Optimal Bounds in Parametric LTL Games *

Martin Zimmermann
Lehrstuhl Informatik 7
RWTH Aachen University, Germany
zimmermann@automata.rwth-aachen.de

We consider graph games of infinite duration with winning conditions in parameterized linear temporal logic, where the temporal operators are equipped with variables for time bounds. In model checking such specifications were introduced as "PLTL" by Alur et al. and (in a different version called "PROMPT-LTL") by Kupferman et al..

We present an algorithm to determine optimal variable valuations that allow a player to win a game. Furthermore, we show how to determine whether a player wins a game with respect to some, infinitely many, or all valuations. All our algorithms run in doubly-exponential time; so, adding bounded temporal operators does not increase the complexity compared to solving plain LTL games.

1 Introduction

Many of todays problems in computer science are no longer concerned with programs that transform data and then terminate, but with non-terminating systems. Model-checking, the automated verification of closed systems (those that do not have to interact with an environment), is nowadays routinely performed in industrial settings. For open system (those that have to interact with a possibly antagonistic environment), the framework of infinite two-player games is a powerful and flexible tool to verify and synthesize such systems. A crucial aspect of automated verification is the choice of a specification formalism, which should be simple enough to be used by practitioners without formal training in automata theory or logics. Here, *Linear Temporal Logic* (LTL) has turned out to be an expressive, but easy to use formalism: its advantages include a compact, variable-free syntax and intuitive semantics. For example, the specification "every request q is answered by a response p" is expressed by $\varphi = \mathbf{G}(q \rightarrow \mathbf{F}p)$.

However, LTL lacks capabilities to express timing constraints, e.g., it cannot express that every request is answered within an unknown, but fixed number of steps. Also, in an infinite game with winning condition φ, Player 0 might have two winning strategies, one that answers every request within m steps, and another one that takes n steps, for some $n > m$. The first strategy is clearly preferable to the second one, but there is no guarantee that the first one is indeed computed, when the game is solved.

To overcome these shortcomings, several parameterized temporal logics [1, 4, 6] where introduced for the verification of closed systems: here one adds parametric bounds on the temporal operators. We are mainly concerned with *Parametric Linear Temporal Logic* (PLTL) [1], 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 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., deciding whether a transition system

*The author's work was supported by the project *Games for Analysis and Synthesis of Interactive Computational Systems (GASICS)* of the *European Science Foundation*.

Giovanna D'Agostino, Salvatore La Torre (Eds.):
Proceedings of the Second International Symposium on
"Games, Automata, Logics and Formal Verification" (GandALF 2011)
EPTCS 54, 2011, pp. 146–161, doi:10.4204/EPTCS.54.11

satisfies a PLTL formula with respect to some, infinitely many, or all variable valuations is **PSPACE**-complete [1], as is LTL model-checking [13]. Similar results hold for parameterized real-time logics [4]. Also, for PLTL one can determine optimal variable valuations for which a formula is satisfied by a given transition system in polynomial space.

In this work, we consider infinite games with winning conditions in PLTL, i.e., we lift the results on model-checking parameterized specifications to synthesis of open systems from parameterized specifications. Our starting point is a result on the fragment of PLTL containing only parameterized eventualities, which was discussed (in a different version called PROMPT–LTL) in [6]: there, the authors show that the realizability problem (an abstract notion of a game) for PROMPT–LTL is **2EXPTIME**-complete. We use this result to solve infinite games with winning conditions in the full logic with parameterized eventualities and always': determining whether a player wins a PLTL game with respect to some, infinitely many, or all variable valuations is also **2EXPTIME**-complete, as is determining the winner of an LTL game [11]. So, we observe the same phenomenon as in model-checking: the addition of parameterized operators does not increase the computational complexity of the problem.

After establishing these results, we consider the problem of finding optimal variable valuations that allow a given player to win the game. If a winning condition contains only parameterized eventualities or only parameterized always', then it makes sense to ask for an optimal valuation that a player can enforce against her opponent and for a winning strategy realizing the optimum. Our main theorem states that this optimization problem can be solved in doubly-exponential time; so even determining an optimal winning strategy for such a game is of the same computational complexity as solving unparameterized games.

The remainder of this paper is structured as follows: in Section 2, we introduce infinite games with winning conditions in parameterized linear temporal logic and fix our notation. In Section 3, we show how the result on the PROMPT–LTL realizability problem can be used to show that determining whether a player wins a PLTL game with respect to some, infinitely many, or all variable valuations can be decided in doubly-exponential time. In Section 4, we use these results to determine optimal winning strategies in games for which a notion of optimality can be defined. Finally, Section 5 gives a short conclusion.

2 Definitions

The set of non-negative integers is denoted by $\mathbb{N}$, the set of positive integers by $\mathbb{N}_+$. The powerset of a set S is denoted by 2^S. Throughout this paper let P be a set of atomic propositions.

Automata. An (non-deterministic) ω-automaton $\mathfrak{A} = (Q,\Sigma,Q_0,\Delta,\mathrm{Acc})$ consists of a finite set of states Q, an alphabet Σ, a set of initial states $Q_0 \subseteq Q$, a transition relation $\Delta \subseteq Q \times \Sigma \times Q$, and an acceptance condition Acc. An ω-automaton is deterministic, if $|Q_0| = 1$ and for every $(q,a) \in Q \times \Sigma$, there is exactly one q' such that $(q,a,q') \in \Delta$. In this case, we denote $Q_0 = \{q_0\}$ by q_0 and Δ as function $\delta\colon Q \times \Sigma \to Q$. The size of $\mathfrak{A}$, denoted by $|\mathfrak{A}|$, is the cardinality of Q. A run of $\mathfrak{A}$ on an ω-word $w \in \Sigma^\omega$ is an infinite sequence of states $q_0q_1q_2\dots$ such that $q_0 \in Q_0$ and $(q_n,w_n,q_{n+1}) \in \Delta$ for every $n \in \mathbb{N}$. We consider different acceptance conditions Acc for ω-automata: (1) Büchi automata with a set of accepting states $F \subseteq Q$. A run $q_0q_1q_2\dots$ is accepting if there are infinitely many n such that $q_n \in F$. (2) Generalized Büchi automata with a family of sets of accepting states $\mathscr{F} \subseteq 2^Q$. A run $q_0q_1q_2\dots$ is accepting if for every $F \in \mathscr{F}$ there are infinitely many n such that $q_n \in F$. (3) Parity automata with a priority function $c\colon Q \to \mathbb{N}$. A run $q_0q_1q_2\dots$ is accepting, if the minimal priority seen infinitely often is even. An ω-word is accepted by an ω-automaton, if there exists an accepting run on it. The language $L(\mathfrak{A})$ of $\mathfrak{A}$ contains the ω-words accepted by $\mathfrak{A}$. An ω-automaton is called unambiguous, if it has at most one accepting run on every ω-word $w \in \Sigma^\omega$. It is called non-confluent, if for every ω-word w and two runs $q_0q_1q_2\dots$

and $q'_0 q'_1 q'_2 \dots$ on w we have for all n that if $q_n = q'_n$, then $q_m = q'_m$ for every $m < n$. In a non-confluent ω-automaton with n states, every finite prefix of an ω-word has at most n finite runs, all of which can be uniquely identified by their last state. Finally, a state of an ω-automaton is unproductive, if it is not reachable from the initial state or if there is no accepting run starting from this state. Removing all unproductive states from a (generalized) Büchi or parity automaton does not change its language.

Remark 1. *An unambiguous (generalized) Büchi or parity automaton without unproductive states is non-confluent.*

Linear Temporal Logics. Let $\mathscr{X}$ and $\mathscr{Y}$ be two disjoint sets of variables[1]. The formulae of Parametric Linear Temporal Logic (PLTL) [1] are given by the grammar

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

where $p \in P$, $x \in \mathscr{X}$ and $y \in \mathscr{Y}$. Also, 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 $\mathbf{G}\varphi := \mathbf{ff}\mathbf{R}\varphi$ [2]. The set of variables occurring in φ is denoted by $\mathrm{var}(\varphi)$ and defined in the obvious way. The size $|\varphi|$ of a formula φ is measured by counting the distinct subformulae of φ. We consider several fragments of PLTL: φ is an LTL formula, if $\mathrm{var}(\varphi) = \emptyset$; φ is a PROMPT–LTL formula [6], if $\mathrm{var}(\varphi)$ is a subset of $\mathscr{X}$ of cardinality at most one; φ is a $\mathrm{PLTL}_{\mathbf{F}}$ formula, if $\mathrm{var}(\varphi) \subseteq \mathscr{X}$; and φ is a $\mathrm{PLTL}_{\mathbf{G}}$ formula, if $\mathrm{var}(\varphi) \subseteq \mathscr{Y}$. A formula in $\mathrm{PLTL}_{\mathbf{F}}$ or $\mathrm{PLTL}_{\mathbf{G}}$ is called unipolar. The semantics of PLTL is defined with respect to an ω-word $w \in \left(2^P\right)^\omega$, a position $i \in \mathbb{N}$, and a variable valuation $\alpha \colon \mathscr{X} \cup \mathscr{Y} \to \mathbb{N}$ as follows:

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

As the satisfaction of an LTL formula φ is independent of the variable valuation α, we omit α and write $(w,i) \models \varphi$ instead of $(w,i,\alpha) \models \varphi$. PLTL and LTL (but not the fragments PROMPT–LTL, $\mathrm{PLTL}_{\mathbf{F}}$ and $\mathrm{PLTL}_{\mathbf{G}}$) are closed under negation, although we only allow formulae in negation normal form. This is due to the duality of $\mathbf{U}$ and $\mathbf{R}$, and $\mathbf{F}_{\leq x}$ and $\mathbf{G}_{\leq y}$. Thus, we use $\neg\varphi$ as shorthand for the equivalent formula obtained by pushing the negation to the atomic propositions. Note that the negation of a $\mathrm{PLTL}_{\mathbf{F}}$ formula is a $\mathrm{PLTL}_{\mathbf{G}}$ formula and vice versa.

Remark 2. *For every* PLTL *formula φ and every valuation α, there exists an* LTL *formula φ_α such that for every $w \in \left(2^P\right)^\omega$ and every $i \in \mathbb{N}$: $(w,i,\alpha) \models \varphi$ if and only if $(w,i) \models \varphi_\alpha$.*

[1] If the sets of variables are not disjoint, already the model-checking problem for PLTL is undecidable [1].

