Energy Games with Resource-Bounded Environments

Orna Kupferman

School of Engineering and Computer Science, Hebrew University, Jerusalem, Israel
orna@cs.huji.ac.il

Naama Shamash Halevy

School of Engineering and Computer Science, Hebrew University, Jerusalem, Israel
naama.shamashhal@mail.huji.ac.il

Abstract

An *energy game* is played between two players, modeling a resource-bounded system and its environment. The players take turns moving a token along a finite graph. Each edge of the graph is labeled by an integer, describing an update to the energy level of the system that occurs whenever the edge is traversed. The system wins the game if it never runs out of energy. Different applications have led to extensions of the above basic setting. For example, addressing a combination of the energy requirement with behavioral specifications, researchers have studied richer winning conditions, and addressing systems with several bounded resources, researchers have studied games with multi-dimensional energy updates. All extensions, however, assume that the environment has no bounded resources.

We introduce and study *both-bounded energy games* (BBEGs), in which both the system and the environment have multi-dimensional energy bounds. In BBEGs, each edge in the game graph is labeled by two integer vectors, describing updates to the multi-dimensional energy levels of the system and the environment. A system wins a BBEG if it never runs out of energy or if its environment runs out of energy. We show that BBEGs are determined, and that the problem of determining the winner in a given BBEG is decidable iff both the system and the environment have energy vectors of dimension 1. We also study how restrictions on the memory of the system and/or the environment as well as upper bounds on their energy levels influence the winner and the complexity of the problem.

2012 ACM Subject Classification Theory of computation

Keywords and phrases Energy Games

Digital Object Identifier 10.4230/LIPIcs.CONCUR.2022.

1 Introduction

A reactive system interacts with its environment and should behave correctly in all environments. Synthesis of a reactive system thus corresponds to finding a winning strategy in a *two-player game* between the system and the environment. The game is played on a graph whose vertices are partitioned between the players. Starting from some initial vertex, the players move a token along the graph: whenever the token is in a vertex owned by the system, the system decides to which successor to move the token, and similarly for the environment. Together, the players generate a path in the graph. The choices of the players correspond to actions that the system and the environment may take, and so the generated path corresponds to a possible outcome of an interaction between the system and its environment.

The winning condition in the game is induced by the correctness criteria for the system. Early work on synthesis focuses on qualitative criteria, typically described by a temporal logic formula that specifies the allowed interactions [27, 3]. There, the essence of the actions that the system and the environment take is the way they modify the truth assignment to input and output signals. Accordingly, the edges of the graph are labeled by such assignments, and the generated path is an infinite word over the alphabet of assignment. The system wins if this word satisfies the specification. Recent work studies also games with quantitative objectives. There, the essence of the actions that the system and the environment take is the way they modify some quantitative measure, such as a budget or an energy level. Accordingly, the edges of the graph are labeled by updates to the quantitative measure, and the winning condition refers to properties like its limit sum or average [17, 32].

Energy games belong to the second class of games: the two players model a *resource-bounded* system and its environment. Accordingly, each edge of the game graph is labeled by an integer, describing an update to the energy level of the system that occurs whenever the edge is traversed. The system wins the game if it never runs out of energy. The term "energy" may refer to a wide range of applications: an actual energy level, where actions involve consumption or charging of energy; storage, where actions involve storing or freeing disc space; money ones, where actions involve costs and rewards to a budget of some economic entity, and more [11].

Different applications have led to extensions of the above basic setting. For example, addressing a combination of the energy requirement with behavioral specifications, researchers have studied *energy parity games*, whose winning conditions combine quantitative and qualitative conditions [9, 2]. Then, addressing systems with several bounded resources, researchers have studied *generalized energy games*, in which the system player has a multi-dimensional energy level, the updates along the edges are vectors of integers, and the system wins if it does not run out of energy in any of its resources.

Two main questions regarding energy games have been studied. The first, called the *unknown initial-credit problem*, is the problem of deciding the existence of an initial energy level that is sufficient for the system to win the game. The second, called the *given initial-credit problem*, is the problem of deciding whether a given initial energy level is sufficient for the system to win. It is shown in [6, 8] that *memoryless strategies*, namely strategies that decide how to direct the token based on its current location, are sufficient to win energy games, and that consequently, both the unknown and the given initial-credit problems are decidable in NP∩coNP. For multi-dimensional energy games, the unknown initial-credit problem is coNP-complete [10], whereas the given initial-credit problem (a.k.a. *Z-reachability* VASS game) is 2EXPTIME-complete [7, 12, 19].

We introduce and study *both-bounded energy games* (BBEGs), in which both the system and the environment have (multi-dimensional) energy bounds. In BBEGs, each edge in the game graph is labeled by two integer vectors, describing updates to the multi-dimensional energy levels of the system and the environment. A system wins a BBEG if it never runs out of energy or if its environment runs out of energy.

Bounded environments are of interest in several paradigms in computer science. For example, in cryptography, one studies the security of a given cryptosystem with respect to attackers with bounded (typically polynomial) computational power [24]. In the analysis of on-line algorithms, one sometimes cares for the competitive ratio of a given on-line algorithm with respect to requests issued by a bounded adversary [5]. Likewise, studying bounded rationality in games, bounds are placed on the power of the players. As shown in [26], such bounds affect the kind of equilibria one gets, and gives in fact a way of getting around some of the problematic cases of equilibria, (e.g., in the Prisoner's Dilemma [28]). Even closer to the work here is the extension of *bounded synthesis* [29] to settings where both the system and the environment have bounds on their size [21]. In addition to better modeling the studied setting, the bounds are sometimes used in order to obtain decidability or better complexity, and they can also serve in heuristics, as in SAT-based algorithms for bounded synthesis [13]. Finally, a setting in which the system and the environment have similar properties (in particular, both are bounded) enjoys *duality* between the players. Adding budget constraints to the environment makes the players in energy games dual up to the player that moves first and the definition of who wins when the game continues forever. From a practical point of view, in many of the scenarios modeled by energy games, the environment is another system, hence with its own bounds. This includes, for example, a robot that interacts with another robot, both having bounded batteries, or a consumer that interacts with a company, both having bounded budgets.

We show that BBEGs are determined, and that the problem of determining the winner in a given BBEG is decidable iff both the system and the environment have energy vectors of dimension 1. This is both bad news, as traditional energy games are decidable for all dimensions [7], and good news, as adding an (unbounded) energy level to the environment causes even the setting with energy vectors of dimension 1 to include two unbounded components, as in two-counter machines [25]. In order to show decidability, we relate the energy level of the environment with the value of a counter in *one-counter energy games* [1], which augment energy games with a counter. Once, however, the system or the environment has an energy vector of dimension 2, we can use the energy level of the other player to store the sum of the counters, which enable us to simulate a two-counter machine by a BBEG in which the dimension of the energy vector of one of the players is strictly bigger than 1.

We continue and study how restrictions on the memory of the system and/or the environment influence the winner and the complexity of the problem. We show that unlike the case of energy games, where memoryless strategies suffice [6, 8], here the situation is more complicated, and is also not symmetric: while infinite memory may be needed for the system, finite-memory strategies are sufficient for the environment. Essentially, this follows from the different winning criteria for the system and the environment, in particular the fact that wins of the environment happen in finite prefixes of the interaction. The memory required for the environment, however, cannot be a-priory bounded. We study the problem of deciding a winner in BBEGs in which the players are restricted to memoryless or finite-memory strategies. We show that such games are not determined, and that when both players are restricted, the problem is Σ_2^P-complete. Also, when only the system is restricted, the problem is strongly related to reachability problems in *vector addition systems with states* (VASS) [18], is decidable, and is in PSPACE for BBEGs in which both the system and the environment have energy vectors of dimension 1.

Finally, we consider settings in which there is an upper bound on the capacity of the bounded resources. Such bounds exist in resources like batteries or disc space. In standard energy games, researchers have extensively studied settings in which the energy level of the system does not exceed a given maximum capacity [6, 15]. This includes both a semantics in which an overflow leads to losing the game and a semantics in which an overflow is truncated. We study this setting in BBEGs, in particular the problem of determining the winner in a BBEG with energy bounds for one of the players. We show that the problem is reducible to deciding standard multi-dimensional energy games, and is thus decidable.

2 Preliminaries

Both-bounded energy game. A *both-bounded energy game* (*BBEG*, for short) is a game played by two players, Player 1 and Player 2, on a weighted game graph. Each of the players has an energy vector, and the edges of the graph are labeled with updates to those vectors, applied when the edge is traversed. The vertices of the graph are partitioned into positions that are owned by Player 1 and positions that are owned by Player 2. The game proceeds as follows. A token is placed on the initial position of the game graph. The players move the token along the graph in rounds. In each round, the player that owns the position the token is placed on chooses an edge from this position, and moves the token along it. Each of the players has an initial energy vector, which is updated according to the updates along the edges. The goal of Player 1 is not to run out of energy. The goal of Player 2 is to make Player 1 run out of energy, without running out of energy herself.

Formally, a BBEG is a tuple $G = \langle S_1, S_2, s_{init}, E, d_1, d_2, x_0^1, x_0^2, \tau\rangle$, where S_1 and S_2 are disjoint finite sets of positions, owned by Player 1 and Player 2, respectively. We use S to denote $S_1 \cup S_2$. Position $s_{init} \in S$ is the initial position; $E \subseteq S \times S$ is a set of edges; for $j \in \{1, 2\}$, we have that $d_j \geq 1$ is the *dimension* of Player j and $x_0^j \in \mathbb{N}^{d_j}$ is the *initial energy vector* of Player j. Finally, $\tau : E \to \mathbb{Z}^{d_1} \times \mathbb{Z}^{d_2}$ is a cost function. Traversing an edge e with $\tau(e) = (x_1, x_2)$, updates to the energy vectors of Player 1 and Player 2 by x_1 and x_2, respectively. We use $\tau(e)[1]$ and $\tau(e)[2]$ to denote x_1 and x_2, respectively. We consider non-blocking games, i.e., for every position $s \in S$, there is at least one edge leaving s, thus $\langle s, s'\rangle \in E$, for some $s' \in S$. We call a BBEG with dimensions d_1 for Player 1 and d_2 for Player 2 a (d_1, d_2)-*BBEG*.

For an integer $n \geq 1$, we denote by $[n]$ the set $\{1, ..., n\}$. For a vector u in $\mathbb{Z}^n$ and $i \in [n]$, we denote by $u[i]$ the i-th component of u. We define the *size* of G to be the size required for storing the cost function τ, that is $|G| = |E| \cdot (d_1 + d_2) \cdot \log(m)$, where m is the largest integer appearing in some energy update vector. Note that since G is non-blocking, the definition takes the position space into account. Note also the definition assumes that the updates are given in binary.

Given a BBEG G, we define a *run* in G to be an infinite sequence $r = s_1, s_2, ... \in S^\omega$ such that $s_1 = s_{init}$ and $\langle s_i, s_{i+1}\rangle \in E$ for all $i \geq 1$. For a run $r = s_1, s_2...$ and $n \geq 0$, we denote by r_n the prefix of r up to its n-th position. That is, $r_n = s_1, s_2, ...s_n$. We say that n is the *length* of r_n. For $j \in \{1, 2\}$, we say that a prefix r_n *belongs* to Player j if $s_n \in S_j$. We define the *energy level of Player j up to the n-th position in r* to be $e_j(r_n) = x_0^j + \sum_{i=0}^{n-1} \tau(\langle s_i, s_{i+1}\rangle)[j]$. Note that $e_j(r_n)$ is a vector in $\mathbb{Z}^{d_j}$. For a vector u in $\mathbb{Z}^n$, We use $u \geq 0$ to indicate that $u[i] \geq 0$ for all $i \in [n]$, and, dually, use $u < 0$ to indicate that $u[i] < 0$ for some $i \in [n]$.

We say that a sequence $c \in S^* + S^\omega$ is a *computation* in G if one of the following holds:

1. c is an infinite *run* in G, and for every $n \geq 1$, we have that $e_1(c_n) \geq 0$ and $e_2(c_n) \geq 0$.
2. There is $n \geq 1$ such that c is a finite prefix of length n of a run in G, $e_1(c) < 0$ or $e_2(c) < 0$, and for every $m < n$, it holds that $e_1(c_m) \geq 0$ and $e_2(c_m) \geq 0$.

We denote by *comp*(G) the set of computations in G. For a finite computation $c \in$ *comp*(G) of length $m \in N$ and $0 \leq n \leq m$, we denote by c_n the prefix of c up to its n-th position. We denote by *comp*(G) the set of computations in G, by *pref*(G) the set of prefixes of *comp*(G), and by *pref*$_j(G)$, for $j \in \{1, 2\}$, the set of prefixes that belong to Player j.

Strategies. A strategy for Player j is a function $\gamma_j : pref_j(G) \to S$, such that for all $p \cdot s \in pref_j(G)$ with $p \in S^*$ and $s \in S_j$, we have that $\langle s, \gamma_j(p \cdot s)\rangle \in E$. That is, a strategy for Player j maps each prefix $p \cdot s$ with $s \in S_j$ to a position that has an incoming edge from s. We say that a computation $c = s_1, s_2, ... \in$ *comp*(G) is *consistent* with a strategy γ_j for Player j, if for every $i \geq 1$ such that $c_i \in pref_j(G)$, it holds that $s_{i+1} = \gamma_j(c_i)$. Given two strategies γ_1 for Player 1 and γ_2 for Player 2, we define the *outcome* of γ_1 and γ_2, denoted *outcome*(γ_1, γ_2), to be the single computation that is consistent with both γ_1 and γ_2. Note that indeed there is exactly one such computation. Note also that since the domain of a strategy may be infinite, a general strategy may require infinite memory.

