Two-Player Boundedness Counter Games

Emmanuel Filiot ✉
Université libre de Bruxelles, Belgium

Edwin Hamel-de le Court ✉
Université libre de Bruxelles, Belgium

Abstract

We consider two-player zero-sum games with winning objectives beyond regular languages, expressed as a parity condition in conjunction with a Boolean combination of boundedness conditions on a finite set of counters which can be incremented, reset to 0, but not tested. A boundedness condition requires that a given counter is bounded along the play. Such games are decidable, though with non-optimal complexity, by an encoding into the logic WMSO with the unbounded and path quantifiers, which is known to be decidable over infinite trees. Our objective is to give tight or tighter complexity results for particular classes of counter games with boundedness conditions, and study their strategy complexity. In particular, counter games with conjunction of boundedness conditions are easily seen to be equivalent to Streett games, so, they are CoNP-c. Moreover, finite-memory strategies suffice for Eve and memoryless strategies suffice for Adam. For counter games with a disjunction of boundedness conditions, we prove that they are in solvable in NP∩CoNP, and in PTime if the parity condition is fixed. In that case memoryless strategies suffice for Eve while infinite memory strategies might be necessary for Adam. Finally, we consider an extension of those games with a max operation. In that case, the complexity increases: for conjunctions of boundedness conditions, counter games are EXPTIME-c.

2012 ACM Subject Classification Theory of computation → Logic and verification; Theory of computation → Automata over infinite objects

Keywords and phrases Controller synthesis, Game theory, Counter Games, Boundedness objectives

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2022.21

Related Version *Full Version*: `https://hal.archives-ouvertes.fr/hal-03626782` [21]

Funding This work was supported by the Fonds de la Recherche Scientifique – F.R.S.-FNRS under the MIS project F451019F. Emmanuel Filiot is a senior research associate at F.R.S.-FNRS.

Acknowledgements We warmly thank Jean-François Raskin for his valuable inputs.

1 Introduction

Games on graphs are a popular mathematical framework to reason on reactive synthesis problems [2, 9]: the system to synthesize is seen as a protagonist which must enforce a given specification (its winning objective) against any adversarial behaviour of its environment. In this framework, executions of reactive systems are modelled as infinite sequences alternating between actions of the systems and actions of its environment. In the ω-regular setting, the set of correct executions of reactive systems is modelled as an automaton, for example, a non-deterministic Büchi automaton, then determinized into a parity automaton. The synthesis problem then boils down to solving a game played on the graph of the parity automaton, where the goal of the protagonist (Eve) is to satisfy, in the long run, the parity condition whatever her opponent (Adam) does. Motivated by the synthesis of more complex systems, the literature is rich in extensions of this basic two-player zero-sum ω-regular setting: multiple players, imperfect information, quantitative objectives, infinite graphs ... (see [2, 9] for some references). In this paper, we follow this line of work and consider an extension of two-player games beyond ω-regularity: counter games with boundedness conditions.

33rd International Conference on Concurrency Theory (CONCUR 2022).
Editors: Bartek Klin, Sławomir Lasota, and Anca Muscholl; Article No. 21; pp. 21:1–21:23
Leibniz International Proceedings in Informatics
Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

Counter games. In this paper, a *two-player counter game with boundedness objectives*, only called counter game hereafter, is given by a finite arena, called counter arena, whose vertices are labelled by counter operations over a finite set of counters C. Those operations can: increment a counter, reset it, or skip it (i.e. leave its value unchanged). We consider objectives given as Boolean combinations of *counter boundedness conditions*. For $c \in C$, the condition $\mathbb{B}(c)$ is satisfied by all infinite paths $\pi = v_0 v_1 \dots$, called plays, such that for some $N \in \mathbb{N}$, the value of c along π is bounded by N. Note that the bound N is not uniform, in the sense that it depends on π, and as a consequence, the set of plays satisfying $\mathbb{B}(c)$ is not ω-regular in general. In this paper, we consider particular classes of Boolean combinations of boundedness conditions. Since they do not necessarily capture all ω-regular objectives, we also, by default, equip counter games with a parity condition.

Given an objective W, as a Boolean formula Φ over atoms $\mathbb{B}(c)$ for all $c \in C$, the goal of the protagonist, Eve, is to enforce plays which satisfy W and the parity condition, whatever the adversary, Adam, does. If she has a strategy to meet this objective, she is said to win the game. Counter games are zero-sum, meaning that the goal of Adam is to enforce the complementary objective. The goal of this paper is to build a fine understanding of counter games, by studying the problem of deciding the winner for important classes of counter games.

Motivations. On infinite words, classes of counter automata with boundedness conditions have appeared in various papers, e.g. in [6, 15, 3, 8]. The most relevant models in the context of counter games are the ωBS-automata of [6] and the max-automata of [8]. They are equipped with the same counter operations as the counter games of this paper, plus a max operation in the case of max-automata, and some boundedness conditions. As a consequence, winning objectives in counter games can naturally be expressed with these automata. However, while they are known to have decidable emptiness problem, not much is known when they are used to define objectives in two-player games. A motivation for this paper is to investigate this question, for games where the winning conditions is not given by such an automaton but where counter operations are explicitly given in the arena.

In the same line of works, max-automata, which are deterministic, are known by [3] to correspond to the logic WMSO+U, which extends weak MSO on infinite words with the *unbounded quantifier* $\mathbb{U}X$. A formula $\mathbb{U}X.\phi(X)$ holds if there are arbitrarily large sets X satisfying ϕ. An important and strong result by Bojańczyk states that the extension of WMSO+U to infinite binary trees and with a path quantifier which allows to quantify over infinite paths, has decidable satisfiability problem [7]. Since strategies are definable, modulo a tree encoding, in this latter logic, a direct consequence of this result is that two-player games with objectives given by max-automata are decidable (see also Example 2 of [7]). As a consequence, counter games with boundedness conditions are decidable, though with non-elementary complexity. We aim here at providing conceptually simpler arguments and insights to prove decidability (with tighter complexity results), for particular instances of boundedness conditions, instead of using the general result of [7].

Contributions. Our contributions are summarized in Fig. 1. We consider objectives given as a conjunction of a parity condition and a formula over atoms $\mathbb{B}(c)$ in the following classes: conjunctions, disjunctions, disjunctions of conjunctions, and negation-free formulas. We also consider the extension of counter games with a max operator which can assign a counter with the maximal value of several counters. The table also mentions the strategy complexity. For conditions in $\bigwedge \mathbb{B}$, counter games are easily proved to be interreducible in polynomial time to Streett games, yielding CoNP-completeness [22]. More interestingly, we prove that when the number of counters is fixed, then, they are interreducible to parity games in polynomial time, using another reduction (Thm 5).

We then prove, in it is our main contribution, that for conditions in $\bigvee \mathbb{B}$, counter games are solvable in NP∩CoNP and in polynomial time when the index of the parity function is fixed. To prove this result, we introduce the notion of *finitely switching strategies* which are, to the best of our knowledge new, and we believe, interesting on their own. This notion is specifically designed for disjunctions of prefix-independent objectives (which is the case of counter boundedness conditions): in a finitely switching strategy, Eve announces which objective from the disjunction she aims to satisfy, and she can change her mind along the play, but only a finite number of times. Eventually, she is bound to satisfy one the objectives. We give general conditions to decide whether Eve has a finitely switching strategy in a two-player game with a disjunction of prefix-independent objectives, and prove that such strategies are sufficient for Eve to win objectives in $\bigvee \mathbb{B}$ and more generally in $\bigvee \bigwedge \mathbb{B}$.

Related works. Two-player games with boundedness conditions have been studied in the literature, first as finitary parity and Streett games [12], then generalized to cost-parity and cost-Streett games [20]. Finitary parity- and Streett-games are request-response games [13], with the additional constraint that the delay (number of edges) between a request and its response is bounded (by a bound which depends on the play). For cost-parity and cost-Streett, instead of the number of edges, costs (including 0) label edges and the delay is defined as the sum of the costs. Cost-parity and cost-Streett games can be encoded as counter games with conditions in $\bigwedge \mathbb{B}$, though with an exponential blowup. The difference between those counter games and finitary- and cost-games can be seen in their complexity: counter games with conditions in $\bigwedge \mathbb{B}$ are CoNP-c, finitary parity games are in PTime, cost-parity in NP∩CoNP, and finitary Streett and cost-Streett are ExpTime-c.

Delay games with objectives given by a max-automaton have been proved to be decidable in [26]. This result is orthogonal to ours: first, those games allow for some delay, here in the sense that Eve has some look-ahead on Adam's future actions. Second, the decision procedure is non-elementary and rely on an encoding into WMSO+UP on infinite trees, some argument we avoid here, but for less expressive boundedness objectives.

Infinite-state games with boundedness conditions have been considered in [11], over pushdown arenas. Finitary games over these arenas are shown to be decidable, as well as (pushdown) counter games with conditions in $\bigwedge \mathbb{B}$, without complexity results. Interestingly, it is shown that those games are equivalent to games where the objective of Eve is to uniformly bound all counters, for a bound which only depends on her strategy, and not on the plays. For counter games in $\bigwedge \mathbb{B}$ over a *finite* arena, this result can easily be seen as a consequence that finite-memory strategies suffice for Eve.

Last but not least, counter boundedness games have appeared implicitly in some existing works on synthesis [1, 19], though the classes considered in these papers are less general and solved using specific techniques. In [19], the authors consider a synthesis problems over infinite alphabets of data. In particular, they study the problem of synthesising Mealy machines with *registers* satisfying specifications given as deterministic register automata over $(\mathbb{N}, <, 0)$. It is shown that this problem is decidable in 2ExpTime, and, even though the decidability proof is not based on counter games, it is proved that the synthesis problem reduces to a game with winning conditions given as a (deterministic) max-automaton whose acceptance is a disjunction of a parity condition and a disjunction of conditions of the form "counter c is unbounded". Although the main technical difficulty in [19] is to prove this reduction, based on it, our results on counter games with max operation yields an alternative procedure to decide the former synthesis problem (with same complexity).

Winning objective parity$\wedge$	Complexity	Memory of Eve	Memory of Adam	Theorem
$\bigwedge \mathbb{B}$	coNP-c	Finite	none	Th 3
$\bigvee \mathbb{B}$	NP $\cap$ CoNP PTime for fixed index	Parity Index	Infinite	Th 12
$\bigvee \bigwedge \mathbb{B}$	coNP-c	Finite	Infinite	Th 13
$\mathrm{Bool}^+(\mathbb{B})$	PSPACE, CoNP-h	Finite	Infinite	Th 14
$\bigwedge \mathbb{B} + max$	EXPTIME-c	Finite	Finite	Th 15
$\mathrm{Bool}(\mathbb{B}) + max$	Decidable	Infinite	Infinite	from [5]

Figure 1 Complexity of deciding whether Eve has a winning strategy in a counter game for various winning objectives, always taken in conjunction with a parity objective. $\mathrm{Bool}^+(\mathbb{B})$ means any negation-free Boolean combination of objectives of the form $\mathbb{B}(c)$. Hardness results hold for any parity function of fixed constant index. The notation $+max$ indicates that counter games are also equipped with a max operation. Since counter games with boundedness objectives are determined, this yields the complexity of deciding whether Eve wins for the complementary objectives: for example, it is NP-c for objectives parity $\vee \bigvee \mathbb{U}$ and memoryless strategies are sufficient for Eve, and in PTIME for parity $\vee \bigwedge \mathbb{U}$ but infinite memory might be necessary for Eve.

The work of [1] considers a parameterized synthesis problem called the population control problem. In this problem, an arbitrary number of processes execute the same NFA, with the goal of reaching an accepting state. The controller picks an action (a letter) common to all of them, while the adversary resolves non-determinism for each of them individually. The problem is to decide whether "controller wins for any number of processes". It is shown that this problem reduces to a finite graph game with a condition of the form "if the play has bounded capacity – where this bound depends on the play –, then the play satisfies some reachability condition" (see Sec 3.2 and Lemma 9 of [1]). Though the authors show that this condition can be equivalently replaced by an ω-regular one, it could also be directly encoded as a counter boundedness condition (with max operation). Our results combined with this reduction would however not provide the optimal complexity found in [1].

While our results on counter games do not provide new decidability results (nor better complexities) with respect to the two applications mentioned before, these two applications show that counter games with boundedness conditions arise naturally in synthesis problems, motivating our general study.

2 Preliminaries

For any set Σ, we denote by Σ^* (Σ^ω) the finite (infinite) sequences of elements of Σ.