[2] In [1], the authors also introduced the operators $\mathbf{U}_{\leq x}$, $\mathbf{R}_{\leq y}$, $\mathbf{F}_{>y}$, $\mathbf{G}_{>x}$, $\mathbf{U}_{>y}$, and $\mathbf{R}_{>x}$. However, they showed that all these operators can be expressed using $\mathbf{F}_{\leq x}$ and $\mathbf{G}_{\leq y}$ only, at the cost of a linear increase of the formula's size. Also, we ignore constant bounds as they do not add expressiveness.

This can be shown by replacing the parameterized operators by disjunctions or conjunctions of nested next-operators. The size of φ_α grows linearly in $\sum_{z\in\mathrm{var}(\varphi)} \alpha(z)$. Due to Remark 2, we do not consider a fixed variable valuation when defining games with winning conditions in PLTL, but ask whether Player 0 can win a game with winning condition φ with respect to some, infinitely many, or all valuations.

Infinite Games. An (initialized and labeled) arena $\mathscr{A} = (V, V_0, V_1, E, v_0, \ell)$ consists of a finite directed graph (V,E), a partition $\{V_0, V_1\}$ of V denoting the positions of Player 0 and Player 1, an initial vertex $v_0 \in V$, and a labeling function $\ell\colon V \to 2^P$. The size $|\mathscr{A}|$ of $\mathscr{A}$ is $|V|$. It is assumed that every vertex has at least one outgoing edge. A play $\rho = \rho_0\rho_1\rho_2\dots$ is an infinite path starting in v_0. The trace of ρ is $t(\rho) = \ell(\rho_0)\ell(\rho_1)\ell(\rho_2)\dots$. A strategy for Player i is a mapping $\sigma\colon V^*V_i \to V$ such that $(\rho_n, \sigma(\rho_0\dots\rho_n)) \in E$ for all $\rho_0\dots\rho_n \in V^*V_i$. A play ρ is consistent with σ if $\rho_{n+1} = \sigma(\rho_0\dots\rho_n)$ for all n with $\rho_n \in V_i$.

A parity game $\mathscr{G} = (\mathscr{A}, c)$ consists of an arena $\mathscr{A}$ and a priority function $c\colon V \to \mathbb{N}$. Player 0 wins a play $\rho_0\rho_1\rho_2\dots$ if the minimal priority seen infinitely often is even. The number of priorities of $\mathscr{G}$ is $|c(V)|$. A strategy σ for Player i is winning for her, if every play that is consistent with σ is won by her. Then, we say Player i wins $\mathscr{G}$.

A PLTL game $\mathscr{G} = (\mathscr{A}, \varphi)$ consists of an arena $\mathscr{A}$ and a PLTL formula φ. Player 0 wins a play ρ with respect to a variable valuation α if $(t(\rho), 0, \alpha) \models \varphi$, otherwise Player 1 wins ρ with respect to α. A strategy 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. Then, we say that Player i wins $\mathscr{G}$ with respect to α. We define the set $\mathscr{W}^i_{\mathscr{G}}$ of winning valuations for Player i in $\mathscr{G} = (\mathscr{A}, \varphi)$ by $\mathscr{W}^i_{\mathscr{G}} = \{\alpha \mid \text{Player } i \text{ wins } \mathscr{G} \text{ with respect to } \alpha\}$. Here (and from now on) we assume that α's domain is restricted to the variables occurring in φ. LTL, PROMPT–LTL, $\mathrm{PLTL}_{\mathbf{F}}$, $\mathrm{PLTL}_{\mathbf{G}}$, and unipolar games are defined by restricting the winning conditions to LTL, $\mathrm{PLTL}_{\mathbf{F}}$, $\mathrm{PLTL}_{\mathbf{G}}$, and unipolar formulae. Again, winning an LTL game is independent of α, hence we are justified to say that Player i wins an LTL game.

Strategies with Memory. A memory structure $\mathscr{M} = (M, m_0, \mathrm{upd})$ for an arena $(V, V_0, V_1, E, v_0, \ell)$ consists of a finite set M of memory states, an initial memory state $m_0 \in M$, and an update function $\mathrm{upd}\colon M \times V \to M$, which can be extended to $\mathrm{upd}^*\colon V^+ \to M$ by $\mathrm{upd}^*(\rho_0) = m_0$ and $\mathrm{upd}^*(\rho_0\dots\rho_n\rho_{n+1}) = \mathrm{upd}(\mathrm{upd}^*(\rho_0\dots\rho_n), \rho_{n+1})$. A next-move function for Player i is a function $\mathrm{nxt}\colon V_i \times M \to V$ that satisfies $(v, \mathrm{nxt}(v,m)) \in E$ for all $v \in V_i$ and all $m \in M$. It induces a strategy σ with memory $\mathscr{M}$ via $\sigma(\rho_0\dots\rho_n) = \mathrm{nxt}(\rho_n, \mathrm{upd}^*(\rho_0\dots\rho_n))$. 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 $\mathscr{M}$ (and, slightly abusive, σ) is $|M|$. An arena $\mathscr{A}$ and a memory structure $\mathscr{M} = (M, m_0, \mathrm{upd})$ for $\mathscr{A}$ induce the expanded arena $\mathscr{A} \times \mathscr{M} = (V \times M, V_0 \times M, V_1 \times M, E', (s_0, m_0), \ell_{\mathscr{A}\times\mathscr{M}})$ where $((s,m),(s',m')) \in E'$ if and only if $(s,s') \in E$ and $\mathrm{upd}(m,s') = m'$, and $\ell_{\mathscr{A}\times\mathscr{M}}(s,m) = l(s)$. A game $\mathscr{G}$ with arena $\mathscr{A}$ is reducible to $\mathscr{G}'$ with arena $\mathscr{A}'$ via $\mathscr{M}$, written $\mathscr{G} \leq_{\mathscr{M}} \mathscr{G}'$, if $\mathscr{A}' = \mathscr{A} \times \mathscr{M}$ and every play $(\rho_0, m_0)(\rho_1, m_1)(\rho_2, m_2)\dots$ in $\mathscr{G}'$ is won by the player who wins the projected play $\rho_0\rho_1\rho_2\dots$ in $\mathscr{G}$.

Remark 3. *If $\mathscr{G} \leq_{\mathscr{M}} \mathscr{G}'$ and Player i has a positional winning strategy for $\mathscr{G}'$, then she also has a finite-state winning strategy with memory $\mathscr{M}$ for $\mathscr{G}$.*

A parity game or a PLTL game $\mathscr{G}$ (with respect to a fixed variable valuation) cannot be won by both players. On the other hand, $\mathscr{G}$ is determined, if one of the players wins it.

Proposition 4.

1. *Parity games are determined with positional strategies [3, 10] and the winner can be determined in time $\mathscr{O}(m(n/d)^{\lceil d/2 \rceil})$ [5], where n, m, and d denote the number of vertices, edges, and priorities.*
2. LTL *games (and therefore also* PLTL *games with respect to a fixed variable valuation) are determined with finite-state strategies. Determining the winner is* **2EXPTIME**-*complete [12] and finite-state winning strategies can be computed in doubly-exponential time.*

3 Solving Prompt and PLTL Games

In this section, we consider several decision problems for PLTL games. Kupferman et al. solved the PROMPT–LTL realizability problem[3] by a reduction to the LTL realizability problem [6], which is complete for doubly-exponential time. We show that this result suffices to prove that even the decision problems for the full logic with non-uniform bounds and parameterized always-operators are in **2EXPTIME**. For games with winning conditions in PLTL we are interested in the following decision problems:

Membership: Given a PLTL game $\mathscr{G}$, $i \in \{0,1\}$, and a valuation α, does $\alpha \in \mathscr{W}^i_{\mathscr{G}}$ hold?
Emptiness: Given a PLTL game $\mathscr{G}$ and $i \in \{0,1\}$, is $\mathscr{W}^i_{\mathscr{G}}$ empty?
Finiteness: Given a PLTL game $\mathscr{G}$ and $i \in \{0,1\}$, is $\mathscr{W}^i_{\mathscr{G}}$ finite?
Universality: Given a PLTL game $\mathscr{G}$ and $i \in \{0,1\}$, does $\mathscr{W}^i_{\mathscr{G}}$ contain all variable valuations?

Our first result is a simple consequence of Remark 2 and Proposition 4.2.

Theorem 5. *The membership problem for* PLTL *games is decidable.*

The realizability problem for PROMPT–LTL is known to be **2EXPTIME**-complete. The proof of this result can easily be adapted to graph-based PROMPT–LTL games as considered here.

Theorem 6 ([6])**.** *The emptiness problem for* PROMPT–LTL *games is* **2EXPTIME***-complete.*

The adapted proof in terms of graph-based games can be found in [14] and is sketched in the next section (see Lemma 13). It proceeds by a reduction to solving LTL games: given a PROMPT–LTL game $\mathscr{G} = (\mathscr{A}, \varphi)$ one constructs an LTL game $\mathscr{G}' = (\mathscr{A}', \varphi')$ with $|\mathscr{A}'| \in \mathscr{O}(|\mathscr{A}|^2)$ and $|\varphi'| \in \mathscr{O}(|\varphi|)$ such that $\mathscr{W}^0_{\mathscr{G}} \neq \emptyset$ if and only if Player 0 wins $\mathscr{G}'$. This proof yields the following corollary, which will be crucial when we determine optimal strategies in the next section: let $f(n) = 2^{2^{75(n+1)}} \in 2^{2^{\mathscr{O}(n)}}$.

Corollary 7 ([14])**.** *Let $\mathscr{G} = (\mathscr{A}, \varphi)$ be a* PROMPT–LTL *game with* $\mathrm{var}(\varphi) = \{x\}$*. If $\mathscr{W}^0_{\mathscr{G}} \neq \emptyset$, then Player* 0 *also has a finite-state winning strategy for $\mathscr{G}$ of size* $2|\mathscr{A}|f(|\varphi|)$ *which is winning with respect to the valuation* $x \mapsto 2(|\mathscr{A}| \cdot f(|\varphi|) + 1)$.

To solve the other decision problems for games with winning conditions in full PLTL, we make use of the duality of unipolar games and the duality of the emptiness and universality problem. For an arena $\mathscr{A} = (V, V_0, V_1, E, v_0, \ell)$, let $\overline{\mathscr{A}} := (V, V_1, V_0, E, v_0, \ell)$ be its dual arena, where the two players swap their positions. Given a PLTL game $\mathscr{G} = (\mathscr{A}, \varphi)$, the dual game is $\overline{\mathscr{G}} := (\overline{\mathscr{A}}, \neg\varphi)$. The dual game of a $\mathrm{PLTL}_{\mathbf{G}}$ game is a $\mathrm{PLTL}_{\mathbf{F}}$ game and vice versa. It is easy to see that Player i wins $\mathscr{G}$ with respect to α if and only if Player $1-i$ wins $\overline{\mathscr{G}}$ with respect to α. The sets $\mathscr{W}^i_{\mathscr{G}}$ enjoy two types of dualities, which we rely on in the following. The first one is due to determinacy of LTL games, the second one due to duality.

