

Trading Bounds for Memory in Games with Counters

Nathanaël Fijalkow^{1,2}, Florian Horn¹, Denis Kuperberg^{2,3}, and Michał Skrzypczak^{1,2}

¹ LIAFA, Université Paris 7

² Institute of Informatics, University of Warsaw

³ Onera/DTIM, Toulouse and IRIT, University of Toulouse

Abstract. We study two-player games with counters, where the objective of the first player is that the counter values remain bounded. We investigate the existence of a trade-off between the size of the memory and the bound achieved on the counters, which has been conjectured by Colcombet and Loeding.

We show that unfortunately this conjecture does not hold: there is no trade-off between bounds and memory, even for finite arenas. On the positive side, we prove the existence of a trade-off for the special case of thin tree arenas. This allows to extend the theory of regular cost functions over thin trees, and obtain as a corollary the decidability of cost monadic second-order logic over thin trees.

1 Introduction

This paper studies finite-memory determinacy for games with counters. The motivation for this investigation comes from the theory of regular cost functions, which we discuss now.

Regular cost functions. The theory of regular cost functions is a *quantitative* extension of the notion of regular languages, over various structures (words and trees). More precisely, it expresses *boundedness questions*. A typical example of a boundedness question is: given a regular language $L \subseteq \{a, b\}^*$, does there exist a bound N such that all words from L contain at most N occurrences of a ?

This line of work has already a long history: it started in the 80s, when Hashiguchi, and then later Leung, Simon and Kirsten solved the *star-height problem* by reducing it to boundedness questions [12, 17, 15, 13]. Both the logics MSO+U and later cost MSO (as part of the theory of regular cost functions) emerged in this context [3, 4, 6, 8, 10], as quantitative extensions of the notion of regular languages allowing to express boundedness questions.

Consequently, developing the theory of regular cost functions comes in two flavours: the first is using it to reduce various problems to boundedness questions, and the second is obtaining decidability results for the boundedness problem for cost MSO over various structures.

For the first point, many problems have been reduced to boundedness questions. The first example is the star-height problem over words [13] and over

trees [10], followed for instance by the boundedness question for fixed points of monadic formulae over finite and infinite words and trees [2]. The most important problem that has been reduced is to decide the Mostowski hierarchy for infinite trees [9].

For the second point, it has been shown that over finite words and trees, a significant part of the theory of regular languages can successfully be extended to the theory of regular cost functions, yielding notions of regular expressions, automata, semigroups and logics that all have the same expressive power, and that extend the standard notions. In both cases, algorithms have been constructed to answer boundedness questions.

However, extending the theory of regular cost functions to infinite trees seems to be much harder, and the major open problem there is the decidability of cost MSO over infinite trees.

LoCo conjecture. Colcombet and Loeding pointed out that the only missing point to obtain the decidability of cost MSO is a finite-memory determinacy result for games with counters. More precisely, they conjectured that there exists a trade-off between the size of the memory and the bound achieved on the counters [7]. So far, this conjecture resisted both proofs and refutations, and the only non-trivial positive case known is due to Vanden Boom [18], which implied the decidability of the weak variant of cost MSO over infinite trees, later generalized to quasi-weak cost MSO in [1]. Unfortunately, weak cost MSO is strictly weaker than cost MSO, and this leaves open the question whether cost MSO is decidable.

Contributions. In this paper, we present two contributions:

- There is no trade-off, even for finite arenas, which disproves the conjecture,
- There is a (non-elementary) trade-off for the special case of thin tree arenas.

Our first contribution does not imply the undecidability of cost MSO, it rather shows that proving the decidability will involve subtle combinatorial arguments that are yet to be understood. As a corollary of the second contribution, we obtain the decidability of cost MSO over thin trees.

Structure of this document. The definitions are given in Section 2. We state the conjecture in Section 3. The Section 4 disproves the conjecture. The Section 5 proves that the conjecture holds for the special case of thin tree arenas.

2 Definitions

Arenas. The games we consider are played by two players, Eve and Adam, over potentially infinite graphs called arenas⁴. Formally, an arena G consists of a directed graph (V, E) whose vertex set is divided into vertices controlled by Eve (V_E) and vertices controlled by Adam (V_A). A token is initially placed on a given

⁴ We refer to [11] for an introduction to games.

initial vertex v_0 , and the player who controls this vertex pushes the token along an edge, reaching a new vertex; the player who controls this new vertex takes over, and this interaction goes on forever, describing an infinite path called a *play*. Finite or infinite plays are paths in the graphs, seen as sequences of edges, typically denoted π . In its most general form, a strategy for Eve is a mapping $\sigma : E^* \cdot V_E \rightarrow E$, which given the history played so far and the current vertex picks the next edge. We say that a play $\pi = e_0 e_1 e_2 \dots$ is consistent with σ if $e_{n+1} = \sigma(e_0 \dots e_n \cdot v_n)$ for every n with $v_n \in V_E$.

Winning conditions. A winning condition for an arena is a set of a plays for Eve, which are called the winning plays for Eve (the other plays are winning for Adam). A strategy for Eve is winning for a condition, or ensures this condition, if all plays consistent with the strategy belong to the condition. For a winning condition W , we denote $\mathcal{W}_E(W)$ the winning region of Eve, *i.e.* the set of vertices from which Eve has a winning strategy.

Here we will consider the classical parity condition as well as *quantitative* bounding conditions.

The parity condition is specified by a colouring function $\Omega : V \rightarrow \{0, \dots, d\}$, requiring that the maximum color seen infinitely often is even. The special case where $\Omega : V \rightarrow \{1, 2\}$ corresponds to Büchi conditions, denoted $\text{Büchi}(F)$ where $F = \{v \in V \mid \Omega(v) = 2\}$. We will also consider the simpler conditions $\text{Safe}(F)$ and $\text{Reach}(F)$, for $F \subseteq V$: the first requires to avoid F forever, and the second to visit a vertex from F at least once.

The bounding condition B is actually a family of winning conditions with an integer parameter $B = \{B(N)\}_{N \in \mathbb{N}}$. We call it a quantitative condition because it is monotone: if $N < N'$, all the plays in $B(N)$ also belong to $B(N')$.

The counter actions are specified by a function $c : E \rightarrow \{\varepsilon, i, r\}^k$, where k is the number of counters: each counter can be incremented (i), reset (r), or left unchanged (ε). The value of a play π , denoted $\text{val}(\pi)$, is the supremum of the value of all counters along the play. It can be infinite if one counter is unbounded. The condition $B(N)$ is defined as the set of play whose value is less than N .

In this paper, we study the condition B -parity, where the winning condition is the intersection of a bounding condition and a parity condition. The value of a play that satisfies the parity condition is its value according to the bounding condition. The value of a play which does not respect the parity condition is ∞ . We often consider the special case of B -reachability conditions, denoted $B \text{ Until } F$. In such cases, we assume that the game stops when it reaches F .