Two-player arenas. A two-player *arena* is a tuple $\mathcal{A} = (V, E, V_\exists, V_\forall, v_0)$, where V is finite set, $E \subseteq V \times V$, and $V_\exists$ and $V_\forall$ are two subsets of V such that $\{V_\exists, V_\forall\}$ is a partition of V, and v_0 is an initial vertex. In this paper, we assume that arenas are deadlock-free, *i.e.* that for any $v \in V$, there exists $v' \in V$ such that $(v, v') \in E$. Given $v \in V$, we denote $\mathcal{A}[v] = (V, E, V_\exists, V_\forall, v)$ the arena $\mathcal{A}$ where v_0 has been substituted by v. A play ρ of $\mathcal{A}$ is a mapping from $\mathbb{N}$ to V such that $(\rho(i), \rho(i+1)) \in E$, for all integers $i \in \mathbb{N}$. The set of plays is denoted by $\mathrm{Plays}(\mathcal{A})$. Any play can also be seen as an element of V^ω, and we call a *history* any finite prefix of a play, and denote by $\mathrm{Hist}(\mathcal{A})$ the set of histories of $\mathcal{A}$.

Strategies and finite-memory. A *strategy* for Eve (resp. Adam) is a function σ from $\mathrm{Hist}(\mathcal{A})$ to V defined for all histories $h = h_0 \cdots h_n$ with $h_n \in V_\exists$ (resp. $h_n \in V_\forall$), and such that $(h_n, \sigma(h)) \in E$. A play ρ is *consistent* with a strategy for Eve (resp. Adam) if, for any integer n such that $\rho(n) \in V_\exists$ (resp. $\rho(n) \in V_\forall$), σ is defined on $\rho(0) \cdots \rho(n)$, and $\rho(n+1) = \sigma(\rho(0) \cdots \rho(n))$. We let $\mathrm{Plays}(\mathcal{A}, \sigma)$ (or just $\mathrm{Plays}(\sigma)$ when $\mathcal{A}$ is clear from the context) the set of plays consistent with σ.

A strategy σ of Eve (resp. Adam) is said to be *finite-memory* if there exists a finite set M, an element $m_I \in M$, a mapping δ from $V \times M$ to V, and a mapping g from $V \times M$ to M such that the following is true. When $h = v_0 v_1 \cdots v_l$ is a prefix of a play consistent with σ such that $v_l \in V_\exists$ (resp. $v_l \in V_\forall$), and the sequence $m_0, m_1, ..., m_l$ is determined by $m_0 = m_I$ and $m_{i+1} = g(v_i, m_i)$, then $\sigma(w) = \delta(v_l, m_l)$. In that case, we say that (δ, g) is a *memory mapping pair* of σ, and that m_l is the memory state of g at move l. We also say that σ is of memory $|M|$, and *memoryless* if it is of memory 1. Note that a memoryless strategy can just be identified with a mapping from V to V.

Two-player games. A *winning condition* for $\mathcal{A}$ is a subset $W \subseteq V^\omega$. A *strategy* σ of Eve or Adam is said to be winning for objective W if $\mathrm{Plays}(\sigma) \subseteq W$. A *two-player game* is a pair $G = (\mathcal{A}, W)$ where $\mathcal{A}$ is an arena and W is a winning condition. We say that a strategy (of Eve or Adam) is winning in G if it is winning for W. A game $G = (\mathcal{A}, W)$ is *determined* if either Eve wins G or Adam wins $(\mathcal{A}, V^\omega \backslash W)$.

In this paper, we consider the problem of deciding, given a game G with a finitely represented winning condition, whether Eve wins G. For a complexity class $\mathcal{C}$ and a class of games $\mathcal{G}$, we say that games in $\mathcal{G}$ are in $\mathcal{C}$ (resp. $\mathcal{C}$-hard, $\mathcal{C}$-complete) if the latter problem for games $G \in \mathcal{G}$ is in $\mathcal{C}$ (resp. $\mathcal{C}$-hard, $\mathcal{C}$-complete).

We also consider the complexity of strategies sufficient or necessary for Eve and Adam to win a game. We say that finite-memory strategies are *sufficient* for Eve (resp. Adam) to win $\mathcal{G}$ if for all $G \in \mathcal{G}$, whenever Eve (resp. Adam) wins G, she has (resp. he has) a finite-memory winning strategy in G. We say that finite-memory is necessary for Eve (resp. Adam) to win $\mathcal{G}$ if memoryless strategies do not suffice for Eve (resp. Adam) to win $\mathcal{G}$. Finally, we say that infinite-memory is necessary for Eve (resp. Adam) to win $\mathcal{G}$ if finite-memory strategies do not suffice for Eve (resp. Adam) to win $\mathcal{G}$.

Parity games. Let $\mathcal{A}$ be an arena with set of vertices V. Let $Q \subseteq \mathbb{N}$ be a finite set of elements called *colours* and $\kappa : V \to Q$ a mapping from vertices to colours called *parity function* or *priority function*. The size $|Q|$ of Q is called the *index* of κ. The mapping κ defines a winning condition denoted $\mathrm{Parity}(\kappa)$, called a *parity condition*, as follows: $\mathrm{Parity}(\kappa)$ is the set of all infinite words $w = w_0 w_1 \cdots \in V^\omega$ such that the greatest colour occurring an infinitely often in $\kappa(w_0)\kappa(w_1)\cdots$ is even. A parity game is a game whose winning condition is a parity condition. We refer to $\mathcal{A}' = (\mathcal{A}, Q, \kappa)$ as a *coloured arena*, and also denote $\mathrm{Parity}(\kappa)$ as $\mathrm{Parity}(\mathcal{A}')$ to avoid an explicit mention of the colouring κ. Note that a coloured arena $\mathcal{A}' = (\mathcal{A}, Q, \kappa)$ uniquely defines a parity game $G = (\mathcal{A}, \mathrm{Parity}(\mathcal{A}'))$. It is well-known that parity games are in $\mathrm{NP} \cap \mathrm{CoNP}$ [17], and even solvable in quasi-polynomial time [10].

Counter operations. Our goal is now to define counter games. First, we introduce counter operations and their semantics. In the rest of the paper, we fix a countable set $\mathcal{C}$ whose elements are called *counters*. A *counter operation* is a mapping from a finite subset C of $\mathcal{C}$ to $\{\mathsf{i}, \mathsf{r}, \mathsf{skip}\}$. We let $\mathsf{Op}(C)$ denote the set of counter operations over $C \subseteq \mathcal{C}$. A *counter valuation* is a mapping ν from C to $\mathbb{N}$. For any infinite word $w \in \mathsf{Op}(C)^\omega$, we define $\lambda(w)$

as the infinite sequence of counter valuations $\nu_0, \nu_1, \nu_2, \dots$ such that for any counter $c \in C$, $\nu_0(c) = 0$ and for any non-negative integer n, $\nu_{n+1}(c) = \nu_n(c) + 1$ if $w_n(c) = \mathsf{i}$, $\nu_{n+1}(c) = 0$ if $w_n(c) = \mathsf{r}$ and $\nu_{n+1}(c) = \nu_n(c)$ if $w_n(c) = \mathsf{skip}$. We define $\lambda(w)$ for $w \in \mathsf{Op}(C)^*$. To ease notations, we write $\lambda(w, c)_i$ instead of $\lambda(w)_i(c)$. We say that λ is the *evaluation* of w.

Counter games with boundedness objectives. Let $\mathcal{A}'$ be an arena with set of vertices V, $C \subseteq \mathcal{C}$ a finite set of counters, and $\zeta : V \to \mathsf{Op}(C)$ a mapping from vertices to counter operations, called *vertex labeling*. Let Q be a set of colours and $\kappa : V \to Q$ be a colouring of V. To avoid cumbersome notations, for any vertex $v \in V$ and counter $c \in C$, we let $\zeta_c(v)$ denote $(\zeta(v))(c)$. We refer to $\mathcal{A} = (\mathcal{A}', C, \zeta, Q, \kappa)$ as a *counter arena*, to $\mathcal{A}'$ as its *underlying arena* and to $(\mathcal{A}, Q, \kappa)$ as its underlying coloured arena. We let $\text{Parity}(\mathcal{A}) = \text{Parity}(\kappa)$.

We consider a particular type of winning objective for counter games, called boundedness conditions, always together with a parity condition. Let $c \in C$. We let $\mathbb{B}(c)$ be an atomic formula which intuitively requires that counter c is bounded along a play, by some constant. Formally, $\mathbb{B}(c)$ is interpreted in $\mathcal{A}$ by the set of plays ρ of $\mathcal{A}$, denoted $\mathit{Plays}(\mathcal{A}, \mathbb{B}(c))$, such that the sequence $\lambda(\zeta(\rho), c)$ is bounded, *i.e.*

$$\mathit{Plays}(\mathcal{A}, \mathbb{B}(c)) = \{\rho \in \text{Plays}(\mathcal{A}) \mid \exists N \in \mathbb{N}, \forall n \in \mathbb{N}, \lambda(\zeta(\rho), c)_n \leq N\}$$

The set $\mathit{Plays}(\mathcal{A}, \mathbb{B}(c))$ is called a *boundedness condition*. To ease readability, we may just write $\mathbb{B}(c)$ to denote $\mathit{Plays}(\mathcal{A}, \mathbb{B}(c))$ when $\mathcal{A}$ is clear from the context. We let $\mathbb{U}(c)$ as a shortcut for $\neg\mathbb{B}(c)$. A *counter condition* for $\mathcal{A}$ is a Boolean formula ϕ over the set of propositions $\{\mathbb{B}(c) \mid c \in C\}$. Its interpretation $\mathit{Plays}(\mathcal{A}, \phi) \subseteq \text{Plays}(\mathcal{A})$ over $\mathcal{A}$ is defined naturally.

Given a counter condition ϕ, the pair $G = (\mathcal{A}, \phi)$ is called a *counter game*. The game *induced* by $G = (\mathcal{A}, \phi)$ is the game $G_\phi = (\mathcal{A}', \mathit{Plays}(\mathcal{A}, \phi) \cap \text{Parity}(\mathcal{A}))$, where $\mathcal{A}'$ is the underlying arena of $\mathcal{A}$. Note that in a counter game, both the counter condition and the parity condition must be satisfied. The notion of strategies and winning strategies carry over to counter games by considering the games they induce. In particular, Eve wins G if she wins G_ϕ, i.e., she has a strategy winning for the objective $\mathit{Plays}(\mathcal{A}, \phi) \cap \text{Parity}(\mathcal{A})$.

In this paper, we consider several classes of counter conditions. The class of counter conditions of the form $\bigwedge_{c \in C} \mathbb{B}(c)$ for some finite set $C \subseteq \mathcal{C}$ is denoted $\bigwedge \mathbb{B}$. Similarly, we denote by $\bigvee \mathbb{B}$, $\bigvee \bigwedge \mathbb{B}$ and $\mathit{Bool}^+(\mathbb{B})$ the classes of counter conditions which are respectively, disjunctions of atoms $\mathbb{B}(c)$, disjunction of conjunctions of atoms $\mathbb{B}(c)$ (DNF), any negation-free Boolean formula.

▶ **Example 1.** First, Fig. 2 illustrates an example (left) with a disjunction of boundedness objectives (see the caption for details). Our second example is given by the 2-counter arena at the right of Fig. 2, where Adam controls all states. Adam has a strategy to win the objective $\bigwedge_{i=1,2} \mathbb{U}(c_i)$. Indeed, he can alternate between q_1 and q_2 by cycling longer and longer in one before cycling to the other. Notice that this strategy requires infinite memory.

▶ **Lemma 2.** *Counter games (with Boolean combinations of boundedness objectives) are determined and decidable.*

Proof. Given a counter arena $\mathcal{A}$ and a counter c of $\mathcal{A}$, the set $\mathit{Plays}(\mathcal{A}, \mathbb{B}(c))$ is a Borel set. Indeed, it is equal to the countable union for all $N \geq 0$ of the sets

$$\text{Plays}_N(\mathcal{A}, \mathbb{B}(c)) = \{\rho \in \text{Plays}(\mathcal{A}) \mid \forall n \in \mathbb{N}, \lambda(\zeta(\rho), c)_n \leq N\}$$

which are ω-regular. Indeed, a Büchi automaton needs $|V| \times N \times |C|$ states to recognize $\text{Plays}_N(\mathcal{A}, \mathbb{B}(c))$. Since ω-regular sets are Borel, so is $\mathit{Plays}(\mathcal{A}, \mathbb{B}(c))$, as well as any Boolean combination of the latter. By Martin's determinacy theorem [24], the result follows.

Figure 2 (Left) Counter arena $\mathcal{A} = (V, E, V_\exists, V_\forall, v)$ with $V_\exists = \{q_1, q_3\}$ and $V_\forall = \{q_2, q_4\}$, $v = 1$. There are two counters (c, d) whose updates are represented on the figure as pairs. We assume no parity condition and a counter condition $\mathbb{B}(c) \vee \mathbb{B}(d)$. From vertex 3, Eve has a memoryless winning strategy σ: always move to 4. Furthermore, she does not have a strategy from 1 to bound counter c, neither does she have a strategy from 1 to bound d. However, she has a memoryless strategy β winning for $\mathbb{B}(c) \vee \mathbb{B}(d)$: from 1, she moves to 2, and from 3 she moves to 4. If the play stays in $\{1, 2\}$, then d is bounded, and if the play eventually moves to 3, then c is bounded. (Right) A 2-counter arena with all states controlled by Adam and no parity condition.