Winning Conditions. A computation c is *winning for Player 1* if one of the following holds:

1. Player 1 never runs out of energy. That is, c is infinite. Note that if c is infinite, then for all $n \geq 1$, we have that $e_1(c_n) \geq 0$. Thus, Player 1 manages to keep her energy level non-negative during the infinite computation c.
2. Player 2 runs out of energy before Player 1. That is, there is $n \geq 1$ such that $c = s_1, s_2, ..., s_n$, it holds that $e_2(c) < 0$, and either $e_1(c) \geq 0$ or $s_{n-1} \in S_2$. We can think of the energy updates along the edges as if traversing an edge leaving position in S_j, for $j \in \{1, 2\}$, updates first the energy vector of Player j, and then updates the energy vector of the other player. Thus, Player 2 runs out of energy before Player 1 if the energy level of Player 2 becomes negative while the energy level of Player 1 is non-negative, or both energy levels become negative together, but as a consequence of a move made by Player 2.

If none of the two conditions above hold, then c is *winning for Player 2*. In other words, c is winning for Player 2 if Player 1 runs out of energy before Player 2. That is, there is $n \geq 1$ such that $c = s_1, s_2, ..., s_n$, $e_1(c) < 0$, and either $e_2(c) \geq 0$ or $s_{n-1} \in S_1$. Note that while a computation winning for Player 2 is always finite, a computation winning for Player 1 may be either finite or infinite.

A strategy γ_1 is winning for Player 1 if for every strategy γ_2 for Player 2, the computation *outcome*(γ_1, γ_2) is winning for Player 1. Dually, a strategy γ_2 is winning for Player 2 if for every strategy γ_1 for Player 1, the computation *outcome*(γ_1, γ_2) is winning for Player 2. For $j \in \{1, 2\}$, we say that Player j wins in G if she has a winning strategy.

▶ **Example 1.** Consider the BBEG G in Figure 1. Drawing BBEGs, we describe positions in S_1 and S_2 by circles and squares, respectively. The initial position is marked by an incoming arrow from the initial energy vectors, and edges are labeled with the energy vectors assigned by the cost function. For example, in G both players start with energy level 0, and the transition from s_2 to s_3 does not change the energy level of Player 1, and decreases by 1 the energy level of Player 2.

We show that Player 1 wins in G. Indeed, if Player 2 always takes the loop on s_1, then Player 1 wins, as the outcome is an infinite computation in which the energy level of Player 1 is always non-negative. Otherwise, Player 2 loops n times in s_1, for some $n \in \mathbb{N}$, and then moves to s_2. At this point, the energy level of both players is n. Player 1 can then take the loop on s_2 exactly n times, setting both energy levels back to 0. At this point, Player 1 can take the transition to s_3 and make Player 2 lose, since her energy level drops below 0. ◀

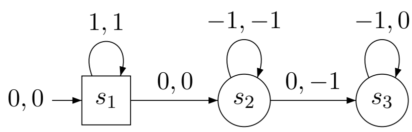

Figure 1 The game graph G.

Determinacy. A game is *determined* if in all instances G of the game, either Player 1 wins in G, or Player 2 wins in G. Since the set of computations that are winning for Player 1 is closed, we have from [23] that BBEGs are determined. Indeed, if Player 2 does not have a winning strategy, one can construct a strategy for Player 1 such that every finite-computation consistent with it is not losing for Player 1. Since the set of winning computations for Player 1 is closed (in the topological sense), this strategy must be winning.

▶ Remark 2. [**Adding structural assumptions**] For simplicity of describing computations and strategies, we define BBEGs without parallel edges. For convenience, we sometimes describe BBEGs with parallel edges (that is, the graph G may have several, yet finitely many, edges between two positions, each with a different update). We sometimes also assume that each transition in the BBEG updates the energy to one player only, or assume that the costs on the transitions are all in $\{-1, 0, 1\}$. As explained in Appendix A.1, these assumptions do not restrict the generality of our results. ◀

3 Deciding BBEGs

In this section we study the problem of determining the winner in a given BBEG. We give a clear border for their decidability: determining the winner in $(1, 1)$-BBEGs is decidable, yet determining the winner in (d_1, d_2)-BBEGs is undecidable when $d_1 \geq 1$ and $d_2 \geq 2$ or when $d_2 \geq 1$ and $d_1 \geq 2$.

▶ **Theorem 3.** *The problem of determining the winner in* $(1, 1)$*-BBEGs is decidable.*

Proof. We reduce $(1, 1)$-BBEGs to *one-counter energy games* of dimension 1.

A one-counter energy game of dimension 1 is $A = \langle Q_1, Q_2, \delta, \delta_0 \rangle$, where Q_1 and Q_2 are distinct finite sets of positions owned by Player 1 and Player 2, respectively. We use Q to denote $Q_1 \cup Q_2$. The game A has two transition relations, $\delta \subseteq Q \times \{-1, 0, 1\}^2 \times Q$ and $\delta_0 \subseteq Q \times \{-1, 0, 1\} \times \{0, 1\} \times Q$. A configuration in A is a triple $\langle p, e, c \rangle \in Q \times \mathbb{Z} \times \mathbb{N}$, which describes a position, energy level, and a counter value. The transition relations δ and δ_0 define a relation between successor configurations as follows. A configuration $\langle p', e', c' \rangle$ is successor of configuration $\langle p, e, c \rangle$ iff one of the following holds:

1. $c' \geq 0$ and $\langle p, e' - e, c' - c, p' \rangle \in \delta$.
2. $c = 0$ and $\langle p, e' - e, c', p' \rangle \in \delta_0$.

Note that δ_0-transitions can be taken only when the value of the counter is 0, and they can not decrease the value. Also, δ-transitions can be taken whenever they do not reduce the value of the counter below 0.

The game proceeds as follows. At each round, the player who owns the current position chooses a transition, and the new configuration is a successor of the current one. Note that during the game, the value of the counter is always non-negative. The game terminates and Player 2 wins if a configuration $\langle p, e, r \rangle$ with $e < 0$ is reached. Player 1 wins every infinite game. It is shown in [1], that given an initial configuration $c = \langle p, e, r \rangle$, determining the winner in A from c is decidable.

Given a $(1, 1)$-BBEG G, we construct a one-counter energy game A with dimension 1, such that Player 1 wins in G iff Player 1 wins in A. Since determining the winner of one-counter energy games with dimension 1 is decidable [1], we get decidability for $(1, 1)$-BBEGs.

Let $G = \langle S_1, S_2, s_{init}, E, 1, 1, x_0^1, x_0^2, \tau \rangle$. For simplicity, we assume that each transition in G updates the energy level of only one player, and that the costs on the transitions are numbers in $\{-1, 0, 1\}$ (see Remark 2).

We define $A = \langle Q_1, Q_2, \delta, \delta_0 \rangle$ so that the energy level in A represents the energy of Player 1 in G, and the counter value represents the energy level of Player 2 in G. For that, we define $Q_1 = S_1 \cup \{sink\}$, and $Q_2 = S_2$. Now, let $Q'_1 = \{s \in S_1 :$ there is $s' \in S$ such that $\langle s, s' \rangle \in E$ and $\tau(\langle s, s' \rangle) = (0, -1)\}$, and $Q'_2 = \{s \in S_2 :$ for all $s' \in S$ such that $\langle s, s' \rangle \in E$, we have that $\tau(\langle s, s' \rangle) = (0, -1)\}$. That is, Q'_1 is the set of positions from which Player 1 can decrease the energy level of Player 2, and Q'_2 is the set of positions from which Player 2 must decrease her own energy level.

We define $\delta = \{\langle s, \tau(\langle s, s' \rangle)[1], \tau(\langle s, s' \rangle)[2], s' \rangle : \langle s, s' \rangle \in E\} \cup \{\langle sink, 0, 0, sink \rangle\}$ and $\delta_0 = (Q'_1 \cup Q'_2) \times \{0\}^2 \times \{sink\}$. In Appendix A.2, we prove that Player 1 wins in A from

$\langle s_{init}, x_0^1, x_0^2 \rangle$ iff Player 1 wins in G. Essentially, this follows from the fact we let Player 1 reach a winning sink whenever she can make Player 2 lose her energy, and we force Player 2 to the sink whenever she runs out of energy. ◀

We now show that the positive result in Theorem 3 is tight.

▶ **Theorem 4.** *The problem of determining the winner of BBEGs is undecidable. Undecidability holds already for* $(1,2)$*-BBEGs or* $(2,1)$*-BBEGs, and when the weights on the transitions are all vectors over* $\{-1,0,1\}$.

Proof. We start with $(1,2)$-BBEGs, and show a reduction from *the halting problem of two-counter machines* to our problem. A two-counter machine is a sequence $M = (l_1, ..., l_n)$ of commands involving two counters x and y. We refer to $\{1, ..., n\}$ as the *locations* of the machine. The command l_n is the halting command, and each command l_i, for $i < n$, is of one of the following forms, where $c \in \{x, y\}$ is a counter and $1 \leq i, j \leq n$ are locations:

- INC : $c \coloneqq c + 1$
- GOTO : goto i
- TEST-DEC : if $c = 0$ then goto i else ($c \coloneqq c - 1$; goto j)

For the TEST-DEC command, we refer to i as the *positive successor* of the command, and refer to j as the *negative successor* of the command. Since we always check whether $c = 0$ before decreasing it, the counters never have negative values. For a two-counter machine M, the question whether M halts is known to be undecidable [25].

Given a machine M, we construct a game G such that M halts iff Player 2 wins in G. The reduction idea is as follows: the dimension of Player 1 is one, and the dimension of Player 2 is two. During a computation in G, the energy level of Player 1 is $x + y$, and the energy level of Player 2 is (x, y), where x and y are the two counters of M. If M never halts, then both energy levels remain non-negative during the infinite computation, and thus Player 1 wins. If M reaches the halting command, then we reach a losing position for Player 1, so Player 2 wins. We now describe the reduction in detail. Given $M = (l_1, ..., l_n)$, we construct $G = \langle S_1, S_2, s_{init}, E, 1, 2, 0, 0^2, \tau \rangle$, such that $S_2 = \{1, ..., n\}$, and $S_1 = L_{td} \times \{1, 2\}$, where $L_{td} \subseteq \{1, .., n\}$ is the set of all locations of the TEST-DEC commands in M. The initial energy levels are 0 for Player 1 and $(0, 0)$ for Player 2, reflecting the fact that the counters are initiated to 0. Now, we introduce a gadget for each command l_i as follows.

1. if l_i is $x \coloneqq x + 1$, then G includes an edge $e = \langle i, i+1 \rangle$ with $\tau(e) = (1, (1, 0))$.
2. if l_i is $y \coloneqq y + 1$, then G includes an edge $e = \langle i, i+1 \rangle$ with $\tau(e) = (1, (0, 1))$.
3. if l_i is goto j, then G includes an edge $e = \langle i, j \rangle$ with $\tau(e) = (0, (0, 0))$.
4. if l_i is if $x = 0$ then goto j else ($x \coloneqq x - 1$; goto k), then G includes the gadget described in Figure 2 (left).
5. if l_i is if $y = 0$ then goto j else ($y \coloneqq y - 1$; goto k), then G includes the gadget described in Figure 2 (right).
6. for the halting command, l_n, the game G includes an edge $e = \langle n, n \rangle$ with $\tau(e) = (-1, (0, 0))$.

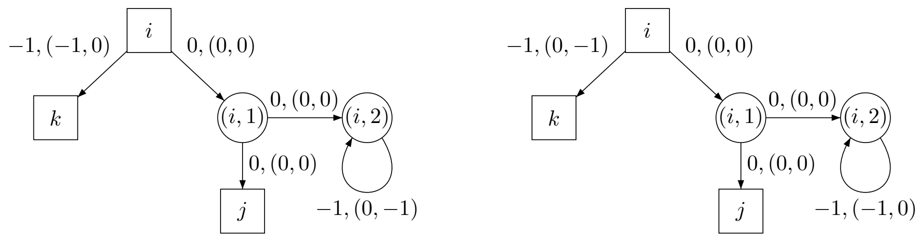

Figure 2 The gadgets for x-TEST-DEC (left) and y-TEST-DEC (right) commands.

These transitions are the only transitions G has. We also define s_{init} to be 1; that is, the state corresponding to l_1.

In Appendix A.3 we prove that the reduction is correct, thus M halts iff Player 2 wins in G. For this, we first prove that if a player has a winning strategy, then she also has a winning strategy that follows the instructions. That is, at every step of the computation, the best move for the current player is the one that leads to the state corresponding to the next command to be read according to M. Then, we show that the outcome of strategies that follow the instruction, is such that the energy level of Player 1 stores $x + y$, and the energy level of Player 2 stores (x, y). Then, as the value of the counters is always non-negative and the position that corresponds to the halting command is losing for Player 1, we get that M halts iff Player 2 wins in G.