Given an initial vertex v_0 , the value $\text{val}(v_0)$ is:

$$\inf_{\sigma} \sup_{\pi} \{ \text{val}(\pi) \mid \pi \text{ consistent with } \sigma \text{ starting from } v_0 \} .$$

Finite-memory strategies. A *memory structure* \mathcal{M} for the arena G consists of a set M of memory states, an initial memory state $m_0 \in M$ and an update function $\mu : M \times E \rightarrow M$. The update function takes as input the current memory state and the chosen edge to compute the next memory state, in a

deterministic way. It can be extended to a function $\mu : E^* \cdot V \rightarrow M$ by defining $\mu^*(v) = m_0$ and $\mu^*(\pi \cdot (v, v')) = \mu(\mu^*(\pi \cdot v), (v, v'))$.

Given a memory structure \mathcal{M} , a strategy is induced by a next-move function $\sigma : V_E \times M \rightarrow E$, by $\sigma(\pi \cdot v) = \sigma(v, \mu^*(\pi \cdot v))$. Note that we denote both the next-move function and the induced strategy σ . A strategy with memory structure \mathcal{M} has finite memory if M is a finite set. It is *memoryless*, or *positional* if M is a singleton: it only depends on the current vertex. Hence a memoryless strategy can be described as a function $\sigma : V_E \rightarrow E$.

An arena G and a memory structure \mathcal{M} for G induce the expanded arena $G \times \mathcal{M}$ where the current memory state is stored explicitly along the current vertex: the vertex set is $V \times M$, the edge set is $E \times \mu$, defined by: $((v, m), (v', m')) \in E'$ if $(v, v') \in E$ and $\mu(m, (v, v')) = m'$. There is a natural one-to-one correspondence between memoryless strategies in $G \times \mathcal{M}$ and strategies in G using \mathcal{M} as memory structure.

3 The conjecture

In this section, we state the conjecture [7], and explain how positive cases of this conjecture imply the decidability of cost MSO.

3.1 Statement of the conjecture

There exists $\text{mem} : \mathbb{N}^2 \rightarrow \mathbb{N}$ and $\alpha : \mathbb{N}^3 \rightarrow \mathbb{N}$ such that for all B -parity games with k counters, $d + 1$ colors and initial vertex v_0 , there exists a strategy σ using $\text{mem}(d, k)$ memory states, ensuring $B(\alpha(d, k, \text{val}(v_0))) \cap \text{Parity}(\Omega)$.

The function α is called a trade-off function: if there exists a strategy ensuring $B(N) \cap \text{Parity}(\Omega)$, then there exists a strategy with *small* memory that ensures $B(\alpha(d, k, N)) \cap \text{Parity}(\Omega)$. So, at the price of increasing the bound from N to $\alpha(d, k, N)$, one can use a strategy using a small memory structure.

To get a better understanding of this conjecture, we show three simple facts:

1. why reducing memory requires to increase the bound,
2. why the memory bound mem depends on the number of counters k ,
3. why a weaker version of the conjecture holds, where mem depends on the value,

For the first point, we present a simple game, represented in Figure 1. It involves one counter and the condition B Until F . Starting from v_0 , the game moves to v and sets the value of the counter to N . The objective of Eve is to take the edge to the right to F . However, this costs N increments, so if she wants the counter value to remain smaller than N she has to set its value to 0 before taking this edge. She has N options: for $\ell \in \{1, \dots, N\}$, the ℓ^{th} option consists in going to u_ℓ , involving the following actions:

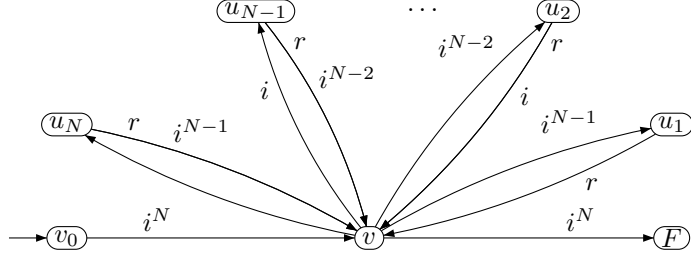


Fig. 1. A trade-off is necessary.

- first, take $N - \ell$ increments,
- then, reset the counter,
- then, take $\ell - 1$ increments, setting the value to $\ell - 1$.

It follows that there is a strategy for Eve to ensure $B(N)$ Until F , which consists in going successively through u_N , u_{N-1} , and so on, until u_1 , and finally to F . Hence to ensure that the bound is always smaller than N , Eve needs $N + 1$ memory states.

However, if we consider the bound $2N$ rather than N , then Eve has a very simple strategy, which consists in going directly to F , using no memory at all. This is a simple example of a trade-off: to ensure the bound N , Eve needs $N + 1$ memory states, but to ensure the worse bound $2N$, she has a positional strategy.

For the second point, consider the following simple game with k counters (numbered cyclically) and only one vertex, controlled by Eve. There are k self-loops, each incrementing a counter and resetting the previous one. Eve has a simple strategy to ensure $B(1)$, which consists in cycling through the loops, and uses k memory states. Any strategy using less than k memory states ensures no bound at all, as one counter would be incremented infinitely many times but never reset. It follows that the memory bound mem in the conjecture has to depend on k , and this example shows that $\text{mem} \geq k$ is necessary.

For the third point, we give an easy result that shows the existence of finite memory strategies whose size depend on the value, even without losing anything on the bound. Unfortunately, this statement is not strong enough; we discuss in the next subsection the implications of the conjecture.

Lemma 1. *For all B -parity games with k counters and initial vertex v_0 , there exists a strategy σ ensuring $B(\text{val}(v_0)) \cap \text{Parity}(\Omega)$ with $(\text{val}(v_0) + 1)^k$ memory states.*

Proof. We consider the memory structure $\mathcal{M} = (\{0, \dots, N\}^k, 0^k, \mu)$ which keeps track of the counter values, where $N = \text{val}(v_0)$. We construct the arena $G \times \mathcal{M}$ and add a new vertex \perp which is reached if a counter reaches the value $N + 1$, according to the memory structure. The condition $\text{Safe}(\perp) \cap \text{Parity}(\Omega)$, which

requires never to reach \perp and to satisfy the parity condition, is equivalent to $B(N) \cap \text{Parity}(\Omega)$. Since parity games are positionally determined, there exists a positional strategy ensuring $\text{Safe}(\perp) \cap \text{Parity}(\Omega)$, which induces a finite-memory strategy using \mathcal{M} as memory structure ensuring $B(N) \cap \text{Parity}(\Omega)$. \square

3.2 The interplay with cost MSO

The conjecture stated above has a purpose: if true, it implies the decidability of cost MSO over infinite trees. More precisely, the technical difficulty to develop the theory of regular cost functions over infinite trees is to obtain effective constructions between variants of automata with counters, and this is what this conjecture is about.

In the qualitative case (without counters), to obtain the decidability of MSO over infinite trees, known as Rabin’s theorem [16], one transforms MSO formulae into equivalent automata. The complementation construction is the technical cornerstone of this procedure. The *key* ingredient for this is games, and specifically positional determinacy for parity games. Similarly, other classical constructions, to simulate either two-way or alternating automata by non-deterministic ones, make a crucial use of positional determinacy for parity games.