To prove decidability, it suffices to notice that winning strategies in counter games are infinite trees such that all of their branches are accepted by a deterministic max-automaton as defined in [3]. Deterministic max-automata corresponds exactly to the logic WMSO+U over infinite words (weak MSO with the unbounding quantifier). WMSO+U has been extended to WMSO+UP on infinite trees with an additional quantifier over infinite paths (P). Therefore, winning strategies of two-player games with winning conditions definable in WMSO+U over infinite words are definable in WMSO+UP (see Ex. 2 of [5]). The result follows since WMSO+UP has decidable satisfiability problem [5]. ◀

3 Counter games with conjunctions of boundedness conditions

In this section, we study games with counter conditions in the class $\bigwedge \mathbb{B}$. Such games are easily shown to be decidable using known results. Indeed, we prove that they are equivalent in polynomial time to Streett games, known to be CoNP-complete [18], and in PTIME for a fixed number of Streett pairs [25]. This allows us to prove the following theorem:

▶ **Theorem 3.** *Counter games with winning conditions in $\bigwedge \mathbb{B}$ are* CONP*-complete, and in PTIME if both the index of the priority function and the number of counters are fixed constants. Finite memory suffices for Eve and memoryless strategies suffice for Adam.* CONP*-hardness holds even if the index of the parity function is any fixed constant.*

Sketch of proof. First, we define Streett games. Given an arena $\mathcal{A}$ with set of vertices V, and a set of k pairs $S = \{(E_i, F_i) \mid 1 \leq i \leq k, E_i, F_i \subseteq V\}$, we let $\mathsf{Streett}(S)$ be the set of words $w \in V^\omega$ such that for all $i = 1, \ldots, k$, if w contains infinitely many occurrences of some $e \in E_i$, then it must contain infinitely many occurrences of some $f \in F_i$. A Streett game is a pair $G = (\mathcal{A}, W)$ where W is given as set of k Streett pairs S, i.e., $W = \mathsf{Streett}(S)$. We prove that $\bigwedge \mathbb{B}$-counter games are interreducible to Streett games in polynomial time. From any counter game G, we construct a Streett game $\Psi(G)$ with the same arena and for each counter c a pair (E_c, F_c) such that E_c is the set of vertices where c is incremented while F_c is the set of vertices where c is reset. The parity function of the counter game can also be split up into Streett pairs. If Eve wins G, it is obvious that Eve wins $\Psi(G)$ with the same winning strategy. For the converse, we use the fact that finite-memory strategies suffice to win Streett games (and memoryless strategies suffice for Adam) as shown in [25]. Any finite-memory Eve's strategy σ winning for $\Psi(G)$ is also winning for G. Indeed, if σ

has some play with some unbounded counter c, then c is necessarily incremented and reset infinitely often. Adam could then find a cycle (both on the arena and the memory structure of σ) containing at least one increment of c and no reset, iterate this cycle *ad infinitum*, and make Eve lose the Streett game if she plays σ. This contradicts that σ is winning. As a consequence, Eve wins G iff she wins $\Psi(G)$. To get the CoNP lower bound, we use the fact that Ψ is actually reversible, in polynomial time. ◀

Theorem 3 does not cover the case where only the number of counters is fixed. We prove that in this case, the complexity is at most NP $\cap$ coNP. Any Streett pair can be seen as a parity condition over colors $\{0, 1, 2\}$. Therefore, if in the latter transformation Ψ we use $\{0, 1, 2\}$-parity conditions instead of Streett pairs and keep exactly the parity condition of G, we obtain that any $\bigwedge\mathbb{B}$-counter game with a fixed number ℓ of counters, is equivalent (in the sense that it preserves the winner) to a game with a winning condition which is a conjunction of a fixed number ℓ of $\{0, 1, 2\}$-parity conditions and a single arbitrary parity condition. We prove that such games are in turn reducible in polynomial time to parity games for $\ell = 1$ in the following lemma, later on applied recursively to show the result the result for any fixed ℓ (Theorem 5).

▶ **Lemma 4.** *Games of the form $G = (\mathcal{A}, W)$ where $W = Parity(\kappa) \cap Parity(\kappa_3)$ for κ an arbitrary colouring of index k and κ_3 a colouring in $\{0, 1, 2\}$, reduce in polynomial time to parity games of index $2k + 1$. Moreover, finite-memory strategies of memory size equal to k are sufficient for Eve to win G.*

Note that Lemma 4 entails that games with a conjunction of a parity condition of index k and a fixed number N of parity conditions over colors $\{0, 1, 2\}$ are solvable in NP $\cap$ CoNP. Indeed, by iterating Lemma 4 N times, the latter games reduce to parity games of index $2^N(k + 1) - 1$ (and number of states exponential in N). Games with Boolean combinations of parity objectives have been studied in [14]. However, the former complexity result is not covered by [14]. As explained before, Lemma 4 implies the following theorem:

▶ **Theorem 5.** *For any fixed positive integer N, counter games of parity index k (which is not supposed to be fixed) with winning conditions in $\bigwedge\mathbb{B}$ and at most N counters, are in NP $\cap$ CoNP (and parity-hard). Finite memory strategies with memory size $2^{N-1}(k + 1) - 1$ suffice for Eve and Adam.*

4 Finitely switching strategies for games with disjunction of prefix-independent objectives

Let $\mathcal{A}$ be an arena, let V be its set of vertices, and let $\mathcal{W}$ be a finite set of prefix-independent[1] winning conditions for $\mathcal{A}$, i.e., $\mathcal{W} \subseteq 2^{V^\omega}$. We let $\bigvee\mathcal{W} = \bigcup\{W \mid W \in \mathcal{W}\}$. In this section, we consider a class of strategies for Eve, called *finitely switching*, whose existence entail that she wins $(\mathcal{A}, \bigvee\mathcal{W})$. We characterize the existence of finitely switching strategies via a least fixpoint and, for some particular classes of winning objectives $\bigvee\mathcal{W}$ of interest in this paper, prove that such strategies suffice for Eve to win $(\mathcal{A}, \bigvee\mathcal{W})$. The complexity of computing the fixpoint for those particular classes of objectives is deferred to Section 5.

Let us first give intuition on the notion of finitely switching strategies. In such a strategy, Eve announces an initial goal $W \in \mathcal{W}$ she wants to satisfy, but she may change her mind during the play, i.e., announce another goal $W' \in \mathcal{W}$, depending on what Adam does. She

[1] A winning condition W is prefix-independent if, for all $(w, u) \in (V^\omega, V^\star)$, $w \in W$ iff $uw \in W$.

can do this only a finite number of times and eventually keep the same goal forever and satisfy it. Formally, for $k \geq 0$, a k-switching strategy for Eve is a strategy σ such that there exists a mapping goal from finite histories of σ to $\mathcal{W}$ such that for all $\pi = v_1 v_2 \cdots \in \text{Plays}(\sigma)$, there exists $W_1, \ldots, W_{k+1} \in \mathcal{W}$ such that $\pi \in W_{k+1}$ and

$$\mathsf{goal}(v_0)\mathsf{goal}(v_0 v_1)\mathsf{goal}(v_0 v_1 v_2) \cdots \in W_1^* W_2^* \ldots W_k^* W_{k+1}^\omega$$

The goal W_{k+1} is called the *ultimate* goal of π. We say that σ is finitely switching if it is k-switching for some $k \geq 0$.

▶ **Example 6.** Consider the example of Fig. 2. The described strategy β is 1-switching for $\mathcal{W} = \{\mathbb{B}(c), \mathbb{B}(d)\}$: initially, her goal is $\mathbb{B}(d)$. If Adam ever tries to make it so that counter d gets unbounded, by going to vertex 3 from vertex 2, Eve can now set her new goal to $\mathbb{B}(c)$.

Consider now the 2-state arena of Example 1 in which Eve wants to satisfy $\bigvee_{c=1,2} \mathbb{U}(c)$. She has no finitely switching strategy: whenever she announces she wants to satisfy $\mathbb{U}(c_i)$ for some i, Adam loops on state q_{3-i} until Eve changes her mind. If her ultimate goal is $\mathbb{U}(c_i)$ for some i, then Adam will loop forever on q_{3-i} and c_i will be bounded, so that Eve does not meet the ultimate goal she announced. By seeing operations on c_1 and c_2 as priority functions, this example also shows that finitely switching strategies are not sufficient to win disjunctions of parity objectives in general. More precisely, for $i = 1, 2$, we can define the priority functions p_i which colors q_i by 0 and q_{3-i} by 1. If she ultimately announces her goal is to satisfy priority p_i, then Adam takes transition q_{3-i} forever and p_i sees infinitely many times color 1.

Since in a finitely switching strategy, any play consistent with that strategy must satisfy its ultimate goal, the following result is immediate:

▶ **Lemma 7** (Soundness). *Any finitely switching strategy for Eve in $\mathcal{A}$ is winning for $(\mathcal{A}, \bigvee \mathcal{W})$.*

We will see later on that the converse holds for some particular classes of boundedness objectives, but for now, let us characterize the existence of finitely switching strategies via some least fixpoint. For a set $X \subseteq V$, we denote the objective of reaching X by $\mathsf{Reach}(X) = V^* X V^\omega$. We let f be the function which associates any $X \subseteq V$ to the set of vertices u from which Eve can win the objective $W \cup \mathsf{Reach}(X)$ for some $W \in \mathcal{W}$. Formally, $f(X) = \{u \in V \mid \exists W \in \mathcal{W},\ \text{Eve wins } (\mathcal{A}[u], W \cup \mathsf{Reach}(X))\}$. Note that $X \subseteq f(X)$ for all $X \subseteq V$. Indeed, if $u \in X$, then Eve has a trivial strategy from u to reach X, and so $u \in f(X)$. Since $(2^V, \subseteq)$ is a complete lattice, by Knaster–Tarski theorem, f has a unique least fixpoint denoted $\mathsf{S}^{\mathcal{W}}$. To compute $\mathsf{S}^{\mathcal{W}}$, it suffices to compute the following sequence of sets until it stabilizes:

- $\mathsf{S}_0^{\mathcal{W}} = \emptyset$,
- for $i \geq 0$, $\mathsf{S}_{i+1}^{\mathcal{W}} = \{u \in V \mid \exists W \in \mathcal{W},\ \text{Eve wins } (\mathcal{A}[u], W \cup \mathsf{Reach}(\mathsf{S}_i^{\mathcal{W}}))\}$.

For all $i \geq 1$ and $u \in \mathsf{S}_i^{\mathcal{W}}$ (if it exists), we denote by $\sigma_{u,i}$ a strategy for Eve winning in the game $(\mathcal{A}[u], W \cup \mathsf{Reach}(\mathsf{S}_{i-1}^{\mathcal{W}}))$ for some $W \in \mathcal{W}$. It exists by definition of $\mathsf{S}_i^{\mathcal{W}}$.

We now prove the following characterization.

▶ **Lemma 8** (Fixpoint characterization of finitely switching strategies). *Let $\mathcal{A}$ be an arena with set of vertices V and $\mathcal{W}$ a finite set of prefix-independent winning conditions for $\mathcal{A}$. For all $u \in V$, the following are equivalent:*

1. *Eve has a finitely switching strategy from u*
2. *Eve has a $|V|$-switching strategy from u*
3. *$u \in S^{\mathcal{W}}$*

Proof. Clearly $2 \Rightarrow 1$. We first prove $1 \Rightarrow 3$ and then $3 \Rightarrow 2$.

Let σ be a k-switching strategy for some $k \geq 0$. By induction on k, we prove that $u \in \mathsf{S}^{\mathcal{W}}_{k+1}$. This implies the claim as $\mathsf{S}^{\mathcal{W}}_{k+1} \subseteq \mathsf{S}^{\mathcal{W}}$.

If $k = 0$, then Eve never changes her mind and therefore all plays of $\text{Plays}(\sigma)$ are in $\mathsf{goal}(u)$ (the history with only the vertex u), so, $u \in \mathsf{S}^{\mathcal{W}}_1$. Suppose that $k > 0$. We take $W = \mathsf{goal}(u)$. Let $\pi \in \text{Plays}\sigma$. We prove that $\pi \in W \cup \mathsf{Reach}(\mathsf{S}^{\mathcal{W}}_k)$. If Eve never changes her mind during π, then $\pi \in W$. Otherwise, let h the smallest prefix of π such that $\mathsf{goal}(h) \neq W$. Let v be the last vertex of h. Note that the strategy[2] $\sigma|_h$ is a $(k-1)$-switching strategy from v. By IH, $v \in \mathsf{Reach}(\mathsf{S}^{\mathcal{W}}_k)$, which means that $\pi \in \mathsf{Reach}(\mathsf{S}^{\mathcal{W}}_k)$ and we are done.

