an automaton as claimed in Lemma 18, let us show that these automata implement winning strategies realizing the optimal values. Due to the reductions, which allow to transfer optimal strategies, we only have to consider games with a single variable. For a PROMPT-LTL game, we determine the optimal variable valuation α_n by reductions to parity games. The automaton $\mathfrak{P}_{\varphi, \alpha_n}$ for this optimum can be turned into a memory structure which implements a finite-state winning strategy for Player 0 for \mathcal{G} with respect to α_n .

Now, consider a PLTL $_{\mathbf{G}}$ game \mathcal{G} whose winning condition φ has a single variable and remember that we reduced the problem to a PROMPT-LTL optimization problem via Equation (1). Hence, after we have determined the optimal variable valuation α_n for $\overline{\mathcal{G}}$, we construct the automaton $\mathfrak{P}_{\varphi, \alpha_{n-1}}$. This automaton can be turned into a memory structure that implements a winning strategy for \mathcal{G} with respect to the optimal variable valuation α_{n-1} .

5.1. Translating PLTL into “Small” Deterministic Automata

In this subsection we translate a PLTL formula with respect to a fixed variable valuation into a deterministic parity automaton while satisfying the requirements formulated in Lemma 18. To this end, we first construct an unambiguous⁵ generalized Büchi automaton⁶, which is then determinized into a parity automaton using a procedure tailored for unambiguous automata. This procedure constructs a parity automaton, while most other determinization procedures only yield Rabin automata, which we would have to translate into a parity automaton first.

Thus, we begin by constructing a generalized Büchi automaton $\mathfrak{A}_{\varphi, \alpha}$ recognizing the language $\{w \in (2^P)^\omega \mid (w, \alpha) \models \varphi\}$ for a given PLTL formula φ and a variable valuation α . The automaton guesses for each position of w

⁵An automaton is unambiguous, if it has at most one accepting run on every input.

⁶A run of a generalized Büchi automaton $(Q, \Sigma, Q_0, \Delta, \mathcal{F})$ with $\mathcal{F} \subseteq 2^Q$ is accepting, if it visits every $F \in \mathcal{F}$ infinitely often. Here, $Q_0 \subseteq Q$ is a set of initial states.

which subformulae of φ are satisfied with respect to α at this position and verifies these guesses while processing w . Since there is only one way to guess right, the automaton is unambiguous. Our construction for the first step is the adaption of the tableaux construction for Metric Temporal Logic (MTL) [3, 4] to PLTL. This logic is defined by adding the operators \mathbf{U}_I (and a corresponding past temporal operator) to LTL, where I is an arbitrary interval of \mathbb{N} whose end-points are integer constants, with the expected semantics. Since we are in this subsection interested in a PLTL formula with respect to a fixed variable valuation, our problem refers to constant bounds as well, and could therefore be expressed in MTL.

The states of the automaton are pairs (B, c) where B is the set of subformulae guessed to be satisfied at the current position and c is a mapping that encodes for every parameterized eventually $\mathbf{F}_{\leq x} \psi$ how many steps it takes before ψ is satisfied for the first time, and for every parameterized always $\mathbf{G}_{\leq y} \psi$ how many steps it takes before ψ is false for the first time.

Formally, let φ be a PLTL formula. A set $B \subseteq \text{cl}(\varphi)$ is consistent, if the following conditions⁷ are satisfied:

- (C1) For all $p, \neg p \in \text{cl}(\varphi)$: $p \in B$ if and only if $\neg p \notin B$.
- (C2) For all $\psi_1 \wedge \psi_2 \in \text{cl}(\varphi)$: $\psi_1 \wedge \psi_2 \in B$ if and only if $\psi_1 \in B$ and $\psi_2 \in B$.
- (C3) For all $\psi_1 \vee \psi_2 \in \text{cl}(\varphi)$: $\psi_1 \vee \psi_2 \in B$ if and only if $\psi_1 \in B$ or $\psi_2 \in B$.
- (C4) For all $\psi_1 \mathbf{U} \psi_2 \in \text{cl}(\varphi)$: $\psi_2 \in B$ implies $\psi_1 \mathbf{U} \psi_2 \in B$.
- (C5) For all $\psi_1 \mathbf{R} \psi_2 \in \text{cl}(\varphi)$: $\psi_1, \psi_2 \in B$ implies $\psi_1 \mathbf{R} \psi_2 \in B$.
- (C6) For all $\mathbf{F}_{\leq x} \psi_1 \in \text{cl}(\varphi)$: $\psi_1 \in B$ implies $\mathbf{F}_{\leq x} \psi_1 \in B$.
- (C7) For all $\mathbf{G}_{\leq y} \psi_1 \in \text{cl}(\varphi)$: $\mathbf{G}_{\leq y} \psi_1 \in B$ implies $\psi_1 \in B$.

These conditions capture the local properties of the semantics of PLTL. The non-local properties are captured by the transition relation of the automaton we are about to define. The set of consistent subsets is denoted by $\mathcal{C}(\varphi)$.