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

1. *$\mathscr{W}^0_{\mathscr{G}}$ is the complement of $\mathscr{W}^1_{\mathscr{G}}$.*
2. *$\mathscr{W}^i_{\mathscr{G}} = \mathscr{W}^{1-i}_{\overline{\mathscr{G}}}$.*

Another useful property is the monotonicity of the parameterized operators: let $\alpha(x) \leq \beta(x)$ and $\alpha(y) \geq \beta(y)$. Then, $(w, i, \alpha) \models \mathbf{F}_{\leq x}\varphi$ implies $(w, i, \beta) \models \mathbf{F}_{\leq x}\varphi$ and $(w, i, \alpha) \models \mathbf{G}_{\leq y}\varphi$ implies $(w, i, \beta) \models \mathbf{G}_{\leq y}\varphi$. Hence, the set $\mathscr{W}^0_{\mathscr{G}}$ is upwards-closed if $\mathscr{G}$ is a $\mathrm{PLTL}_{\mathbf{F}}$ game, and downwards-closed if $\mathscr{G}$ is a $\mathrm{PLTL}_{\mathbf{G}}$ game (valuations are compared componentwise). Now, we prove the main result of this section.

Theorem 9. *The emptiness, finiteness, and universality problems for* PLTL *games are* **2EXPTIME***-complete.*

[3] An abstract game without underlying arena in which two players alternatingly pick letters from 2^P. The first player wins if the ω-word produced by the players satisfies the winning condition φ.

Proof. Let $\mathscr{G} = (\mathscr{A}, \varphi)$. Due to Lemma 8.2 it suffices to consider $i = 0$.

Emptiness of $\mathscr{W}^0_{\mathscr{G}}$: Let $\varphi_{\mathbf{F}}$ be the formula obtained from φ by inductively replacing every subformula $\mathbf{G}_{\leq y}\psi$ by ψ, and let $\mathscr{G}_{\mathbf{F}} := (\mathscr{A}, \varphi_{\mathbf{F}})$. Note that $\mathscr{G}_{\mathbf{F}}$ is a $\mathrm{PLTL}_{\mathbf{F}}$ game. Applying downwards-closure, we obtain that $\mathscr{W}^0_{\mathscr{G}}$ is empty if and only if $\mathscr{W}^0_{\mathscr{G}_{\mathbf{F}}}$ is empty.

The latter problem can be decided by a reduction to PROMPT–LTL games. Fix a variable $x \in \mathscr{X}$ and let φ' be the formula obtained from $\varphi_{\mathbf{F}}$ by replacing every variable z in ψ by x. Then, $\mathscr{W}^0_{\mathscr{G}_{\mathbf{F}}} \neq \emptyset$ if and only if $\mathscr{W}^0_{\mathscr{G}'} \neq \emptyset$, where $\mathscr{G}' = (\mathscr{A}, \varphi')$. The latter problem can be decided in doubly-exponential time by Theorem 6. Since we have $|\varphi'| \leq |\varphi|$, the emptiness of $\mathscr{W}^0_{\mathscr{G}}$ can be decided in doubly-exponential time.

Universality of $\mathscr{W}^0_{\mathscr{G}}$: Applying both statements of Lemma 8 we get that $\mathscr{W}^0_{\mathscr{G}}$ is universal if and only if $\mathscr{W}^1_{\mathscr{G}} = \emptyset$ if and only if $\mathscr{W}^0_{\mathscr{G}} = \emptyset$. The latter is decidable in doubly-exponential time, as shown above.

Finiteness of $\mathscr{W}^0_{\mathscr{G}}$: If φ contains at least one $\mathbf{F}_{\leq x}$, then $\mathscr{W}^0_{\mathscr{G}}$ is infinite, if and only if it is non-empty, due to monotonicity of $\mathbf{F}_{\leq x}$. The emptiness of $\mathscr{W}^0_{\mathscr{G}}$ can be decided in doubly-exponential time as discussed above. Otherwise, $\mathscr{G}$ is a $\mathrm{PLTL}_{\mathbf{G}}$ game whose finiteness problem can be decided in doubly-exponential time by a reduction to the universality problem for a (simpler) $\mathrm{PLTL}_{\mathbf{G}}$ game. We assume that φ has at least one parameterized temporal operator, since the problem is trivial otherwise. The set $\mathscr{W}^0_{\mathscr{G}}$ is infinite if and only if there is a variable $y \in \mathrm{var}(\varphi)$ that is mapped to infinitely many values by the valuations in $\mathscr{W}^0_{\mathscr{G}}$. 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 $\mathbf{G}_{\leq z}\psi$ for $z \neq y$ by ψ and define $\mathscr{G}_y := (\mathscr{A}, \varphi_y)$. Then, $\mathscr{W}^0_{\mathscr{G}}$ is infinite, if and only if there exists some variable $y \in \mathrm{var}(\varphi)$ such that $\mathscr{W}^0_{\mathscr{G}_y}$ is universal. So, deciding whether $\mathscr{W}^0_{\mathscr{G}}$ is infinite can be done in doubly-exponential time by solving $|\mathrm{var}(\varphi)|$ many universality problems for $\mathrm{PLTL}_{\mathbf{G}}$ games, which were discussed above.

Finally, hardness follows directly from **2EXPTIME**-hardness of solving LTL games. ☐

4 Optimal Winning Strategies for unipolar PLTL Games

For unipolar games, it is natural to view synthesis of winning strategies as an optimization problem: which is the *best* variable valuation α such that Player 0 can win with respect to α? We consider two quality measures for a valuation α for φ: the maximal parameter $\max_{z \in \mathrm{var}(\varphi)} \alpha(z)$ and the minimal parameter $\min_{z \in \mathrm{var}(\varphi)} \alpha(z)$. For a $\mathrm{PLTL}_{\mathbf{F}}$ game, Player 0 tries to minimize the waiting times. Hence, we are interested in minimizing the minimal or maximal parameter. Dually, for $\mathrm{PLTL}_{\mathbf{G}}$ games, we are interested in maximizing the quality measures. The dual problems, i.e., maximizing the waiting times in a $\mathrm{PLTL}_{\mathbf{F}}$ game and minimizing the satisfaction time in a $\mathrm{PLTL}_{\mathbf{G}}$ game, are trivial due to upwards-respectively downwards- closure of the set of winning valuations. Again, we only consider Player 0 as one can dualize the game to obtain similar results for Player 1. The main result of this section states that all these optimization problems are not harder than solving LTL games.

Theorem 10. *Let $\mathscr{G}_{\mathbf{F}} = (\mathscr{A}_{\mathbf{F}}, \varphi_{\mathbf{F}})$ be a $\mathrm{PLTL}_{\mathbf{F}}$ game and $\mathscr{G}_{\mathbf{G}} = (\mathscr{A}_{\mathbf{G}}, \varphi_{\mathbf{G}})$ be a $\mathrm{PLTL}_{\mathbf{G}}$ game. Then, the following values (and winning strategies realizing them) can be computed in doubly-exponential time.*

1. $\min_{\alpha \in \mathscr{W}^0_{\mathscr{G}_{\mathbf{F}}}} \min_{x \in \mathrm{var}(\varphi_{\mathbf{F}})} \alpha(x)$.
2. $\min_{\alpha \in \mathscr{W}^0_{\mathscr{G}_{\mathbf{F}}}} \max_{x \in \mathrm{var}(\varphi_{\mathbf{F}})} \alpha(x)$.
3. $\max_{\alpha \in \mathscr{W}^0_{\mathscr{G}_{\mathbf{G}}}} \max_{y \in \mathrm{var}(\varphi_{\mathbf{G}})} \alpha(y)$.
4. $\max_{\alpha \in \mathscr{W}^0_{\mathscr{G}_{\mathbf{G}}}} \min_{y \in \mathrm{var}(\varphi_{\mathbf{G}})} \alpha(y)$.

We begin the proof by showing that all four problems can be reduced to the optimization problem for PROMPT–LTL games: let $\mathscr{G} = (\mathscr{A}, \varphi)$ be a PROMPT–LTL game with $\mathrm{var}(\varphi) = \{x\} \subseteq \mathscr{X}$. The goal is to determine $\min_{\alpha \in \mathscr{W}^0_{\mathscr{G}}} \alpha(x)$.

The latter three reductions are simple applications of the monotonicity of the parameterized operators, while the first one requires substantial work.

1.) For each $x \in \mathrm{var}(\varphi)$, we replace eventualities parameterized by $z \neq x$ by an unparameterized formula, thereby constructing the projection of $\mathscr{W}^0_{\mathscr{G}_\mathbf{F}}$ to the values of x. However, we cannot just remove the parameters from an eventuality, as we have to ensure that the waiting times are still bounded by some unknown, but fixed value. This is achieved by applying the alternating-color technique for PROMPT–LTL [6].

Let $p \notin P$ be a fixed proposition. An ω-word $w' = w'_0 w'_1 w'_2 \ldots \in \left(2^{P \cup \{p\}}\right)^\omega$ is a p-coloring of $w = w_0 w_1 w_2 \ldots \in \left(2^P\right)^\omega$ 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 $p \notin w'_n$. Given $k \in \mathbb{N}$ we say that w' is k-spaced, if the colors in w' change infinitely often, but not twice in any infix of length k. Dually, w' is k-bounded, if the colors change at least once in every infix of length $k+1$.

The formula $alt_p := \mathbf{GF} p \wedge \mathbf{GF} \neg p$ is satisfied if the colors change infinitely often. Given a PLTL formula φ and $X \subseteq var(\varphi)$, let φ_X denote the formula obtained by inductively replacing every subformula $\mathbf{F}_{\leq x} \psi$ with $x \notin X$ by $(p \rightarrow (p \mathbf{U} (\neg p \mathbf{U} \psi))) \wedge (\neg p \rightarrow (\neg p \mathbf{U} (p \mathbf{U} \psi)))$. Finally, consider the formula $\varphi_X \wedge alt_p$. It forces a coloring to have infinitely many color changes and every subformula $\mathbf{F}_{\leq x} \psi$ with $x \notin X$ to be satisfied within one color change. We have $\mathrm{var}(\varphi_X) = X$ and $|\varphi_X| \in \mathscr{O}(|\varphi|)$.