We now prove $3 \Rightarrow 2$. Let $u \in \mathsf{S}^{\mathcal{W}}$. Let i be smallest index such that $u \in \mathsf{S}^{\mathcal{W}}_i$. Note that $i \leq |V|$. We prove by induction on i that Eve has an $(i-1)$-switching strategy $\beta_{u,i}$ witnessed by a goal function $\mathsf{goal}_{u,i}$. If $u \in \mathsf{S}^{\mathcal{W}}_1$, then $\sigma_{u,1}$ wins $(\mathcal{A}[u], W)$ for some $W \in \mathcal{W}$ and so we let $\mathsf{goal}_{u,1}(h) = W$ for any history h of $\sigma_{u,1}$.

Suppose that $i > 1$ and $u \in \mathsf{S}^{\mathcal{W}}_i$. Remind that the strategy $\sigma_{u,i}$ wins $(\mathcal{A}[u], W \cup \mathsf{Reach}(\mathsf{S}^{\mathcal{W}}_{i-1}))$. We modify $\sigma_{u,i}$ into a strategy $\beta_{u,i}$ as follows: $\beta_{u,i}$ is the same as $\sigma_{u,i}$ as long as $\mathsf{S}^{\mathcal{W}}_{i-1}$ has not been reached. If eventually $\mathsf{S}^{\mathcal{W}}_{i-1}$ is reached, say at a vertex v, then $\beta_{u,i}$ plays according to $\beta_{v,i-1}$ (which exists by IH).

We prove that $\beta_{u,i}$ is $(i-1)$-switching. We let $\mathsf{goal}_{u,i}(h) = W$ for any history h which does not visit $\mathsf{S}^{\mathcal{W}}_{i-1}$. For any history $h = h_1 v h_2$ such that $|h_1|$ is minimal and $v \in \mathsf{S}^{\mathcal{W}}_{i-1}$, we let $\mathsf{goal}_{u,i}(h) = \mathsf{goal}_{v,i-1}(vh_2)$. Let $\pi \in \text{Plays}(\beta_{u,i})$. If $\pi = v_0 v_1 \ldots$ never visits $\mathsf{S}^{\mathcal{W}}_{i-1}$, then $\mathsf{goal}(v_0)\mathsf{goal}(v_0 v_1) \cdots \in W^\omega$, and $\pi \in W^\omega$. If there exists j minimal such that $v_j \in \mathsf{S}^{\mathcal{W}}_{i-1}$, then, by HI, there exists $W_1, \ldots, W_i \in \mathcal{W}$ such that $\mathsf{goal}_{v_j,i-1}(v_j)\mathsf{goal}_{v_j,i-1}(v_j v_{j+1}) \cdots \in W_1^* \ldots W_{i-1}^* W_i^\omega$. By definition of $\mathsf{goal}_{u,i}$, we obtain that $\mathsf{goal}_{u,i}(v_0)\mathsf{goal}_{u,i}(v_0 v_1) \cdots \in W^* W_1^* \ldots W_{i-1}^* W_i^\omega$. Finally, it remains to prove that $\pi \in W_i$: by IH, its suffix $v_j v_{j+1} \ldots$ is in W_i, and since W_i is prefix-independent, so is π, concluding the proof. ◀

According to Lemma 8, when Eve has a finitely switching strategy, then she has a $|V|$-switching strategy. Interestingly, observe that the number of times she possibly needs to change her mind does not depend on the number of winning objectives in $\mathcal{W}$.

The proof of Lemma 8 constructs, for all $1 \leq i \leq |V|$ and $u \in \mathsf{S}^{\mathcal{W}}_i$, a finitely switching strategy $\beta_{u,i}$, which either mimics $\sigma_{u,i}$ or switch to a strategy $\beta_{v,i-1}$. So, Eve needs to remember the current vertex u and index i, in order to know whether she must play according to $\sigma_{u,i}$ or to switch to a strategy $\beta_{v,i-1}$. So, even if for some N, all the strategies $\sigma_{u,i}$ are finite-memory of size at most N, $\beta_{u,i}$ needs memory $O(N.|V|^2)$ in general. We now prove that Eve can do better.

▶ **Lemma 9** (Memory transfer). *Let $\mathcal{A}$ be a counter arena, V be its set of vertices, and $\mathcal{W}$ a finite set of prefix-independent winning conditions for $\mathcal{A}$. Let $N \in \mathbb{N}$ and suppose that for all $X \subseteq V$, $u \in V$ and $W \in \mathcal{W}$, strategies of memory size at most N suffice for Eve to win $(\mathcal{A}[u], W \cup \mathsf{Reach}(X))$. Then for all $u \in \mathsf{S}^{\mathcal{W}}$, Eve wins $(\mathcal{A}[u], \bigvee \mathcal{W})$ with memory at most N.*

The converse of Lemma 7 does not hold in general, as illustrated in Example 1 for disjunction of unboundedness objectives. However, we show here that it holds for disjunctions of conjunctions of boundedness objectives.

▶ **Lemma 10** (Completeness for boundedness conditions in DNF). *Let $\mathcal{A}$ be a counter arena and C its set of counters. Let $\mathcal{W}$ be a finite subset of counter conditions for $\mathcal{A}$ in $\bigwedge \mathbb{B}$. If Eve wins the counter game $(\mathcal{A}, \bigvee \mathcal{W})$, then she has a finitely switching strategy from the initial vertex v_0.*

[2] The restriction $\sigma|_h$ is defined by $\sigma|_h(h') = \sigma(hh')$ for all h'.

Proof. Let $C_1, \dots, C_p$ be subsets of C such that $\mathcal{W}$ is the set of all counter conditions $\bigwedge_{c \in C_i} \mathbb{B}(c)$, for $i \in \{1, \dots, p\}$. Suppose that Eve does not have a finitely switching strategy from the initial vertex v_0. This means, by Lemma 8, that $v_0 \not\in \mathsf{S}^{\mathcal{W}}$. We construct a strategy for Adam winning the complementary objective $Comp = \bigcap_{i \in \{1,\dots,p\}} \left(\bigcup_{c \in C_i} \mathbb{U}(c) \cup \overline{\text{Parity}(\mathcal{A})} \right)$. By definition of $\mathsf{S}^{\mathcal{W}}$, $f(\mathsf{S}^{\mathcal{W}}) = \mathsf{S}^{\mathcal{W}}$. Therefore, by definition of f, for any $v \in V \backslash \mathsf{S}^{\mathcal{W}}$ and $i \in \{1, \dots, p\}$, Eve does not win the game $(\mathcal{A}[v], (\text{Parity}(\mathcal{A}[v]) \cap \bigcap_{c \in C_i} \mathbb{B}(c)) \cup \mathsf{Reach}(\mathsf{S}^{\mathcal{W}}))$. Moreover, notice that since $\text{Parity}(\mathcal{A}[v]) \cap \bigcap_{c \in C_i} \mathbb{B}(c)$ is a Borel set, so is $(\text{Parity}(\mathcal{A}[v]) \cap \bigcap_{c \in C_i} \mathbb{B}(c)) \cup \mathsf{Reach}(\mathsf{S}^{\mathcal{W}})$. Thus, by Martin's theorem, Adam has a winning strategy $\sigma_{v,i}$ in $\mathcal{A}[v]$ for the complementary objective $(\bigcup_{c \in C_i} \mathbb{U}(c) \cup \overline{\text{Parity}(\mathcal{A}[v])}) \cap \overline{\mathsf{Reach}(\mathsf{S}^{\mathcal{W}})}$, for all $i \in \{1, \dots, p\}$. Let us now explain how intuitively we build a strategy for Adam winning for *Comp*. It is defined by breaking it down into the following steps:

- Adam begins by step $(1, 1)$: he follows strategy $\sigma_{v_0,1}$ until the play of the game reaches a vertex where the value of a counter of C_1 is 1. If that is never the case, then Adam follows $\sigma_{v_0,1}$ *ad. infinitum.* Notice that, if the value of every counter of C_1 is bounded by a certain integer, Adam wins, since the play does not belong to $\text{Parity}(\mathcal{A}[v_0]) = \text{Parity}(\mathcal{A})$.
- After completing step (i, j) in a vertex v, two cases arise:
  - If $j < p$, then Adam carries out step $(i, j + 1)$ by following $\sigma_{v,j+1}$ until the play of the game reaches a vertex where the value of a counter of C_{j+1} is i. If that is never the case, Adam follows $\sigma_{v,j+1}$ *ad. infinitum*, and he wins since the play then satisfies $\overline{\text{Parity}(A[v])}$ starting from v, and thus $\overline{\text{Parity}(A)}$ globally.
  - If $j = p$, then Adam carries out step $(i + 1, 1)$ by following $\sigma_{v,1}$ until the play of the game reaches a vertex where the value of a counter of C_1 is $i + 1$. If that is never the case, Adam follows $\sigma_{v,1}$ *ad. infinitum.* ◀

5 Complexity of games with disjunctions of boundedness conditions

The next result gives sufficient conditions on a class of games $\mathcal{G}$, to guarantee decidability of the problem of deciding if Eve has a finitely switching strategy for a disjunction of objectives in the class. In this result, we assume that the winning objectives of $\mathcal{G}$ are finitely represented in some way. This is the case of all classes to which we apply this lemma in the paper.

▶ **Lemma 11.** *Let $\mathcal{C} \in \{PTIME, NP, \text{co}NP, EXPTIME\}$. Let $\mathcal{G}$ be a class of games with prefix-independent objectives, such that deciding whether, given $(\mathcal{A}, W) \in \mathcal{G}$, a vertex v of $\mathcal{A}$, and a subset X of vertices of $\mathcal{A}$, Eve wins $(\mathcal{A}[v], W \cup \mathsf{Reach}(X))$, is in $\mathcal{C}$. Then, deciding, given an arena $\mathcal{A}$ and a finite subset of winning conditions $\mathcal{W}$ such that $\{(\mathcal{A}, W) \mid W \in \mathcal{W}\} \subseteq \mathcal{G}$, whether Eve has a winning finitely switching strategy for $(\mathcal{A}, \bigvee \mathcal{W})$, is in $\mathcal{C}$.*

Proof. Suppose first that $\mathcal{C} = \text{PTIME}$. From Lemma 8, Eve has a winning finitely switching strategy for $(\mathcal{A}, \bigvee \mathcal{W})$ if and only if the initial vertex v_0 of $\mathcal{A}$ is in $\mathsf{S}^{\mathcal{W}}$. Thus, we can decide whether Eve has a finitely switching strategy by recursively computing the $\mathsf{S}_i^{\mathcal{W}}$, one after the other, until $\mathsf{S}_i^{\mathcal{W}} = \mathsf{S}_{i+1}^{\mathcal{W}} = \mathsf{S}^{\mathcal{W}}$. In order to compute $\mathsf{S}_{i+1}^{\mathcal{W}}$ from $\mathsf{S}_i^{\mathcal{W}}$, we check for every vertex v of $\mathcal{A}$ whether Eve wins the game $(\mathcal{A}[v], W \cup \mathsf{Reach}(\mathsf{S}_i^{\mathcal{W}}))$. Thus, since $\mathsf{S}_{|V|}^{\mathcal{W}} = \mathsf{S}^{\mathcal{W}}$, in order to compute $\mathsf{S}^{\mathcal{W}}$, we only need to check, in ptime, whether Eve wins a game of the form $(\mathcal{A}[v], W \cup \mathsf{Reach}(X))$ at most $|V| \times |V| \times |\mathcal{W}|$ times. As a consequence, the problem of deciding whether Eve has a winning finitely switching strategy for $(\mathcal{A}, \bigvee \mathcal{W})$ is in PTIME. We present this generic fixpoint algorithm in Algorihtm 1, as it is useful to treat the other complexity cases. In that figure, slv is an algorithm that terminates in polynomial time, and such that $\text{slv}(\mathcal{A}, v, W, H)$ returns true if and only if Eve wins $(\mathcal{A}[v], W \cup \mathsf{Reach}(H))$. The case where $\mathcal{C} = \text{EXPTIME}$ is similar to the case where $\mathcal{C} = \text{PTIME}$.

Algorithm 1 Generic algorithm to check $v \in \mathsf{S}^{\mathcal{W}}$.

SOLVE($\mathcal{A}$,$\mathcal{W}$)
//v_0 is the initial vertex of $\mathcal{A}$
//$V = \{v_1, \ldots, v_n\}$
//$\mathcal{W} = \{W_1, \ldots, W_p\}$
1. $N \leftarrow n^2 \times p$
2. $\alpha \leftarrow 0$
3. $H_0, H_1, \ldots, H_N \leftarrow \emptyset$
4. While $\alpha < n$
5.   For $i = 1, \ldots, n$
6.   For $j = 1, \ldots, p$
7.   If slv$(\mathcal{A}, v_i, W_j, H_\alpha)$
8. $H_{\alpha+1} \leftarrow \{v_i\} \cup H_{\alpha+1}$
9. $\alpha \leftarrow \alpha + 1$
10. Return $(v_0 \in H_\alpha)$

In the case where $\mathcal{C} = \mathrm{NP}$, we transform the algorithm SOLVE into an ptime algorithm VERIF, which is defined as the algorithm SOLVE, except that line 7 is replaced by a call to a ptime verifier that Eve wins $(\mathcal{A}[v_i], W_j \cup \mathsf{Reach}(H_\alpha))$ given a certificate. Notice that the algorithm given is directly written as an algorithm in NP, *i.e.* an algorithm that verifies if Eve wins given a certificate, and *not* as an algorithm in P with oracle NP. All the certificates needed for each call at line 7 are taken as input of the algorithm VERIF. This approach works because the algorithm VERIF returns **True** if and only if the answers to some well-chosen questions of the type "Does Eve win $(\mathcal{A}[v], W \cup \mathsf{Reach}(X))$?" are true. The case where $\mathcal{C} = \mathrm{CoNP}$ is done in a similar way, but this time by guessing the complement of $\mathsf{S}^{\mathcal{W}}$. ◀

We are now ready to prove complexity results for solving counter games with disjunction of boundedness objectives. We start with the case of $\bigvee \mathbb{B}$.