Let us denote the set of parameterized subformulae of φ by $\text{cl}_p(\varphi)$. The states of our automaton are pairs (B, c) where $B \in \mathcal{C}(\varphi)$ and $c: \text{cl}_p(\varphi) \rightarrow \mathbb{N} \cup \{\perp\}$. The counter $c(\mathbf{F}_{\leq x} \psi_1)$ is used to verify that every position at which $\mathbf{F}_{\leq x} \psi_1$ is guessed to be satisfied, is followed after exactly $\alpha(x)$ steps by a position at which ψ_1 is guessed to be satisfied (and no position in between at which ψ_1 is guessed to be satisfied). Dually, $c(\mathbf{G}_{\leq y} \psi_1)$ is used to verify that

⁷Note that the set of conditions (C1) - (C11) is not minimal. But, for the sake of exposition, we prefer to work with these redundant conditions.

at the next $\alpha(y)$ positions ψ_1 is guessed to be satisfied, whenever $\mathbf{G}_{\leq y} \psi_1$ is guessed to be satisfied. The value \perp denotes that a counter is inactive. A pair (B, c) is consistent, if the following properties are satisfied:

- (C8) For all $\mathbf{F}_{\leq x} \psi_1 \in \text{cl}(\varphi)$: $\psi_1 \in B$ if and only if $c(\mathbf{F}_{\leq x} \psi_1) = 0$.
- (C9) For all $\mathbf{F}_{\leq x} \psi_1 \in \text{cl}(\varphi)$: $\mathbf{F}_{\leq x} \psi_1 \in B$ if and only if $c(\mathbf{F}_{\leq x} \psi_1) \neq \perp$.
- (C10) For all $\mathbf{G}_{\leq y} \psi_1 \in \text{cl}(\varphi)$: $\mathbf{G}_{\leq y} \psi_1 \in B$ if and only if $c(\mathbf{G}_{\leq y} \psi_1) = \alpha(y)$.
- (C11) For all $\mathbf{G}_{\leq y} \psi_1 \in \text{cl}(\varphi)$: $\psi_1 \in B$ if and only if $c(\mathbf{G}_{\leq y} \psi_1) \neq \perp$.

These conditions capture the relation between a parameterized subformula and its associated counter: (C8) requires the counter for the formula $\mathbf{F}_{\leq x} \psi_1$ to be zero if and only if the formula ψ_1 is guessed to be satisfied while (C9) requires the counter to be active if and only if the formula $\mathbf{F}_{\leq x} \psi_1$ is guessed to be satisfied. In this situation, the counter will be decremented in each step until it reaches value zero. At such a position, ψ_1 has to be guessed to be satisfied due to the first condition. The requirements on the counters for parameterized always operator are dual: if $\mathbf{G}_{\leq y} \psi_1$ is guessed to be satisfied, then the counter has to have value $\alpha(y)$ and is decremented in each step until it reaches value zero. Furthermore, the formula ψ_1 has to be guessed to be satisfied at every position at which the counter is active. This ensures that ψ_1 is satisfied for $\alpha(y)$ consecutive positions. Decrementing the counters is implemented in the transition relation.

Finally, given a variable valuation α , we say that a pair (B, c) is α -bounded, if we have:

- (C12) For all $\mathbf{F}_{\leq x} \psi_1 \in \text{cl}(\varphi)$: $c(\mathbf{F}_{\leq x} \psi_1) \neq \perp$ implies $c(\mathbf{F}_{\leq x} \psi_1) \leq \alpha(x)$.
- (C13) For all $\mathbf{G}_{\leq y} \psi_1 \in \text{cl}(\varphi)$: $c(\mathbf{G}_{\leq y} \psi_1) \neq \perp$ implies $c(\mathbf{G}_{\leq y} \psi_1) \leq \alpha(y)$.

Construction 19. Given a PLTL formula φ and a variable valuation α , we define the generalized Büchi automaton $\mathfrak{A}_{\varphi, \alpha} = (Q, 2^P, Q_0, \Delta, \mathcal{F})$ with the following components.

- Q is the set of pairs (B, c) , where $B \in \mathcal{C}(\varphi)$ and $c: \text{cl}_p(\varphi) \rightarrow \mathbb{N} \cup \{\perp\}$, such that (B, c) satisfies (C1) up to (C13).
- $Q_0 = \{(B, c) \in Q \mid \varphi \in B\}$.
- $((B, c), a, (B', c')) \in \Delta$ if and only if
 - (T1) $B \cap P = a$,
 - (T2) $\mathbf{X} \psi_1 \in B$ if and only if $\psi_1 \in B'$,