The challenging part in the construction and its proof is to construct the TEST-DEC gadgets so that a strategy that follows the instruction is indeed dominating, and that the energy levels indeed maintain the values of the the counters and their sum. Note that excluding positions induced by the TEST-DEC gadgets, all positions in G belong to Player 2. In order to understand the idea behind the gadget, consider for example the x-TEST-DEC gadget, associated with the command if $x = 0$ then goto j else ($x := x - 1$; goto k). As the energy level of Player 2 is (x, y), taking the transition from position i to position k when $x = 0$ is a losing action for Player 2, as it updates the x-component of her energy level to -1. Thus, when $x = 0$, a dominating strategy for Player 2 takes the transition from position i to position $(i, 1)$. Then, as the energy level of Player 1 is $x + y$, taking the transition from $(i, 1)$ to $(i, 2)$ when $x = 0$ is a loosing action for Player 1. Indeed, after y traversals in the loop in position $(i, 2)$, the energy levels of the players become 0 and $(0, 0)$, causing Player 1 to lose in the next round. Thus, when $x = 0$, a dominating strategy for Player 1 takes the transition from position $(i, 1)$ to position j. In addition, the energy levels of the players does not change when the token moves from position i to j. Similar considerations show that when $x \neq 0$, a dominating strategy for Player 2 takes the transition from position i to position k, which involves an update to the energy levels that corresponds to the decrement of x by 1.

We continue and prove undecidability for $(2, 1)$-BBEGs. We show a similar reduction from the halting problem of two-counter machines. Take $G = \langle S_1, S_2, s_{init}, E, 1, 2, 0, (0, 0), \tau \rangle$ the BBEG used above, and consider the BBEG $G' = \langle S_2, S_1, s_{init}, E, 2, 1, (0, 0), 0, \tau' \rangle$, where $\tau'(\langle s, s' \rangle) = (\tau(\langle s, s' \rangle)[2], \tau(\langle s, s' \rangle)[1])$ for all $\langle s, s' \rangle \in E$, $s \neq n$, and $\tau'(n, n) = ((-1, 0), 0)$. That is, G' obtained from G by switching the dimensions of the players, their initial energy vectors, the updates on the edges and the sets of positions. Consequently, also in G', a dominating strategy for the players is consistent with the commands, it implies that the energy level of Player 1 is (x, y), the energy level of Player 2 is $x + y$, and since the sink n is losing for Player 1, we get that M halts if and only if Player 2 wins in G'. ◀

It follows from Theorem 3 and Theorem 4 that determining the winner of (d_1, d_2)-BBEGs is decidable iff $d_1 = d_2 = 1$. In particular, it is easy to extend Theorem 4 to bigger dimensions, by adding to the energy vectors components whose energy values are not updated during the whole computation.

4 BBEGs with finite-memory strategies

In this section we study BBEGs in which the memory used in the strategies of the players is bounded. Following [13], we consider two types of finite-memory strategies. The first type bounds the number of states of a *transducer* that induces the strategy. The second type is *position-based*, and bounds the number of memory states with which we can refine each position of the BBEG. In particular, a *memoryless* strategy is a position-based strategy in which no refinement is allowed. Below we describe the two types formally.

An *I/O-transducer* is a tuple $\mathcal{M} = \langle I, O, Q, q_0, \delta, L\rangle$, for an input alphabet I, an output alphabet O, a finite set of states Q, an initial state $q_0 \in Q$, a transition function $\delta : Q \times I \to Q$, and a labelling function $L : Q \to O$. We extend the transition function δ to words in I^* in the expected way, thus $\delta^* : Q \times I^* \to Q$ is such that for all $q \in Q$, $p \in I^*$, and $i \in I$, we have that $\delta^*(q, \epsilon) = q$, and $\delta^*(q, p \cdot i) = \delta(\delta^*(q, p), i)$. The transducer $\mathcal{M}$ induces a strategy $\gamma_{\mathcal{M}} : I^* \to O$, where for all $p \in I^*$, we have that $\gamma_{\mathcal{M}}(p) = L(\delta^*(q_0, p))$.

Consider a BBEG $G = \langle S_1, S_2, s_{init}, E, d_1, d_2, x^1_0, x^2_0, \tau\rangle$. Let $S = S_1 \cup S_2$. We say that a strategy γ_j for Player j in G has *finite-memory* if it can be defined by an S/S-transducer (or transducer, when S is clear from the context). The strategy corresponding to $\mathcal{M}$ is defined by $\gamma_j(p) = L(\delta^*(q_0, p))$, for all $p \in \mathit{pref}_j(G)$. We say that an S/S-transducer $\mathcal{M} = \langle S, S, Q, q_0, \delta, L\rangle$ *refines* G, if the states of $\mathcal{M}$ refine the positions of G. Formally, $Q = S \times M$ for some finite set of *memory states* M, $q_0 = \langle s_{init}, m_0\rangle$ for some $m_0 \in M$, and for all $s_1, s_2 \in S$ and $m_1 \in M$, it holds that $\delta(\langle s_1, m_1\rangle, s_2) = \langle s_2, m_2\rangle$ for some $m_2 \in M$. We say that a strategy for Player j is *memoryless*, if it is induced by a transducer that refines G with $|M| = 1$, thus, $Q = S$. Note that one can refer to a memoryless strategy for Player j as a function $\gamma_j : S_j \to S$.

For $m_1, m_2 \geq 1$, we say that Player 1 (m_1, m_2)-*wins* in G, if she has a strategy induced by a transducer with m_1 states, that is winning against all strategies for Player 2 that are induced by a transducer with m_2 states. The definition for Player 2 (m_1, m_2)-winning is similar. All our results on (m_1, m_2)-winning apply also to transducers that refine G (see Remark 15). Note that a general BBEG corresponds to $m_1 = m_2 = \infty$. Of special interest are also settings in which only one of m_1 or m_2 is ∞, corresponding to BBEGs where only one player has a memory bound.

4.1 Properties of BBEGs with finite-memory strategies

In this section we study properties of BBEGs with finite-memory strategies. Recall that in energy games with no resource-bounds on the environment, it is sufficient to consider memoryless strategies. We first show that the situation in BBEGs is more complicated, and is also not symmetric: while infinite memory may be needed for Player 1, finite-memory strategies are sufficient for Player 2. Essentially, this follows from the fact that a win of Player 2 is a *co-safety* property: when Player 2 wins, she does so in a finite computation.

▶ **Theorem 5.** *There is a game G such that Player 1 (∞, ∞)-wins G, but for all $m_1 \geq 1$, Player 2 (m_1, ∞)-wins G. On the other hand, for every BBEG G, if Player 2 (∞, ∞)-wins G, then there is $m_2 \in \mathbb{N}$ such that Player 2 (∞, m_2)-wins G.*

Proof. For the first claim, consider the game G described in Example 1 with initial energy levels 0 for both players. We saw that Player 1 has a (general) winning strategy. On the other hand, for every strategy γ_1 for Player 1 that is based on a transducer with m_1 states, the (finite-memory) strategy γ_2 for Player 2 that loops $m_1 + 1$ times in s_1 and then moves to s_2 is winning for Player 2 (see Appendix A.4 for the full proof). We continue to the second claim. Intuitively, the claim follows from the fact that all the computations in which Player 2 wins are finite. Formally, let G be a BBEG in which Player 2 wins, and let γ_2 be a winning strategy. Consider the unfolding of the game G in which Player 2 plays γ_2. The unfolding is a tree $T_G^{\gamma_2}$ in which each node is a prefix of a computation that is consistent with γ_2. Since Player 2 wins, every such a computation is finite, thus every path in $T_G^{\gamma_2}$ is finite. Since the degree of $T_G^{\gamma_2}$ is bounded, we get that $T_G^{\gamma_2}$ is a finite tree, which induces a finite-memory winning strategy for Player 2. ◀

Since finite-memory strategies are sufficient for Player 2 to win, a natural question is whether there is a "bounded-size property" for Player 2's strategy, in particular whether she can win with a memoryless strategies. Such properties exist in many other settings. For example, in games with a Streett winning condition, only Player 2 can win with a memoryless strategy [30], and, more relevant to our study here, in synthesis of an LTL formula ψ, we know that if there is an infinite system that realizes ψ, then there is also a system with at most $2^{2^{|\psi|}}$ states that does it, and the same for the environment [21, 27, 14]. Thus, (∞, ∞)-realizability coincides with $(\infty, 2^{2^{|\psi|}})$-realizability, $(2^{2^{|\psi|}}, \infty)$-realizability, and $(2^{2^{|\psi|}}, 2^{2^{|\psi|}})$-realizability. As we now show, in the case of BBEGs, no bounded-size property exists.

▶ **Theorem 6.** *There is no computable function $f : BBEGs \to \mathbb{N}$ such that for every BBEG G, we have that Player 2 (∞, ∞)-wins G iff Player 2 $(\infty, f(G))$-wins G.*

Proof. In Section 4.2, we are going to show that the problem of deciding whether Player 2 (∞, m_2)-wins a BBEG G is decidable for all given BBEGs and bounds $m_2 \in \mathbb{N}$. Hence, the existence of a computable function f would lead to decidability of BBEGs of all dimensions, contradicting Theorem 4. ◀

Recall that BBEGs are determined. As finite-state and memoryless strategies need not be sufficient to winning a BBEG, we now study determinancy of BBEGs when both players have bounds on their memory. Formally. we say that a game is *determined under finite-memory strategies* or *determined under memoryless strategies*, if in all instances G of the game, either Player 1 wins in G, or Player 2 wins in G, when the strategies of both players are restricted to finite-memory or memoryless strategies, respectively. Note that since the restriction applies to both players, the two types of determinancy need not imply each other.

▶ **Theorem 7.** *BBEGs are not determined under finite-memory or memoryless strategies.*

Proof. We start with finite-memory strategies. Consider the game G described in Example 1. In Appendix A.4, we show that when both players are restricted to finite-memory strategies, there is no winning player in G.

We continue to memoryless strategies. Consider the $(1, 1)$-BBEG G described in Figure 3. In Appendix A.5, we show that there is no winning strategy in G when both players are restricted to play memoryless strategies. ◀

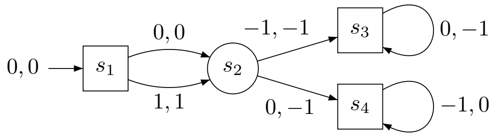

Figure 3 No player has a memoryless winning strategy

4.2 Deciding BBEGs with finite-memory strategies

In this section we study the problem of deciding the winner in a given BBEG in which at least one player is restricted to finite-memory strategies. We show that the problem is decidable for BBEGs of all dimensions. We start with BBEGs with memoryless strategies and show that deciding whether Player 1 has a memoryless strategy that is winning against every memoryless strategy for Player 2 is Σ_2^P-complete. We first prove the following lemma, about deciding the winner given strategies for the players. The proof, in Appendix A.6, is based on the fact that $outcome(\gamma_1, \gamma_2)$ is a simple lasso, and one can determine the winner by analyzing the updates to the energy levels along the prefix and the cycle of the lasso.

▶ **Lemma 8.** *Given a BBEG and two memoryless strategies γ_1 and γ_2 for Player 1 and Player 2, respectively, deciding the winner in outcome(γ_1, γ_2) can be done in polynomial time.*

Lemma 8 suggests that deciding whether Player 1 has a memoryless strategy that is winning against every memoryless strategy for Player 2 can proceed by guessing a Player 1 strategy and challenging it against a guessed Player 2 strategy. Thus, the problem can be solved by a nondeterministic polynomial-time Turing machine with an oracle to a nondeterministic polynomial-time Turing machine. Below we formalize this intuition and provide also a matching lower bound.

▶ **Theorem 9.** *Deciding whether Player 1 has a memoryless strategy that is winning against every memoryless strategy for Player 2 is Σ_2^P-complete.*

Proof. The upper bound follows directly from Lemma 8 (see details in Appendix A.7). For the lower bound, we describe a reduction from QBF_2, the problem of determining the truth of quantified Boolean formulas with two alternations of quantifiers, where the external quantifier is “exists". Let ψ be a Boolean propositional formula over the variables $x_1, ..., x_l, y_1, ..., y_m$, and let $\theta = \exists x_1, ..., x_l \forall y_1, ..., y_m \psi$. Also, let $X = \{x_1, ..., x_l\}, Y = \{y_1, ..., y_m\}, \bar{X} = \{\overline{x_1}, ..., \overline{x_l}\}, \bar{Y} = \{\overline{y_1}, ..., \overline{y_m}\}$, and $Z = X \cup \bar{X} \cup Y \cup \bar{Y}$. By [31], we may assume that ψ is given in 3DNF. That is, $\psi = (z_1^1 \wedge z_1^2 \wedge z_1^3) \vee ... \vee (z_n^1 \wedge z_n^2 \wedge z_n^3)$, where for all $1 \le i \le 3$ and $1 \le j \le n$, we have that $z_j^i \in Z$. For $1 \le j \le n$, we denote the clause $(z_j^1 \wedge z_j^2 \wedge z_j^3)$ by c_j.