In the quantitative case now, Colcombet and Loeding [10] showed that to extend these constructions, one needs a similar result on parity games with counters, which is the conjecture we stated above.

So far, there is only one positive instance of this conjecture, which is the special case of B-Büchi games over chronological arenas⁵.

Theorem 1 ([18]). *For all B-Büchi games with k counters and initial vertex v_0 over chronological arenas, Eve has a strategy ensuring $B(2 \cdot \text{val}(v_0)) \cap \text{Büchi}(F)$ with $2 \cdot k!$ memory states.*

It leads to the following decidability result.

Corollary 1 ([18]). *Weak cost MSO over infinite trees is decidable.*

4 No trade-off over Finite Arenas

In this section, we show that the conjecture does not hold, even for finite arenas.

Theorem 2. *For all K , for all N , there exists a finite B-reachability game $G_{K,N}$ with one counter such that:*

- *there exists a 3^K memory states strategy ensuring $B(K(K+3))$ Until F ,*
- *no $K+1$ memory states strategy ensure $B(N)$ Until F .*

We proceed in two steps. The first is an example giving a lower bound of 3, and the second is a nesting of this first example.

⁵ The definition of chronological arenas is given in Section 5.

4.1 A first lower bound of 3

We start with a game G_1 , which gives a first lower bound of 3. It is represented in Figure 2. The condition is B Until F . In this game, Eve is torn between going to the right to reach F , which implies incrementing the counter, and going to the left, to reset the counter. The actions of Eve from the vertex u_N are:

- *increment*, and go one step to the right, to v_{N-1} ,
- *reset*, and go two steps to the left, to v_{N+2} .

The actions of Adam from the vertex v_N are:

- *play*, and go down to u_N ,
- *skip*, and go to v_{N-1} .

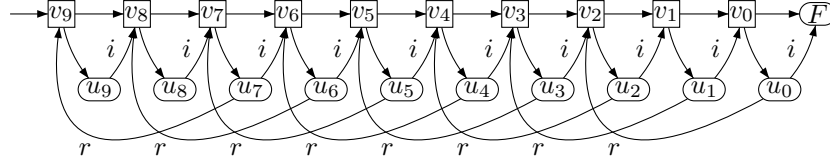


Fig. 2. Part of the game G_1 , where Eve needs 3 memory states.

Formally:

$$V = \begin{cases} V_E = \{u_n \mid n \in \mathbb{N}\} \\ V_A = \{v_n \mid n \in \mathbb{N}\} \end{cases}$$

$$E = \begin{cases} \{v_{n+1} \rightarrow v_n \mid n \in \mathbb{N}\} \\ \cup \{v_n \rightarrow u_n \mid n \in \mathbb{N}\} \\ \cup \{u_{n+1} \xrightarrow{i} v_n \mid n \in \mathbb{N}\} \cup \{u_0 \xrightarrow{i} F\} \\ \cup \{u_n \xrightarrow{r} v_{n+2} \mid n \in \mathbb{N}\} \end{cases}$$

Theorem 3. In G_1 :

- Eve has a 4 memory states strategy ensuring $B(3)$ Until F ,
- Eve has a 3 memory states strategy ensuring $B(4)$ Until F ,
- For all N , no 2 memory states strategy ensures $B(N)$ Until F from v_N .

The first item follows from Lemma 1. However, to illustrate the properties of the game G_1 we will provide a concrete strategy with 4 memory states that ensures $B(3)$ Until F . The memory states are i_1, i_2, i_3 and r , linearly ordered by $i_1 < i_2 < i_3 < r$. With the memory states i_1, i_2 and i_3 , the strategy chooses to increment, and updates its memory state to the next memory state. With the

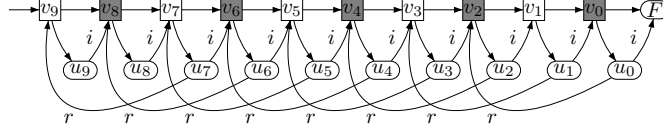


Fig. 3. Illustration of the 3 memory states strategy in G_1 .

memory state r , the strategy chooses to reset, and updates its memory state to i_1 . This strategy satisfies a simple invariant: it always resets to the right of the previous reset, if any.

We show how to save one memory state, at the price of increasing the bound by one: we construct a 3 memory states strategy ensuring $B(4)$ Until F . The idea, as represented in Figure 3, is to color every second vertex and to use this information to track progress. The 3 memory states are called i , j and r . The update is as follows: the memory state is unchanged in uncoloured (white) states, and switches from i and j and from j to r on gray states. The strategy is as follows: in the two memory states i and j , Eve chooses to increment, and in r she chooses to reset. As for the previous strategy, this strategy ensures that it always resets to the right of the previous reset, if any.

We now show that 2 memory states is not enough. Assume towards contradiction that there exists a 2 memory states strategy ensuring $B(N)$ Until F from v_{2N} , for some N , using the memory structure $\mathcal{M} = (M, \mu, m_0)$.

We first argue that without loss of generality we can assume that the strategy σ is normalized, *i.e.* satisfies the following three properties:

1. for all $n \leq 2N$, there is at least one memory state that chooses increment from u_n ,
2. for all $n \leq 2N$ but at most N of them, there is at least one memory state that chooses reset from u_n ,
3. no play from (v_{2N}, m_0) consistent with σ comes back to v_{2N} .

Indeed:

1. Assume towards contradiction that this is not the case, then there exists n such that Eve resets from u_n with both memory states; Adam can loop around this u_n , contradicting that σ ensures to reach F .
2. Assume towards contradiction that there are at least $N + 1$ vertices u_n from which Eve increments from u_n with both memory states; Adam can force $N + 1$ increments without a reset, contradicting that σ ensures $B(N)$.
3. For m and m' two memory states, we say that $m < m'$ if there exists a play from (v_{2N}, m') consistent with σ which reaches (v_{2N}, m) . Since σ ensures to reach F , the graph induced by $<$ is acyclic. We can take as initial memory state from v_{2N} the smallest memory state which is smaller or equal to m_0 .

We fix the strategy of Adam which skips if, and only if, both memory states of σ choose to increment. Consider the play from v_{2N} consistent with σ and this strategy of Adam. This means that for all vertices u_n that are reached, there is one memory state that resets, and one that increments. Since σ ensures $B(N)$ and there are at most N positions skipped, at some point Eve chooses to reset. From there two scenarios are possible:

- Either Eve keeps resetting until she reaches v_{2N} , contradicting that σ is normalized,
- Or she starts incrementing again, which means that she uses the same memory state than she did before the reset, implying that there is a loop, contradicting that σ ensures to reach F .

4.2 General lower bound

We now push the example above further.

A first approach is to modify G_1 by increasing the length of the resets, going ℓ steps to the left rather than only 2. However, this does not give a better lower bound: there exists a 3 memory states strategy in this modified game that ensures twice the value, following the same ideas as presented above.