- (T3) $\psi_1 \mathbf{U} \psi_2 \in B$ if and only if $\psi_2 \in B$ or $(\psi_1 \in B$ and $\psi_1 \mathbf{U} \psi_2 \in B')$,
- (T4) $\psi_1 \mathbf{R} \psi_2 \in B$ if and only if $\psi_2 \in B$ and $(\psi_1 \in B$ or $\psi_1 \mathbf{R} \psi_2 \in B')$,
- (T5) if $\alpha(x) > 0$ and $c(\mathbf{F}_{\leq x} \psi_1) = \perp$, then $c'(\mathbf{F}_{\leq x} \psi_1) \in \{\alpha(x), \perp\}$,
- (T6) if $\alpha(x) > 0$ and $c(\mathbf{F}_{\leq x} \psi_1) > 0$, then $c'(\mathbf{F}_{\leq x} \psi_1) = c(\mathbf{F}_{\leq x} \psi_1) - 1$,
- (T7) if $\alpha(y) > 0$ and $c(\mathbf{G}_{\leq y} \psi_1) = 0$, then $c'(\mathbf{G}_{\leq y} \psi_1) = \perp$,
- (T8) if $\alpha(y) > 0$ and $0 < c(\mathbf{G}_{\leq y} \psi_1) < \alpha(y)$, then $c'(\mathbf{G}_{\leq y} \psi_1) = c(\mathbf{G}_{\leq y} \psi_1) - 1$, and
- (T9) if $\alpha(y) > 0$ and $c(\mathbf{G}_{\leq y} \psi_1) = \alpha(y)$, then $\alpha(y) - 1 \leq c'(\mathbf{G}_{\leq y} \psi_1) \leq \alpha(y)$.

• $\mathcal{F} = \mathcal{F}_{\mathbf{U}} \cup \mathcal{F}_{\mathbf{R}}$ where

- $\mathcal{F}_{\mathbf{U}} = \{F_{\psi_1 \mathbf{U} \psi_2} \mid \psi_1 \mathbf{U} \psi_2 \in \text{cl}(\varphi)\}$ with $F_{\psi_1 \mathbf{U} \psi_2} = \{(B, c) \in Q \mid \psi_1 \mathbf{U} \psi_2 \notin B \text{ or } \psi_2 \in B\}$, and
- $\mathcal{F}_{\mathbf{R}} = \{F_{\psi_1 \mathbf{R} \psi_2} \mid \psi_1 \mathbf{R} \psi_2 \in \text{cl}(\varphi)\}$ with $F_{\psi_1 \mathbf{R} \psi_2} = \{(B, c) \in Q \mid \psi_1 \mathbf{R} \psi_2 \in B \text{ or } \psi_2 \notin B\}$.

Let us explain the definition of Δ : the conditions (T1) up to (T4) are standard for LTL and reflect the semantics of these operators. Hence, we focus on the latter conditions for the parameterized operators. So, consider a formula $\mathbf{F}_{\leq x} \psi_1$ with $\alpha(x) > 0$. If the counter for this subformula is inactive, then the counter is either also inactive at the next position, or it is started with value $\alpha(x)$ (which means $\mathbf{F}_{\leq x} \psi_1$ is guessed to be satisfied). In the second case, ψ_1 has to be guessed true after exactly $\alpha(x)$ steps. This is captured by (T5). If the counter is active, but not zero, then it is decremented in the next step, which is captured by (T6). Finally, if the counter is zero (which is equivalent to ψ_1 is guessed to be satisfied at the current position, due to (C8)), then it is zero in the next step (ψ_1 is guessed to be satisfied in the next step as well), inactive, or can be restart with any value k (meaning that ψ_1 is guessed to be satisfied for the next time in exactly k) positions. Hence, there is no requirement on the counter in this case, any value is allowed. Note that we require a counter to start with value $\alpha(x)$ after it is \perp in the previous step, i.e., $\mathbf{F}_{\leq x} \psi_1$ has to be guessed to be satisfied as soon as possible. This property is crucial to obtain an unambiguous automaton.

The conditions for formulae $\mathbf{G}_{\leq y} \psi_1$ are dual: as long as $\mathbf{G}_{\leq y} \psi_1$ is guessed to be satisfied, $c(\mathbf{G}_{\leq y} \psi_1)$ has value $\alpha(y)$ due to (C10). Beginning at the first position where $\mathbf{G}_{\leq y} \psi_1$ is no longer guessed to hold, the counter has to be

decremented in each step (due to (T8) and (T9)) and checks that the next $\alpha(y)$ positions satisfy ψ_1 due to (C11). Since the counter has to be inactive after it has reached value zero (due to (T7)), the automaton cannot start the decrement phase too early or too late.

The requirements on c and c' in the definition are only phrased for parameterized formulae with variable z such that $\alpha(z) > 0$. This is because $\mathbf{F}_{\leq x} \psi_1$ and $\mathbf{G}_{\leq y} \psi_1$ are both equivalent to ψ_1 if we have $\alpha(x) = 0$ or $\alpha(y) = 0$, respectively. This is modeled by the fact that we have $c(\mathbf{F}_{\leq x} \psi_1) \in \{0, \perp\}$ for such a formula. Hence, the consistency properties make sure that we have $\mathbf{F}_{\leq x} \psi_1 \in B$ if and only if $\psi_1 \in B$ and $\mathbf{G}_{\leq y} \psi_1 \in B$ if and only if $\psi_1 \in B$.