▶ **Theorem 12.** *Counter games with counter conditions in $\bigvee \mathbb{B}$ are in NP $\cap$ coNP, and are in PTIME if the index of the colouring is fixed. A memory of size equal to the index of the colouring suffices for Eve, and infinite memory is required for Adam.*

Proof. Let G be a game over counter arena $\mathcal{A}$ with set of counters C, initial vertex v_0 and objective $\bigvee \mathcal{W}$ where $\mathcal{W} = \{\mathrm{Parity}(\mathcal{A}) \cap \mathbb{B}(c) \mid c \in C'\}$ for some $C' \subseteq C$. It should be clear that those conditions are prefix-independent, therefore, by Lemma 8 and Lemma 10, Eve wins G iff she has a finitely switching strategy iff $v_0 \in \mathsf{S}^{\mathcal{W}}$. So, to check whether Eve wins G, it suffices to compute the fixpoint $\mathsf{S}^{\mathcal{W}}$. We prove that each step of the fixpoint computation (line 7 in algorithm SOLVE) is done in NP $\cap$ coNP, and in PTIME if the index of the colouring is fixed. By Lemma 11, the complexity statement of the theorem follows. It remains to show that for all subset $X \subseteq V$, any vertex $u \in V$ and any counter $c \in C'$, it is decidable in NP $\cap$ coNP (and in ptime for fixed parity) whether Eve wins the game $(\mathcal{A}[u], (\mathrm{Parity}(\mathcal{A}) \cap \mathbb{B}(c)) \cup \mathsf{Reach}(X))$. First, we evacuate the reachability condition, i.e., reduce in ptime the latter problem to solving a game $(\mathcal{A}', \mathrm{Parity}(\mathcal{A}') \cap \mathbb{B}(c))$. This is easily done by adding a sink state to $\mathcal{A}$ reached whenever X is visited, with operation skip on c and priority 0. This reduction works for more general boundedness conditions. Finally, the game $(\mathcal{A}', \mathrm{Parity}(\mathcal{A}') \cap \mathbb{B}(c))$ is solvable in NP $\cap$ coNP by Theorem 5, and in ptime for fixed parity, which is the case of $\mathcal{A}'$ when the index of $\mathcal{A}$ is fixed, because they have the same colours.

For Adam, infinite memory might be necessary to enforce the complementary objective, as illustrated by Example 1. For Eve, Theorem 5 states that a memory of size the index of the parity function is sufficient to solve the "local" games $(\mathcal{A}', \text{Parity}(\mathcal{A}') \cap \mathbb{B}(c))$, which can be translated back to strategies of same size in $(\mathcal{A}[u], (\text{Parity}(\mathcal{A}) \cap \mathbb{B}(c)) \cup \mathsf{Reach}(X))$. Therefore, the memory transfer lemma (Lemma 9) yields the result. ◀

We now turn to games on arenas $\mathcal{A}$ with conditions in $\bigvee\bigwedge\mathbb{B}$, i.e., where $\mathcal{W} = \{\text{Parity}(\mathcal{A}) \wedge \bigwedge_{c\in C_i} \mathbb{B}(c) \mid i = 1, \ldots, n\}$ for $C_1, \ldots, C_n$ finite subsets of counters. The same reasoning as in the proof of Theorem 12 applies. The only difference here is that, to solve the "local" games of the fixpoint computation (line 7 of algorithm SOLVE), we rely on Theorem 3.

▶ **Theorem 13.** *Counter games with winning conditions in $\bigvee\bigwedge\mathbb{B}$ are* coNP*-complete. Finite memory suffices for Eve, and infinite memory is required for Adam.*

We conclude this section by the case of Boolean combination of boundedness objectives.

▶ **Theorem 14.** *Counter games with winning conditions in $Bool^+(\mathbb{B})$ are in PSPACE and* CoNP*-hard. Finite memory suffices for Eve, and infinite memory is required for Adam.*

Proof. Any counter condition which is a positive boolean combination $\phi \in Bool^+(\mathbb{B})$ can be written in disjunctive normal form $\psi = \bigvee_{i\in\{1,\ldots,p\}} \bigwedge_{c\in C_i} \mathbb{B}(c)$, where the C_i are subsets of $\mathcal{C}$. Let $\mathcal{W} = \{\text{Parity}(\mathcal{A}) \wedge \bigwedge_{c\in C_i} \mathbb{B}(c) \mid i = 1, \ldots, n\}$. A direct application of Theorem 13 yields a $\mathsf{CoNExpTime}$, because p might be exponential. Instead, we do not construct ψ explicitely. Recall that, from Theorem 3, counter games with counter conditions in $\bigwedge\mathbb{B}$ are in coNP, and thus in PSPACE. Thus, since it is well-known that, even if p may be exponential in the size of ϕ, we can enumerate $\mathcal{W}$ in polynomial space, we can use this enumeration algorithm at line 6 of algorithm SOLVE in Algorithm 1 to compute the fixpoint $\mathsf{S}^{\mathcal{W}}$ in polynomial space. As a consequence, the problem of deciding whether Eve has a winning finitely switching strategy for counter games with winning conditions in $Bool^+(\mathbb{B})$ is in PSPACE. Hence, the result follows because, as for Theorem 13, these strategies suffice for Eve. ◀

6 Extensions of counter games with max operation

In this section, we consider counter games where the players can, in addition, put into a counter the maximum value of a subset of counters. In other words, max-counter games are defined in the same exact way as counter games, the only difference being *counter operations* are now mappings from a finite subset C of $\mathcal{C}$ to $\{\mathsf{i}, \mathsf{r}, \mathsf{skip}\} \cup \{\max_{c\in S}(c) \mid S \subseteq \mathcal{C}\}$.

▶ **Theorem 15.** *Let $\mathcal{G}$ be the class of counter games G with counter condition $\bigwedge_{c\in C} \mathbb{B}(c)$, where C is the set of counters of G. Given a game G in $\mathcal{G}$, the problem of deciding whether Eve wins G is EXPTIME-c. Finite memory is sufficient for Eve and Adam.*

Proof. For hardness, we reduce the emptiness problem of the intersection of n deterministic top-down tree automata, which is known to be EXPTIME-hard [16]. We first show PSPACE-hardness in the case of arenas where Adam plays no role, i.e., $V_\forall = \varnothing$. The proof is by reduction from the emptiness problem of the intersection of n DFA. The latter reduction is inspired from the proof that deterministic min-automata have PSPACE-c emptiness problem [8]. Using the fact that strategies are trees, we lift the latter reduction to tree automata. It is non-trivial but standard. The detailed proof is in Appendix, in Lemma 16.

It remains to show that solving a game in $\mathcal{G}$ can be done in exponential time. The difficulty for solving a game G of $\mathcal{G}$ comes from the fact that counters interact with each other, since the value of counters can "flow" from one to another via the max operation.

That was not case for $\bigwedge\mathbb{B}$-counter games without max, which are CoNP-c, and we could track each counter separately, replacing each boundedness condition by a condition of the form "if c is incremented infinitely often, then it is reset infinitely often". Here, we need to track sequences of counters that flow one into another, called traces. We rather solve games with the complementary objective, which is correct since max-counter games are determined (see Lemma 19 in Appendix). To formalize this idea, we use the notion of $\mathbb{U}$-automata, *i.e.* automata with counters accepting some positive boolean combination of unboundedness conditions, that is a notion very close to the notion of S-automata described in [6]. We define a (non-deterministic) $\mathbb{U}$-automaton $\mathcal{B}$ with a single counter d and acceptance condition $\mathbb{U}(d)$ that guesses either a new trace, or a valid continuation to the current trace, at every move of a play of $\mathcal{G}$. Every operation on the counters of the trace are mimicked on d, and it accepts a play iff there exists a run such that d is unbounded. That same idea is already used in the proof of Theorem 1 of [4], from which this proof is inspired. So, solving G boils down to solving a game on the same arena but with objective given by the language $L(\mathcal{B})$.

We show that the class of games G with an objective given by a non-deterministic $\mathbb{U}$-automaton with an acceptance condition of the form $\bigvee\mathbb{U}$ is in EXPTIME. To that end, we convert $\mathcal{B}$ into a non-deterministic parity automaton $\mathcal{T}$, which does not preserve the language, but preserves the existence of winning strategy for Eve: when playing on the arena of G, Eve wins the objective $\mathcal{L}(\mathcal{B})$ if and only if she wins the objective $\mathcal{L}(\mathcal{T})$. Correctness is ensured by a pumping-like argument based on the fact that finite-memory strategies are sufficient to win ω-regular games. The automata $\mathcal{B}$ and $\mathcal{T}$ are constructed in ptime from G. Then we determinize $\mathcal{T}$ in exponential time, take its product with G, and obtain a classical parity game of exponential size and linear index. We can conclude since parity games with m edges, n vertices and index k can be solved in $O(mn^k)$ (see e.g. [14]). The detailed proof is in the Appendix, in Lemma 15. ◀

7 Future work

In this paper, we have proved new complexity results for important classes of counter games, with the aim of finely understanding why they are decidable. We observe that they are mainly two types of boundedness conditions, which require different techniques: conjunctions of boundedness conditions, which are equivalent to Streett games, and disjunction of boundedness conditions (for which we introduce the notion of finitely switching strategies). To emphasize this dichotomy, we note that even for a parity function of fixed index, counters games with conjunctions of boundedness conditions are CoNP-c, while they are in PTime for disjunctions. By determinacy, those results also yield complexity bounds for the complementary classes of *un*boundedness objectives. For example, we get that games with conjunctions of objectives of the form $\mathbb{U}(c)$ can be solved in NP∩CoNP and that infinite memory is required. However, note that our counter games are always taken in conjunction with a parity condition. Therefore, in the complementary objectives, this parity condition is now taken in disjunction. We leave conjunction of parity and unboundedness objectives as future work. Another important direction is to consider classes of conditions that mix boundedness and unboundedness objectives. Since the techniques used to solve them individually are different, this would require new techniques. More generally, the only known upper bound for any Boolean combination (not necessarily negation-free) of boundedness objective is non-elementary. We believe there is space for improvement.

References

1 Nathalie Bertrand, Miheer Dewaskar, Blaise Genest, Hugo Gimbert, and Adwait Amit Godbole. Controlling a population. *Log. Methods Comput. Sci.*, 15(3), 2019. doi:10.23638/LMCS-15(3:6)2019.

2 Roderick Bloem, Krishnendu Chatterjee, and Barbara Jobstmann. Graph games and reactive synthesis. In Edmund M. Clarke, Thomas A. Henzinger, Helmut Veith, and Roderick Bloem, editors, *Handbook of Model Checking*, pages 921–962. Springer, 2018. doi:10.1007/978-3-319-10575-8_27.

3 Mikolaj Bojanczyk. Weak MSO with the unbounding quantifier. *Theory Comput. Syst.*, 48(3):554–576, 2011. doi:10.1007/s00224-010-9279-2.

4 Mikolaj Bojanczyk. Weak MSO with the unbounding quantifier. *Theory Comput. Syst.*, 48(3):554–576, 2011.

5 Mikolaj Bojanczyk. Weak MSO+U with path quantifiers over infinite trees. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, volume 8573 of *Lecture Notes in Computer Science*, pages 38–49. Springer, 2014. doi:10.1007/978-3-662-43951-7_4.

6 Mikolaj Bojanczyk and Thomas Colcombet. Bounds in w-regularity. In *21th IEEE Symposium on Logic in Computer Science (LICS 2006), 12-15 August 2006, Seattle, WA, USA, Proceedings*, pages 285–296. IEEE Computer Society, 2006. doi:10.1109/LICS.2006.17.

7 Mikolaj Bojanczyk, Tomasz Gogacz, Henryk Michalewski, and Michal Skrzypczak. On the decidability of MSO+U on infinite trees. In Javier Esparza, Pierre Fraigniaud, Thore Husfeldt, and Elias Koutsoupias, editors, *Automata, Languages, and Programming - 41st International Colloquium, ICALP 2014, Copenhagen, Denmark, July 8-11, 2014, Proceedings, Part II*, volume 8573 of *Lecture Notes in Computer Science*, pages 50–61. Springer, 2014. doi:10.1007/978-3-662-43951-7_5.