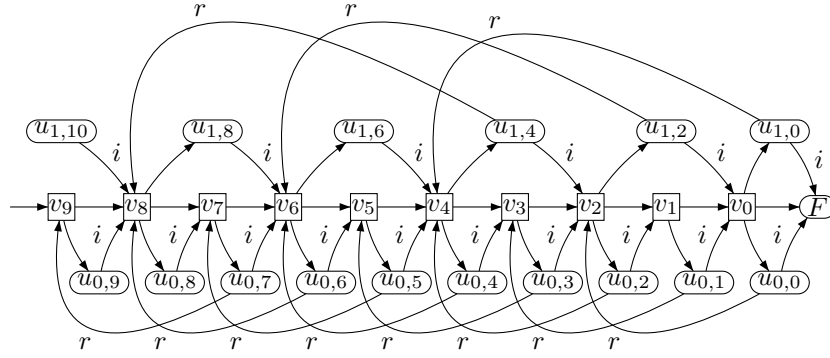


Fig. 4. The game with two levels.

We construct $G_{K,N}$, a nesting of the game G_1 with K levels. Unlike G_1 , it is finite, as we only keep a long enough “suffix”. In figure 4, we represented the interaction between two levels. Roughly speaking, the two levels are independent, so we play both games at the same time. Those two games use different timeline. For instance, in Figure 4, the bottom level is based on $(+1, -2)$ (an increment goes one step to the right, a reset two steps to the left), and the top level is based on $(+2, -4)$. This difference in timeline ensures that a strategy for Eve needs to take care somehow independently of each level, ensuring that the number of memory states depends on the number of levels.

To give the formal definition of $G_{K,N}$, we need two functions, $d(K, N) = (N + 1)^{K-1}$ and

$$n(K + 1, N) = \begin{cases} 2N & \text{if } K = 0, \\ (N + 1)^{K+1} + (N + 1) \cdot n(K, N) & \text{otherwise.} \end{cases}$$

We now define $G_{K,N}$.

$$V = \begin{cases} V_E = \{u_{p,n} \mid p \in \{1, \dots, K\}, n \leq n(K, N)\} \\ V_A = \{v_n \mid n \leq n(K, N)\} \end{cases}$$

$$E = \begin{cases} \{v_{n+1} \rightarrow v_n \mid n\} \\ \cup \{v_n \rightarrow u_{p,n} \mid p, n\} \\ \cup \{u_{p,n+d(p,N)} \xrightarrow{i} v_n \mid p, n\} \\ \cup \{u_{p,0} \xrightarrow{i} F \mid p\} \\ \cup \{u_{p,n} \xrightarrow{r} v_{n+(p+1) \cdot d(p,N)} \mid p, n\} \end{cases}$$

Observe that $G_{1,N}$ is the “suffix” of length $n(1, N)$ of G_1 , for all N .

Theorem 4. *In $G_{K,N}$:*

- Eve has a 3^K memory states strategy ensuring $B(K(K + 3))$ Until F ,
- No $K + 1$ memory states strategy ensures $B(N)$ Until F from $v_{n(K,N)}$.

We first construct a strategy with 3^K memory states ensuring $B(K(K + 3))$ Until F . To this end, we construct for the p^{th} level a strategy with 3 memory states ensuring $B(2(p + 1))$ Until F , using the same ideas as for G_1 , colouring every $(p + 1) \cdot d(p, N)$ vertices. Now we construct the general strategy by playing independently in each copy, except that when a reset is taken, all memory structures update to the (initial) memory state i . This way, it ensures that it always resets to the right of the previous reset, if any. It uses 3^K memory states, and ensures $B(\sum_{p=1}^K 2 \cdot (p + 1))$ Until F , *i.e.* $B(K(K + 3))$ Until F .

We now show that $K + 1$ memory states is not enough. We proceed by induction on K . The case $K = 1$ follows from Theorem 3.

Consider a strategy ensuring $B(N)$ Until F from $v_{n(K+1,N)}$ in $G_{K+1,N}$, for some N , using the memory structure $\mathcal{M} = (M, \mu, m_0)$. We will prove that it has at least $K + 2$ memory states. To this end, we will show that it implies a strategy ensuring $B(N)$ Until F in $G_{K,N}$, which uses one less memory state. The induction hypothesis will conclude.

We first argue that without loss of generality we can assume that no play from $(v_{n(K+1,N)}, m_0)$ consistent with σ comes back to $v_{n(K+1,N)}$. (The proof is the same as for G_1 .) For m and m' two memory states, we say that $m < m'$ if there exists a play from $(v_{n(K+1,N)}, m')$ consistent with σ which reaches $(v_{n(K+1,N)}, m)$. Since σ ensures to reach F , the graph induced by $<$ is acyclic. We can take as initial memory state from $v_{n(K+1,N)}$ the smallest memory state which is smaller or equal to m_0 .

We now argue that there exists $n \leq n(K+1, N)$ and a play from (v_n, m) (for some memory state $m \in M$) to $u_{K+1, n-n(K, N)}$ consistent with σ , which does not use the topmost level (level $K+1$), and such that from there σ chooses to reset.

Assume towards contradiction that this is not the case. Consider the following strategy of Adam, from $v_{n(K+1, N)}$. It alternates ($N+1$ times) between skipping for $n(K, N)$ steps and going to the topmost level. By assumption, σ chooses to increment. This implies $N+1$ increments without resets, contradicting that σ ensures $B(N)$.

Let v_n given by the above property. For every n' such that $n - n(K, N) \leq n' \leq n$ and $p \leq K$, for every vertex $v_{n'}$ and $u_{p, n'}$, there exists a memory state that leads to $u_{K+1, n-n(K, N)}$ such that from there σ chooses to reset. Up to renaming, we can assume that it is always the same memory state, denoted m .

Consider the game obtained by restricting to the first $n(K, N)$ moves from v_n and excluding the topmost level; it is equal to the game $G_{K, N}$ from $v_{n(K, N)}$ for the condition $B(N)$ Until $v_{n-n(K, N)}$. Observe now that the strategy σ restricted to this game ensures $B(N)$ Until $v_{n-n(K, N)}$. Furthermore, it does not make use of the memory state m ; indeed, the functions have been chosen such that $(K+1) \cdot d(K+1, N) \geq n(K, N)$, so resetting in $u_{K+1, n-n(K, N)}$ leads to the left of v_n . Using the memory state m at any point would allow Adam to force to reach $u_{K+1, n-n(K, N)}$ and reset from there, which would contradict the fact that σ is normalized. This concludes.

5 Existence of a trade-off for thin tree arenas

In this section, we prove that the conjecture holds for the special case of thin tree arenas⁶.