Example 20. Consider the subformulae $\mathbf{F}_{\leq x} p$ and $\mathbf{G}_{\leq y} q$ of some formula φ and the variable valuation α with $\alpha(x) = 2$ and $\alpha(y) = 3$. Table 2 shows how the counters evolve during a run of the generalized Büchi automaton $\mathfrak{A}_{\varphi, \alpha}$.

w	$\{q\}$	$\{p, q\}$	\emptyset	$\{p, q\}$	$\{q\}$	$\{q\}$	$\{q\}$	$\{p, q\}$	\emptyset^ω
$p \in B$	n	y	n	y	n	n	n	y	n
$\mathbf{F}_{\leq x} p \in B$	y	y	y	y	n	y	y	y	n
$c(\mathbf{F}_{\leq x} p)$	1	0	1	0	\perp	2	1	0	\perp
$q \in B$	y	y	n	y	y	y	y	y	n
$\mathbf{G}_{\leq y} q \in B$	n	n	n	y	y	n	n	n	n
$c(\mathbf{G}_{\leq y} q)$	1	0	\perp	3	3	2	1	0	\perp

Figure 2: A run of the automaton $\mathfrak{A}_{\varphi, \alpha}$

Lemma 21. *Let φ be a PLTL formula, let α be a variable valuation, and let $\mathfrak{A}_{\varphi, \alpha}$ be the automaton obtained in Construction 19. Then,*

1. $L(\mathfrak{A}_{\varphi, \alpha}) = \{w \in (2^P)^\omega \mid (w, \alpha) \models \varphi\}$,
2. $\mathfrak{A}_{\varphi, \alpha}$ is unambiguous, and
3. $|\mathfrak{A}_{\varphi, \alpha}| \leq 2^{|\varphi|} \cdot (\max_{z \in \text{var}(\varphi)} \alpha(z) + 2)^{|\varphi|}$ and $|\mathcal{F}| < |\varphi|$.

Proof. 1.) First, we show $L(\mathfrak{A}_{\varphi, \alpha}) \subseteq \{w \in (2^P)^\omega \mid (w, \alpha) \models \varphi\}$. Thus, let $(B_0, c_0)(B_1, c_1)(B_2, c_2) \cdots$ be an accepting run of $\mathfrak{A}_{\varphi, \alpha}$ on w . A straightforward structural induction over the construction of φ shows that $\psi \in B_n$ if and only if $(w, n, \alpha) \models \psi$. This suffices to show $(w, \alpha) \models \varphi$, since we have $\varphi \in B_0$ by definition of Q_0 .

Now, let us prove the second inclusion $\{w \in (2^P)^\omega \mid (w, \alpha) \models \varphi\} \subseteq L(\mathfrak{A}_{\varphi, \alpha})$. Let $(w, \alpha) \models \varphi$ and define for each n

$$B_n = \{\psi \in \text{cl}(\varphi) \mid (w, n, \alpha) \models \psi\}$$

to be the set of subformulae that are satisfied at position n with respect to α . Now, we define a counter c_n for each n : for every $\mathbf{F}_{\leq x} \psi_1 \in \text{cl}_p(\varphi)$ let

$$c_n(\mathbf{F}_{\leq x} \psi_1) = \min\{k \mid 0 \leq k \leq \alpha(x) \text{ and } (w, n+k, \alpha) \models \psi_1\} ,$$

where we set $\min \emptyset = \perp$, be the minimal waiting times for the parameterized eventually operators at position n . Dually, for every $\mathbf{G}_{\leq y} \psi_1 \in \text{cl}_p(\varphi)$ let

$$c_n(\mathbf{G}_{\leq y} \psi_1) = \max\{k \mid 0 \leq k \leq \alpha(y) \text{ and } (w, n+j, \alpha) \models \psi_1 \text{ for every } j \leq k\} ,$$

where we set $\max \emptyset = \perp$, be the maximal satisfaction times for the parameterized always operators at position n .

It is again straightforward to show that $(B_0, c_0)(B_1, c_1)(B_2, c_2) \cdots$ is an accepting run of $\mathfrak{A}_{\varphi, \alpha}$. The semantics of PLTL guarantees that each set B_n and each pair (B_n, c_n) is consistent and the c_n are α -bounded by definition. Thus, each pair (B_n, c_n) is a state and we have $(B_0, c_0) \in Q_0$ due to $(w, 0, \alpha) \models \varphi$. The semantics of PLTL also guarantee that we have $((B_n, c_n), w_n, (B_{n+1}, c_{n+1})) \in \Delta$ for every n , i.e., $(B_0, c_0)(B_1, c_1)(B_2, c_2) \cdots$ is a run of $\mathfrak{A}_{\varphi, \alpha}$ on w . It is accepting due to the semantics of PLTL, which guarantee that each $F_{\psi_1 \cup \psi_2}$ and each $F_{\psi_1 \mathbf{R} \psi_2}$ is visited infinitely often.