Given a formula $\theta = \exists x_1, ..., x_l \forall y_1, ..., y_m \psi$, we construct a $(1, 1)$-BBEG G such that θ is true iff Player 1 wins G with a memoryless strategy. In the game G, we describe the energy levels of the players and updates to the energy levels by bit-vectors in $\{-2, -1, 0, 1, 2, 3\}^n$. Updates to the bit-vectors are done in a bit-wise manner, thus $\langle b_n, b_{n-1}, ..., b_1\rangle + \langle b'_n, b'_{n-1}, ..., b'_1\rangle = \langle b_n + b'_n, b_{n-1} + b'_{n-1}, ..., b_1 + b'_1\rangle$. Our games are defined so that all reachable energy levels are in $\{-2, -1, 0, 1, 2, 3\}^n$. Each bit vector $v = \langle b_n, b_{n-1}, ..., b_1\rangle$ represents a single value in $\mathbb{Z}$, namely $\sum_{i=1}^n b_i \cdot (10)^{i-1}$. For example, the value of $\langle 1, -2, 0, 3\rangle$ is $3 \cdot 1 + 0 \cdot 10 + (-2) \cdot 100 + 1 \cdot 1000 = 803$. We say that v is positive (negative) iff the value it represents is positive (negative), respectively.

The idea behind the reduction is as follows. Each assignment $g : X \cup Y \to \{T, F\}$ induces a bit-vector $v_g = \langle b_n, b_{n-1}, ..., b_1\rangle \in \{0, 1, 2, 3\}^n$, such that for all $1 \leq i \leq n$, the bit b_i indicates how many literals in c_i are satisfied by the assignment g. Note that this number is indeed in $\{0, 1, 2, 3\}$. For example, take $\psi = (x_1 \wedge x_2 \wedge y_1) \vee (x_1 \wedge x_2 \wedge \overline{y_1})$, with the assignment g in which $g(x_1) = g(x_2) = T$, and $g(y_1) = F$. Since g satisfies two literals in c_1 and three literals in c_2, we have that $v_g = \langle 3, 2\rangle$.

The game G consists of two parts: an *assignment* part, and a *check* part. In the assignment part, Player 1 assigns values to the variables in X, and then Player 2 assigns values to the variables in Y. Together, the players generate an assignment $g : X \cup Y \to \{T, F\}$, and the energy level of both players is updated in the same way, so that by the end of this part, it is v_g. Note that the assignment g satisfies ψ iff the vector v_g contains the bit 3; thus there is $1 \leq i \leq n$ with $b_i = 3$. At the check part, we let Player 2 win if v_g does not contain such a bit. We do this by allowing Player 2 to decrease each bit (in the energy level of both players) by 0, 1 or 2. Accordingly, if no bit in v_g is 3, then Player 2 has a strategy so that by the end of this process, the energy level of the players is represented by the bit-vector 0^n, in which case Player 2 can force a win. On the other hand, if some bit in v_g is 3, then for all strategies of Player 2, at least one bit is not 0 at the end of this process. In this case, Player 2 loses.

In Appendix A.7, we describe the two parts in detail and prove the correctness of the reduction. ◀

Note that since under memoryless strategies BBEGs are not determined, Π_2^P-completeness for the dual problem does not follow from Theorem 9. In fact, as we show below, the dual problem is also Σ_2^P-complete. The proof, in Appendix A.8, is similar to the proof of Theorem 9. In particular, for the lower bound, the game we construct here is obtained from the game constructed there by switching the ownership of positions, switching between the cost functions of the players, and by changing the sink to be a winning position for Player 2.

▶ **Theorem 10.** *Deciding whether Player 2 has a memoryless strategy that is winning against every memoryless strategy for Player 1 is Σ_2^P-complete.*

We now show that Σ_2^P-completeness holds also when both players are restricted to finite-state strategies. Note that while the considerations are similar to these in the proof of Theorem 9, the lower bound for the memoryless case implies only a lower bound for the finite-memory case with transducers that refine the game G. There, we can use the reduction from the proof of Theorem 9 as is, with $m_1 = |S_1|$ and $m_2 = |S_2|$. For general finite-state strategies, a transducer with $|S_j|$ states, for $j \in \{1, 2\}$, does not necessarily induce a memoryless strategy for Player j. In the proof of the theorem, in Appendix A.9, we show that for the specific game G described in the reduction in Theorem 9, Player 1 $(|S_1|, |S_2|)$-wins G iff she wins with a memoryless strategy, and similarly for Player 2 and the game described in the reduction in Theorem 10. Hence, the same reduction can be used.

▶ **Theorem 11.** *Given a BBEG G and $m_1, m_2 \in \mathbb{N}$ (given in unary), the problems of deciding whether Player 1 (m_1, m_2)-wins and deciding whether Player 2 (m_1, m_2)-wins in G are Σ_2^P-complete.*

Note that the reductions used in Theorems 9, 10, and 11 generate a $(1, 1)$-BBEG, thus Σ_2^P-hardness holds already for them.

We continue and consider BBEGs in which only Player 1 has a memory bound. We show that the setting is strongly related to *vector addition systems with states* (VASS), defined below.

For $d \geq 1$, a d-VASS is a finite $\mathbb{Z}^d$-labeled directed graph $V = \langle Q, T \rangle$, where Q is a finite set of *states*, and $T \subseteq Q \times \mathbb{Z}^d \times Q$ is a finite set of *transitions*. The set of *configurations* of V is $C = Q \times \mathbb{N}^d$. For a pair of configurations $\langle p_1, v_1 \rangle, \langle p_2, v_2 \rangle \in C$ and $t = \langle p_1, z, p_2 \rangle \in T$ such that $v_2 = v_1 + z$, we write $\langle p_1, v_1 \rangle \to^t \langle p_2, v_2 \rangle$. For $c, c' \in C$ we write $c \to^* c'$ if $c = c'$, or if there is $m \geq 1$ such that $c_0 \to^{t_1} c_2 \to^{t_2} .... \to^{t_m} c_m$, for some $t_1, ..., t_m \in T$ and $c_0, ..., c_m \in C$, with $c_0 = c$ and $c_m = c'$. That is, $c \to^* c'$ indicates that there is a sequence of successive configurations from c to c' in V, and the vector is non-negative in all the configurations along the sequence. The *d-VASS reachability problem* is to decide, given a d-VASS V and configurations $c, c' \in C$, whether $c \to^* c'$.

We are going to reduce questions about (m_1, ∞)-winning in BBEGs to questions about VASSs. The underlying idea is as follows. First, once we bound the memory of Player 1, we can guess a transducer that generates her strategy. The product of the BBEG with such a transducer results in a *one-player BBEG*, in which all positions belong to Player 2. As the evolution of a one-player BBEG does not involve alternation between players, we can model it by a VASS. Essentially, the configurations of the VASS correspond to positions in the game along with energy vectors of the players. The winning condition in the BBEG induces requirement on the VASS, as formalized in the following lemma (see proof in Appendix A.10).

▶ **Lemma 12.** *Given a (d_1, d_2)-BBEG G in which all the positions are owned by Player 2, the winner in G can be decided by solving at most d_1 instances of $(d_2 + 1)$-VASS reachability.*

We can now use Lemma 12 in order to decide whether Player 1 (m_1, ∞)-wins a given BBEG.

▶ **Theorem 13.** *Given a BBEG G and $m_1 \in \mathbb{N}$, determining whether Player 1 (m_1, ∞)-wins G is decidable.*

Proof. Let G be a (d_1, d_2)-BBEG, for some $d_1, d_2 \geq 1$, and consider a transducer T with state space Q of size m_1 that maintains a strategy for Player 1. Let $S = S_1 \cup S_2$ be the state space of G. When Player 1 follows T, the possible outcomes of the game are embedded in the product $G \times T$. The product has state space $S \times Q$. Each positions in $S_1 \times Q$ has a single successor: its S-component is determined by the output function of T and its Q-component is determined by the transition function of T. Therefore, we can refer to the product $G \times T$ as a BBEG all whose positions belong to Player 2. The updates on the edges of the product BBEG are induced by these in G, and so it is a (d_1, d_2)-BBEG. By Lemma 12, determining the winner in $G \times T$ can be reduced to solving d_1 instances of $(d_2 + 1)$-VASS-reachability, which is decidable [22].

It follows that determining whether Player 1 (m_1, ∞)-wins G can be decided by going over the finitely many candidates transducers T of size m_1, and applying the above check to each of them. ◀

▶ Remark 14. **[Complexity]** While Theorems 13 only refer to decidability, known complexity results on VASS can be used in order to give complexity upper bounds in some cases. Specifically, as 2-VASS reachability is PSPACE-complete [4], and the candidate transducers T are polynomial in m_1, we get that determining whether Player 1 (m_1, ∞)-wins G is decidable in PSPACE for $(1, 1)$-BBEGs with m_1 given in unary. ◀

We note that while similar considerations can be used in order to decide whether Player 2 (∞, m_2)-wins a given BBEG, for $m_2 \in \mathbb{N}$ (see proof in Appendix A.11), the latter does not provide a solution to the problem of deciding whether Player 1 (∞, m_2)-wins a given BBEG, which we leave open. Indeed, BBEGs are not (∞, m_2)-determined, in the sense that there is a BBEG G and $m_2 \in \mathbb{N}$ such that neither Player 1 (∞, m_2)-wins nor Player 2 (∞, m_2)-wins G. For example, by switching the vertices owned by Player 1 and Player 2 in the BBEG appearing in Figure 3, we get a BBEG such that Player 1 does not (∞, m_2)-wins for all $m_2 \in \mathbb{N}$, and Player 2 does not wins with a memoryless strategy, and in particular does not $(\infty, 1)$-wins.

Finally, we note that, unsurprisingly, even when we fix the size of the strategy of Player 2, the size of the strategy required for Player 1 to win depends on both the number of positions in the game and the updates in its transitions, inducing a strict hierarchy. Specifically, in Appendix A.12, we show that for all $m_1 \in \mathbb{N}$, there is a BBEG G_{m_1} with 3 states as well as a BBEG G'_{m_1} in which all updates are in $\{-1, 0, 1\}$, such that Player 1 $(m_1 + 2, 0)$-wins G_{m_1} and G'_{m_1}, yet Player 2 $(m_1 + 1, 0)$-wins G_{m_1} and G'_{m_1}. Similar results can be shown for the size of the strategy for Player 2.

▶ Remark 15. **[From general to position-based strategies]** Our positive decidability and complexity results are based on going over candidate strategies for the players. By restricting attention to strategies that refine the BBEG, these results apply also to position-based finite-state strategies. In addition, our lower bounds apply already for memoryless strategies, and so apply also for position-based finite-state strategies. ◀

5 BBEG with Bounded Energy Capacities

So far we studied BBEGs in which the players must keep their energy level non-negative, but there is no upper bound on the energy they may accumulate. This corresponds to systems in which there is no bound on the capacity of the energy resource. In many cases (c.f., battery, disc space), such a bound exists. In this section we study the problem of determining the winner in BBEGs in which one of the players has a bounded energy capacity. We consider both a semantics in which an overflow leads to losing the game (losing semantics, for short) and a semantics in which an overflow is truncated (truncated semantics, for short).

Formally, a *one-player-bounded BBEG* is $G = \langle S_1, S_2, s_{init}, E, d_1, d_2, x_0^1, x_0^2, \tau, j, b\rangle$, which extends a BBEG by specifying a player $j \in \{1, 2\}$ and a bound vector $b \in \mathbb{Z}^{d_j}$. In the losing semantics, the definition of a winning computation in a one-player-bounded BBEG is similar to the definition in the case of a BBEG, except that the requirement for the energy to stay non-negative is replaced, for Player j, by a requirement to stay both non-negative and below the bound b. Formally, a computation c that is winning for Player j has to satisfy, in addition to the winning condition of a BBEG, the requirement $e_j(c_n)[i] \leq b[i]$ for all $n \geq 1$ and $i \in [d_j]$. In the truncated semantics, the winning condition is as in the underlying BBEG, yet the energy level of Player j up to the n-th position in a run $r = s_1, s_2, \ldots$ is defined inductively for all $i \in [d_j]$ as follows: $e_j(r_n)[i] = \min\{b[i], e_j(r_{n-1})[i] + \tau(\langle s_i, s_{i+1}\rangle)[j][i]\}$, where $e_j(r_0)[i] = x_0^j[i]$.

In Theorem 16 below we show that the problem of deciding whether Player 1 wins a one-player-bounded BBEG is decidable for BBEGs of all dimensions. Essentially, our solution is based on expanding the position space of the game to maintain the energy level of Player j. Consequently, the cost function in the transitions updates the energy level of the other player only. When $j = 2$, thus the energy of Player 2 is bounded, we are left with updates to the energy level of Player 1. Thus, we obtain a standard multi-dimensional energy game, except that we add a sink that is winning for Player 1 and corresponds to positions in which the energy level of Player 2 is negative or, in the losing semantics, is above the bound b.

When $j = 1$, thus the energy of Player 1 is bounded, we obtain a multi-dimensional energy game in which transitions update the energy level Player 2 only. The game contains a sink, which is losing for Player 1, and Player 2 wins the game if she can reach the sink without her energy becoming negative. Thus, the setting is similar to that of multi-dimensional reachability energy games. By [16], one-dimensional energy-reachability games can be decided in NP∩coNP, and so our proof boils down to extending their algorithm to the multi-dimensional case. The full details can be found in Appendix A.13.