Theorem 5. *There exists two functions $\text{mem} : \mathbb{N}^2 \rightarrow \mathbb{N}$ and $\alpha : \mathbb{N}^4 \rightarrow \mathbb{N}$ such that for all B -parity games with k counters and $d+1$ colors over thin tree arenas of width W with initial vertex v_0 , Eve has a strategy to ensure $B(\text{val}(v_0))^k \cdot \alpha(d, k, W, \text{val}(v_0)) \cap \text{Parity}(\Omega)$, with $W \cdot 3^k \cdot k! \cdot \text{mem}(d, k)$ memory states.*

The functions α and mem are defined as follows.

$$\alpha(d, k, W, N) = \begin{cases} 2N & \text{if } d = 1, \\ \alpha(d-2, k+1, 6W, K \cdot (N+1)^k) & \text{otherwise.} \end{cases}$$

$$\text{mem}(d, k) = \begin{cases} 2 \cdot k! & \text{if } d = 1, \\ 4 \cdot \text{mem}(d-2, k+1) & \text{otherwise.} \end{cases}$$

As an intermediate result, we will prove that the conjecture holds for the special case of word arenas.

⁶ The definitions of word and thin tree arenas are given in Subsection 5.1.

Theorem 6. *There exists two functions $\text{mem} : \mathbb{N}^2 \rightarrow \mathbb{N}$ and $\alpha : \mathbb{N}^4 \rightarrow \mathbb{N}$ such that for all B -parity games over word arenas of width W with initial vertex v_0 , Eve has a strategy to ensure $B(\alpha(d, W, k, \text{val}(v_0))) \cap \text{Parity}(\Omega)$, with $\text{mem}(d, k)$ memory states.*

5.1 Word and thin tree arenas

A (non-labelled binary) *tree* is a subset $T \subseteq \{0, 1\}^*$ which is prefix-closed and non-empty. The elements of T are called *nodes*, and we use the natural terminology: for $n \in \{0, 1\}^*$ and $\ell \in \{0, 1\}$, the node $n \cdot \ell$ is a child of n , and a descendant of n if $\ell \in \{0, 1\}^*$.

A (finite or infinite) branch π is a word in $\{0, 1\}^*$ or $\{0, 1\}^\omega$. We say that π is a branch of the tree T if $\pi \subseteq T$ (or every prefix of π belongs to T when π is infinite) and π is maximal satisfying this property. A tree is called *thin* if it has only countably many branches. For example, the full binary tree $T = \{0, 1\}^*$ has uncountably many branches, therefore it is not thin.

Given a thin tree T , we can associate to each node n a rank, denoted $\text{rank}(n)$, which is a countable ordinal number, satisfying the following properties.

Fact 7 ([5])

1. *If n' is a child of n , then $\text{rank}(n') \leq \text{rank}(n)$.*
2. *The set of nodes having the same rank is either a single node or an infinite branch of T .*

Definition 1. *An arena is:*

- *chronological if there exists a function $r : V \rightarrow \mathbb{N}$ which increases by one on every edge: for all $(v, v') \in E$, $r(v') = r(v) + 1$.*
- *a word arena of width W if it is chronological, and for all $i \in \mathbb{N}$, the set $\{v \in V \mid r(v) = i\}$ has cardinal at most W .*
- *a tree arena of width W if there exists a function $R : V \rightarrow \{0, 1\}^*$ such that*
 1. *for all $n \in \{0, 1\}^*$, the set $\{v \in V \mid R(v) = n\}$ has cardinal at most W .*
 2. *for all $(v, v') \in E$, we have $R(v') = R(v) \cdot \ell$ for some $\ell \in \{0, 1\}$.*

It is a thin tree arena if $R(V)$ is a thin tree.

To avoid a possible confusion: in a (thin) tree arena, “vertices” refers to the arena and “nodes” to $R(V)$, hence if the arena has width W , then a node is a bundle of at most W vertices.

The notions of word and tree arenas naturally appear in the study of automata over infinite words and trees. Indeed, the acceptance games of such automata, which are used to define their semantics, are played on word or tree arenas. Furthermore, the width corresponds to the size of the automaton.

5.2 Existence of a trade-off for word arenas

We prove Theorem 6 by induction on the number of colors in the parity condition. The base case is given by Büchi conditions, and follows from Theorem 1.

Consider a B -parity game G with k counters and $d + 1$ colors over a word arena of width W with initial vertex v_0 . Denote $N = \text{val}(v_0)$.

We examine two cases, depending whether the least important color (*i.e.* the smallest) that appears is odd or even. In both cases we construct an equivalent B -parity game G' using one less color; from the induction hypothesis we obtain a winning strategy using small memory in G' , which we use to construct a winning strategy using small memory in G .

Removing the least important color: the odd case The first case we consider is when the least important color is 1. The technical core of the construction is motivated by the technique used in [18]. Note that if this is the only color, then Eve cannot win and the result is true; we now assume that the color 2 also appears in the arena. Without loss of generality we restrict ourselves to vertices reachable with σ from v_0 .

Consider a vertex v and T_v the tree of plays consistent with σ . The strategy σ ensures the parity condition, so in particular every branch in T_v contains a vertex of color greater than 1. We prune the tree T_v by cutting paths when they first meet a vertex of color greater than 1. Since the arena is finite-branching, so is T_v and by Koenig's Lemma the tree obtained is finite. Thus, to every vertex v , we can associate $S(v)$ a rank such that the strategy σ ensures that all paths from v contain a vertex of color greater than 1 before reaching the rank $S(v)$.

We define by induction an increasing sequence of integers $(S_k)_{k \in \mathbb{N}}$ called slices, such that σ ensures that between two slices, a vertex of color greater than 1 is reached. We first set $S_0 = S(v_0)$ (recall that v_0 is the initial vertex). Assume S_k has been defined, we define S_{k+1} as $\max\{S(v) \mid r(v) = S_k\}$. (Note that this is well-defined since $\{v \mid r(v) = S_k\}$ is finite.)

Now we equip G with a memory structure \mathcal{M} of size 2 which keeps track of the boolean information whether or not a vertex of color greater than 1 has been reached since the last slice.

We equip the arena $G \times \mathcal{M}$ with the colouring function Ω' defined by

$$\Omega'(v, m) = \begin{cases} \Omega(v) & \text{if } \Omega(v) \neq 1, \\ 2 & \text{otherwise.} \end{cases}$$

Remark that Ω' uses one less color than Ω .

Define $L = \{(v, 1) \mid v \in S_k \text{ for some } k \in \mathbb{N}\}$, and equip $G \times \mathcal{M}$ with the condition $B(N) \cap \text{Parity}(\Omega') \cap \text{Safe}(L)$.

Lemma 2.

1. The strategy σ in G induces a strategy σ' in $G \times \mathcal{M}$ that ensures $B(N) \cap \text{Parity}(\Omega') \cap \text{Safe}(L)$.

2. Let σ' be a strategy in $G \times \mathcal{M}$ ensuring $B(N') \cap \text{Parity}(\Omega') \cap \text{Safe}(L)$ with K memory states, then there exists σ a strategy in G that ensures $B(N') \cap \text{Parity}(\Omega)$ with $2K$ memory states.

Proof.