2.) Assume $\mathfrak{A}_{\varphi, \alpha}$ has two accepting runs $(B_0, c_0)(B_1, c_1)(B_2, c_2) \cdots$ and $(B'_0, c'_0)(B'_1, c'_1)(B'_2, c'_2) \cdots$ on a word w , i.e., there is a position n such that $B_n \neq B'_n$ or $c_n \neq c'_n$. We have shown above that we have $\psi \in B_n$ if and only if $(w, n, \alpha) \models \psi$. Since the same holds true for the sets B'_n , we conclude $B_n = B'_n$ for every n . This leaves us with $c_n \neq c'_n$. The consistency requirements (C9) and (C11) and the fact $B_n = B'_n$ imply $c_n(\psi) = \perp$ if and only if $c'_n(\psi) = \perp$ for every parameterized formula $\psi \in \text{cl}_p(\varphi)$. Hence, we must have $c_n(\psi) = k \neq k' = c'_n(\psi)$ for some $\psi \in \text{cl}_p(\varphi)$.

First, we consider the case $\psi = \mathbf{F}_{\leq x} \psi_1$, which implies $0 \leq k, k' \leq \alpha(x)$. We assume without loss of generality $k < k'$. Then, applying (T6) inductively yields $c_{n+j}(\mathbf{F}_{\leq x} \psi_1) = k - j$ for every j in the range $0 \leq j \leq k$ and similarly $c'_{n+j}(\mathbf{F}_{\leq x} \psi_1) = k' - j$ for every j in the range $0 \leq j \leq k'$. Hence, we have $c_{n+k}(\mathbf{F}_{\leq x} \psi_1) = 0$, which implies $\psi_1 \in B_{n+k}$ due to (C8). On the other hand,

we have $c'_{n+k}(\mathbf{F}_{\leq x} \psi_1) = k' - k > 0$, which implies $\psi_1 \notin B'_{n+k}$, again due to (C8). This yields the desired contradiction, since we have $B_{n+k} = B'_{n+k}$.

Now, we consider the case $\psi = \mathbf{G}_{\leq y} \psi_1$, which implies $0 \leq k, k' \leq \alpha(y)$. Again, we assume without loss of generality $k < k'$. Then, applying (T6) inductively yields $c_{n+j}(\mathbf{G}_{\leq y} \psi_1) = k - j$ for every j in the range $0 \leq j \leq k$, since we have $k < \alpha(y)$. Hence, we have $c_{n+k}(\mathbf{G}_{\leq y} \psi_1) = 0$, which implies $c_{n+k+1}(\mathbf{G}_{\leq y} \psi_1) = \perp$ due to (T7). Thus, we have $\psi_1 \notin B_{n+k+1}$ due to (C11).

Similarly, we have $c'_{n+j}(\mathbf{G}_{\leq y} \psi_1) \geq k' - j$ for every j in the range $0 \leq j \leq k'$ due to (T8) and (T9). Thus, $c'_{n+k}(\mathbf{G}_{\leq y} \psi_1) \geq k' - k > 0$. Condition (T8) implies $c'_{n+k+1}(\mathbf{G}_{\leq y} \psi_1) \geq 0$, which in turn implies $\psi_1 \in B'_{n+k+1}$ due to (C11). Again, we have derived a contradiction, due to $B_{n+k+1} = B'_{n+k+1}$.

3.) The number of consistent subsets is bounded by $2^{|\varphi|}$ and we have $c(\mathbf{F}_{\leq x} \psi_1) \in \{0, \dots, \alpha(x)\} \cup \{\perp\}$ and $c(\mathbf{G}_{\leq y} \psi_1) \in \{0, \dots, \alpha(y)\} \cup \{\perp\}$, if c is α -bounded. This yields the desired upper bound on $|\mathfrak{A}_{\varphi, \alpha}|$. Finally, $|\mathcal{F}| < |\varphi| = |\text{cl}(\varphi)|$, as $\text{cl}(\varphi)$ contains at least one atomic proposition. \square

Using a standard construction to turn a generalized Büchi automaton into a Büchi automaton [16] (which preserves unambiguity) and the determinization of Morgenstern and Schneider [17], which is applicable to unambiguous Büchi automata, we obtain a deterministic parity automaton with the required properties.

Theorem 22. *Let φ be a PLTL formula and let α be a variable valuation. We denote $\max_{z \in \text{var}(\varphi)} \alpha(z) + 2$ by m . There exists a deterministic parity automaton $\mathfrak{P}_{\varphi, \alpha}$ of size $|\mathfrak{P}_{\varphi, \alpha}| \leq 2^{\mathcal{O}(|\varphi|^2 \cdot (2m)^{2|\varphi|})}$ with $2 \cdot |\varphi| \cdot (2m)^{|\varphi| + 1}$ priorities and $L(\mathfrak{P}_{\varphi, \alpha}) = \{w \in (2^P)^\omega \mid (w, \alpha) \models \varphi\}$.*

Note that this proves Lemma 18, since we can construct the automaton $\mathfrak{P}_{\varphi, \alpha}$ in triply-exponential time. Furthermore, we can use the automaton $\mathfrak{P}_{\varphi, \alpha}$ to solve the membership problem for the set of winning valuations in doubly-exponential time: $\mathfrak{P}_{\varphi, \alpha}$ has doubly exponential size and has exponentially many colors (both measured in $|\varphi| + |\alpha|$), hence we can apply Lemma 2 to reduce the membership problem to a parity game which can be solved in doubly-exponential time.