8 Mikolaj Bojanczyk and Szymon Torunczyk. Deterministic automata and extensions of weak MSO. In Ravi Kannan and K. Narayan Kumar, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2009, December 15-17, 2009, IIT Kanpur, India*, volume 4 of *LIPIcs*, pages 73–84. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2009. doi:10.4230/LIPIcs.FSTTCS.2009.2308.

9 Véronique Bruyère. Computer aided synthesis: A game-theoretic approach. In Émilie Charlier, Julien Leroy, and Michel Rigo, editors, *Developments in Language Theory - 21st International Conference, DLT 2017, Liège, Belgium, August 7-11, 2017, Proceedings*, volume 10396 of *Lecture Notes in Computer Science*, pages 3–35. Springer, 2017. doi:10.1007/978-3-319-62809-7_1.

10 Cristian S. Calude, Sanjay Jain, Bakhadyr Khoussainov, Wei Li, and Frank Stephan. Deciding parity games in quasipolynomial time. In *STOC*, pages 252–263, 2017.

11 Krishnendu Chatterjee and Nathanaël Fijalkow. Infinite-state games with finitary conditions. In Simona Ronchi Della Rocca, editor, *Computer Science Logic 2013 (CSL 2013), CSL 2013, September 2-5, 2013, Torino, Italy*, volume 23 of *LIPIcs*, pages 181–196. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2013. doi:10.4230/LIPIcs.CSL.2013.181.

12 Krishnendu Chatterjee, Thomas A. Henzinger, and Florian Horn. Finitary winning in omega-regular games. *ACM Trans. Comput. Log.*, 11(1):1:1–1:27, 2009. doi:10.1145/1614431.1614432.

13 Krishnendu Chatterjee, Thomas A. Henzinger, and Florian Horn. The complexity of request-response games. In *LATA*, volume 6638, pages 227–237, 2011.

14 Krishnendu Chatterjee, Thomas A. Henzinger, and Nir Piterman. Generalized parity games. In Helmut Seidl, editor, *Foundations of Software Science and Computational Structures, 10th International Conference, FOSSACS 2007, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2007, Braga, Portugal, March 24-April 1, 2007, Proceedings*, volume 4423 of *Lecture Notes in Computer Science*, pages 153–167. Springer, 2007. doi:10.1007/978-3-540-71389-0_12.

15 Thomas Colcombet and Christof Löding. The non-deterministic mostowski hierarchy and distance-parity automata. In Luca Aceto, Ivan Damgård, Leslie Ann Goldberg, Magnús M. Halldórsson, Anna Ingólfsdóttir, and Igor Walukiewicz, editors, *Automata, Languages and Programming, 35th International Colloquium, ICALP 2008, Reykjavik, Iceland, July 7-11, 2008, Proceedings, Part II - Track B: Logic, Semantics, and Theory of Programming & Track C: Security and Cryptography Foundations*, volume 5126 of *Lecture Notes in Computer Science*, pages 398–409. Springer, 2008.

16 Hubert Comon, Max Dauchet, Rémi Gilleron, Florent Jacquemard, Denis Lugiez, Christof Löding, Sophie Tison, and Marc Tommasi. Tree Automata Techniques and Applications, 2008. URL: `https://hal.inria.fr/hal-03367725`.

17 E. Allen Emerson and Charanjit S. Jutla. Tree automata, mu-calculus and determinacy (extended abstract). In *32nd Annual Symposium on Foundations of Computer Science, San Juan, Puerto Rico, 1-4 October 1991*, pages 368–377. IEEE Computer Society, 1991.

18 E. Allen Emerson and Charanjit S. Jutla. The complexity of tree automata and logics of programs. *SIAM J. Comput.*, 29(1):132–158, 1999.

19 Léo Exibard, Emmanuel Filiot, and Ayrat Khalimov. Church synthesis on register automata over linearly ordered data domains. In Markus Bläser and Benjamin Monmege, editors, *38th International Symposium on Theoretical Aspects of Computer Science, STACS 2021, March 16-19, 2021, Saarbrücken, Germany (Virtual Conference)*, volume 187 of *LIPIcs*, pages 28:1–28:16. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2021.

20 Nathanaël Fijalkow and Martin Zimmermann. Cost-parity and cost-streett games. In Deepak D'Souza, Telikepalli Kavitha, and Jaikumar Radhakrishnan, editors, *IARCS Annual Conference on Foundations of Software Technology and Theoretical Computer Science, FSTTCS 2012, December 15-17, 2012, Hyderabad, India*, volume 18 of *LIPIcs*, pages 124–135. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2012. `doi:10.4230/LIPIcs.FSTTCS.2012.124`.

21 Emmanuel Filiot and Edwin Hamel-de Le Court. Two-player Boundedness Counter Games. working paper or preprint, March 2022. `doi:10.4230/LIPIcs`.

22 Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata Logics, and Infinite Games: A Guide to Current Research*. Springer-Verlag, Berlin, Heidelberg, 2002.

23 Orna Kupferman, Nir Piterman, and Moshe Y. Vardi. Pushdown specifications. In *LPAR*, volume 2514, pages 262–277, 2002.

24 Donald A. Martin. Borel determinacy. *Annals of Mathematics*, 102(2):363–371, 1975.

25 Nir Piterman and Amir Pnueli. Faster solutions of rabin and streett games. In *LICS*, pages 275–284, 2006.

26 Martin Zimmermann. Delay games with WMSO+U winning conditions. *RAIRO Theor. Informatics Appl.*, 50(2):145–165, 2016. `doi:10.1051/ita/2016018`.

A Detailed proofs of Section 6

In this section, we prove the following theorem:

▶ **Theorem 15.** *Let $\mathcal{G}$ be the class of counter games G with counter condition $\bigwedge_{c \in C} \mathbb{B}(c)$, where C is the set of counters of G. Given a game G in $\mathcal{G}$, the problem of deciding whether Eve wins G is EXPTIME-c. Finite memory is sufficient for Eve and Adam.*

The proof of Theorem 15 is split into two parts, each covered by a different lemma. Lemma 16 gives the EXPTIME-hardness, and Lemma 22 gives the EXPTIME-easyness.

▶ **Lemma 16.** *Max-counter games with a single winning condition $\mathbb{B}(c)$ for some counter c, and no parity condition, are EXPTIME-hard.*

Proof of Lemma 16. We prove EXPTIME-hardness of max-counter games with no parity condition and a conjunction of boundedness conditions $\bigwedge_{c\in C}\mathbb{B}(c)$. This entails the result because one can always add a counter c_m which takes the maximal value of all other counters $c \in C$ at each step, so that $\bigwedge_{c\in C}\mathbb{B}(c)$ is satisfied iff $\mathbb{B}(c_m)$ is satisfied.

To prove the theorem for conjunctions of boundedness conditions, we reduce the problem, called $\bigcap_n DTOP$, of deciding if the intersection of n languages recognized by deterministic top-down tree automata (DTOP) is empty, which is known to be EXPTIME-c [23]. Before giving the EXPTIME-hardness proof, we first prove PSPACE-hardness for the particular class of counter games where $V_\forall = \varnothing$, i.e., where Adam plays no role. We reduce the problem of deciding if the intersection of n languages recognized by deterministic finite-automata (DFA) is empty. We call the latter problem $\bigcap_n DFA$. The proof is inspired by a PSPACE-hardness proof of deciding non-emptiness of the language recognized by a deterministic min-automaton [8]. Then we lift the reduction from $\bigcap_n DFA$ to the problem $\bigcap_n DTOP$, i.e., to trees, by using the branching nature of counter games induced by Adam.

Consider an alphabet Σ and n complete DFA $D_i = (\Sigma, Q_i, q_0^i, F_i, \delta_i)$ such that all Q_i are pairwise disjoint. We construct a counter arena $\mathcal{A}[D_1,\dots,D_n]$ with $V_\forall = \varnothing$ and a set C of $n+1$ counters, and no parity condition, such that Eve has a strategy to satisfy objective $\bigwedge_{c\in C}\mathbb{B}(c)$ iff $\bigcap_i L(D_i) \neq \varnothing$. This construction is similar to that of [8], which is a reduction from the universality problem for NFA. We assume that Σ contains a symbol $\# \in \Sigma$ and for all i, $L(D_i) \subseteq (\Sigma - \#)^*\#$. The counter arena $\mathcal{A}[D_1,\dots,D_n]$ is defined by $V_\exists = \Sigma$ and $V_\forall = \varnothing$, and the set of transitions is $E = V_\exists \times V_\exists$. The vertex $\#$ is initial. The set of counters is $C = \{c_0\} \cup \{c_q \mid q \in Q_i, i = 1,\dots,n\}$, and they are updated as follows for $i = 1,\dots,n$, where $\max(\varnothing) = 0$:

- on vertex $f \neq \#$: for all $q \in Q_i$, $c_q := \max\{c_{q'}+1 \mid \exists q' \in Q_i, \delta(q', f) = q\}$ and $c_0 := c_0+1$
- on vertex $\#$: $c_{q_O^i} := \max\{c_q \mid q \in Q_{i'}$ for some i' and $\delta_{i'}(q,\#) \notin F_{i'}\}$, and the counters c_q for all $q \in Q_i \setminus \{q_0^i\}$ are reset, as well as c_0.

Note that for $f \neq \#$, two operations are performed at once: increment counters $c_{q'}$ and take the max. This is done to simplify the presentation and can be simulated by doubling the number of vertices of the arena.

Now, observe that $\mathrm{Plays}(\mathcal{A}[D_1,\dots,D_n]) = \#\Sigma^\omega$ and a strategy for Eve is nothing but an infinite word w in $\#\Sigma^\omega$. We prove the following claims:

▷ Claim 17. For all non-empty finite set $X \subseteq \bigcap_{i=1}^n L(D_i)$, any play in $\#.X^\omega$ satisfies $\bigwedge_{c\in C}\mathbb{B}(c)$.

▷ Claim 18. No play in $\#.(\bigcup_{i=1}^n((\Sigma-\#)^*\#)\setminus L(D_i))^\omega$ satisfies $\bigwedge_{c\in C}\mathbb{B}(c)$.

Proof of Claim 17. Let $m = \max\{|u| \mid u \in X\}$. Let $w = \#u_1u_2\dots$ such that for all $j \geq 1$, $u_j \in X$. We prove that w, which is a play of $\mathcal{A}[D_1,\dots,D_n]$) satisfies that all the counters are bounded by $2m$. First, note that each u_j is of the form $v_j\#$, because $u_j \in \bigcap_i L(D_i)$ and the DFA D_i are assumed to accept words where $\#$ is an endmarker. First, consider counter c_0: it is reset every time $\#$ is read, so, its maximal value is bounded by m. Now, for all $j \geq 1$ and $q \in \bigcup_i Q_i$, we let $in_{j,q}$ be the value of counter c_q after prefix $\#u_1\dots u_{j-1}$ and $out_{j,q}$ is value after prefix $\#u_1\dots u_{j-1}v_j$. By definition of the counter updates, we have:

1. $in_{j,q} = 0$ for all $j \geq 1$ and q not initial
2. $in_{j,q_0^i} = \max\{out_{j-1,q} \mid q \in Q_{i'}$ for some i' and $\delta_{i'}(q,\#) \notin F_{i'}\}$ for all $j \geq 1$
3. $out_{j,q} = in_{j,q_0^i} + |v_j|$ if $q \in Q_i$ for some i and there exists a run of D_i from q_0^i to q on v_j
4. otherwise, $out_{j,q} = |r|$ where r is a run of maximal length on a prefix of v_j, ending in q.

Let $q \in Q_i$ for some i such that $\delta_i(q, \#) \notin F_i$. For all $j \geq 1$, there is no run from q_0^i to q on v_j, since $u_j = v_j\# \in L(D_i)$. So, we are in case 4 above and we have $out_{j,q} \leq |v_j| \leq m$. From the latter fact and 2, we get that $in_{j,q_0^i} \leq m$ for all i, j. From that and 3, we get that $out_{j,q} \leq m + |v_j| \leq 2m$ for all j. So, all the counter have value at most $2m$ after each v_j, which concludes the proof that they are bounded. ◁

Proof of Claim 18. Let w be a play of $\mathcal{A}[D_1, \ldots, D_n]$ in $\#.(\bigcup_{i=1}^n((\Sigma-\#)^*\#)\backslash L(D_i))^\omega$. Then, $w = \#w_1\#w_2\#w_3\#\ldots$ such that $w_j \in (\Sigma - \#)^*$ for all $j \geq 1$. Moreover, for all $j \geq 1$, there exists $i_j \in \{1, \ldots, n\}$ such that $w_j\# \notin L(D_{i_j})$ and there exists a run of D_{i_j} on w_j from $q_0^{i_j}$ to some non-accepting state q_{i_j}. Denote by $in(i_j)$ the value of counter $c^0_{q_{i_j}}$ before reading $w_j\#w_{j+1}\ldots$ in w, and by $out(i_j)$ the value of counter $c_{q_{i_j}}$ before reading $\#w_{j+1}\#w_{j+1}\ldots$ in w. By definition of the counter updates, we have $out(i_1) \geq in(i_1) + |u|$, $out(i_2) \geq in(i_2) + |u|$, and so on. Moreover, $in(i_2) \geq out(i_1)$, $in(i_3) \geq out(i_2)$, and so on, since the states q_{i_j} are non-accepting. This yields that the sequence $(in(i_j))_j$ is unbounded, concluding the proof. ◁

As a side note, observe that the two claims imply the following: $\bigcap_{i=1}^n L(D_i) \neq \varnothing$ iff there exists a word $w \in \#\Sigma^\omega$ which satisfies $\bigwedge_{c\in C} \mathbb{B}(c)$. Indeed, if there exists $u \in \bigcap_{i=1}^n L(D_i)$, then it suffices to apply Claim 1 to $X = \{u\}$. Conversely, if $\bigcap_{i=1}^n L(D_i) = \varnothing$, then $(\bigcup_{i=1}^n(\Sigma^*\backslash L(D_i)))^\omega = \Sigma^\omega$ and Claim 2 implies that no word of Σ^ω satisfy $\bigwedge_{c\in C} \mathbb{B}(c)$.