1. The strategy σ' that mimics σ , ignoring the memory structure \mathcal{M} , ensures $B(N) \cap \text{Parity}(\Omega')$.
2. Let σ' be a strategy in $G \times \mathcal{M}$ ensuring $B(N') \cap \text{Parity}(\Omega') \cap \text{Safe}(L)$ using \mathcal{M}' as memory structure. We define σ using $\mathcal{M} \times \mathcal{M}'$ as memory structure, simply by $\sigma(v, (m, m')) = \sigma'((v, m), m')$. Since plays of σ and of σ' are in one-to-one correspondence, σ ensures $B(N') \cap \text{Parity}(\Omega')$. Further, the safety condition satisfied by σ' ensures that infinitely often a vertex of color greater than 1 is seen (specifically, between each consecutive slices), so σ satisfies $\text{Parity}(\Omega)$.

Removing the least important color: the even case The second case we consider is when the least important color is 0.

We explain the intuition for the case of CoBüchi conditions, *i.e* if there are only colors 0 and 1. Let $F = \{v \mid \Omega(v) = 0\}$. Define $X_0 = Y_0 = \emptyset$, and for $i \geq 1$:

$$\begin{cases} X_{i+1} = \mathcal{W}_E(\text{Safe}(F) \text{ WeakUntil } Y_i) \\ Y_{i+1} = \mathcal{W}_E(\text{Reach}(X_{i+1})) \end{cases}$$

The condition $\text{Safe}(F) \text{ WeakUntil } Y_i$ is satisfied by plays that do not visit F before Y_i : they may never reach Y_i , in which case neither F , or they reach Y_i , in which case they did not visit F before that.

We have $\bigcup_i Y_i = \mathcal{W}_E(\text{CoBüchi}(F))$. A winning strategy based on these sets has two aims: in X_i it avoids F (“Safe” mode) and in Y_i it attracts to the next X_i (“Attractor” mode). The key property is that since the arena is a word arena of width W where Eve can bound the counters by N , she only needs to alternate between modes a number of times bounded by a function of N and W . In other words, the sequence $(Y_i)_{i \in \mathbb{N}}$ stabilizes after a number of steps bounded by a function of N and W . A remote variant of this bounded-alternation fact can be found in [14]. Hence the CoBüchi condition can be checked using a new counter and a Büchi condition, as follows.

There are two modes: “Safe” and “Attractor”. The Büchi condition ensures that the “Safe” mode is visited infinitely often. In the “Safe” mode, only vertices of colors 0 are accepted; visiting a vertex of color 1 leads to the “Attractor” mode and increments the new counter. At any time, she can reset the mode to “Safe”. The counter is never reset, so to ensure that it is bounded, Eve must change modes finitely often. Furthermore, the Büchi condition ensures that the final mode is “Safe”, implying that the CoBüchi condition is satisfied.

For the more general case of parity conditions, the same idea is used, but as soon as a vertex of color greater than 1 is visited, then the counter is reset.

Define G' :

$$V' = \begin{cases} V'_E = V_E \times \{A, S\} \cup \overline{V} \\ V'_A = V_A \times \{A, S\} . \end{cases}$$

After each edge followed, Eve is left the choice to set the flag to S . The set of choice vertices is denoted \overline{V} . We define E' and the counter actions.

$$E' = \begin{cases} (v, A) \xrightarrow{c(v, v'), \varepsilon} \overline{v'} & \text{if } (v, v') \in E, \\ (v, S) \xrightarrow{c(v, v'), \varepsilon} (v', S) & \text{if } (v, v') \in E \text{ and } \Omega(v') = 0, \\ (v, S) \xrightarrow{c(v, v'), i} (v', A) & \text{if } (v, v') \in E \text{ and } \Omega(v') = 1, \\ (v, S) \xrightarrow{c(v, v'), r} (v', S) & \text{if } (v, v') \in E \text{ and } \Omega(v') > 1, \\ \overline{v} \xrightarrow{\varepsilon} (v, A) \text{ and } \overline{v} \xrightarrow{\varepsilon} (v, S) \end{cases}$$

Equip the arena G' with the colouring function Ω' defined by

$$\Omega'(v, m) = \begin{cases} 1 & \text{if } m = A, \\ 2 & \text{if } \Omega(v) = 0 \text{ and } m = S, \\ \Omega(v) & \text{otherwise.} \end{cases}$$

We do not color the choice vertices, which does not matter as all plays contain infinitely many non-choice vertices; we could give them the least important color, that is 1. Remark that Ω' uses one less color than Ω , since no vertices have color 0 for Ω' .

Before stating and proving the equivalence between G and G' , we formalise the property mentioned above, that in word arenas Eve does not need to alternate an unbounded number of times between the modes “Safe” and “Attractor”.

Lemma 3. *Let G be a word arena of width W , and a subset F of vertices such that every path in G contains finitely many vertices in F . Define the following sequence of subsets of vertices $X_0 = \emptyset$, and for $i \geq 0$*

$$\begin{cases} X_{2i+1} = \left\{ v \mid \begin{array}{l} \text{all paths from } v \text{ contain no vertices in } F \\ \text{before the first vertex in } X_{2i}, \text{ if any} \end{array} \right\}, \\ X_{2i+2} = \{ v \mid \text{all paths from } v \text{ are finite or lead to } X_{2i+1} \}. \end{cases}$$

We have $X_0 \subseteq X_1 \subseteq X_2 \cdots$, and X_{2W} covers the whole arena.

Proof. We first argue that the following property, denoted (\dagger) , holds: “for all $i \geq 0$, if X_{2i} does not cover the whole arena, then $X_{2i+1} \setminus X_{2i-1}$ contains an infinite path”. (For technical convenience $X_{-1} = \emptyset$.)

Let $v \notin X_{2i}$. We consider G_v where v is the initial vertex, and prune it by removing the vertices from X_{2i-1} , as well as vertices which do not have an infinite path after removing X_{2i-1} ; denote by G'_v the graph obtained. Note that for any $u \notin X_{2i}$, the vertex u belongs to G'_v , so G'_v contains an infinite path. We claim that there exists a vertex v' in G'_v such that all paths from v' contain no

vertices F . Indeed, assume towards contradiction that from every node in G'_v , there exists a path to a vertex in F . Then there exists a path that visits infinitely many vertices in F , contradicting the assumption on G . Any infinite path from v' is included into $X_{2i+1} \setminus X_{2i-1}$, hence the latter contains an infinite path.

We conclude using (†): assume towards contradiction that X_{2W} does not cover the whole arena. Then G contains $W + 1$ pairwise disjoint paths, contradicting that it has width W . \square

Lemma 4.

1. *There exists a strategy σ' in G' that ensures $B(W \cdot (N + 1)^k) \cap \text{Parity}(\Omega')$.*
2. *Let σ' be a strategy in G' ensuring $B(N') \cap \text{Parity}(\Omega')$ with K memory states, then there exists σ a strategy in G that ensures $B(N') \cap \text{Parity}(\Omega)$ with $2K$ memory states.*

Proof.

1. Thanks to Lemma 1, there exists a strategy σ in G ensuring $B(N) \cap \text{Parity}(\Omega)$ using a memory structure \mathcal{M} of size $(N + 1)^k$. We construct a strategy σ' in G' by mimicking σ . We now explain when does σ' chooses to set the flag to value S , *i.e.* sets the “Safe” mode.