Corollary 23. *The membership problem for PLTL games is **2EXPTIME**-complete.*

$$\underbrace{000}_{c_0} 0' \quad \underbrace{001}_{c_1} 1' \quad \underbrace{010}_{c_2} 1' \quad \underbrace{011}_{c_3} 0' \quad \underbrace{100}_{c_4} 0' \quad \underbrace{101}_{c_5} 1' \quad \underbrace{110}_{c_6} 0' \quad \underbrace{111}_{c_7} 1'$$

Figure 3: Binary encoding of the counter value 101 with positions in front of each bit

6. Lower Bounds on Optimal Variable Valuations

A consequence of our algorithm for the $\text{PLTL}_{\mathbf{F}}$ emptiness problem is a $2^{2^{\mathcal{O}(|\mathcal{G}|)}}$ upper bound on optimal variable valuations which allow Player 0 to win a $\text{PLTL}_{\mathbf{F}}$ game \mathcal{G} . In this section, we present a $2^{2^{\sqrt{|\mathcal{G}|}}}$ lower bound.

Theorem 24. *For every $n \geq 1$, there exists a PROMPT-LTL game \mathcal{G}_n with winning condition φ_n with $|\mathcal{G}_n| \in \mathcal{O}(n^2)$ and $\text{var}(\varphi_n) = \{x\}$ such that $\mathcal{W}_0(\mathcal{G}_n) \neq \emptyset$, but Player 1 wins \mathcal{G}_n with respect to every variable valuation α such that $\alpha(x) \leq 2^{2^n}$.*

The proof idea is to encode a binary counter with range $\{0, \dots, 2^{2^n} - 1\}$ and require Player 0 to satisfy some obligation expressed by a parameterized eventually operator, but only after the counter has reached value $2^{2^n} - 1$. Player 1 is in charge of maintaining the counter and Player 0 has to check whether Player 1 increments the counter correctly. If he does not, Player 0 may fulfill her obligation earlier. Hence, by always incrementing correctly, Player 1 is able to prevent Player 0 from satisfying the parameterized eventually operator in less than 2^{2^n} steps, but not longer than that.

We denote the *values* the counter assumes by d_0, d_1, d_2, \dots and encode each of them by 2^n bits. To enable a small winning condition to check the faulty increment claimed by Player 0, Player 1 has to precede every bit of d_ℓ by a binary representation of its position $c_j \in [2^n]$, which is of length n . Figure 3 shows an example: the primed bits constitute the binary representation of the counter value $d_\ell = 101$, and each bit is preceded by its position (here we have $n = 3$).

Player 0 can mark the position of a single bit of some $d_{\ell+1}$ to claim $d_{\ell+1} \neq d_\ell + 1$. Using the addresses c_j , the winning condition can verify whether the claim is correct or not. This idea is formalized in the following.

We begin the proof of Theorem 24 by fixing an $n \geq 1$. The arena \mathcal{A}_n is depicted in Figure 4. The trace w of a play in \mathcal{A}_n starting in the initial

3. Reset: if a number c_j is equal to $2^n - 1$, then it is followed by a dollar.

If w satisfies the conjunction ψ_1 of these three requirements and the formula $\mathbf{GF}\$,$ then infinitely many dollars occur in w and there are exactly 2^n primed bits between each adjacent pair of dollars. Hence, we have $0 \leq d_\ell \leq 2^{2^n} - 1$ for every ℓ . Furthermore, we can construct ψ_1 such that its size is quadratic in n .

All previous formulae express requirements on Player 1's behavior since he is in charge of producing the c_j and d_ℓ . Now, we turn our attention to Player 0. Her only task is to decide whether to move to f_0 or to f_1 , thereby producing a position at which the proposition f holds or does not hold. The formula $\psi_f = \mathbf{G}(f \rightarrow \mathbf{XG} \neg f)$ expresses that there is at most one position at which f holds. As last formula, consider $\psi_{\text{err}} = \mathbf{F} \psi'_{\text{err}}$ where⁸

$$\begin{aligned} \psi'_{\text{err}} = & s \wedge (\neg \$ \mathbf{U} (\$ \wedge \neg \$ \mathbf{U} f)) \wedge \\ & \mathbf{X} \left(\bigwedge_{j=0}^{n-1} (\mathbf{X}^j 0 \leftrightarrow \mathbf{F}(0 \wedge \mathbf{X}^{(n-j)+2} f)) \right) \wedge (\psi_{\text{max}} \vee \psi_{\text{faulty-inc}}) \end{aligned}$$

where $\psi_{\text{max}} = \neg \$ \mathbf{U} (\$ \wedge (\neg 0' \mathbf{U} \$))$ and