For a variable valuation α and a subset X of α's domain, we denote the restriction of α to X by $\alpha_{\restriction X}$.

Lemma 11 ([6]). *Let φ be a* PLTL *formula, $X \subseteq \mathrm{var}(\varphi)$, and let $w \in \left(2^P\right)^\omega$.*

1. *If $(w, 0, \alpha) \models \varphi$, then $(w', 0, \alpha_{\restriction X}) \models \varphi_X \wedge alt_p$ for every k-spaced p-coloring w' of w, where $k = \max_{x \in \mathrm{var}(\varphi) \setminus X} \alpha(x)$.*
2. *Let $k \in \mathbb{N}$. If w' is a k-bounded p-coloring of w with $(w', 0, \alpha) \models \varphi_X$, then $(w, 0, \beta) \models \varphi$ where* $\beta(x) = \begin{cases} \alpha(x) & \text{if } x \in X, \\ 2k & \text{else.} \end{cases}$

The previous lemma shows how replace (on suitable p-colorings) a parameterized eventuality by an LTL formula, while still ensuring a bound on the satisfaction of the parameterized eventuality. To apply the alternating-color technique, we have to transform the original arena $\mathscr{A}$ into an arena $\mathscr{A}'$ in which Player 0 produces p-colorings of the plays of the original arena, i.e., $\mathscr{A}'$ will consist of two disjoint copies of $\mathscr{A}$, one labeled with 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 vertex for every edge of $\mathscr{A}$ which allows Player 0 to choose the component. Formally, given an arena $\mathscr{A} = (V, V_0, V_1, E, v_0, \ell)$, define the expanded arena $\mathscr{A}' := (V', V'_0, V'_1, E', v'_0, \ell')$ by

- $V' = V \times \{0,1\} \cup E$,
- $V'_0 = V_0 \times \{0,1\} \cup E$,
- $V'_1 = V_1 \times \{0,1\}$,
- $E' = \{((v,0),e), ((v,1),e), (e,(v',0)), (e,(v',1)) \mid e = (v,v') \in E\}$,

- $v'_0 = (v_0, 0)$,
- $\ell'(e) = \emptyset$ for all $e \in E$ and $\ell'(v,b) = \begin{cases} \ell(v) \cup \{p\} & \text{if } b = 0, \\ \ell(v) & \text{if } b = 1. \end{cases}$

$\mathscr{A}'$ is bipartite with partition $\{V \times \{0,1\}, E\}$, so a play has the form $(\rho_0, b_0)e_0(\rho_1, b_1)e_1(\rho_2, b_2)\dots$ where $\rho_0\rho_1\rho_2\dots$ is a play in $\mathscr{A}$, $e_n = (\rho_n, \rho_{n+1})$, and the b_n are in $\{0,1\}$. Also, we have $|\mathscr{A}'| \in \mathscr{O}(|\mathscr{A}|^2)$.

Finally, this construction necessitates a modification of the semantics of the game: only every other vertex is significant when it comes to determining the winner of a play in $\mathscr{A}'$, the choice vertices have to be ignored. This motivates blinking semantics for PLTL games. Let $\mathscr{G} = (\mathscr{A}, \varphi)$ be a PLTL game and $\rho = \rho_0\rho_1\rho_2\dots$ be a play. Player 0 wins ρ with respect to α under blinking semantics, if $(t(\rho_0\rho_2\rho_4\dots), 0, \alpha) \models \varphi$. Analogously, Player 1 wins ρ with respect to α under blinking semantics if $(t(\rho_0\rho_2\rho_4\dots), 0, \alpha) \not\models \varphi$. The notions of winning strategies and winning $\mathscr{G}$ with respect to α under blinking semantics are defined in the obvious way.

Remark 12. PLTL *games with respect to a fixed variable valuation under blinking semantics are determined with finite-state strategies.*

Now, we can state the connection between a $\text{PLTL}_\mathbf{F}$ game $(\mathscr{A}, \varphi)$ and its counterpart in $\mathscr{A}'$ with blinking semantics. The proof relies on the existence of finite-state winning strategies which necessarily produce only k-bounded plays for some fixed k, since alt_p is part of the winning condition.

Lemma 13. *Let* $(\mathscr{A}, \varphi)$ *be a* $\text{PLTL}_\mathbf{F}$ *game and* $X \subseteq \text{var}(\varphi)$.

1. *Let* $\alpha\colon \text{var}(\varphi) \to \mathbb{N}$ *be a variable valuation. If Player i wins* $(\mathscr{A}, \varphi)$ *with respect to* α*, then she wins* $(\mathscr{A}', \varphi_X \wedge alt_p)$ *with respect to* $\alpha_{\restriction X}$ *under blinking semantics.*
2. *Let* $\alpha\colon X \to \mathbb{N}$ *be a variable valuation. If Player i wins* $(\mathscr{A}', \varphi_X \wedge alt_p)$ *with respect to* α *under blinking semantics, then there exists a variable valuation* β *with* $\beta(x) = \alpha(x)$ *for every* $x \in X$ *such that she wins* $(\mathscr{A}, \varphi)$ *with respect to* β.

Applying the lemma to our problem, we have

$$\min_{\alpha \in \mathscr{W}^0_{\mathscr{G}}} \min_{x \in \text{var}(\varphi)} \alpha(x) = \min_{x \in \text{var}(\varphi)} \min\{\alpha(x) \mid \text{ Player 0 wins } (\mathscr{A}', \varphi_{\{x\}} \wedge alt_p) \text{ w.r.t. } \alpha \text{ u. blinking semantics}\} \ .$$

Since $\varphi_{\{x\}} = \{x\}$, we have reduced the minimization problem to $|\text{var}(\varphi)|$ many PROMPT–LTL optimization problems, albeit under blinking semantics. However, the proof presented in the following can easily be adapted to deal with blinking semantics.

2.) 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 $\mathscr{G}' := (\mathscr{A}_\mathbf{F}, \varphi'_\mathbf{F})$. Then, $\min_{\alpha \in \mathscr{W}^0_{\mathscr{G}_\mathbf{F}}} \max_{x \in \text{var}(\varphi_\mathbf{F})} \alpha(x) = \min_{\alpha \in \mathscr{W}^0_{\mathscr{G}'}} \alpha(z)$, due to upwards-closure of $\mathscr{W}^0_{\mathscr{G}_\mathbf{F}}$.

3.) For every $y \in \text{var}(\varphi_\mathbf{G})$ let φ_y be obtained from $\varphi_\mathbf{G}$ by replacing every subformula $\mathbf{G}_{\leq z}\psi$ for $z \neq y$ by ψ and let $\mathscr{G}_y := (\mathscr{A}_\mathbf{G}, \varphi_y)$. Then, we have $\max_{\alpha \in \mathscr{W}^0_{\mathscr{G}_\mathbf{G}}} \max_{y \in \text{var}(\varphi_\mathbf{G})} \alpha(y) = \max_{y \in \text{var}(\varphi_\mathbf{G})} \max_{\alpha \in \mathscr{W}^0_{\mathscr{G}_y}} \alpha(y)$, due to downwards-closure of $\mathscr{W}^0_{\mathscr{G}_\mathbf{G}}$. Hence, we have reduced the original problem to $|\text{var}(\varphi_\mathbf{G})|$ maximization problems for a $\text{PLTL}_\mathbf{G}$ game with a single variable, which are discussed below.

4.) Let $\varphi'_\mathbf{G}$ be obtained from $\varphi_\mathbf{G}$ by renaming every variable in $\varphi_\mathbf{G}$ to z and let $\mathscr{G}' = (\mathscr{A}_\mathbf{G}, \varphi'_\mathbf{G})$. Then, $\max_{\alpha \in \mathscr{W}^0_{\mathscr{G}_\mathbf{G}}} \min_{y \in \text{var}(\varphi_\mathbf{G})} \alpha(y) = \max_{\alpha \in \mathscr{W}^0_{\mathscr{G}'}} \alpha(z)$, again due to downwards-closure of $\mathscr{W}^0_{\mathscr{G}_\mathbf{G}}$. Again, we have reduced the original problem to a maximization problem for a $\text{PLTL}_\mathbf{G}$ game with a single variable.

To finish the reductions we translate a $\text{PLTL}_\mathbf{G}$ optimization problem with a single variable into a PROMPT–LTL optimization problem: let $\mathscr{G} = (\mathscr{A}, \varphi)$ be a $\text{PLTL}_\mathbf{G}$ game with $\text{var}(\varphi) = \{y\} \subseteq \mathscr{Y}$.

Then, we have $\max_{\alpha \in \mathscr{W}^0_{\mathscr{G}}} \alpha(y) = \max_{\alpha \in \mathscr{W}^1_{\overline{\mathscr{G}}}} \alpha(y) = \min_{\alpha \in \mathscr{W}^0_{\overline{\mathscr{G}}}} \alpha(y) + 1$, due to the closure properties and Lemma 8. As $\overline{\mathscr{G}}$ is a PROMPT–LTL game, we achieved our goal.

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 $\text{PLTL}_{\mathbf{F}}$ game and to maximize the maximal parameter value in a $\text{PLTL}_{\mathbf{G}}$ 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, it suffices to show that a PROMPT–LTL optimization problem can be solved in doubly-exponential time.

So, let $\mathscr{G} = (\mathscr{A}, \varphi)$ be a PROMPT–LTL game with $\text{var}(\varphi) = \{x\}$. If $\mathscr{W}^0_{\mathscr{G}} \neq \emptyset$, then Corollary 7 yields $\min_{\alpha \in \mathscr{W}^0_{\mathscr{G}}} \alpha(x) \leq \; k := 2(|\mathscr{A}| \cdot f(|\varphi|) + 1) \in |\mathscr{A}| \cdot 2^{2^{\mathscr{O}(|\varphi|)}}$. Let α_n be the valuation mapping x to n. To determine $\min_{\alpha \in \mathscr{W}^0_{\mathscr{G}}} \alpha(x)$, it suffices to find the smallest $n < k$ such that $\alpha_n \in \mathscr{W}^0_{\mathscr{G}}$. As the number of such valuations α_n is equal to k, it suffices to show that $\alpha_n \in \mathscr{W}^0_{\mathscr{G}}$ can be decided in doubly-exponential time in the size of $\mathscr{G}$, provided that $n < k$. This is achieved by a game reduction to a parity game.