We consider the arena $G \times \mathcal{M}$, it is a word arena of width $W \cdot (N + 1)^k$, and restrict it to the moves prescribed by σ , obtaining the word arena G_σ of width $W \cdot (N + 1)^k$. Without loss of generality we restrict G_σ to vertices reachable with σ from the initial vertex $(v_0, 0)$. Consider a vertex v of color 0 or 1, and G_σ^v the word arena obtained by considering v as initial vertex and pruned by cutting paths when they first meet a vertex of color greater than 1. Since the strategy σ ensures that the parity condition is satisfied, every infinite path in G_σ^v contains finitely many vertices of color 1. Relying on Lemma 3 for the word arena G_σ^v and F the set of vertices of color 1, we associate to each vertex v' in G_σ^v a rank, which is a number between 1 and $2W \cdot M$, the minimal i such that $v' \in X_i(G_\sigma^v)$.

Now consider a play consistent with σ , and a suffix of this play starting in a vertex v of color 0 or 1. By definition, from this position on, the rank (with respect to v) is non-increasing until a vertex of color greater than 1 is visited, if any. Furthermore, if the rank is even then no vertices of color 1 are visited, and the rank does not remain forever odd.

The strategy σ' in G' mimics σ , and at any point of a play remembers the first vertex v that has not been followed by a vertex of color greater than 1. As observed above, the rank with respect to v is non-increasing; the strategy σ' switches to the “Safe” mode when the rank goes from even to odd. By definition, the new counter is incremented only when the rank goes from odd to even, which happens at most $W \cdot (N + 1)^k$ times, and it is reset when a vertex of color greater than 1 is visited, so σ' ensure that it remains bounded by $W \cdot (N + 1)^k$.

Also, since σ ensures to bound the counters by N , then so does σ' . For the parity condition, there are two cases. Consider a play consistent with σ' .

Either from some point onwards the only colors seen are 0 and 1 (with respect to Ω), then the new counter is not reset after this point, but it is incremented only when the rank decreases from odd to even, which corresponds to switches of mode from “Safe” to “Attractor”. Since this counter is bounded, the mode stabilizes, which by definition of the ranks imply that the stabilized rank is odd, so the mode is “Safe”, and from there on only vertices of color 0 (with respect to Ω) are visited, hence $\text{Parity}(\Omega')$ is satisfied. Or infinitely many vertices of color greater than 1 are seen (with respect to Ω), but since they coincide for Ω and Ω' , the condition $\text{Parity}(\Omega')$ is satisfied. It follows that σ' ensures $B(W \cdot (N + 1)^k) \cap \text{Parity}(\Omega')$.

2. Let σ' be a strategy in G' ensuring $B(N') \cap \text{Parity}(\Omega')$ using \mathcal{M}' as memory structure of size K . We construct σ that mimics σ' ; to this end, we need a memory structure which simulates both \mathcal{M}' and the boolean flag, of size $2K$. By definition, plays of σ and plays of σ' are in one-to-one correspondence, so σ ensures $B(N')$. For the parity condition, there are two cases. Consider a play consistent with σ' . Either from some point onwards the only colors seen are 0 and 1 (with respect to Ω), then the new counter is not reset after this point, but it is incremented each time the mode switches from “Safe” to “Attractor”; since this counter is bounded, the mode stabilizes, and since the play in G' satisfies $\text{Parity}(\Omega')$, the stabilized mode is “Safe”, implying that from there on only vertices of color 0 (with respect to Ω) are visited, hence satisfy $\text{Parity}(\Omega)$. Or infinitely many vertices of color greater than 1 are seen (with respect to Ω), but since they coincide for Ω and Ω' , the condition $\text{Parity}(\Omega)$ is satisfied.

□

5.3 Extending to thin tree arenas

In this subsection, we extend the results for word arenas to thin tree arenas, proving Theorem 5.

Consider a B -parity game G with k counters and $d + 1$ colors over a thin tree arena of width W with initial vertex v_0 . Define $N = \text{val}(v_0)$. Let $R : V \rightarrow \{0, 1\}^*$ witnessing that G is a thin tree arena. We rely on the decomposition of the thin tree $R(V)$ to locally replace σ by strategies using small memory given by Theorem 6.

It follows from Fact 7 that along a play, the rank is non-increasing and decreases only finitely many times. Since the parity condition is prefix-independent, if for each rank Eve plays a strategy ensuring the parity condition, then the resulting strategy ensures the parity condition; however, a closer attention to the counters is required.

We summarize counter actions as follows: let $w \in (\{\varepsilon, i, r\}^k)^*$, its summary $\text{sum}(w) \in \{\varepsilon, i, r\}^k$ is, for each counter, r if the counter is reset in w , i if the counter is incremented by not reset in w , and ε otherwise.

Fact 8 Consider $w = w_1 w_2 \cdots w_n w_\infty$, where $w_1, \dots, w_n \in (\{\varepsilon, i, r\}^k)^*$ and $w_\infty \in (\{\varepsilon, i, r\}^k)^\omega$. Denote $u = \text{sum}(w_1) \text{sum}(w_2) \cdots \text{sum}(w_n) \text{sum}(w_\infty)$, then:

1. $\text{val}(u) \leq \text{val}(w)$,
2. if for all $i \in \{1, \dots, n, \infty\}$ we have $\text{val}(w_i) \leq N'$ and $\text{val}(u) \leq N$, then $\text{val}(w) \leq N \cdot N'$.

We define a B -game G' , where the plays that remain in vertices of the same rank are summarized in one step. It has k counters (as does G). Let $\text{rank}(V)$ denote the set of ranks (subset of the countable ordinals), and \mathbb{S} the set of all strategies in G ensuring $B(N) \cap \text{Parity}(\Omega)$. Define:

$$V = \begin{cases} V_E = \text{rank}(V) \times \{1, \dots, W\} \times \{\varepsilon, i, r\}^k \\ V_A = \text{rank}(V) \times \{1, \dots, W\} \times \mathbb{S} \end{cases}$$

We explain how a couple $(\nu, \ell) \in \text{rank}(V) \times \{1, \dots, W\}$ uniquely determines a vertex in G . First, the rank ν corresponds in R either to a node or to an infinite branch, in the second case we consider the first node in this branch. Second, the component ℓ identifies a vertex in this node.

We say that (ν, ℓ', a) is an outcome of (μ, ℓ, σ) if there exists a play from the vertex corresponding to (μ, ℓ) consistent with σ ending in the vertex corresponding to (ν, ℓ') whose summarized counter actions are a .

$$E = \begin{cases} \{((\nu, \ell, a), (\nu, \ell, \sigma)) \mid \ell, \nu, \sigma\} & \text{counter action: } a \\ \{((\mu, \ell, \sigma), (\nu, \ell', a)) \mid \text{if } (\nu, \ell', a) \text{ is an outcome of } (\mu, \ell, \sigma)\} \end{cases}$$

By definition, G' is well-founded *i.e.* there are no infinite plays in G' .