$$\begin{aligned} \psi_{\text{faulty-inc}} = & [(\neg 0' \mathbf{U} \$) \rightarrow \mathbf{F}(1' \wedge \mathbf{X}^2 f)] \wedge \\ & [((\neg 0' \wedge \neg 1') \mathbf{U} (0' \wedge (\neg 0' \mathbf{U} \$))) \rightarrow \mathbf{F}(0' \wedge \mathbf{X}^2 f)] \wedge \\ & [((\neg 0' \wedge \neg 1') \mathbf{U} (0' \wedge (\neg \$ \mathbf{U} 0')))] \rightarrow \mathbf{F}(1' \wedge \mathbf{X}^2 f)] \wedge \\ & [((\neg 0' \wedge \neg 1') \mathbf{U} (1' \wedge (\neg \$ \mathbf{U} 0')))] \rightarrow \mathbf{F}(0' \wedge \mathbf{X}^2 f)] \quad . \end{aligned}$$

Let us dissect the formula ψ_{err} : assume we have $(w, \alpha) \models \psi_{\text{err}}$, i.e., there exists a position m such that $(w, m, \alpha) \models \psi'_{\text{err}}$. At this position, s holds true. Hence, the next n positions encode a number c_j . Furthermore, after the next dollar, f holds true at least once before the next dollar occurs. If we assume ψ_f to be satisfied by w , then this is the only f occurring in w . This f is preceded by another sequence of bits which encode a number $c_{j'}$. The next subformula of ψ'_{err} requires the values c_j and $c_{j'}$ to be equal: here, we use the fact that at these positions either 0 or 1 holds, but not both at the same time. Hence, the next primed bit after the position n is the c_j -th bit of some number d_ℓ and the primed bit two positions before the position

⁸Here, we use \mathbf{X}^k as shorthand for k nested next operators.

at which f holds is the c_j -th bit of $d_{\ell+1}$. The final disjunction is satisfied, if these primed bits witness $d_{\ell+1} \neq d_\ell + 1$ (by the disjunct $\psi_{\text{faulty-inc}}$) or if we have $d_{\ell+1} = 2^{2^n} - 1$ (by the disjunct ψ_{max}): ψ_{max} is satisfied, if there is no primed zero between the next two dollars, which implies $d_{\ell+1} = 2^{2^n} - 1$, since we still assume ψ_1 to be satisfied. Now consider the conjuncts of $\psi_{\text{faulty-inc}}$: the first one is satisfied if the bits right of (including) the c_j -th one of d_ℓ are all ones, but the c_j -th bit of $d_{\ell+1}$ is not flipped to zero. The second one is satisfied if the bits right of (excluding) the c_j -th one of d_ℓ , which is zero, are all ones, but the c_j -th bit of $d_{\ell+1}$ is not flipped to one. The last two formulae are symmetric, thus we only explain the third one: it is satisfied, if the c_j -th bit of d_ℓ is a zero and is followed by another zero before the next dollar occurs, and the c_j -th bit of $d_{\ell+1}$ is flipped. Thus, ψ'_{err} is indeed satisfied at a position m if the next primed bit and the primed bit before the (only) occurrence of f witness $d_{\ell+1} \neq d_\ell + 1$ or if we have $d_{\ell+1} = 2^{2^n} - 1$.

Let us wrap things up and prove Theorem 24.

Proof. Consider the game $\mathcal{G}_n = (\mathcal{A}_n, d_1, \varphi)$ with

$$\varphi = \psi_1 \rightarrow (\psi_f \wedge \psi_{\text{err}} \wedge \mathbf{F}_{\leq x} f) .$$

The arena has $2n + 8$ vertices and the size of φ_n is quadratic in n .

Next, we show that $\mathcal{W}_0(\mathcal{G}_n)$ is non-empty. Let w be the trace of a play of \mathcal{G}_n . If it does not satisfy ψ_1 , then it is winning for Player 0. So, assume we have $w \models \psi_1$. Then, w has the form as described above: the c_j 's count from 0 to $2^n - 1$ and the numbers d_ℓ are in the range $0 \leq d_\ell \leq 2^{2^n} - 1$. In this situation, Player 0 has to ensure that f holds exactly once, at a position as described above: either after $d_\ell = 2^{2^n} - 1$ or after a primed bit that witnesses a faulty increase by Player 1. Player 0 is always able to find such a position since Player 1 can produce at most 2^{2^n} numbers d_ℓ without introducing a faulty increment. Hence, Player 0 wins \mathcal{G}_n with respect to some α . On the other hand, by always incrementing the d_ℓ correctly until they reach $2^{2^n} - 1$, Player 1 is able to win \mathcal{G}_n with respect to (at least) every α such that $\alpha(x) \leq n \cdot 2^n \cdot 2^{2^n}$, since there are 2^{2^n} values for d , each having 2^n bits which are encoded by one round through the arena, each of which visits more than n vertices. \square

7. Conclusion

We have shown the membership, emptiness, finiteness, and universality problem for PLTL games to be **2EXPTIME**-complete. Thus, these problems

are not harder than solving LTL games without parameterized operators. Furthermore, all but the finiteness problem for PLTL_G games can be reduced to solving a single LTL game.