Fix a valuation α and remember that φ_α is an LTL formula (see Remark 2). Now, observe that a deterministic parity automaton $\mathfrak{P} = (Q, 2^P, q_0, \delta, c)$ with $L(\mathfrak{P}) = \{w \in (2^P)^\omega \mid (w,0) \models \varphi_\alpha\}$ can be turned into a memory structure $\mathscr{M} = (Q, q_0, \text{upd})$ for $(\mathscr{A}, \varphi_\alpha)$ by defining $\text{upd}(q,v) = \delta(q, \ell(v))$. Then, we have $(\mathscr{A}, \varphi_\alpha) \leq_{\mathscr{M}} (\mathscr{A} \times \mathscr{M}, c')$, where $c'(v,q) = c(q)$. Hence, the Remarks 2 and 3 yield $\alpha \in \mathscr{W}^0_{\mathscr{G}}$ if and only if Player 0 wins $(\mathscr{A} \times \mathscr{M}, c')$.

Lemma 14. *Let α be a variable valuation and φ be a* PROMPT–LTL *formula with* $\text{var}(\varphi) = \{x\}$. *There exists a deterministic parity automaton $\mathfrak{P}$ recognizing the language $\{w \in (2^P)^\omega \mid (w,0) \models \varphi_\alpha\}$ such that $|\mathfrak{P}| \in 2^{2^{\mathscr{O}(|\varphi|)}} \cdot (\alpha(x)+1)^{2^{\mathscr{O}(|\varphi|)}}$ and $\mathfrak{P}$ has $2^{\mathscr{O}(|\varphi|)}$ many colors.*

For a valuation α_n with $n < k$, we have $|\mathfrak{P}| \in 2^{2^{\mathscr{O}(|\mathscr{A}|+|\varphi|)}}$ with $2^{\mathscr{O}(|\varphi|)}$ many colors. Thus, Proposition 4.1 implies that $(\mathscr{A} \times \mathscr{M}, c')$ can be solved in doubly-exponential time in the size of $\mathscr{G}$, which suffices to prove Theorem 10, as we have to solve at most doubly-exponentially many parity games[4], each of which can be solved in doubly-exponential time. Thus, it remains to prove Lemma 14.

Furthermore, we have seen that the automaton $\mathfrak{P}$ for the *minimal* α_n can easily be turned into a finite-state winning strategy for $\mathscr{G}$ realizing $\min_{\alpha \in \mathscr{W}^0_{\mathscr{G}}} \alpha(x)$. To obtain a winning strategy for the general case of an $\text{PLTL}_{\mathbf{F}}$ (respectively $\text{PLTL}_{\mathbf{G}}$) game it is necessary to construct a deterministic parity automaton for the $\text{PLTL}_{\mathbf{F}}$ formula φ (respectively $\neg\varphi$) as described below. In case of a $\text{PLTL}_{\mathbf{G}}$ game, we need to complement the automaton, which is achieved by incrementing the priority of each state by one.

We construct an automaton as required in Lemma 14 in the remainder of this section. Note that the naive approach of constructing a deterministic parity automaton for the LTL formula φ_{α_n} yields an automaton that recognizes the desired language, but is of quadruply-exponential size, if n is close to k. The problem arises from the fact that φ_{α_n} uses a disjunction of nested next-operators of depth n to be able to count up to n. This (doubly-exponential) *counter* is hardwired into the formula φ_{α_n} and thus leads to a quadruply-exponential blowup when turning φ_{α_n} into a deterministic parity automaton, since turning LTL formulae into deterministic parity automata necessarily incurs a doubly-exponential blowup [7].

To obtain our results, we decouple the counter from the formula by relaxing parameterized eventualities to plain eventualities. We translate the relaxed formula into a generalized Büchi automaton, which is then turned in a Büchi automaton. By placing an additional constraint on accepting runs we take care of the bound on the (now relaxed) parameterized operators. As these automata are unambiguous, we also end up with a non-confluent Büchi automaton, which is then determinized into a parity automaton. Only then, the additional constraint is added to the parity automaton in the form of a counter that tracks

[4] This can be improved to exponentially many by binary search.

(and aborts, if the counter is overrun) different runs of the Büchi automaton. This way, we obtain an automaton that is equivalent to the (unrelaxed) PROMPT–LTL formula with respect to α_n. To add these counters, it is crucial to have a non-confluent Büchi automaton, as such an automaton has at most $|Q|$ runs which have to be tracked by the counter.

In the following we extend known constructions for translating an LTL formula into a non-deterministic Büchi automaton and for translating a non-deterministic Büchi automaton into a deterministic parity automaton. In the first step we have to deal with the additional constraints, which do not appear in the classical translation problem. In the second step, we have to simulate these constraints with the states of the parity automaton, which requires changes to this translation as well. Since our proof technique can deal with several parameters, we consider the more general case of a $\mathrm{PLTL}_{\mathbf{F}}$ formula instead of a PROMPT–LTL formula.

From $\mathrm{PLTL}_{\mathbf{F}}$ to generalized Büchi Automata. We begin by constructing a generalized Büchi automaton from a $\mathrm{PLTL}_{\mathbf{F}}$ formula using a slight adaptation of a standard textbook method (see [2]). We ignore the parameters when defining the transition relation, i.e., we treat a parameterized eventually as a plain eventually. The bounds are taken care of by additional constraints on accepting runs.

Given a $\mathrm{PLTL}_{\mathbf{F}}$ formula φ we define its closure $\mathrm{cl}(\varphi)$ to be the set of subformulae of φ. A set $B \subseteq \mathrm{cl}(\varphi)$ is consistent, if the following properties are satisfied:

- $p \in B$ if and only if $\neg p \notin B$ for every $p \in P$.
- $\psi_1 \wedge \psi_2 \in B$ if and only if $\psi_1 \in B$ and $\psi_2 \in B$.
- $\psi_1 \vee \psi_2 \in B$ if and only if $\psi_1 \in B$ or $\psi_2 \in B$.
- $\psi_2 \in B$ implies $\psi_1 \mathbf{U} \psi_2 \in B$.
- $\psi_1, \psi_2 \in B$ implies $\psi_1 \mathbf{R} \psi_2 \in B$.
- $\psi_1 \in B$ implies $\mathbf{F}_{\leq x} \psi_1 \in B$.

The set of consistent subsets is denoted by $\mathscr{C}(\varphi) \subseteq 2^{\mathrm{cl}(\varphi)}$.

Construction 15. *Given a $\mathrm{PLTL}_{\mathbf{F}}$ formula φ, we define the generalized Büchi automaton $\mathfrak{A}_\varphi = (Q, 2^P, Q_0, \Delta, \mathscr{F})$ by*

- $Q = \mathscr{C}(\varphi)$ *and* $Q_0 = \{B \in \mathscr{C}(\varphi) \mid \varphi \in B\}$,
- $(B, a, B') \in \Delta$ *if and only if*
  - $B \cap P = a$,
  - $\mathbf{X}\psi_1 \in B$ *if and only if* $\psi_1 \in B'$,
  - $\psi_1 \mathbf{U} \psi_2 \in B$ *if and only if* $\psi_2 \in B$ *or* ($\psi_1 \in B$ *and* $\psi_1 \mathbf{U} \psi_2 \in B'$),
  - $\psi_1 \mathbf{R} \psi_2 \in B$ *if and only if* $\psi_2 \in B$ *and* ($\psi_1 \in B$ *or* $\psi_1 \mathbf{R} \psi_2 \in B'$), *and*
  - $\mathbf{F}_{\leq x} \psi_1 \in B$ *if and only if* $\psi_1 \in B$ *or* $\mathbf{F}_{\leq x} \psi_1 \in B'$.
- $\mathscr{F} = \mathscr{F}_{\mathbf{U}} \cup \mathscr{F}_{\mathbf{R}} \cup \mathscr{F}_{\mathbf{F}_\leq}$ *where*
  - $\mathscr{F}_{\mathbf{U}} = \{F_{\psi_1 \mathbf{U} \psi_2} \mid \psi_1 \mathbf{U} \psi_2 \in \mathrm{cl}(\varphi)\}$ *with* $F_{\psi_1 \mathbf{U} \psi_2} = \{B \in \mathscr{C}(\varphi) \mid \psi_1 \mathbf{U} \psi_2 \notin B$ *or* $\psi_2 \in B\}$,
  - $\mathscr{F}_{\mathbf{R}} = \{F_{\psi_1 \mathbf{R} \psi_2} \mid \psi_1 \mathbf{R} \psi_2 \in \mathrm{cl}(\varphi)\}$ *with* $F_{\psi_1 \mathbf{R} \psi_2} = \{B \in \mathscr{C}(\varphi) \mid \psi_1 \mathbf{R} \psi_2 \in B$ *or* $\psi_2 \notin B\}$, *and*
  - $\mathscr{F}_{\mathbf{F}_\leq} = \{F_{\mathbf{F}_{\leq x} \psi_1} \mid \mathbf{F}_{\leq x} \psi_1 \in \mathrm{cl}(\varphi)\}$ *with* $F_{\mathbf{F}_{\leq x} \psi_1} = \{B \in \mathscr{C}(\varphi) \mid \mathbf{F}_{\leq x} \psi_1 \notin B$ *or* $\psi_1 \in B\}$.

Lemma 16. *Let $\varphi \in \mathrm{PLTL}_{\mathbf{F}}$ and let $\mathfrak{A}_\varphi$ be defined as in Construction 15.*

1. $(w, 0, \alpha) \models \varphi$ *if and only if* $\mathfrak{A}_\varphi$ *has an accepting run* ρ *on* w *such that each* $F_{\mathbf{F}_{\leq x} \psi_1} \in \mathscr{F}_{\mathbf{F}_\leq}$ *is visited at least once in every infix of* ρ *of length* $\alpha(x) + 1$.
2. $\mathfrak{A}_\varphi$ *is unambiguous.*
3. $|\mathfrak{A}_\varphi| \leq 2^{|\varphi|}$ *and* $|\mathscr{F}| < |\varphi|$.