▶ **Theorem 16.** *The problem of determining whether Player 1 wins a one-player-bounded BBEG is decidable.*

▶ Remark 17. [**Bounding only some of the energy components**] In the multi-dimensional setting, we can consider games in which each player has energy bounds for some of the components in her energy vector. It is easy to see for for $d_1, d_2 \geq 1$ determining the winner of a (d_1, d_2)-BBEG is decidable iff each player has at most one unbounded component. Indeed, one can extend the position space of a BBEG to remember the value of the $(d-1)+(d-1)$ bounded components, and then deciding $(1,1)$-BBEG. ◀

References

1. P.A. Abdulla, M.F. Atig, P. Hofman, R. Mayr, K.N. Kumar, and P. Totzke. Infinite-state energy games. In *Proc. 29th IEEE Symp. on Logic in Computer Science*, pages 7:1–7:10. ACM, 2014.
2. G. Amram, S. Maoz, O. Pistiner, and J. O. Ringert. Efficient algorithms for omega-regular energy games. In *24th International Symposium on Formal Methods*, volume 13047 of *Lecture Notes in Computer Science*, pages 163–181. Springer, 2021.
3. R. Bloem, K. Chatterjee, and B. Jobstmann. Graph games and reactive synthesis. In *Handbook of Model Checking.*, pages 921–962. Springer, 2018.
4. M. Blondin, A. Finkel, S. Göller, C. Haase, and P. McKenzie. Reachability in two-dimensional vector addition systems with states is pspace-complete. In *Proc. 30th IEEE Symp. on Logic in Computer Science*, pages 32–43. IEEE, 2015.
5. A. Borodin and R. El-Yaniv. *Online Computation and Competitive Analysis.* Cambridge University Press, 1998.
6. P. Bouyer, U. Fahrenberg, K.G. Larsen, N. Markey, and J. Srba. Infinite runs in weighted timed automata with energy constraints. In *6th International Conference on Formal Modeling and Analysis of Timed Systems*, volume 5215 of *Lecture Notes in Computer Science*, pages 33–47. Springer, 2008.
7. T. Brázdil, P. Jančar, and A. Kučera. Reachability games on extended vector addition systems with states. In *Proc. 37th Int. Colloq. on Automata, Languages, and Programming*, pages 478–489. Springer, 2010.
8. A. Chakrabarti, L. de Alfaro, T.A. Henzinger, and M. Stoelinga. Resource interfaces. In *International Workshop on Embedded Software*, pages 117–133. Springer, 2003.
9. K. Chatterjee and L. Doyen. Energy parity games. *Theoretical Computer Science*, 458:49–60, 2012.
10. K. Chatterjee, L. Doyen, T.A. Henzinger, and J-F. Raskin. Generalized mean-payoff and energy games. In *Proc. 30th Conf. on Foundations of Software Technology and Theoretical Computer Science*, volume 8 of *LIPIcs*, pages 505–516. Schloss Dagstuhl - Leibniz-Zentrum für Informatik, 2010.
11. K. Chatterjee, A.K. Goharshady, and Y. Velner. Quantitative analysis of smart contracts. In *27th European Symposium on Programming Languages and Systems*, volume 10801 of *Lecture Notes in Computer Science*, pages 739–767. Springer, 2018.
12. J-B. Courtois and S. Schmitz. Alternating vector addition systems with states. In *39th Int. Symp. on Mathematical Foundations of Computer Science*, pages 220–231. Springer, 2014.
13. R. Ehlers. Symbolic bounded synthesis. In *Proc. 22nd Int. Conf. on Computer Aided Verification*, volume 6174 of *Lecture Notes in Computer Science*, pages 365–379. Springer, 2010.
14. E.A. Emerson. Automata, tableaux, and temporal logics. In *Proc. Workshop on Logic of Programs*, volume 193 of *Lecture Notes in Computer Science*, pages 79–87. Springer, 1985.
15. U. Fahrenberg, L. Juhl, K. G. Larsen, and J. Srba. Energy games in multiweighted automata. In *8th International Colloquium on Theoretical Aspects of Computing*, volume 6916 of *Lecture Notes in Computer Science*, pages 95–115. Springer, 2011.
16. L. Hélouët, N. Markey, and R. Raha. Reachability games with relaxed energy constraints. *Information and Computation*, page 104806, 2021.
17. T.A. Henzinger. From Boolean to quantitative notions of correctness. In *Proc. 37th ACM Symp. on Principles of Programming Languages*, pages 157–158, 2010.
18. J.E. Hopcroft and J-J. Pansiot. On the reachability problem for 5-dimensional vector addition systems. *Theoretical Computer Science*, 8:135–159, 1979.
19. M. Jurdzinski, R. Lazic, and S. Schmitz. Fixed-dimensional energy games are in pseudo-polynomial time. In *Proc. 42nd Int. Colloq. on Automata, Languages, and Programming*, volume 9135 of *Lecture Notes in Computer Science*, pages 260–272. Springer, 2015.

20 M. Jurdziński, R. Lazić, and S. Schmitz. Fixed-dimensional energy games are in pseudo-polynomial time. In *Proc. 42nd Int. Colloq. on Automata, Languages, and Programming*, pages 260–272. Springer, 2015.
21 O. Kupferman, Y. Lustig, M.Y. Vardi, and M. Yannakakis. Temporal synthesis for bounded systems and environments. In *Proc. 28th Symp. on Theoretical Aspects of Computer Science*, pages 615–626, 2011.
22 J. Leroux and S. Schmitz. Reachability in vector addition systems demystified. In *Proc. 35th IEEE Symp. on Logic in Computer Science*, 2015.
23 D.A. Martin. Borel determinacy. *Annals of Mathematics*, 65:363–371, 1975.
24 A. Menezes, P.C. van Oorschot, and S.A. Vanstone. *Handbook of Applied Cryptography.* CRC Press, 1996.
25 M.L. Minsky. *Computation: Finite and Infinite Machines.* Prentice Hall, 1 edition, 1967.
26 C.H. Papadimitriou and M. Yannakakis. On complexity as bounded rationality (extended abstract). In *Proc. 26th ACM Symp. on Theory of Computing*, pages 726–733, 1994.
27 A. Pnueli and R. Rosner. On the synthesis of a reactive module. In *Proc. 16th ACM Symp. on Principles of Programming Languages*, pages 179–190, 1989.
28 A. Rubinstein. Finite automata play the repeated prisoner's dilemma. *J. Economic Theory*, 39(1):83 – 96, 1986.
29 S. Schewe and B. Finkbeiner. Bounded synthesis. In *5th Int. Symp. on Automated Technology for Verification and Analysis*, volume 4762 of *Lecture Notes in Computer Science*, pages 474–488. Springer, 2007.
30 W. Thomas. On the synthesis of strategies in infinite games. In *Proc. 12th Symp. on Theoretical Aspects of Computer Science*, volume 900 of *Lecture Notes in Computer Science*, pages 1–13. Springer, 1995.
31 C. Wrathall. Complete sets and the polynomial-time hierarchy. *Theoretical Computer Science*, 3(1):23–33, 1976.
32 U. Zwick and M.S. Paterson. The complexity of mean payoff games on graphs. *Theoretical Computer Science*, 158:343–359, 1996.

A Proofs

A.1 Proof of the assumptions in Remark 2

It is easy to see that every BBEG with parallel edges has an equivalent BBEG of linear size without parallel edges. Indeed, let $s, t \in S$ be two positions and let A be the set of edges from s to t, with updates $l_1, ..., l_{|A|}$. We can add new positions $s_1^{(s,t)}, ..., s_{|A|}^{(s,t)}$, and edges $\{(s, s_i^{(s,t)}) \ : \ 1 \leq i \leq |A|\} \cup \{(s_i^{(s,t)}, t) \ : \ 1 \leq i \leq |A|\}$ instead of the parallel edges, with updates $\tau(\langle s, s_i^{(s,t)} \rangle) = l_i$ and $\tau(\langle s_i^{(s,t)}, t \rangle) = (0^{d_1}, 0^{d_2})$, for all $1 \leq i \leq |A|$.

It is also easy to see that every BBEG with has an equivalent BBEG of linear size in which each transition updates the energy to one player only. The only nontrivial issue in the decomposition of a transition is that we should first update the energy of the player that owns the source position. Thus, an edge leaving $s \in S_1$, labeled with (x_1, x_2) and leading to $t \in S$, can be replaced the two edges $\langle s, u_{s,t} \rangle$ with $\tau(\langle s, u_{s,t} \rangle) = (x_1, 0^{d_2})$, and $\langle u_{s,t}, t \rangle$ with $\tau(\langle u_{s,t}, t \rangle) = (0^{d_1}, x_2)$, for a new position $u_{s,t}$. For the case $s \in S_2$, the new edges update first the energy of Player 2.

Finally, we can translate a BBEG to a BBEG in which the updates on the transitions are all in $\{-1, 0, 1\}$. We describe the translation for $(1,1)$-BBEGs. A similar translation works for BBEGs of higher dimensions. Indeed, one can first convert a BBEG to one in which every transition updates the energy to one player only, as described above, and then replace an edge labeled with $(x_1, 0^{d_2})$ with $|x_1|$ edges that update x_1 to the energy of Player 1, while not affecting the energy of Player 2. Similarly, we can handle edges labeled with $(0^{d_1}, x_2)$. Note, however, that since we define the size of a BBEG with the costs on the edges of given in binary, the resulting BBEG is of size exponential in the size of the original BBEG. Since we consider BBEGs with updates in $\{-1, 0, 1\}$ only in contexts of decidability, this does not affect our results.

A.2 Correctness of the upper-bound reduction in Theorem 3

We prove that Player 1 wins in A from $\langle s_{init}, x_0^1, x_0^2\rangle$ iff Player 1 wins in G. First, an infinite computation in G induces an infinite game in A that never reaches the sink. Also, a finite computation in G in which Player 1 runs out of energy before Player 2, induces a finite game in A that is losing for Player 1. Finally, a finite computation in G that reaches a configuration in which Player 1 can make Player 2 lose, or Player 2 has no choice but to lose her energy, reaches a position in $Q_1' \cup Q_2'$ with the energy level of Player 2 being 0. The corresponding game in A reaches $Q_1' \cup Q_2'$ with the counter being 0. If the current position is in Q_1', Player 1 can use the δ_0-transition to the sink and stay there forever. If the current position is in Q_2', Player 2 has no choice but to use the δ_0-transition and reach the sink. Thus, Player 1 wins in G iff Player 1 can force an infinite game in A.

A.3 Correctness of the lower-bound reduction in Theorem 4

We prove that the reduction is correct, i.e., the machine M halts iff Player 2 wins in G. We describe a computation of M by an infinite sequence $f = f_0, f_1, f_2, ... \in (\{1, ..., n\} \times \mathbb{N} \times \mathbb{N})^\omega$, such that $f_0 = (1, 0, 0)$ and for all $i \geq 1$, we have that $f_i[1]$ is the location of the i-th command in the computation, and $f_i[2]$ and $f_i[3]$ are the values of the counters x and y, respectively, after reading that command. If for some $i \geq 0$ we have that $f_i[1] = n$, then $f_{i+1} = f_i$. Consider a computation $\pi \in comp(G)$, and let $v = v_0, v_1, ...$ be the projection of π on S_2. We say that π is *consistent* if for all $i \in \mathbb{N}$, we have that $e_1(v_i) = f_i[2] + f_i[3]$ and $e_2(v_i) = (f_i[2], f_i[3])$. That is, π is consistent if the energy level of Player 1 stores $x + y$, and the energy level of Player 2 stores $\langle x, y\rangle$.

First, we show that if a player has a winning strategy, then she also has a winning strategy that follows the instructions. That is, at every step of the computation, the best move for the current player is the one that leads to the state corresponding to the next command to be read according to M. For $c \in \{x, y\}$, denote by $L_{td}^c \subseteq L_{td}$ the set of locations of TEST-DEC commands that examine counter c. Note that excluding positions induced by the TEST-DEC gadgets, all positions in G belong to Player 2, and that the position corresponding to the halting command is losing for Player 1. Also note that all positions except some positions in the TEST-DEC gadgets are deterministic, that is, have a single transition leaving them.

Recall that for a consistent prefix p, the energy level $e_2(p)$ stores $\langle x, y\rangle$. Accordingly, for $c \in \{x, y\}$, we use $e_2^c(p)$ to refer to $e_2(p)[1]$ when $c = x$, and to refer to $e_2(p)[2]$ when $c = y$. Also, we use $\bar{c}$ to refer to y when $c = x$, and to refer to x when $c = y$.

We say that a strategy γ_1 for Player 1 is *consistent* if for every $p \in pref_1(G)$ ending in position $(i, 1)$ for $i \in L_{td}^c$, if $e_1(p) > e_2^{\bar{c}}(p)$, then $\gamma_1(p) = (i, 2)$, and if $e_1(p) \leq e_2^{\bar{c}}(p)$, then $\gamma_1(p) = j$, for j that is the positive successor of l_i. Similarly, we say that a strategy γ_2 for Player 2 is *consistent* if for every $p \in pref_2(G)$ ending in position $i \in L_{td}^c$, if $e_2^c(p) = 0$, then $\gamma_2(p) = (i, 1)$, and if $e_2^c(p) > 0$, then $\gamma_2(p) = k$, for k that is the negative successor of l_i.

Note that every player has a unique consistent strategy. Let γ_1 and γ_2 be the consistent strategies for Player 1 and Player 2, respectively. Let $r = outcome(\gamma_1, \gamma_2)$. We argue that r is consistent. Let $v = v_0, v_1, ...$ be the projection of r on S_2. We prove that for all $i \in \mathbb{N}$, it holds that $e_1(v_i) = f_i[2]+f_i[3]$ and $e_2(v_i) = (f_i[2], f_i[3])$. The proof proceeds by an induction on i. Initially, $f_0 = (1,0,0)$, and indeed for all runs in G, the initial position is 1 and the initial energy levels are 0 for Player 1 and $(0,0)$ for Player 2.