We now lift the latter reduction to (binary) trees. We let Σ be a finite alphabet containing a symbol $\#$ called a constant symbol, and all other symbols are called *binary* symbols. We let $\Sigma_2 = \Sigma - \#$ be the set of binary symbols. A Σ-tree is defined as a term where terms t are inductively defined by $t, t_1, t_2 ::= \# \mid f(t_1, t_2), f \in \Sigma_2$. The set of branches of a Σ-tree t is inductively defined as $br(\#) = \{\#\}$, and $br(f(t_1, t_2)) = \{(f, d).b \mid d \in \{1, 2\}, b \in br(t_d)\}$.

A *deterministic top-down tree automaton* is a tuple $\mathcal{T} = (Q, q_0, F, \delta)$ where Q is a finite set of states, $q_0 \in Q$ the initial state, $F \subseteq Q$ the final states, and $\delta : Q\times(\{\#\}\cup(\Sigma_2\times\{1, 2\})) \to Q$ is a (total) transition function. We see $\mathcal{T}$ as a DFA $DFA(\mathcal{T})$ recognizing a language in $(\Sigma_2 \times \{1, 2\})^*\#$ naturally as follows: $DFA(\mathcal{T}) = (Q, q_0, F, \delta')$ where for all $q \in Q$, for all $(f, d) \in \Sigma_2 \times \{1, 2\}$, $\delta'(q, f) = \mathsf{proj}_d(\delta(q, f))$, with proj_d the dth projection, and $\delta'(q, \#) = \delta(q, \#)$, and we denote by $L_{br}(\mathcal{T})$ the language recognized by this DFA. The language of Σ-trees accepted by $\mathcal{T}$ is the set

$$L(\mathcal{T}) = \{t \in Trees_\Sigma \mid br(t) \subseteq L_{br}(\mathcal{T})\}$$

Deciding[3], given n DTOP $\mathcal{T}_1, \ldots, T_n$, whether $\bigcap_{i=1}^n L(\mathcal{T}_i) = \varnothing$ is EXPTIME-c [16].

Given $\mathcal{T}_1, \ldots, T_n$ such that $\mathcal{T}_i = (Q_i, q_0^i, F_i, \delta_i)$ for all i, we construct a max-counter game G winnable by Eve iff $\bigcap_{i=1}^n L(\mathcal{T}_i) \neq \varnothing$. The main idea of the proof is construct a game where Adam picks a direction $d \in \{1, 2\}$ (1 means left and 2 right), while Eve picks the labels in Σ. The arena $\mathcal{A}[\mathcal{T}_1, \ldots, \mathcal{T}_n]$ of G (without the counters) is depicted on Fig. 3.

We now define counter conditions which make sure that if Eve has a strategy to keep all the counters bounded iff there exists $t \in \bigcap_i L(\mathcal{T}_i)$. For all i, let $\mathcal{T}_i = (Q_i, q_0^i, F_i, \delta_i)$. The set of counters is $C = \{c_q \mid q \in \bigcup_i Q_i\} \cup \{c_0\}$ (we assume wlog that all the sets Q_i are pairwise disjoint). Let us define counter updates. They are defined as for the arena $\mathcal{A}[DFA(\mathcal{T}_1), \ldots, DFA(\mathcal{T}_n)]$. To simplify the presentation (and in particular the structure of the arena), we perform several operations at once. Let us define the updates, for all $1 \leq i \leq n$:

[3] In [16], the definition of DTOP is slightly different, but less general: there are no accepting states but the transition function can be partial. A tree is accepted if there is a run on it which traverses the whole tree (it is not in an inner node). Those automata can easily be encoded into (our) DTOP by completing the transition function into a sink state q_s, declaring all states to be final but q_s.

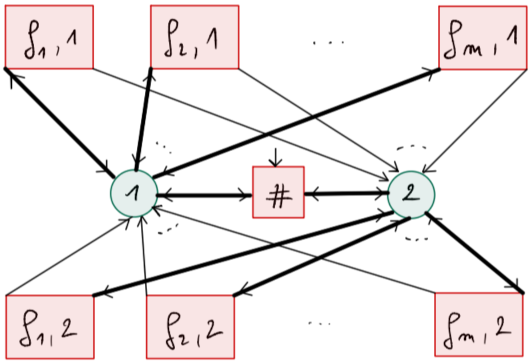

Figure 3 Arena for the proof of Theorem 16, where $\Sigma = \{\#, f_1, \dots, f_m\}$. Transitions in bold are in both directions. Square vertices are controlled by Adam, and the initial vertex is $\#$. When Adam picks a direction $d \in \{1, 2\}$, then Eve is forced to pick a vertex in $\Sigma_2 \times \{d\}$, or $\#$.

- on vertex $(f, j) \in \Sigma_2 \times \{1, 2\}$: for all $q_j \in Q_i$, $c_{q_j} := \max\{c_q + 1 \mid \exists q, q_{3-j} \in Q_i, \delta(q, f) = (q_1, q_2)\}$ and $c_0 := c_0 + 1$
- on vertex $\#$: $c_{q_O^i} := \max\{c_q \mid q \in Q_{i'}$ for some i' and $\delta_{i'}(q, \lambda) \notin F_{i'}\}$, and the counters c_q for all $q \in Q_i \setminus \{q_0^i\}$ are reset, as well as c_0.
- on vertices $i \in \{1, 2\}$: counters are unchanged.

There is no parity condition and the counter condition is that the counters in C must be bounded. Let G be the constructed max-counter game. Before showing correctness, let us introduce some useful notation. Note that the histories and plays of G are elements of $\{\#\} \cup \Sigma_2 \times \{1, 2\}$ alternating with directions in $\{1, 2\}$. The following function removes the intermediate directions. Given $w = \lambda_1 d_1 \lambda_2 d_2 \dots \lambda_n d_n$ such that for all i, $\lambda_i \in \{\#\} \cup \Sigma_2 \times \{1, 2\}$ and $d_i \in \{1, 2\}$, we let $lab(w) = \lambda_1 \lambda_2 \dots \lambda_n$.

We now show correctness of the reduction. Suppose that there exists some $t \in \bigcap_i L(\mathcal{T}_i)$. We first define a strategy σ_t for Eve and then show it is winning in G. The strategy σ_t just mimics t: it plays as t dictates when a leaf of t is reached, its behaviour is reset to the root of t. Formally, the construction of σ_t satisfies the following invariant: all histories ending with an Eve vertex are words of the form $h = \# h_1 h_2 \dots h_k p d$ where:

- all h_i are such that $lab(h_i) \in br(t)$,
- $lab(p)$ is a prefix of a branch of t
- $d \in \{1, 2\}$

Given such a history h, we consider two cases: if $lab(p) \in br(t)$, then σ_t is reset to the root of t, which means that $\sigma_t(h) = (f, d)$ such that f is the label of the root of t. Otherwise, $\sigma_t(h) = \#$ if $lab(p)\# \in br(t)$, and $\sigma_t(h) = (f, d)$ if $lab(p)(f, d) \in br(t)$. Let us show that σ_t is winning. Let $\pi \in \text{Plays}(\sigma_t)$. First, we observe that $lab(\pi)$ is a play of $\mathcal{A}[DFA(\mathcal{T}_1), \dots, DFA(\mathcal{T}_n)]$. Let C' be the set of counters of the latter arena. By definition of σ_t, $lab(\pi)$ is of the form $\# b_1 \# b_2 \# \dots$ with infinitely many $\#$ such that for all $j \geq 1$, $b_j \#$ is a branch of t. Since $t \in \bigcap_i L(\mathcal{T}_i)$, we also get that $b_j \# \in \bigcap_i L(DFA(\mathcal{T}_i))$. The set $X = \{b_j \# \mid j \geq 1\}$ is finite since its elements correspond to branches of t. Therefore, by Claim 1, π satisfies $\bigwedge_{c \in C'} \mathbb{B}(c)$. We conclude by observing that $C = C'$, that $\mathcal{A}[\mathcal{T}_1, \dots, \mathcal{T}_n]$ has the same vertices as $\mathcal{A}[DFA(\mathcal{T}_1), \dots, DFA(\mathcal{T}_n)]$ plus the two vertices 1 and 2, with the same counter updates for their common vertices and no update on 1 and 2. Therefore, π satisfies $\bigwedge_{c \in C} \mathbb{B}(c)$ in $\mathcal{A}[\mathcal{T}_1, \dots, \mathcal{T}_n]$.

Conversely, suppose that $\bigcap_i L(\mathcal{T}_i) = \varnothing$. Take an arbitrary strategy σ of Eve. We show it is not winning. Intuitively, σ can be seen as an infinite tree. If there is a branch of the tree which visits # finitely many times, then σ is not winning because by following the directions corresponding to that branch, Adam can guarantee that counter c_0 is unbounded. So, we can assume that σ is such that all plays consistent with it sees infinitely many #. We construct a play π of the form $\#h_1\#h_2\#\dots$ such that for all $j \geq 1$, there exists i such that $lab(h_j)\# \notin L(DFA(\mathcal{T}_i))$, and we conclude by Claim 2.

Consider the set of histories H_1 of σ which contains a # symbol only at their end. Clearly, H_1 can be identified with a Σ-tree t_1. Since $t_1 \notin \bigcap_i L(\mathcal{T}_i)$, there exists i such that $t_1 \notin L(\mathcal{T}_i)$ and therefore, a history $h_1\# \in H_1$ such that $lab(h_1)\# \notin L(DFA(\mathcal{T}_i))$. To construct $h_2, h_3, \dots$, we proceed similarly. Let us explain how to construct h_2. We let H_2 be the set of histories of the form $h_1\#g_2\#$ such that $h_1\#g_2\#$ is a history of σ such that g_2 does not contain #. The set $(h_1\#)^{-1}H_2$ can be identified with a Σ-tree t_2. Now, it suffices to take $h_2\# \in (h_1\#)^{-1}H_2$ such that $lab(h_2\#) \notin L(DFA(\mathcal{T}_i))$ for some $i = 1, \dots, n$. It exists since $t_2 \notin \bigcap_i L(\mathcal{T}_i)$. This concludes the proof. ◀

In order to prove Lemma 22, we first prove the following, in a very similar way to the proof of Lemma 2.

▶ **Lemma 19.** *Max-counter games (with Boolean combinations of boundedness objectives) are determined.*

Proof. Given a counter arena $\mathcal{A}$ and a counter c of $\mathcal{A}$, the set $Plays\,(\mathcal{A}, \mathbb{B}(c))$ is a Borel set. Indeed, it is equal to the countable union for all $N \geq 0$ of the sets

$$\mathrm{Plays}_N(\mathcal{A}, \mathbb{B}(c)) = \{\rho \in \mathrm{Plays}(\mathcal{A}) \mid \forall n \in \mathbb{N}, \lambda(\zeta(\rho), c)_n \leq N\}$$

which are ω-regular. Indeed, a Büchi automaton that stores, in every state, the maximums between N and the value of each counter of C needs $|V| \times N^{|C|}$ states to recognize $\mathrm{Plays}_N(\mathcal{A}, \mathbb{B}(c))$. Since ω-regular sets are Borel, so is $Plays\,(\mathcal{A}, \mathbb{B}(c))$, as well as any Boolean combination of the latter. By Martin's determinacy theorem [24], the result follows. ◀

Furthermore, to make the proof of Lemma 22 clearer, we now define two models of automata: non-deterministic $\mathbb{U}$-automata, and Parity-Rabin automata.