Proof. 1.) Let $(w,0,\alpha) \models \varphi$. For each n define $B_n = \{\psi \in \mathrm{cl}(\varphi) \mid (w,n,\alpha) \models \psi\}$ and show that $\rho = B_0B_1B_2\dots$ is an accepting run of $\mathfrak{A}_\varphi$ such that each $F_{\mathbf{F}_{\leq x}\psi_1} \in \mathscr{F}_{\mathbf{F}_\leq}$ is visited at least once in every infix of ρ of length $\alpha(x)+1$. The semantics of PLTL guarantee that each B_n is consistent, $B_0 \in Q_0$ follows from $(w,0,\alpha) \models \varphi$, and $(B_n, w_n, B_{n+1}) \in \Delta$ for every n is due to the semantics of PLTL. Thus, the sequence $B_0B_1B_2\dots$ is a run. Assume that some $F_{\psi_1\mathbf{U}\psi_2}$ is visited only finitely often, i.e., there exists an index n such that for every $n' \geq n$ we have $(w,n',\alpha) \models \psi_1\mathbf{U}\psi_2$ and $(w,n',\alpha) \not\models \psi_2$. This contradicts the semantics of the until-operator, which guarantee a position $m \geq n$ such that $(w,m,\alpha) \models \psi_2$, if $(w,n,\alpha) \models \psi_1\mathbf{U}\psi_2$. Now, assume that some $F_{\psi_1\mathbf{R}\psi_2}$ is visited only finitely often, i.e., there exists an index n such that for every $n' \geq n$ we have $(w,n',\alpha) \not\models \psi_1\mathbf{R}\psi_2$ and $(w,n',\alpha) \models \psi_2$. This contradicts the semantics of the release-operator, which state $(w,n,\alpha) \models \psi_1\mathbf{R}\psi_2$, if ψ_2 holds at every position $n' \geq n$. Finally, assume that some $F_{\mathbf{F}_{\leq x}\psi_1} \in \mathscr{F}_{\mathbf{F}_\leq}$ is not visited in an infix of $B_0B_1B_2\dots$ of length $\alpha(x)+1$, i.e., there is some index n such that $(w,n,\alpha) \models \mathbf{F}_{\leq x}\psi_1$ and $(w,n+j,\alpha) \not\models \psi_1$ for every j in the range $0 \leq j \leq \alpha(x)$. This contradicts the semantics of the parameterized eventually, which guarantee the existence of an index k in the range $0 \leq k \leq \alpha(x)$ such that $(w,n+k,\alpha) \models \psi_1$. Hence, $B_0B_1B_2\dots$ is an accepting run such that each $F_{\mathbf{F}_{\leq x}\psi_1} \in \mathscr{F}_{\mathbf{F}_\leq}$ is visited at least once in every infix of $B_0B_1B_2\dots$ of length $\alpha(x)+1$.

For the other direction, let $\rho = B_0B_1B_2\dots$ be an accepting run of $\mathfrak{A}_\varphi$ on w such that each $F_{\mathbf{F}_{\leq x}\psi_1} \in \mathscr{F}_{\mathbf{F}_\leq}$ is visited at least once in every infix of ρ of length $\alpha(x)+1$. A structural induction over φ shows that $\psi \in B_n$ if and only if $(w,n,\alpha) \models \psi$. This suffices, since we have $\varphi \in B_0$.

2.) Let $e(\varphi)$ be the formula obtained from $\varphi \in \mathrm{PLTL}_\mathbf{F}$ by replacing every parameterized eventually $\mathbf{F}_{\leq x}$ by an eventually $\mathbf{F}$. The automata $\mathfrak{A}_\varphi$ and $\mathfrak{A}_{e(\varphi)}$ are isomorphic. Thus, it suffices to show that $\mathfrak{A}_{e(\varphi)}$ is unambiguous. So, assume there are two accepting runs $B_0B_1B_2\dots$ and $B'_0B'_1B'_2\dots$ on an ω-word w and let n be an index such that $B_n \neq B'_n$, i.e., there exists $\psi \in \mathrm{cl}(e(\varphi))$ such that (w.l.o.g.) $\psi \in B_n$, but $\psi \notin B'_n$. In 1.), we have shown that we have $\psi \in B_n$ (respectively $\psi \in B'_n$) if and only if $(w,n) \models \psi$ (note that ψ is an LTL formula, hence we do not need to care about a variable valuation). Thus, we have $(w,n) \models \psi$ (due to $\psi \in B_n$) and $(w,n) \not\models \psi$ (due to $\psi \notin B'_n$), which yields the desired contradiction.

3.) Clear. □

From generalized Büchi Automata to Büchi Automata. Now, we use a standard construction (see [2]) to turn a generalized Büchi automaton $\mathfrak{A} = (Q,\Sigma,Q_0,\Delta,\{F_1,\dots,F_k\})$ into a Büchi automaton $\mathfrak{A}' = (Q',\Sigma,Q'_0,\Delta',F')$ while preserving its language (even under the additional constraints) and its unambiguity. The state set of $\mathfrak{A}'$ is $Q \times \{0,1,\dots,k\}$, where the first component is used to simulate the behavior of $\mathfrak{A}$, while the second component is used to ensure that every set F_j is visited infinitely often.

Lemma 17. *Let $\mathfrak{A} = (Q,\Sigma,Q_0,\Delta,\{F_1,\dots,F_k\})$ be a generalized Büchi automaton. There exists a Büchi automaton $\mathfrak{A}'$ with state set $Q \times \{0,1\dots,k\}$ such that the following holds:*

1. *Let $\mathfrak{A} = \mathfrak{A}_\varphi$ for some $\mathrm{PLTL}_\mathbf{F}$ formula φ as in Construction 15. Then, $(w,0,\alpha) \models \varphi$ if and only if $\mathfrak{A}'$ has an accepting run $(q_0,i_0)(q_1,i_1)(q_2,i_2)\dots$ on w such that each $\mathbf{F}_{\mathbf{F}_{\leq x}\psi_1} \in \mathscr{F}_{\mathbf{F}_\leq}$ is visited at least once in every infix of $q_0q_1q_2\dots$ of length $\alpha(x)+1$.*
2. *$\mathfrak{A}'$ is unambiguous, if $\mathfrak{A}$ is unambiguous.*
3. $|\mathfrak{A}'| = |\mathfrak{A}| \cdot (k+1)$

From Büchi Automata to Deterministic Parity Automata. Now, we have to determinize an unambiguous (and therefore non-confluent) Büchi automaton while incorporating the additional constraints on accepting runs. Abstractly, we are given a non-confluent Büchi automaton $\mathfrak{A}$ and a finite set of tuples $(F_j,b_j) \in 2^Q \times \mathbb{N}_+$ and are only interested in runs ρ that visit a state from F_j in every infix of ρ of length b_j, while visiting the accepting states of the Büchi automaton infinitely often. Remember that a non-confluent automaton has at most $|Q|$ finite runs on a finite word $w_0 \cdots w_n$, which can be uniquely identified by their last state. Furthermore, for every last state q of such a run, there is a unique state p such that p is the last state of a run of the automaton on $w_0 \cdots w_{n-1}$ and $(p, w_n, q) \in \Delta$. Thus, to check the additional constraints on the runs, we can use counters $d(q, j)$ to abort the run ending in q if it did not visit F_j for b_j consecutive states. The state space of the deterministic automaton we construct is the cartesian product of the state space of $\mathfrak{P}$ and the counters $d(q, j)$ for every q and j, where $\mathfrak{P}$ is a deterministic automaton recognizing the language of $\mathfrak{A}$ without additional constraints. To prove Theorem 10, we want to use the deterministic automaton with counters as memory structure in a game reduction, which imposes additional requirements on its size and its acceptance condition.

The Büchi automaton we need to determinize is already of exponential size. Hence, we can spend another exponential for determinization, which is the typical complexity of a determinization procedure for Büchi automata. However, we have to carefully choose the acceptance condition of the deterministic automaton we construct: to prove the main theorem, we need an acceptance condition Acc such that a game with arena $\mathscr{A} \times \mathscr{M}$ and winning condition Acc can be solved in doubly-exponential time, even if $\mathscr{M}$ is already of doubly-exponential size. Furthermore, it is desirable to use a condition Acc that guarantees Player 0 positional winning strategies: in this case, $\mathscr{M}$ implements a finite-state winning strategy for her in the original PLTL$_\mathbf{F}$ game.

The parity condition satisfies all our requirements. Thus, we adapt a determinization construction [8, 9] tailored for non-confluent Büchi automata yielding a parity automaton. The automata obtained by this construction are slightly larger than the ones obtained by optimal constructions, but still small enough to satisfy our requirements on them. Another advantage of this construction is the fact that it is conceptually simpler than the constructions for arbitrary Büchi automata based on trees labeled with state sets. Nevertheless, it is possible to use another determinization construction, as long as it satisfies the requirements in terms of size and winning condition described above.

Given a transition relation $\Delta \subseteq Q \times \Sigma \times Q$, define $\Delta(S, a) = \{q' \in Q \mid (q, a, q') \in \Delta \text{ for some } q \in S\}$.

Construction 18 ([9]). *Given a non-confluent Büchi automaton $\mathfrak{A} = (Q, \Sigma, Q_0, \Delta, F)$ and a finite set $\{(F_1, b_1), \dots, (F_k, b_k)\} \subseteq 2^Q \times \mathbb{N}_+$ construct the deterministic parity automaton $\mathfrak{P} = (Q', \Sigma, q_0', \delta, c)$ as follows: let $n = |Q|$ and define*

- $Q' = \{((S_0, m_0), \dots, (S_n, m_n), d) \mid S_i \subseteq Q, m_i \in \{0, 1\}, d \colon Q \times \{1, \dots, k\} \to \mathbb{N} \cup \{\bot\}$ *with* $d(q, j) < b_j$ *or* $d(q, j) = \bot\}$,
- $q_0' = ((S_0, 0), (\emptyset, 0), \dots, (\emptyset, 0), d_0)$ *with* $d_0(q, j) = 0$ *if* $q \in Q_0 \cap F_j$; $d_0(q, j) = 1$ *if* $q \in Q_0 \setminus F_j$ *and* $1 < b_j$; *and* $d_0(q, j) = \bot$ *otherwise; and* $S_0 = \{q \in Q_0 \mid d(q, j) \neq \bot \text{ for every } j\}$.
- *We define the transition function δ only for reachable states:* $\delta(((S_0, m_0), \dots, (S_n, m_n), d), a) = ((S_0', m_0'), \dots, (S_n', m_n'), d')$ *where*
  - $$d'(q, j) = \begin{cases} 0 & \text{if } q \in \Delta(S_0, a) \text{ and } q \in F_j, \\ d(p, j) + 1 & \text{if } q \in \Delta(S_0, a),\ q \notin F_j, \text{ and } d(p, j) + 1 < b_j, \\ \bot & \text{if } q \in \Delta(S_0, a),\ q \notin F_j, \text{ and } d(p, j) + 1 = b_j, \\ \bot & \text{if } q \notin \Delta(S_0, a), \end{cases}$$
    where p is the unique (due to non-confluence, see Lemma 20.1) state in S_0 with $(p, a, q) \in \Delta$. Define $T = \{q \in Q \mid d'(q, j) \neq \bot \text{ for every } j\}$.
  - *For the update of the state sets consider the sequence $(S_0, m_0), \dots, (S_n, m_n)$ as a list containing tuples (S, m). Remark 19.2 yields that there are at most n non-empty sets S_i. First, we delete all elements of the list containing the empty set by moving the non-empty state sets to*

the left, without changing their order. Then, we replace every S_i by $\Delta(S_i,a)\cap T$. Finally, we append the state set $S_0\cap F$ to the end of the list. Denote the length of the updated list by ℓ. Now, we clean up states. For $i=0,\dots,\ell-1$ do: if $S_i\setminus F$ is a subset of $\bigcup_{i'=i+1}^{\ell-1} S_{i'}$ and $S_i\neq\emptyset$, then set $m'_i=1$, otherwise $m'_i=0$. Now, if $m_i=1$, then remove the states contained in S_i from every $S_{i'}$ with $i'>i$. As we have $\ell\le n+1$, we can retranslate the updated list into a unique state tuple $((S'_0,m'_0),\dots,(S'_n,m'_n))$ (if the list is too short, we pad it with $(\emptyset,0)$ at the end).