We first argue that there exists a strategy in G' ensuring $B(N)$. Indeed, it is induced by the strategy σ . A play consistent with this strategy is of the form $u = \text{sum}(w_1)\text{sum}(w_2) \cdots \text{sum}(w_n)\text{sum}(w_\infty)$, where $w = w_1 w_2 \cdots w_n w_\infty$ is a play consistent with σ , following the notations of Fact 8. This fact, item 1., implies that $\text{val}(u) \leq \text{val}(w)$, so $\text{val}(u) \leq N$. Hence the induced strategy ensures $B(N)$.

Lemma 5. *For all B -games with k counters over a well-founded arena with initial vertex v_0 , Eve has a strategy to ensure $B(\text{val}(v_0)^k)$ with $k!$ memory states.*

Thanks to Lemma 5, there exists a strategy σ' in G' ensuring $B(N^k)$ and using the memory structure $\mathcal{M} = (M, m_0, \mu)$, of size $k!$. We construct a strategy in G ensuring $B(N^k \cdot \alpha(d, W, k, N)) \cap \text{Parity}(\Omega)$ using $W \cdot 3^k \cdot k! \cdot \text{mem}(d, k)$ memory states. The memory structure is the product of four memory structures:

- a memory structure that keeps track of the $\ell \in \{1, \dots, W\}$ we used when entering the current rank,
- a memory structure that keeps track of the summary since we entered the current rank, of size 3^k ,
- the memory structure \mathcal{M} ,
- a memory structure of size $\text{mem}(d, k)$, which is used to simulate the strategies obtained from Theorem 6.

Consider $\nu \in \text{rank}(V)$ that corresponds to an infinite branch of $R(V)$. For every $\ell \in \{1, \dots, W\}$ and $m \in M$, the strategy σ' picks a strategy $\sigma'(\nu, \ell, m)$

to play in this infinite branch, ensuring $B(N) \cap \text{Parity}(\Omega)$. When playing this strategy, two scenarios are possible: either the play stays forever in the infinite branch, or an outcome is selected and the game continues from there. Consider the game obtained by starting from the vertex corresponding to (ν, ℓ) and restricted to the infinite branch corresponding to ν , plus a vertex for each outcome. It is played over a word arena of width W with $d + 1$ colors and k counters. Denote by $O(\sigma'(\nu, \ell, m))$ the set of outcomes of this game that are not consistent with $\sigma'(\nu, \ell, m)$. The strategy $\sigma'(\nu, \ell, m)$ ensures $B(N) \cap \text{Parity}(\Omega) \cap \text{Safe}(O(\sigma'(\nu, \ell, m)))$. Thanks to Theorem 6, there exists a strategy $\sigma(\nu, \ell, m)$ ensuring $B(\alpha(d, W, k, N)) \cap \text{Parity}(\Omega) \cap \text{Safe}(O(\sigma'(\nu, \ell, m)))$ using $\text{mem}(d, k)$ memory states.

The strategy σ simulates the strategies $\sigma(\nu, \ell, m)$ in the corresponding parts of the game. Observe that this requires to keep track of both the value ℓ and the summary of the current rank, which is done by the memory structure. We argue that σ ensures $B(N^k \cdot \alpha(d, W, k, N)) \cap \text{Parity}(\Omega)$. A play consistent with this strategy is of the form $w = w_1 w_2 \cdots w_n w_\infty$, where for all $i \in \{1, \dots, n, \infty\}$, we have $\text{val}(w_i) \leq \alpha(d, W, k, N)$. Denote $u = \text{sum}(w_1) \text{sum}(w_2) \cdots \text{sum}(w_n) \text{sum}(w_\infty)$, we have $\text{val}(u) \leq N^k$, since it corresponds to a play consistent with σ' . It follows from Fact 8, item 2., that $\text{val}(w) \leq N^k \cdot \alpha(d, W, k, N)$. Hence the strategy σ ensures $B(N^k \cdot \alpha(d, W, k, N)) \cap \text{Parity}(\Omega)$.

Conclusion

We studied the existence of a trade-off between bounds and memory in games with counters, as conjectured by Colcombet and Loeding. We proved that there is no such trade-off in general, but that under some structural restrictions, as thin tree arenas, the conjecture holds.

We believe that the conjecture holds for all tree arenas, which would imply the decidability of cost MSO over infinite trees. A proof of this result would probably involve advanced combinatorial arguments, and require a deep understanding of the structure of tree arenas.

Acknowledgments

The unbounded number of fruitful discussions we had with Thomas Colcombet and Mikołaj Bojańczyk made this paper possible.

References

1. Achim Blumensath, Thomas Colcombet, Denis Kuperberg, Paweł Parys, and Michael Vanden Boom. Two-way cost automata and cost logics over infinite trees. In *CSL-LICS*, pages 16–26, 2014.
2. Achim Blumensath, Martin Otto, and Mark Weyer. Decidability results for the boundedness problem. *Logical Methods in Computer Science*, 10(3), 2014.

3. Mikołaj Bojańczyk. A bounding quantifier. In *CSL*, pages 41–55, 2004.
4. Mikołaj Bojańczyk and Thomas Colcombet. Bounds in ω -regularity. In *LICS*, pages 285–296, 2006.
5. Mikołaj Bojańczyk, Tomasz Idziaszek, and Michał Skrzypczak. Regular languages of thin trees. In *STACS*, pages 562–573, 2013.
6. Thomas Colcombet. The theory of stabilisation monoids and regular cost functions. In *ICALP (2)*, pages 139–150, 2009.
7. Thomas Colcombet. Fonctions régulières de coût. Habilitation Thesis, 2013.
8. Thomas Colcombet. Regular cost functions, part I: logic and algebra over words. *Logical Methods in Computer Science*, 9(3), 2013.
9. Thomas Colcombet and Christof Löding. The non-deterministic Mostowski hierarchy and distance-parity automata. In *ICALP (2)*, pages 398–409, 2008.
10. Thomas Colcombet and Christof Löding. Regular cost functions over finite trees. In *LICS*, pages 70–79, 2010.
11. Erich Grädel, Wolfgang Thomas, and Thomas Wilke, editors. *Automata, Logics, and Infinite Games*, volume 2500 of *Lecture Notes in Computer Science*. Springer, 2002.
12. Kosaburo Hashiguchi. Improved limitedness theorems on finite automata with distance functions. *Theoretical Computer Science*, 72(1):27–38, 1990.
13. Daniel Kirsten. Distance desert automata and the star height problem. *ITA*, 39(3):455–509, 2005.
14. Orna Kupferman and Moshe Y. Vardi. Weak alternating automata are not that weak. In *5th Israeli Symposium on Theory of Computing and Systems*, pages 147–158. IEEE Computer Society Press, 1997.
15. Hing Leung. Limitedness theorem on finite automata with distance functions: An algebraic proof. *Theoretical Computer Science*, 81(1):137–145, 1991.
16. Michael O. Rabin. Decidability of second-order theories and automata on infinite trees. *Transactions of the AMS*, 141:1–23, 1969.
17. Imre Simon. On semigroups of matrices over the tropical semiring. *ITA*, 28(3-4):277–294, 1994.
18. Michael Vanden Boom. Weak cost monadic logic over infinite trees. In *MFCSS*, pages 580–591, 2011.