A (non-deterministic) $\mathbb{U}$-*automaton* B is a nine-tuple $(\Sigma, S, s_i, \Delta, Q, \kappa, C, \zeta, C_1)$, where Σ is an alphabet, S is a finite set of states, $s_i \in S$ is the initial state, $\Delta \in S \times S \times \Sigma$ is a transition function, Q is finite set of colors, κ is an alphabet colouring from Σ to Q, C is a finite set of counters, ζ is a state labeling from S to $\mathsf{Op}(C)$, and C_1 is a subset of C. A *run* in B is an infinite word $\pi = y_0 y_1 \cdots \in \Delta^\omega$ such that $y_0 = s_i$, and such that the second element of each y_i is the first element of y_{i+1} for any non-negative integer i. We let $\mathrm{States}(\pi)$ denote the word $v_0 v_1 \cdots$, where each v_i is the first element of y_i, and we let $\mathrm{Input}(\pi)$ denote the word $z_0 z_1 \cdots$, where each z_i is the third element of y_i (*i.e.* the label of the edge y_i). A word w is *accepted* by B if either $w \in \mathrm{Parity}(\kappa)$ (*i.e.* if the greatest color seen infinitely often in w is even), or if there exists a run π of B such that $\mathrm{Input}(\pi) = w$ and such that $\mathrm{States}(\pi)$ satisfies $\bigvee_{c \in C_1} \mathbb{U}(c)$. The language accepted by B is the set of accepted words.

A (non-deterministic) *Parity-Rabin automaton* D is a variant of a Rabin automaton, and is defined as a seven-tuple $(\Sigma, S, q_i, \Delta, Q, \kappa, \{\kappa_i\}_{i \in \{1,\dots,\ell\}})$ where Σ, S, q_i, Δ, Q and κ are defined in the same way as in the the case of $\mathbb{U}$-automata, where and $\{\kappa_i\}_{i \in \{1,\dots,\ell\}}$ is a finite set of colourings from S to $\{1, 2, 3\}$. Furthermore, a word w is *accepted* by D if and only if either w is in $\mathrm{Parity}(\kappa)$, or there exists an integer $i \in \{1, \dots, \ell\}$ and a run ρ of D such that $\mathrm{Input}(\rho) = w$ and such that $\mathrm{States}(\rho)$ is in $\mathrm{Parity}(\kappa_i)$. The language recognized by D, denoted $\mathcal{L}(D)$, is the set of words accepted by D.

▶ **Lemma 20.** *The language recognized by a non-deterministic Parity-Rabin automaton is ω-regular. Furthermore, games with an objective given by a non-deterministic Parity-Rabin automaton are solvable in EXPTIME.*

Proof. A Parity-Rabin automaton D can be converted into a non-deterministic automaton D_1, with $\ell + 1$ colours whose domains are the set of states, by copying each state for every transition that goes to it, and transferring the colour κ to the states depending on which incoming transition the copy represents. The acceptation condition of D_1 is expressed by the union of the parity conditions induced by its colourings. The automaton D_1 can be further converted into a non-deterministic parity automaton D_2 with a single colouring, by copying it for every colouring it has, colouring the first copy with the first colouring, the second copy with the second colouring, etc. Thus, there exists a parity automaton D_2 that recognizes the same language as D, with a size polynomial in the size of D. One of the consequences of that statement is that $\mathcal{L}(D)$ is thus an ω-regular language. Furthermore, it is well-known that we can determinize D_2 into a deterministic parity automaton D_3 with exponential size and linear index, in exponential time. In addition, if G' is the game obtained from the product of a game G and the deterministic parity automaton D_3, G' is a parity game of exponential size in the size of G and D, and index linear in the number of colours of D, such that Eve wins G' if and only if Eve wins G. Thus, since parity games with m edges, n vertices and index k can be solved in $O(mn^k)$ (see e.g. [14]), the class of games with an objective given by a non-deterministic Parity-Rabin automaton is in EXPTIME. ◀

We now show that the class of counter games with a counter condition given by a non-deterministic $\mathbb{U}$-automaton with an acceptance condition of the form $\bigvee \mathbb{U}$ is also decidable in EXPTIME, by converting them into Parity-Rabin automata.

▶ **Lemma 21.** *Let B be a non-deterministic $\mathbb{U}$-automaton with acceptance condition of the form $\bigvee_{c\in C_1} \mathbb{U}(c)$, and $\mathcal{A}$ be a two-player arena. We can decide if Eve wins $(\mathcal{A}, \mathcal{L}(B))$ in EXPTIME.*

Proof. Let $B = (S, \Delta, i, \zeta, \bigvee_{c\in C_1} \mathbb{U}(c), \kappa)$. We construct in polynomial time a Parity-Rabin automaton D such that Eve wins $(\mathcal{A}, \mathcal{L}(B))$ if and only if Eve wins $(\mathcal{A}, \mathcal{L}(D))$. The idea is to keep the same automata structure as B, the same parity function, and to replace each atom $\mathbb{U}(c)$ by a parity function which is satisfied iff there is infinitely many increase of c and finitely many reset of c. So, for each counter c we introduce the parity function κ_c defined by:

$$\kappa_c(x) = \begin{cases} 1 & \text{if } \zeta_c(x) = \mathsf{skip} \\ 2 & \text{if } \zeta_c(x) = \mathsf{i} \\ 3 & \text{if } \zeta_c(x) = \mathsf{r} \end{cases}$$

We show that Eve wins $(\mathcal{A}, \mathcal{L}(B))$ if and only if Eve wins $(\mathcal{A}, \mathcal{L}(D))$. If Eve wins $(\mathcal{A}, \mathcal{L}(D))$, then she wins $(\mathcal{A}, \mathcal{L}(B))$ with the same winning strategy, as $\mathcal{L}(D) \subseteq \mathcal{L}(B)$. Suppose now that σ is a winning strategy of Eve for $(\mathcal{A}, \mathcal{L}(B))$, and that Eve does not win $(\mathcal{A}, \mathcal{L}(D))$. However, by Lemma 20, $\mathcal{L}(D)$ is ω-regular and $(\mathcal{A}, \mathcal{L}(D))$ is thus determined. Therefore, Adam has a finite memory winning strategy τ for $(\mathcal{A}, \overline{\mathcal{L}(D)})$. We exhibit a contradiction. Let ρ be a play of $\mathcal{A}$ consistent with σ and τ. Then ρ satisfies both of the following properties:

1. $\rho \notin \text{Parity}(\kappa)$, and for any run π of D over ρ, for any counter c, if π sees infinitely many increase of c, then it sees infinitely many reset of c (because τ is winning)
2. either $\rho \in \text{Parity}(\kappa)$, or there exists a counter $c_0 \in C_1$, and there exists a run π of B over ρ such that $\text{States}(\pi)$ satisfies $\mathbb{U}(c)$ (it is because σ is winning)

Now, since $\rho \notin \text{Parity}(\kappa)$ from property 1, along ρ, c_0 is unbounded from property 2, so it sees infinitely many increase, and by property 1 it must see infinitely many reset. Intuitively, it implies that they are longer and longer segments in between two consecutive resets with more and more increase of c_0. Since τ is finite-memory, Eve can find a cycle (both cycling on the arena, the memory-structure of the strategy and the automaton B) which contains at least one increase of c_0 and no reset. By iterating this cycle *ad infinitum*, she creates a play which is consistent with τ and a run of D over that new play, which sees infinitely many increase of c_0 but finitely many reset, contradicting Property 1.

Since D can be computed in polynomial time from B, and since the class of games with an objective given by a non-deterministic Parity-Rabin automaton is in EXPTIME, we can decide if Eve wins $(\mathcal{A}, \mathcal{L}(B))$ in EXPTIME. ◀

▶ **Lemma 22.** *Given a game in $\mathcal{G}$, the problem of deciding whether Eve wins $\mathcal{G}$ is in EXPTIME. Finite memory is sufficient for Eve and Adam.*

Proof. We show that counter games G with counter condition of the form

$$Plays\left(\mathcal{A}, \bigvee_{c \in C} \mathbb{U}(c)\right) \cup \text{Parity}(\mathcal{A}),$$

where C is the set of counters of G, and $\mathcal{A}$ its underlying two-player arena, can be solved in EXPTIME, which implies the lemma by Lemma 19.

We construct, from a max-counter game G, a game G' whose acceptance condition is $\mathbb{U}$-automaton D of size polynomial in the size of G.

Let G be a counter game with underlying two-player arena $\mathcal{A} = (V, E, V_\exists, V_\forall, v)$, vertex labeling ζ, set of colors Q, colouring κ, and winning condition $Plays\left(\mathcal{A}, \bigvee_{c \in C} \mathbb{U}(c)\right) \cup \text{Parity}(\mathcal{A})$. We construct a $\mathbb{U}$-automaton B, of size polynomial in $|C|$, with a single counter denoted d (we assume $d \notin C$), that recognizes the language of all words $w \in V^\omega$ such that either $w \in \text{Parity}(\kappa)$, or $\zeta(w)$ satisfies the condition $\bigvee_{c \in C} \mathbb{U}(c)$. To make the construction more easily understood, we first introduce the notion of trace. A *trace* of a word $w = z_0 z_1 \cdots \in \mathbf{Op}(C)^\omega$ is a mapping θ from $\{i, \ldots, j\}$ to C, where $i \leq j$ are two integers, such that, for any $l \in \{i, \ldots, j-1\}$,

- either $\theta(l) = \theta(l+1)$ and $z_l(\theta(l)) \in \{\mathsf{i}, \mathsf{r}, \mathsf{skip}\}$,
- or $\theta(l+1) \neq \theta(l)$ and $z_l(\theta(l+1)) = \max_{c \in S}(c)$ with $S \subseteq C$ and $\theta(l) \in S$.

The *value* of θ at move $t \in \{i, \ldots, j\}$ is defined inductively as 0 if $t = i$, one plus the value at move $t-1$ if $z_{t-1}(\theta(t-1)) = \mathsf{i}$, 0 if $z_{t-1}(\theta(t-1)) = \mathsf{r}$, and the value at move $t-1$ otherwise. If a counter c reaches a value $N \geq 1$ at some point in w, then it is always possible to "track back", with a trace of w, the sequence of counter operations which led to c having that value, by choosing, every time we go back to a previous counter operation of the type $c' = \max_{d \in S}(d)$ with $S \subseteq C$, the good counter d of S (the one with the maximum value), until reaching a counter whose value is 0. Thus, there exists a counter $c \in C$ and two integers t and N such that $\lambda(w, c)_t = N$ if and only if there exists a trace θ of w, such $\theta(t) = c$, and such that the value of θ at move t is N. As a consequence, there exists counter c such that $\lambda(w, c)$ is unbounded if and only if the values of the traces of w are unbounded.

This result allows us to define $\mathcal{B}$ in the following way. The $\mathbb{U}$-automaton B works, on input w, by guessing all the possible traces of $\zeta(w)$, by using non-determinism. The value of a trace is stored inside the counter d. More precisely, every time B reads a letter, it either guesses a new trace, or guesses the next counter c' of C of the trace it is following, while applying, if c' is equal to the current counter c of the trace, the operation over c induced by the letter read, to counter d. Thus, the $\mathbb{U}$-automaton B is constructed so that the value of d is unbounded if and only if there are traces of its input of arbitrarily large values. Moreover, we set the colouring of B as κ. Thus, B recognizes the language of all words $w \in V^\omega$ such that either $w \in \text{Parity}(\kappa)$, or $\zeta(w)$ satisfies the condition $\bigvee_{c \in C} \mathbb{U}(c)$, *i.e.* the language recognized by B is the winning condition of the game G. The precise definition of B is given below.

We let $V_1 = C \times \{\mathsf{i}, \mathsf{r}, \mathsf{skip}\}$, and $v_1 = (c, \mathsf{r})$ where c is any counter in C. Furthermore, we let ζ^1 denote the mapping from V_1 to $\mathbf{Op}(\{d\})$ such that $(\zeta^1(c, \alpha))(d) = \alpha$, and E_1 denote the the smallest subset of $V_1 \times V_1 \times V$ such that, for any $\alpha \in \{\mathsf{i}, \mathsf{r}, \mathsf{skip}\}$ and any $v \in V$, we have

- for any $c, c' \in C$, $((c, \alpha), (c', \mathsf{r}), v) \in E_1$ (this comes from the fact that B should be able to guess a new trace at any time),
- for any $c \in C$, if $\zeta_c(v) \in \{\mathsf{i}, \mathsf{skip}\}$, $((c, \alpha), (c, \zeta_c(v)), v) \in E_1$ (the trace follows the increment or skip operation of a counter while updating d),
- for any $c \in C$, if $\zeta_c(v) = \max_{c' \in S}(c)$, then $((c', \alpha), (c, \mathsf{skip}), v)) \in E_1$, for any $c' \in S$ (the trace changes counters on a max operation while leaving d unchanged).

The $\mathbb{U}$-automaton B is the $\mathbb{U}$-automaton $(V, V_1, v_1, E_1, Q, \kappa, \{d\}, \zeta_1, \{d\})$. By Lemma 21 and Lemma 20, since B can be computed in a polynomial time from G, and since Eve wins G if and only if Eve wins $(\mathcal{A}, \mathcal{L}(B))$, we can decide if Eve wins G in EXPTIME. ◀