- *To define c consider a reachable state $q=((S_0,m_0),\dots,(S_n,m_n),d)$. Let e be the minimal i such that $S_i=\emptyset$ and let m be the minimal i such that $m_i=1$. Note that e is always defined for reachable states (due to Remark 19.2) and that $e\neq m$. We define*

$$c(q)=\begin{cases}1 & \text{if } e=0,\\ 2m & \text{if } m<e,\\ 2e-1 & \text{if } 0<e<m \text{ or if } m \text{ undefined}.\end{cases}$$

Note that in the definition of δ, cleaning up the sets might introduce new empty sets in the middle of the list. Also, note that p in the definition of d' is only well defined when considering reachable states. To prove the correctness of this construction, we need some properties of the states of $\mathfrak{P}$.

Remark 19. *Let $q'=((S_0,m_0),\dots,(S_n,m_n),d)$ be a reachable state of $\mathfrak{P}$.*

1. *$S_i\subseteq S_0$ for every i.*
2. *For every non-empty set S_i there is a state $q_i\in S_i$ such that $q_i\notin S_{i'}$ for every $i'>i$.*
3. *$S_0=\{q\in Q\mid d(q,j)\neq\bot$ for every $j\}$.*

To improve readability, we say that a finite or infinite run ρ satisfies $\mathcal{O}=\{(F_1,b_1),\dots,(F_k,b_k)\}\subseteq 2^Q\times\mathbb{N}_+$, if for every j we have that every infix of ρ of length b_j contains at least one state from F_j. Next, we show that $d(q,j)$ counts the time since the unique simulated run of $\mathfrak{A}$ ending in q has visited F_j.

Lemma 20. *Let $q'_0q'_1q'_2\dots$ be the run of $\mathfrak{P}$ on $w_0w_1w_2\dots\in\Sigma^\omega$ with $q'_t=((S^t_0,m^t_0),\dots,(S^t_n,m^t_n),d^t)$.*

1. *If $q_t\in S^t_i$, then there exists a (unique) finite run $q_0q_1\dots q_t$ of $\mathfrak{A}$ on $w_0w_1\dots w_{t-1}$ that satisfies $\mathcal{O}$.*
2. *Let $t_0<t_1$ be positions of $q'_0q'_1q'_2\dots$ and let i be in the range $0\le i\le n$ such that*
   - *$m^{t_0}_i=m^{t_1}_i=1$,*
   - *$S^t_i\neq\emptyset$ for every t in the range $t_0\le t\le t_1$, and*
   - *$m^t_{i'}=0$ and $S^t_{i'}\neq\emptyset$ for every t in the range $t_0\le t\le t_1$ and every $i'<i$.*

   Then, every finite run $q_{t_0}\dots q_{t_1}$ of $\mathfrak{A}$ on $w_{t_0}\dots w_{t_1-1}$ satisfying $q_t\in S^t_i$ for every t in the range $t_0\le t\le t_1$ visits a state in F at least once.
3. *Let $q_0q_1q_2\dots$ be a run of $\mathfrak{A}$ on $w_0w_1w_2\dots$ that satisfies $\mathcal{O}$. Then, we have $q_t\in S^t_0$ for every t.*

Proof. 1.) We show a stronger statement by induction over t: if $q_t\in S^t_i$ for some i, then there exists a finite run $q_0q_1\dots q_t$ of $\mathfrak{A}$ on $w_0w_1\dots w_{t-1}$ that satisfies $\mathcal{O}$ and for every j in the range $0\le j\le k$ we have $d^t(q_t,j)=\min\{t-t'\mid t'\le t$ and $q_{t'}\in F_j\}$ or (in case there is no such $q_{t'}\in F_j$) we have $d^t(q_t,j)=|q_0q_1\cdots q_t|=t+1$. Uniqueness of the run is then implied by non-confluence of $\mathfrak{A}$.

Due to Remark 19.1 it suffices to consider $i=0$. The claim holds for $t=0$ by definition of q'_0. Now, let $t>0$: as $q_t\in S^t_0$, there is a unique (due to non-confluence) state $q_{t-1}\in S^{t-1}_0$ such that $(q_{t-1},w_{t-1},q_t)\in\Delta$. Applying the inductive hypothesis, we obtain a run $q_0q_1\dots q_{t-1}$ of $\mathfrak{A}$ on $w_0w_1\dots w_{t-2}$ that satisfies $\mathcal{O}$ and $d^{t-1}(q_{t-1},j)=\min\{(t-1)-t'\mid t'\le t-1$ and $q_{t'}\in F_j\}$ or $d^{t-1}(q_{t-1},j)=|q_0q_1\cdots q_{t-1}|=t$. Furthermore, Remark 19.3 yields $d^t(q_t,j)<b_j$. We consider two cases: if $q_t\in F_j$, then $q_0q_1\cdots q_t$ satisfies $\mathcal{O}$

and we have $d^t(q_t, j) = 0$, by definition of d^t, which is equal to $\min\{t - t' \mid t' \le t \text{ and } q_{t'} \in F_j\}$. Now, suppose $q_t \notin F_j$. Then, we have $d^{t-1}(q_{t-1}, j) < b_j - 1$, since we have $d^t(q_t, j) = d^{t-1}(q_{t-1}, j) + 1 < b_j$ by the definition of d^t in case $q_t \notin F_j$. We consider the two choices for the value of $d^{t-1}(q_{t-1}, j)$. If $d^{t-1}(q_{t-1}, j) = \min\{(t-1) - t' \mid t' \le t-1 \text{ and } q_{t'} \in F_j\} < b_j - 1$, then the suffix of $q_0q_1 \cdots q_{t-1}$ of length $b_j - 1$, contains a vertex from F_j. Thus, also the suffix of $q_0q_1 \cdots q_t$ of length b_j contains a vertex from F_j and hence $d^t(q_t, j) = \min\{(t-1) - t' \mid t' \le t-1 \text{ and } q_{t'} \in F_j\} + 1 = \min\{t - t' \mid t' \le t \text{ and } q_{t'} \in F_j\}$ and $q_0q_1 \cdots q_t$ satisfies $\mathcal{O}$, since the induction hypothesis applies to every infix but the last one, which has a vertex from F_j. Otherwise, if $d^{t-1}(q_{t-1}, j) = |q_0q_1 \cdots q_{t-1}| = t < b_j - 1$, then $d^t(q_t, j) = t + 1 = |q_0q_1 \cdots q_t|$ by definition of d^t. Then, $q_0q_1 \cdots q_t$ trivially satisfies $\mathcal{O}$, as it has no infix of length b_j.

2.) We assume $q_{t_1} \notin F$, since we are done otherwise. We have $q_{t_0} \notin S_{i'}^{t_0}$ for every $i' > i$, due to $m_i^{t_0} = 1$, which means all states from $S_i^{t_0}$ are deleted from the sets $S_{i'}^{t_0}$ for every $i' > i$. Let t' in the range $t_0 < t' \le t_1$ be the first position such that $q_{t'} \in \bigcup_{i'=i+1}^{n} S_{i'}^{t'}$. Such a position exists, as we have $m_i^{t_1} = 1$, which implies $q_{t_1} \in S_{i'}^{t_1}$ for some $i' > i$. Since $q_{t'} \in S_{i'}^{t'}$, either $q_{t'} \in \Delta(S_{i'}^{t'-1}, w_{t'-1})$ or $q_{t'} \in \Delta(S_0^{t'-1}, w_{t'-1}) \cap F$. Thus, it suffices to derive a contradiction in the first case: $q_{t'} \in \Delta(S_{i'}^{t'-1}, w_{t'-1})$ implies the existence of a $p \in S_{i'}^{t'-1}$ such that $(p, w_{t'-1}, q_{t'}) \in \Delta$. We have $p \ne q_{t'-1}$ due to the minimality of the position t'. But then Lemma 20.1 yields two different runs of $\mathfrak{A}$ from q_0 to $q_{t'}$ on $w_0 \dots w_{t'-1}$, which gives the desired contradiction to the non-confluence of $\mathfrak{A}$.

3.) Again, we show a stronger statement by induction over t: let $q_0q_1q_2\dots$ be a run of $\mathfrak{A}$ on $w_0w_1w_2\dots$ that satisfies $\mathcal{O}$. Then, for every t we have $q_t \in S_0^t$ and for every j in the range $1 \le j \le k$ we have $d^t(q_t, j) = \min\{t - t' \mid t' \le t \text{ and } q_{t'} \in F_j\}$ or (in case there is no such $q_{t'} \in F_j$) we have $d^t(q_t, j) = |q_0q_1 \cdots q_t| = t + 1$. Note that this statement is only well-defined for a non-confluent automaton.

The induction start $t = 0$ follows from the definition of q_0'. Now, let $t > 0$: the induction hypothesis yields $q_{t-1} \in S_0^{t-1}$ and for every j in the range $1 \le j \le k$ we have $d^{t-1}(q_{t-1}, j) = \min\{(t-1) - t' \mid t' \le t-1 \text{ and } q_{t'} \in F_j\}$ or $d^{t-1}(q_{t-1}, j) = |q_0q_1 \cdots q_{t-1}| = t$. We consider two cases. If $q_t \in F_j$, then we have $q_t \in S_0^t$ and $d^t(q_t, j) = 0$, which is equal to $\min\{t - t' \mid t' \le t \text{ and } q_{t'} \in F_j\}$ by definition of S_0^t and d^t. Otherwise, if $q_t \notin F$, then we have $d^{t-1}(q_{t-1}, j) < b_j - 1$, by induction hypothesis and the fact that $q_0q_1 \cdots q_{t-1}$ satisfies $\mathcal{O}$. Due to Remark 19.3, it suffices to show $d^t(q_t, j) < b_j$. We consider the two choices for the value of $d^{t-1}(q_{t-1}, j)$. If $d^{t_1}(q_{t-1}, j) = \min\{(t-1) - t' \mid t' \le t-1 \text{ and } q_{t'} \in F_j\} < b_j - 1$, then $d^t(q_t, j) = \min\{(t-1) - t' \mid t' \le t-1 \text{ and } q_{t'} \in F_j\} + 1 = \min\{t - t' \mid t' \le t \text{ and } q_{t'} \in F_j\} < b_j$. On the other hand, if $d^{t-1}(q_{t-1}, j) = |q_0q_1 \cdots q_{t-1}| = t < b_j - 1$, then $d^t(q_t, j) = t + 1 = |q_0q_1 \cdots q_t| < b_j$. ☐

We are now able to prove the correctness of Construction 18. Our proof proceeds along the lines of the proof for the original construction without counters [9].