This has to be contrasted with the status of the PLTL optimization problems for which we presented an algorithm with triply-exponential running time. It is open whether the optimization problems for games can also be solved in doubly-exponential time. We have complemented our algorithm for these problems by a doubly-exponential lower bound on the value of an optimal variable valuation for a unipolar game.

A challenging open problem concerns the memory requirements of winning strategies realizing optimal variable valuations: these strategies are finite-state, but in some cases being optimal requires more memory than just being winning. The exact tradeoff between quality and size of a winning strategy remains to be investigated. Note that this question is very general and can be posed for many other winning conditions with an induced quality measure as well.

Finally, we propose to investigate the following variant of PLTL games: according to our definition, the emptiness problem for PLTL games asks whether there *exists* a strategy σ and a variable valuation α such that *every* play that is consistent with σ is a model of the winning condition with respect to α , i.e., the order of quantifiers is $\exists\sigma\exists\alpha\forall\rho$. If we change the order to $\exists\sigma\forall\rho\exists\alpha$, we ask whether there is a strategy such that the winning condition is satisfied on every consistent play, but with a variable valuation that may depend on the play. Thus, instead of guaranteeing uniform bounds for all plays consistent with a strategy, Player 0 only has to guarantee some bound on each play. This non-uniform variant of PLTL games is reminiscent of finitary objectives [18].

Acknowledgments. I want to thank Marcin Jurdziński, Andreas Morgenstern, and Wolfgang Thomas for helpful discussions, and Roman Rabinovich for coming up with the name *blinking semantics*. Furthermore, I am very grateful to Christof Löding for a fruitful discussion which resulted in Theorem 24.

References

- [1] H. W. Kamp, Tense Logic and the Theory of Linear Order, Ph.D. thesis, Computer Science Department, University of California at Los Angeles, USA, 1968.

- [2] A. P. Sistla, E. M. Clarke, The complexity of propositional linear temporal logics, *J. ACM* 32 (1985) 733–749.
- [3] R. Koymans, Specifying real-time properties with metric temporal logic, *Real-Time Systems* 2 (1990) 255–299.
- [4] R. Alur, T. A. Henzinger, Real-time logics: Complexity and expressiveness, *Inf. Comput.* 104 (1993) 35–77.
- [5] R. Alur, K. Etessami, S. L. Torre, D. Peled, Parametric temporal logic for “model measuring”, *ACM Trans. Comput. Log.* 2 (2001) 388–407.
- [6] O. Kupferman, N. Piterman, M. Y. Vardi, From liveness to promptness, *Formal Methods in System Design* 34 (2009) 83–103.
- [7] B. D. Giampaolo, S. L. Torre, M. Napoli, Parametric metric interval temporal logic, in: A. H. Dediú, H. Fernau, C. Martín-Vide (Eds.), *LATA*, volume 6031 of *LNCS*, Springer, 2010, pp. 249–260.
- [8] A. Pnueli, R. Rosner, On the synthesis of an asynchronous reactive module, in: G. Ausiello, M. Dezani-Ciancaglini, S. R. D. Rocca (Eds.), *ICALP*, volume 372 of *LNCS*, Springer, 1989, pp. 652–671.
- [9] M. Zimmermann, Optimal bounds in parametric LTL games, in: G. D’Agostino, S. L. Torre (Eds.), *GandALF*, volume 54 of *EPTCS*, pp. 146–161.
- [10] E. A. Emerson, C. S. Jutla, Tree automata, mu-calculus and determinacy (extended abstract), in: *FOCS*, IEEE, 1991, pp. 368–377.
- [11] A. Mostowski, Games with Forbidden Positions, Technical Report 78, University of Gdańsk, 1991.
- [12] M. Jurdziński, Small progress measures for solving parity games, in: H. Reichel, S. Tison (Eds.), *STACS*, volume 1770 of *LNCS*, Springer, 2000, pp. 290–301.
- [13] E. Grädel, W. Thomas, T. Wilke (Eds.), Automata, Logics, and Infinite Games: A Guide to Current Research, volume 2500 of *LNCS*, Springer, 2002.

- [14] A. Pnueli, R. Rosner, On the synthesis of a reactive module, in: POPL, pp. 179–190.
- [15] R. Rosner, Modular Synthesis of Reactive Systems, Ph.D. thesis, Weizmann Institute of Science, 1991.
- [16] C. Baier, J.-P. Katoen, Principles of Model Checking, The MIT Press, 2008.
- [17] A. Morgenstern, K. Schneider, From LTL to symbolically represented deterministic automata, in: F. Logozzo, D. Peled, L. D. Zuck (Eds.), VMCAI, volume 4905 of *LNCS*, Springer, 2008, pp. 279–293.
- [18] K. Chatterjee, T. A. Henzinger, F. Horn, Finitary winning in omega-regular games, *ACM Trans. Comput. Log.* 11 (2009).