Let $m \geq 1$, and assume that the claim holds for all $0 \leq i < m$. If $v_m \notin L_{td}$, then Player 2 has a single successor, which corresponds to $f_{m+1}[1]$, and the energy levels are updated correctly. We now consider the case $v_m \in L^x_{td}$. Denote $f_{m-1}[1] = i$, $f_{m-1}[2] = x$, and $f_{m-1}[2] = y$. By the induction hypothesis, we have that $e_1(v_{m-1}) = x + y$ and $e_2(v_{m-1}) = (x, y)$. We distinguish between two cases:

1. If $x = 0$, then following γ_2, Player 2 chooses to go to position $(i, 1)$. This move does not affect the energy level. Since $x = 0$, then $x + y = y$, and following γ_1, Player 1 chooses to go to position j that is the positive successor of l_i. This transition does not affect the energy levels either. So, we have that $v_m = j$, $e_1(v_m) = x + y$, and $e_2(v_m) = (x, y)$, as required.
2. If $x > 0$, then, following γ_2, Player 2 chooses to go to position k that is the negative successor of l_i. This transition decreases by one the the energy level of Player 1 and the first component in the energy level of Player 2. So, $v_m = k$, $e_1(v_m) = x + y - 1$, and $e_2(v_m) = (x - 1, y)$, as required.

The case where $i \in L^y_{td}$ is similar.

Let γ_1, γ_2 be the consistent strategies for Player 1 and Player 2, respectively, and denote $r = \mathit{outcome}(\gamma_1, \gamma_2)$. We show that if Player 2 plays a strategy δ_2 that is not consistent, then she loses against the consistent strategy γ_1 of Player 1.

Assume that Player 1 plays γ_1 and Player 2 plays δ_2, which is not consistent. Let m be the minimal index in $\mathit{outcome}(\gamma_1, \delta_2)$ that deviates from r. That is, m is the minimal index t such that $\delta_2(r_t) \neq \gamma_2(r_t)$. Let i be the last position in r_m. Since all positions in $S_2 \setminus L_{td}$ are deterministic, it must be that $i \in L_{td}$. Assume that $i \in L^x_{td}$. Then, either $e_2(r_m)[0] = 0$ and $\delta_2(r_m) = k$, for k that is the negative successor of l_i, or $e_2(r_m)[0] > 0$ and $\delta_2(r_m) = (i, 1)$. Since m is minimal and r is consistent, we get that $e_1(r_m) = x + y$ and $e_2(r_m) = (x, y)$ for some $x, y \in \mathbb{N}$. If $x = 0$ and $\delta_2(r_m) = k$, then the first component in the energy level of Player 2 is decreased below 0, so she loses. If $x > 0$ and $\delta_2(r_m) = (i, 1)$, then according to γ_1, Player 1 chooses from $(i, 1)$ to go to $(i, 2)$. Since $x + y > y$, Player 1 wins at the sink $(i, 2)$. Hence, $\mathit{outcome}(\gamma_1, \delta_2)$ is winning for Player 1. The case where $i \in L^y_{td}$ is similar.

Since δ_2 is not winning for every $\delta_2 \neq \gamma_2$, we get that if Player 2 wins, her winning strategy must be consistent.

Now, we show that if Player 1 wins, then she can win with γ_1. Assume that Player 1 has a winning strategy $\delta_1 \neq \gamma_1$. We show that γ_1 is winning for Player 1 too. We already showed that $\mathit{outcome}(\gamma_1, \delta_2)$ is winning for Player 1 for every $\delta_2 \neq \gamma_2$. It is left to show that $\mathit{outcome}(\gamma_1, \gamma_2)$ is winning for Player 1. Let m be the minimal index t in $\mathit{outcome}(\delta_1, \gamma_2)$ such that $\delta_1(r_t) \neq \gamma_1(r_t)$. Since all positions in $S_1 \setminus (L_{td} \times \{1\})$ are deterministic, it must be that r_m ends in position $i \in L_{td} \times \{1\}$. Assume that $i \in L^x_{td} \times \{1\}$. Then, either $e_1(r_m) > e_2(r_m)[2]$ and $\delta_1(r_m) = j$ for j that is the positive successor of l_i, or $e_1(r_m) \leq e_2(r_m)[2]$ and $\delta_1(r_m) = (i, 2)$. Since m is minimal and r is consistent, we get that $e_1(r_m) = x + y$ and $e_2(r_m) = (x, y)$ for some $x, y \in \mathbb{N}$. If it is the case that $e_1(r_m) > e_2(r_m)[2]$, we have that $\delta_1(r_m) = j$ and $\gamma_1(r_m) = (i, 2)$. By going to $(i, 2)$, since $x + y > y$, we get that Player 2 loses at $(i, 2)$. Hence, $\mathit{outcome}(\gamma_1, \gamma_2)$ is winning for Player 1. Also, it cannot be the case that $e_1(r_m) \leq e_2(r_m)[2]$ and $\delta_1(r_m) = (i, 2)$: since $x + y \leq y$, we get that Player 1 loses at $(i, 2)$, in contradiction to the fact that δ_1 is winning. The case where $i \in L^y_{td} \times \{1\}$ is similar.

By the above, if Player 2 has a winning strategy, it must be consistent, and if Player 1 wins, her consistent strategy is winning. Therefore, the question of determining the winner in G is reduced to determining the winner of $outcome(\gamma_1, \gamma_2)$. When both players play their consistent strategies, we have that the energy levels are updated according to the values of the counters in f. Since the value of every counter is non-negative during the run, so are the energy levels of the players during the computation. Since the state corresponding to the HALT command is a rejecting sink for Player 1, we have that if M halts, then Player 2 wins in G. Otherwise, the energy levels of both players, in particular Player 1, remain non-negative during the infinite computation, and Player 1 wins.

A.4 Proof of Theorem 7 – finite-memory strategies

We prove that when both players are restricted to finite-memory strategies, there is no winning player in the BBEG G described in Example 1.

First, we show that for every (finite-memory) strategy γ_2 for Player 2, there is finite-memory strategy γ_1 of Player 1, such that $outcome(\gamma_1, \gamma_2)$ is winning for Player 1. Let γ_2 be a strategy for Player 2, and let n be the number of times Player 2 loops at s_1 before moving to s_3 (clearly, if she loops at s_1 forever, she loses). Let γ_1 be finite-memory strategy for Player 1 that loops at s_2 exactly n times, and then moves to s_3. As was shown in Example 1, $outcome(\gamma_1, \gamma_2)$ is winning for Player 1.

We now show that Player 1 does not have a finite-memory strategy that is winning against every finite-memory strategy of Player 2. Moreover, for every finite-memory strategy γ_1 for Player 1, there is a finite-memory strategy γ_2 for Player 2 such that $outcome(\gamma_1, \gamma_2)$ is winning for Player 2. Let γ_1 be a finite memory strategy for Player 1, modelled by a transducer $\mathcal{M} = \langle I, O, Q, q_0, \delta, L\rangle$, and let $m = |Q|$. Consider the finite-memory strategy γ_2 for Player 2 that loops $m+1$ times in s_1 and then moves to s_2. Let t be the number of times Player 1 loops at s_2 before moving to s_3 in $outcome(\gamma_1, \gamma_2)$. We show that $outcome(\gamma_1, \gamma_2)$ is winning for Player 2. Indeed, If $t < m+1$, then after looping t times in s_2, the energy level of Player 2 is strictly positive. Thus, when reaching s_3, the energy levels are non-negative, and s_3 is a losing position for Player 1. Now, assume $t \geq m+1$. For $i \in [m+1]$, Let p^i be the prefix of $outcome(\gamma_1, \gamma_2)$ after which s_2 is visited for the $(i+1)$-th time. That is, $p^i = (s_1)^{m+1} \cdot (s_2)^{i+1}$. Since $m+1 > m$, there are two indices $1 \leq k < l \leq m+1$ such that $\delta^*(q_0, p^k) = \delta^*(q_0, p^l)$. That is, after Player 2 loops $m+1$ times at s_1, looping k or l times in s_2 takes Player 1 to the same state of $\mathcal{M}$. Since $k < l$, we get that there is a loop in the strategy of Player 1, making her loop at s_2 forever. Thus, Player 1 makes both energy levels negative at the same time, and loses.

A.5 Proof of Theorem 7 – memoryless strategies

We prove that when both players are restricted to memoryless strategies, there is no winning player in the BBEG G described in Figure 3.

First, we show that for every memoryless strategy γ_1 for Player 1, there is a memoryless strategy γ_2 for Player 2 such that $outcome(\gamma_1, \gamma_2)$ is winning for Player 2. Note that Player 1 has to choose an outgoing edge only from s_2. Let us consider a memoryless strategy γ_1 for Player 1. If $\gamma_1(s_2) = s_3$, then for the strategy γ_2 for Player 2 that chooses to go from s_1 to s_2 by the edge labeled $(0, 0)$, it holds that $outcome(\gamma_1, \gamma_2)$ is winning for Player 2: when the computation reaches s_2, the energy level of Player 1 is 0, so the transition to s_3 makes her lose. If $\gamma_1(s_2) = s_4$, then the strategy γ_2 for Player 2 that chooses to go from s_1 to s_2 by the edge labeled $(1, 1)$ is such that $outcome(\gamma_1, \gamma_2)$ is winning for Player 2: when the computation reaches s_4, the energy level of Player 2 is 1, so she can pay 1 to reach s_5, which is a rejecting sink for Player 1.

We continue and show that for every strategy γ_2 for Player 2 (in particular a memoryless strategy), there is a memoryless strategy γ_1 for Player 1 such that $outcome(\gamma_1, \gamma_2)$ is winning for Player 1. Consider a strategy γ_2 for Player 2. If by following γ_2 Player 2 goes from s_1 to s_2 by the edge labeled $(0, 0)$, then a memoryless strategy γ_1 for Player 1 with $\gamma_1(s_2) = s_4$ is such that $outcome(\gamma_1, \gamma_2)$ is winning for Player 1: the energy level of Player 2 becomes negative at the transition to s_4. If by following γ_2 Player 2 goes from s_1 to s_2 by the edge labeled $(1, 1)$, then the strategy γ_1 for Player 1 with $\gamma_1(s_2) = s_3$ is such that $outcome(\gamma_1, \gamma_2)$ is also winning for Player 1: until the computation reaches s_3, the energy level of Player 1 remains non-negative, and s_3 is a winning sink for Player 1.

A.6 Proof of Lemma 8

As explained in Remark 2, we can assume that every edge in G updates the energy of one player only. It is easy to see that the strategies γ_1 and γ_2 induce a simple lasso in G. We can ignore all positions and edges in G that are not part of this lasso and consider the graph G' that is induced by this lasso. Denote by p the part of the computation before the loop, and by q the computation that is a single traversal of the loop. If some component decreases below 0 in $p \cdot q$, then the winner can be determined easily: the losing player is the one that owns the component that becomes negative earliest, which can be found in polynomial time. Otherwise, all $d_1 + d_2$ components remain non-negative during $p \cdot q$. If $e_1(q) \geq 0$, then Player 1 wins. Indeed, Player 1 survives the part before the loop and a single traversal of the loop, and since the loop does not have a negative effect on her energy level, she can take the loop forever. If $e_1(q) < 0$ and $e_2(q) \geq 0$, then Player 2 wins, since the energy level of Player 1 becomes negative at some point of the computation, while the energy level of Player 2 remains non-negative. If $e_1(q) < 0$ and $e_2(q) < 0$, we can check which player wins as follows. We can think of the energy of the players and the updates as vectors in $\mathbb{Z}^{d_1+d_2}$, where the first d_1 components belong to Player 1, and the last d_2 components belong to Player 2. We can check in polynomial time which component becomes negative earliest, and the loser is the player owns this component. Note that since every transition in G' updates the energy vector of a single player at a time, if there is more than one component that becomes negative at the same step, all of those components belong to the same player.

A.7 Missing details in the proof of Theorem 9

For the upper bound, consider a BBEG $G = \langle S_1, S_2, s_{init}, E, d_1, d_2, x_0^1, x_0^2, \tau \rangle$. Memoryless strategies for the players can be represented by polynomial-length strings. Then, given a memoryless strategy γ_1 for Player 1, the problem of checking whether there is a memoryless strategy γ_2 for Player 2 such that $outcome(\gamma_1, \gamma_2)$ is winning for Player 2 is in NP. Indeed, given a memoryless strategy γ_1 for Player 1, we can decide by a non-deterministic Turing Machine whether there is a memoryless strategy γ_2 for Player 2 such that $outcome(\gamma_1, \gamma_2)$ is winning for Player 2, by guessing γ_2 and applying Lemma 8. So, deciding whether there is a memoryless strategy γ_1 for Player 1 such that for every memoryless strategy γ_2 for Player 2 it holds that $outcome(\gamma_1, \gamma_2)$ is winning for Player 1, can be done by a nondeterministic polynomial-time Turing machine with an oracle to a nondeterministic polynomial-time Turing machine, and we are done.

We continue to the lower bound and describe the two parts of the BBEG in detail. For convenience, we describe the BBEG with parallel edges (see Remark 2). Both players start with the initial energy level 0, which is represented by the bit-vector 0^n. The assignment part is described in Figure 4.