Lemma 21. *Let $\mathfrak{A} = (Q, \Sigma, q_0, \Delta, F)$ be a non-confluent Büchi automaton, let $\{(F_1, b_1), \dots, (F_k, b_k)\} \subseteq 2^Q \times \mathbb{N}_+$, and let $\mathfrak{P}$ be the deterministic parity automaton obtained from Construction 18.*

1. *$\mathfrak{P}$ accepts w if and only if $\mathfrak{A}$ has an accepting run ρ on w such that every F_j is visited at least once in every infix of ρ of length b_j.*
2. *$|\mathfrak{P}| \le 2^{(|\mathfrak{A}|+1)^2} \cdot \left(\Pi_{j=1}^{k}(b_j + 1)\right)^{|\mathfrak{A}|}$ and $|c(Q')| = 2|\mathfrak{A}| + 1$.*

Proof. 1.) Let $q_0'q_1'q_2'\dots$ be an accepting run of $\mathfrak{P}$ on w, with $q_t' = ((S_0^t, m_0^t), \dots, (S_n^t, m_n^t), d^t)$. Then, there exists a position t_0 and an i such that $c(q_t') = 2i$ for infinitely many t and $c(q_t') \ge 2i$ for every $t \ge t_0$. Thus, $S_{i'}^t \ne \emptyset$ for every $t \ge t_0$ and every $i' \le i$ and $m_{i'}^t = 0$ for every $t \ge t_0$ and every $i' < i$. Since S_i^{t+1} is a non-empty subset of $\Delta(S_i^t, w_t)$ for every $t \ge t_0$, König's Lemma yields an infinite run $q_{t_0}q_{t_0+1}q_{t_0+2}\dots$

(not necessarily starting in an initial state) of $\mathfrak{A}$ on $w_{t_0} w_{t_0+1} w_{t_0+2} \dots$ such that $q_t \in S_i^t$ for every $t \geq t_0$. Furthermore, there exists a finite run of $\mathfrak{A}$ on $w_0 \dots w_{t_0-1}$ starting in an initial state and ending in q_{t_0} due to Lemma 20.1. These runs can be concatenated to an infinite run $q_0 q_1 q_2 \dots$ of $\mathfrak{A}$ on w such that $q_t \in S_0^t$ for every t. Hence, $q_0 q_1 q_2 \dots$ satisfies $\mathscr{O}$ due to Lemma 20.1. Let $t_1 < t_2 < t_3 < \cdots$ be the positions after t_0 such that $c(q'_{t_s}) = 2i$, i.e., $m_i^{t_s} = 1$. The run $q_0 q_1 q_2 \dots$ is accepting due to Lemma 20.2, as the run visits an accepting state in between any t_s and t_{s+1}, of which there are infinitely many.

Now, let $q_0 q_1 q_2 \dots$ be an accepting run of $\mathfrak{A}$ on w that satisfies $\mathscr{O}$ and let $q'_0 q'_1 q'_2 \dots$ be the run of $\mathfrak{P}$ on w with $q'_t = ((S_0^t, m_0^t), \dots, (S_n^t, m_n^t), d^t)$. We have $q_t \in S_0^t$ for every t due to Lemma 20.3. Assume there are only finitely many t such that $m_0^t = 1$. Then, there is a minimal index i_1 such that an infinite suffix of $q_0 q_1 q_2 \dots$ is tracked by S_{i_1} and $S_{i'}^t \neq \emptyset$ for every $i' \leq i_1$ from some point onwards. This is due to the fact that for every t with $q_t \in F$ the set $S_0 \cap F$ (which contains q_t) is appended to the list of state sets. Furthermore, this set can be moved to the left (in case other sets are empty) only a finite number of times. Finally, if the state q_t is deleted from this set, then there is a smaller set which tracks this run, for which the same reasoning applies. Again, assume there are only finitely many t such that $m_{i_1}^t = 1$. Then, there exists a minimal index $i_2 > i_1$ such that an infinite suffix of $q_0 q_1 q_2 \dots$ is tracked by S_{i_2} and $S_{i'}^t \neq \emptyset$ for every $i' \leq i_2$ from some point onwards. This can be iterated until we have that the sets S_{n-1}^t track the suffix of $q_0 q_1 q_2 \dots$ and all smaller sets are always non-empty. But as S_{n-1}^t is in this situation always a singleton (see Remark 19.2), it gets marked every time an accepting state is visited by $q_0 q_1 q_2 \dots$. Hence, the run of $\mathfrak{P}$ on w is accepting.

2.) Clear. □

The Lemmata 16, 17, and 21 imply the existence of a deterministic parity automaton with the properties required in Lemma 14. Hence, this finishes the proof of Theorem 10. To compute a finite-state strategy realizing the optimal value (witnessed by a valuation α) in a PLTL$_\mathbf{F}$ game with winning condition φ, one has to compute a deterministic parity automaton recognizing the ω-words w satisfying φ_α, as explained above Lemma 14. Dually, in a PLTL$_\mathbf{G}$ game with winning condition φ, one computes a deterministic parity automaton recognizing the ω-words w satisfying $\neg\varphi_\alpha$, which is then complemented by incrementing the priorities. This complement automaton is a memory structure for the PLTL$_\mathbf{G}$ game.

5 Conclusion

We presented **2EXPTIME**-algorithms for computing optimal strategies in a PLTL game and to determine whether a given player wins with respect to some, infinitely many, or all variable valuations. The decision problems for PROMPT–LTL and PLTL (with the exception of the finiteness problem for PLTL) are decidable by solving a single LTL game of the same size. Hence, adding parameterized operators does not increase the asymptotic computational complexity of solving these games. Furthermore, even the optimization problems for unipolar games can be solved in doubly-exponential time, so they are of the same computational complexity as solving LTL games. However, it takes an exponential number of parity games to solve to determine an optimal strategy. It is open whether this can be improved.

An interesting open question concerns the tradeoff between the size of a finite-state strategy and the quality of the bounds it is winning for.

Acknowledgments. The author wants to thank Marcin Jurdziński, Christof Löding, Andreas Morgenstern, and Wolfgang Thomas for helpful discussions, and Roman Rabinovich for coming up with the name *blinking semantics*. Also, valuable comments on an earlier paper [14] by anonymous referees are gratefully acknowledged.

References

[1] Rajeev Alur, Kousha Etessami, Salvatore La Torre & Doron Peled (2001): *Parametric Temporal Logic for "Model Measuring"*. *ACM Trans. Comput. Log.* 2(3), pp. 388–407. Available at `http://doi.acm.org/10.1145/377978.377990`.

[2] Christel Baier & Joost-Pieter Katoen (2008): *Principles of Model Checking*. The MIT Press.

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

[4] Barbara Di Giampaolo, Salvatore La Torre & Margherita Napoli (2010): *Parametric Metric Interval Temporal Logic*. In Adrian Horia Dediu, Henning Fernau & Carlos Martín-Vide, editors: *LATA*, *Lecture Notes in Computer Science* 6031, Springer, pp. 249–260, doi:`10.1007/978-3-642-13089-2_21`. Available at `http://dx.doi.org/10.1007/978-3-642-13089-2`.

[5] Marcin Jurdzinski (2000): *Small Progress Measures for Solving Parity Games*. In Horst Reichel & Sophie Tison, editors: *STACS*, *Lecture Notes in Computer Science* 1770, Springer, pp. 290–301, doi:`10.1007/3-540-46541-3_24`.

[6] Orna Kupferman, Nir Piterman & Moshe Y. Vardi (2009): *From Liveness to Promptness*. *Formal Methods in System Design* 34(2), pp. 83–103. Available at `http://dx.doi.org/10.1007/s10703-009-0067-z`.

[7] Orna Kupferman & Moshe Y. Vardi (1998): *Freedom, Weakness, and Determinism: From Linear-Time to Branching-Time*. In: *LICS*, pp. 81–92, doi:`10.1007/s10703-009-0067-z`.

[8] Andreas Morgenstern (2010): *Symbolic Controller Synthesis for LTL Specifications*. Ph.D. thesis, Department of Computer Science, University of Kaiserslautern, Germany, Kaiserslautern, Germany.

[9] Andreas Morgenstern & Klaus Schneider (2008): *From LTL to Symbolically Represented Deterministic Automata*. In Francesco Logozzo, Doron Peled & Lenore D. Zuck, editors: *VMCAI*, *Lecture Notes in Computer Science* 4905, Springer, pp. 279–293, doi:`10.1007/978-3-540-78163-9_24`.

[10] Andrzej Mostowski (1991): *Games with Forbidden Positions*. Technical Report 78, University of Gdańsk.

[11] Amir Pnueli & Roni Rosner (1989): *On the Synthesis of a Reactive Module*. In: *POPL*, pp. 179–190. Available at `http://doi.acm.org/10.1145/75277.75293`.

[12] Amir Pnueli & Roni Rosner (1989): *On the Synthesis of an Asynchronous Reactive Module*. In Giorgio Ausiello, Mariangiola Dezani-Ciancaglini & Simona Ronchi Della Rocca, editors: *ICALP*, *Lecture Notes in Computer Science* 372, Springer, pp. 652–671, doi:`10.1007/BFb0035790`.

[13] A. Prasad Sistla & Edmund M. Clarke (1985): *The Complexity of Propositional Linear Temporal Logics*. *J. ACM* 32(3), pp. 733–749. Available at `http://doi.acm.org/10.1145/3828.3837`.

[14] Martin Zimmermann (2010): *Parametric LTL Games*. Technical Report AIB 2010-20, RWTH Aachen University. Available at `http://aib.informatik.rwth-aachen.de/2010/2010-20.ps.gz`.