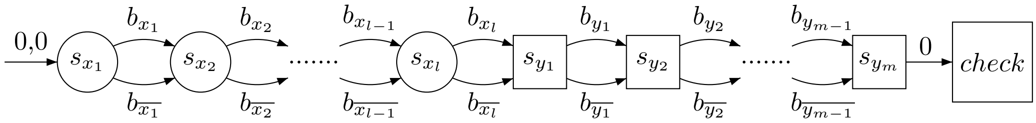

Figure 4 The assignment part.

For every literal $z \in Z$, let $b_z = \langle b_z^n, \ldots, b_z^1 \rangle \in \{0,1\}^n$ describe how the bit-vector v_g should be updated when z is assigned T. That is, for all $1 \leq i \leq n$, if the literal z appears in the clause c_i, then $b_z^i = 1$, and otherwise $b_z^i = 0$. For our example formula $(x_1 \wedge x_2 \wedge y_1) \vee (x_1 \wedge x_2 \wedge \overline{y_1})$, we have $b_{\overline{x_1}} = \langle 0, 0 \rangle$ and $b_{y_1} = \langle 0, 1 \rangle$. Since in this part, the energy levels of both players are updated in the same way, we label each transition in the figure by a single update. As described in the figure, first Player 1 assigns values to the variables in X and then Player 2 assigns values to the variable in Y. An assignment is reflected in the energy levels of both players being updated according to the literal that is chosen. In our example, if from s_{y_1} Player 2 chooses the transition that corresponds to assigning T to y_1, then the energy level of both players is increased by $\langle 0, 1 \rangle$.

We continue to the check part, where all the positions belong to Player 2. The check part is described in Figure 5. Here too, except for the transition to the sink, the updates to the energy levels of Player 1 and Player 2 coincide, and we label the transitions in the figure by a single update.

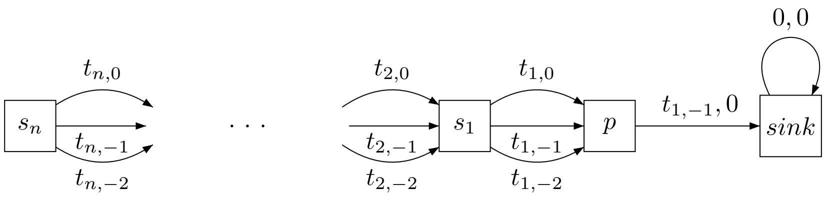

Figure 5 The check part.

For every $1 \leq i \leq n$ and $d \in \{0, -1, -2\}$, let $t_{i,d} = 0^{i-1} \cdot \{d\} \cdot 0^{n-(i+1)}$. That is, all the bits in $t_{i,d}$ are 0, except for the i-th bit, which is d. As described in Figure 5, the check part consists of a chain of positions s_i, for $n \geq i \geq 1$, where from s_{i+1} Player 2 proceeds to s_i while updating the energy levels by $t_{i,0}$, $t_{i,-1}$, or $t_{i,-2}$. Then, from position p, there is a single transition with updates $t_{1,-1}, 0$ to the energy levels. Thus, the least significant bit of the energy level of Player 1 is decreased by 1, and the energy level of Player 2 is not changed.

We now prove that θ is true iff Player 1 wins in G with a memoryless strategy.

Assume first that θ is true. Then, there is an assignment f_X for X such that for every assignment f_Y for Y, we have that ψ is true under $f_X \cup f_Y$. We show that there is a memoryless strategy for Player 1 that is winning against every (not necessarily memoryless) strategy for Player 2. An assignment f_X for X induces a memoryless strategy γ_{f_X} for Player 1 in which for every variable x_i such that $f_X(x_i) = T$, Player 1 chooses from s_{x_i} the transition labeled b_{x_i}, and for every variable x_i such that $f_X(x_i) = F$, Player 1 chooses from s_{x_i} the transition labeled $b_{\overline{x_i}}$. We show that γ_{f_X} is winning for Player 1. Let γ be a strategy for Player 2, and let f_Y be the assignment for Y induced by γ_{f_X} and γ. That is, $f_Y(y_i) = T$ if γ proceeds from s_{y_i} with the transition labeled b_{y_i} in the computation in which Player 1 follows γ_{f_X}, and $f_Y(y_i) = F$ if γ proceeds from s_{y_i} with the transition labeled $b_{\overline{y_i}}$. When the computation that is consistent with γ_{f_X} and γ reaches the check part, the energy level of both players is $v_{f_X \cup f_Y}$. Since $f_X \cup f_Y$ satisfies ψ, we have that there is $1 \leq i \leq n$ such that the i-th bit of $v_{f_X \cup f_Y}$ is 3. Let v^p be the bit-vector the players own when reaching p. It is easy to verify that v^p is not all-zero. Let j be the most significant bit in v^p that is not 0. We distinguish between two cases. If the j-th bit of v_p is positive, then v^p is positive. In this case, $v^p + t_{1,-1}$ is not negative, and Player 1 can loop in the sink forever and win the game. Otherwise, the j-th bit of v_p is negative, so v^p is negative. So, at some point at the check part, the current bit-vector of the players becomes negative, as a consequence of step made by Player 2. So Player 2 loses.

For the second direction, assume that θ is false, and consider a strategy γ for Player 1. Note that every strategy for Player 1 in G is memoryless. Let f_X be the assignment for X induced by γ. Then, there is an assignment f_Y for Y such that ψ is false under $f_X \cup f_Y$. Let γ_{f_Y} be the following memoryless strategy for Player 2. First, at the assignment part, the strategy γ_{f_Y} is consistent with f_Y. That is, as detailed above, for a position s_{y_i} the strategy γ_{f_Y} proceeds with the transition labeled with the update that corresponds to $f_Y(y_i)$. Let $v = \langle b_n, b_{n-1}, ..., b_1\rangle$ be the energy level of both players at the end of the assignment part. Since ψ is false under $f_X \cup f_Y$, then $b_i \in \{0, 1, 2\}$ for all $n \geq i \geq 1$. Accordingly, in the check part, the strategy γ_{f_Y} can choose from s_i a transition labeled $t_{i,-b_i}$, namely a transition that decreases the i-th bit of the energy levels of both players to 0. Consequently, the computation of G that is consistent with γ and γ_{f_Y} reaches the state p with energy level 0, and reaches the sink with a negative energy level for Player 1, which loses.

A.8 Proof of Theorem 10

The upper bound follows from Lemma 8 by arguments similar to these in Theorem 9. For the lower bound, we show a reduction similar to the one in the proof of Theorem 9. Let $G = \langle S_1, S_2, s_{init}, E, 1, 1, 0, 0, \tau\rangle$ be the BBEG in the proof of Theorem 9, and consider the BBEG $G' = \langle S_2, S_1, s_{init}, E, 1, 1, 0, 0, \tau'\rangle$, where for all $\langle s, s'\rangle \in E$ such that $s \neq sink$, we have that $\tau'(\langle s, s'\rangle) = (\tau(\langle s, s'\rangle)[2], \tau(\langle s, s'\rangle)[1])$, and $\tau'(\langle sink, sink\rangle) = (-1, 0)$. Note that G' is obtained from G by switching between S_1 and S_2, between the cost functions of the players, and by changing the sink to be a winning position for Player 2.

We claim that Player 2 wins in G' iff θ is true: at the assignment part, Player 2 assigns values to the variables in X, and then Player 1 assigns values to the variable in Y. As in Theorem 9, the assignment is reflected in the energy levels of both players being updated according to the literal that is chosen. Then, at the check part, we let Player 1 win iff there is an unsatisfied clause under the assignment induced by the first part of the play.

A.9 Proof of Theorem 11

We start with the upper bounds. Let $G = \langle S_1, S_2, s_{init}, E, d_1, d_2, x_0^1, x_0^2, \tau\rangle$. An S/S-transducer with m states can be represented by a string polynomial in m and $|G|$. Also, given two transducers $\mathcal{M}_1$ and $\mathcal{M}_2$ with m_1 and m_2 states, respectively, one can build a BBEG G' that is the product of G with $\mathcal{M}_1$ and $\mathcal{M}_2$, and is in size polynomial of G, m_1, and m_2. Note that since in G' is a product of a game with strategies for both players, it is a simple lasso that correspond to the outcome of the game when the players follow their strategies. Then, by arguments similar to those in Lemma 8, one can determine the winner in G' in time polynomial in G, m_1 and m_2. Hence, for $j \in \{1, 2\}$, deciding whether Player j (m_1, m_2)-wins can be done by a nondeterministic polynomial-time Turing machine with an oracle to a nondeterministic polynomial-time Turing machine, and we are done.

We continue to the lower bound. In Theorem 9, we described a reduction from QBF_2 that proves Σ_2^P-hardness for determining whether Player 1 wins when both players play memoryless strategies, and similarly for Player 2. Taking $m_1 = |S_1|$ and $m_2 = |S_2|$, the reduction also proved Σ_2^P-hardness for deciding whether Player 1 (m_1, m_2)-wins with a finite-state strategy that refines G, and for deciding whether Player 2 (m_1, m_2)-wins with a finite-state strategy that refines G. We argue that the same reduction proves hardness also when the players have general finite-state strategy. That is, given a formula $\theta = \exists x_1, ..., x_l \forall y_1, ..., y_m \psi$, we construct the same game G as in Theorem 9, and claim that θ is true iff Player 1 $(|S_1|, |S_2|)$-wins in G.

For the first direction, assume θ is true. Assume $S_1 = \{s_1^1, ..., s_{|S_1|}^1\}$. As we saw in Theorem 9, Player 1 has a memoryless winning strategy γ_1. This memoryless strategy is winning against any general strategy for Player 2. Consider the strategy $\gamma_{\mathcal{M}_1}$ induced by a transducer $\mathcal{M}_1$ with states $q_1, .., q_{|S_1|}$, where each state q_i defines the behaviour of Player 1 in s_i^1 according to γ_1. That is, $L(q_i) = \delta(q_i, s) = \gamma_1(s_i^1)$, for all $i \in [n]$ and $s \in S$. For every $p \in \mathit{pref}_1(G)$, it holds that $\gamma_{\mathcal{M}_1}(p) = \gamma_1(p)$. Hence, $\gamma_{\mathcal{M}_1}$ is a winning strategy.

For the second direction, recall that the proof of Theorem 9 shows that if θ is not true, then for every strategy (not necessarily memoryless) γ_1 for Player 1, Player 2 has a memoryless strategy γ_2 such that $\mathit{outcome}(\gamma_1, \gamma_2)$ is winning for Player 2. Assume $S_2 = \{s_1^2, ..., s_{|S_2|}^2\}$. So, given a transducer $\mathcal{M}_1$ with $|S_1|$ states, let $\gamma_{\mathcal{M}_1}$ be the strategy for Player 1 that corresponds to $\mathcal{M}_1$. Let γ_2 be a memoryless strategy for Player 2 such that $\mathit{outcome}(\gamma_{\mathcal{M}_1}, \gamma_2)$ is winning for Player 2. Then the transducer $\mathcal{M}_2$ with states $q_1, .., q_{|S_2|}$, in which each state q_i defines the behaviour of Player 2 in s_i^2 according to γ_2, induces a winning strategy.

The arguments for the problem of deciding whether Player 2 (m_1, m_2)-wins are similar, applied to the BBEG G' in the proof of Theorem 9.

A.10 Proof of Lemma 12

Let $G = \langle \emptyset, S_2, s_{init}, E, d_1, d_2, x_0^1, x_0^2, \tau \rangle$ be a BBEG. We construct a VASS V with configurations that represent a position and energy vectors in G, with target configuration that represents a position and energy vectors from which Player 2 can win in one move. The idea is that Player 2 wins in G iff she can force the game to an edge in which the energy level of Player 1 is low enough at some component to drop below 0, and her own energy level is high enough to stay non-negative after taking this edge.

Formally, for all $k \in [d_1]$, we construct the $(d_2 + 1)$-VASS $V_k = \langle Q_k, T_k \rangle$ as follows. Let $Q_k = S \cup \{s_{sink}\}$ for some $s_{sink} \notin S$, and $T_k^{'} = \{\langle u, z, v \rangle : \langle u, v \rangle \in E$, for all $i \in [d_2]$ we have that $z[i] = \tau(\langle u, v \rangle)[2][i]$, and $z[d_2 + 1] = \tau(\langle u, v \rangle)[1][k]\}$. That is, the vectors on the transitions in $T_k^{'}$ represent the update to the energy vector of Player 2 in their first d_2 components, and the update of the k-th component of Player 1 in their last component. We define the set of transitions $T_k^{''} = \{\langle u, z, s_{sink} \rangle :$ there is $v \in S$ such that $\langle u, z, v \rangle \in T_k^{'}\}$. That is, for every transition in $T_k^{'}$ leaving a state u, there is a transition in $T_k^{''}$ leaving u with the same update and entering s_{sink}. For $i \in [d_2 + 1]$ and $z \in Z$, let b_i^z to be the vector of dimension $d_2 + 1$ with z in the i-th component, and 0 in all other components. We define the set of transitions $T_k^{'''} = \{\langle s_{sink}, b_i^{-1}, s_{sink} \rangle : i \in [d_2]\} \cup \{\langle u, b_{d_2+1}^1, u \rangle : u \in V \setminus \{s_{sink}\}\}$. That is, s_{sink} has self loops that can decrease the components that belong to Player 2. Also, every state but the sink has a self loop that increases the component that belongs to Player 1. We define the set of transitions of V to be $T_k = T_k^{'} \cup T_k^{''} \cup T_k^{'''} \cup \{\langle s_{sink}, 0^{d_2+1}, s_{sink} \rangle\}$. Let $v_{init}^k \in \mathbb{Z}^{d_2+1}$ be the vector with $v_{init}^k[i] = x_0^2[i]$ for all $i \in [d_2]$, and $v_{init}^k[d_2 + 1] = x_0^1[k] + 1$. That is, v_{init}^k represents x_0^2 in its first d_2 components, and $x_0^1[k] + 1$ in its last component. Note that we added 1 to $x_0^1[k]$. That is because in V_k we want to let the last component reach 0, if in the corresponding computation in G it becomes negative.

We claim that Player 2 wins G iff there is $k \in [d_1]$ such that $\langle s_{init}, v_{init}^k \rangle \to^* \langle s_{sink}, 0^{d_2+1} \rangle$ in V_k. It is clear that Player 2 wins G iff there is $k \in [d_1]$ such that Player 2 can lead the computation in G to a configuration in which Player 1 runs out of energy in her k-th component, while keeping her own energy vector non-negative during the whole computation. Let $p \cdot s$ be the computation before the final step in which Player 1 drops below 0 in the k-th component. Thus, there is an edge $e = \langle s, s' \rangle \in E$ for some $s' \in S$, such that $e_2(p \cdot s) + \tau(e)[2] \geq 0$, but $e_1(p \cdot s)[k] + \tau(e)[1][k] < 0$. The sequence of successive configurations in V_k that witnesses $\langle s_{init}, v_{init}^k \rangle \to^* \langle s_{sink}, 0^{d_2+1} \rangle$ starts with $p \cdot s$. When reaching s, the current vector v has $e_2(p \cdot s)$ for its first d_2 components, and $e_1(p \cdot s)[k]$ in its last one. Now, we can take the self loop $\langle s, b_{d_2+1}^1, s \rangle$ exactly $-(e_1(p \cdot s)[k] + \tau(e)[1][k])$ times, setting the k-th component to $-\tau(e)[1][k]$. Then, we move to the sink with the transition that adds $\tau(e)[2]$ in the first d_2 components, and $\tau(e)[1][k]$ in the last one. Note that now the vector has $e_2(p \cdot s) + \tau(e)[2]$ in its first d_2 components, and 0 in its last one. Now, for all $i \in [d_2]$, we take the loop $\langle s_{sink}, b_i^{-1}, s_{sink} \rangle$ exactly $e_2(p \cdot s)[i] + \tau(e)[2][i]$ times, setting to 0 the i-th component. After that, we reached the configuration $\langle s_{sink}, 0^{d_2+1} \rangle$. Note that the only way to reach $\langle s_{sink}, 0^{d_2+1} \rangle$ is by reaching a position $s \in S$ with a vector in which the first d_2 components are big enough to survive the transition to the sink, and the last one is small enough to be set to 0, after a (maybe empty) sequence of incrementation-loops at s. Since we start with $x_0^1[k] + 1$ at the $(d_2 + 1)$-th component, this sequence of configuration induces a computation in G in which the k-th component of Player 1 drops below 0.

A.11 Deciding whther Player 2 (∞, m_2)-wins

▶ **Theorem 18.** *Given a BBEG G and $m_2 \in \mathbb{N}$, determining whether Player 2 (∞, m_2)-wins G is decidable.*

Proof. Assume that G is a (d_1, d_2)-BBEG. As in (m_1, ∞)-winning for Player 1, we can consider the product of G with a transducer T for Player 2 with m_2 states. This product is a BBEG all whose positions are owned by Player 1. It is easy to see that Player 1 wins in this product iff it contains infinite computation in which her energy level is always non-negative, or a finite prefix of a computation that leads to a position in which the energy level of Player 2 is negative in some component while the energy vector of Player 1 along this prefix is always non-negative. Checking the second condition can be done by a reduction to VASS, with a construction similar to the one in the proof of Lemma 12. Checking the first condition can also be reduced to VASS, but is more complicated. So, for the sake of decidability, it is sufficient to note that the first condition can also be solved by solving a d_1-dimensional energy game, in which we ignore the components that belong to Player 2. From [20, 7], the given initial-credit problem of d_1-dimensional energy game can be solved in $(d_1 - 1)-$EXPTIME, and is thus decidable.

It follows that for every transducer with m_2 states for Player 2, we can check whether Player 1 wins when Player 2 follows this transducer. Moreover, if Player 1 does not win, Player 2 does, and so the transducer T induces a winning strategy for her. Thus, Player 2 (∞, m_2)-wins G iff she wins with some transducer with m_2 states, that is, iff she wins in at least on of these products, which is decidable. ◀

A.12 On the size of the strategy required for Player 1

The two theorems below show that even when we fix the size of the strategy of Player 2, the size of the strategy of Player 1 depends on both the number of positions in the game and the updates in its transitions.

▶ **Theorem 19.** *There are $n, m_2 \in \mathbb{N}$ such that for all $m_1 \in \mathbb{N}$, there is a BBEG G_{m_1} with n states such that Player 1 $(m_1 + 2, m_2)$-wins G_{m_1}, but Player 2 $(m_1 + 1, m_2)$-wins G_{m_1}.*

Proof. We show the claim holds with $n = 3$ and $m_2 = 0$. For all $m_1 \in \mathbb{N}$, consider the BBEG G_{m_1} in Figure 6, with initial vector 0 for each player. Moving fron s_1 to s_2, Player 1 updates the two energy vectors to m_1. Since s_3 is a losing position for Player 1, Player 1 must take the transition from s_2 to s_3 when the energy vector of Player 2 is 0. In order to do so, she must loop at s_2 exactly m_1 times. If she loops less than m_1 times, the computation reaches s_3, which makes Player 1 lose. If she loops at s_2 more than m_1 times, she makes herself loose at s_2. A transducer with $m_1 + 2$ states enables Player 1 remember how many s_2-loops are left before moving to s_3 (the first state is for moving from s_1 to s_2, and the last one is for moving to s_3 and stay there), while every transducer with less than $m_1 + 2$ states induces a losing strategy for Player 1. ◀

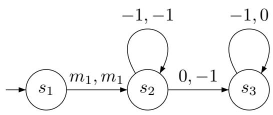

Figure 6 The BBEG G_{m_1}

We now show that the strategy size of Player 1 depends on number of positions. This stays valid even if the updates on the edges and the strategy size of Player 2 are fixed.

▶ **Theorem 20.** *There are $w, m_2 \in \mathbb{N}$ such that for all $m_1 \in \mathbb{N}$, there is a BBEG G'_{m_1} in which all updates are in $\{-w, -(w-1), ..., w-1, w\}$, such that Player 1 $(m_1 + 2, m_2)$-wins G'_{m_1}, but does not $(m_1 + 1, m_2)$-wins G'_{m_1}.*

Proof. We show the claim holds with $w = 1$ and $m_2 = 0$. For all $m_1 \in \mathbb{N}$, consider the BBEG G'_{m_1} in Figure 7, with initial energy 0 for each player. Note that G'_{m_1} is essentially the BBEG G_{m_1} in Figure 6, except that the first transition is replaced by m_1 transitions labeled by $(1, 1)$. The proof is similar to the proof of Theorem 19. ◀

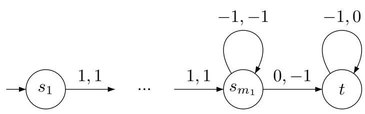

Figure 7 The BBEG G'_{m_1}

A.13 Proof of Theorem 16

Let $G = \langle S_1, S_2, s_{init}, E, d_1, d_2, x_0^1, x_0^2, \tau, j, b\rangle$ be a one-player-bounded BBEG. Assume first that $j = 2$, thus $b \in \mathbb{Z}^{d_2}$ is a bound vector for Player 2. We start with the losing semantics and define the d_1-dimensional energy game $G' = \langle S'_1, S'_2, \langle s_{init}, x_0^2\rangle, E', \tau'\rangle$ as follows. Let V be the set of all non-negative vectors in $\mathbb{Z}^{d_2}$ that are bounded by b. That is, $V = \{v \in \mathbb{Z}^{d_2} : 0 \le v[i] \le b[i]$ for all $i \in [d_2]\}$. Let $S'_1 = S_1 \times V$ and $S'_2 = S_2 \times V$. Also, let $S = S'_1 \cup S'_2 \cup \{s_{sink}\}$, for some $s_{sink} \notin S_1 \cup S_2$. We now define a set of edges $E' \subseteq S' \times S'$ and a cost function $\tau' : E' \to \mathbb{Z}^{d_1}$. For all $e = \langle s, s'\rangle \in E$ and $v, v' \in V$ such that $v' = v + \tau(e)[2]$, we have the edge $e' = \langle\langle s, v\rangle, \langle s', v'\rangle\rangle$ in E', with $\tau'(e') = \tau(e)[1]$. For all $e = \langle s, s'\rangle \in E$ and $v \in V$ such that $v + \tau(e)[2] \notin V$, we have the edge $e' = \langle\langle s, v\rangle, s_{sink}\rangle$ in E', with $\tau'(e') = \tau(e)[1]$.

We also have an edge $\langle s_{sink}, s_{sink}\rangle$ in E', with $\tau'(\langle s_{sink}, s_{sink}\rangle) = 0^{d_1}$. Note that the cost function τ' defines the cost for Player 1 only, while S' maintains the energy level of Player 2.

We claim Player 1 wins in G iff Player 1 wins in G' with initial energy x_0^1. Indeed, every computation c in G induces a computation c' in G', such that the current energy level of Player 2 in c' is maintained at the second component of the current position in c', and the energy level of Player 1 in c is the same as in c'. Thus, if c is infinite, so is c'. Also, if at some point during c, Player 2 exceeds her boundaries (by going below 0 or above b at some component), then c' reaches s_{sink}, which is a winning position for Player 1. Finally, if at some point during c, the energy level of Player 1 drops below 0, then so it does in c'. Hence, in order to decide the winner in G, we can determine the winner in G'. Since the given initial-credit problem for d_1-dimensional energy game is decidable in $(d_1 - 1)-$EXPTIME [20, 7], we can decide the winner of a one-player-bounded BBEG with $j = 2$.

Now, in the truncated semantics, since there are finitely-many possible energy vectors for Player 2, we can also expand the position space to maintain them. The only difference is that when an overflow in the energy of Player 2 occurs in some components, instead of reaching the sink, the computation stays in positions that correspond to the maximum bound of those components.

We continue to the case $j = 1$, thus $b \in \mathbb{Z}^{d_1}$ is a bound vector for Player 1. We describe the construction for the losing semantics, the extension to the truncated semantics is exactly as in the $j = 2$ case.

We define the d_2-dimensional energy-reachability game $G' = \langle S_1', S_2', \langle s_{init}, x_0\rangle, E', \tau'\rangle$ as follows. Let V be the set of all non-negative vectors in $\mathbb{Z}^{d_1}$ that are bounded by b. That is, $V = \{v \in \mathbb{Z}^{d_1} \ : \ 0 \leq v[i] \leq b[i] \text{ for all } i \in [d_1]\}$. Let $S_1' = S_1 \times V$ and $S_2' = S_2 \times V$. Also, let $S = S_1' \cup S_2' \cup \{s_{sink}\}$, for some $s_{sink} \notin S_1 \cup S_2$. We now define a set of edges $E' \subseteq S' \times S'$ and a cost function $\tau' : E' \to \mathbb{Z}^{d_2}$. For all $e = \langle s, s'\rangle \in E$ and $v, v' \in V$ such that $v' = v + \tau(e)[1]$, we have the edge $e' = \langle\langle s, v\rangle, \langle s', v'\rangle\rangle$ in E' with $\tau'(e') = \tau(e)[2]$. For all $e = \langle s, s'\rangle \in E$ and $v \in V$ such that $v + \tau(e)[1] \notin V$, we have the edge $e' = \langle\langle s, v\rangle, s_{sink}\rangle$ in E' with $\tau'(e') = \tau(e)[2]$. We also have an edge $\langle s_{sink}, s_{sink}\rangle$ in E' with $\tau'(\langle s_{sink}, s_{sink}\rangle) = 0^{d_2}$. Note that the cost function τ' defines the cost for Player 2 only, while S' maintains the energy level of Player 1. In G', Player 2 wins if she can reach s_{sink}, while keeping her own energy vector non-negative. Otherwise, Player 1 wins.

By [16], one-dimensional energy-reachability games can be decided in NP∩coNP. Since we are interested in the multi-dimensional case, we give here a brief description of an algorithm that determines the winner in multi-dimensional energy-reachability games: First, note that without the energy constraints, thus in a plain reachability game played on the game graph G' with objective s_{sink}, one can compute in polynomial time the set *Attr* of winning positions for the reacher, namely for Player 2. From every position in *Attr*, Player 2 has a memoryless winning strategy, called the *attractor strategy*. Since the strategy is winning a memoryless, it includes no cycles, and so we can assume that every play that is consistent with this strategy is a simple path in the graph. Now, adding the energy constraint to the picture, we get that if Player 2 reaches a position in *Attr* with energy level that is sufficient for traversing a simple path in G' she can win by using her attractor strategy. Moreover, such a sufficient energy level can be computed, for example $|E| \cdot |W|^{d_2}$, where $|W|$ is the largest absolute value of an update, is sufficient. Hence, we can extend the position-space of G' to maintain the energy level of Player 2 (with the bound of $|E| \cdot |W|^{d_2}$), and then determine the winner of a plain reachability game on this extended graph.