Games with bound guess actions

Thomas Colcombet *
Institut de Recherche en Informatique Fondamentale,
CNRS and Université Paris Diderot
thomas.colcombet@liafa.univ-paris-diderot.fr

Stefan Göller †
Laboratoire Spécification et Vérification,
ENS de Cachan & CNRS
goeller@lsv.fr

Abstract

We introduce games with (bound) guess actions. These are games in which the players may be asked along the play to provide numbers that need to satisfy some bounding constraints. These are natural extensions of domination games occurring in the regular cost function theory. In this paper we consider more specifically the case where the constraints to be bounded are regular cost functions, and the long term goal is an ω-regular winning condition. We show that such games are decidable on finite arenas.

Categories and Subject Descriptors F.1.1 [*Models of computation*]: Computation by abstract devices

1. Introduction

The study of games with regular objectives, as well as their specialized variants (Streett, Rabin, Muller), play a key role in modern automata theory and many model-checking techniques. The success of this approach relies in the versatility of ω-regular languages. These can be used to encode many intricate phenomena, and the toolbox for solving related problems has been extensively developed. In particular, it gives immediate answers to almost any reasonable question in this context.

In recent years, the focus of the community has been more and more attracted to quantitative analysis. Indeed, it is desirable to check if a system can achieve a task, but questions arise such as "how much time does the task take" or "what is the quantity of resources the task will consume"? Several models of games and automata have been introduced for addressing such questions. The approach here is in particular based on the theory of regular cost functions.

Regular cost functions (Colcombet 2013b, 2011) offer a quantitative extension to regular languages, in which languages are replaced by functions from inputs (words, trees, ...) to $\mathbb{N} \cup \{\infty\}$. In language theoretic terms, 0 can be understood as 'in the language', while ∞ stands for 'outside the language'. The other values offer intermediates between these two extremal situations. For instance, regular cost functions can be used to measure time or count events, and indeed a rich panel of possibilities are offered for combining such quantities. The automaton models used in this context are the continuation of the ones used by Hashiguchi (Hashiguchi 1988), Leung (Leung 1988), Simon (Simon 1994), Kirsten (Kirsten 2005), as well as Bojańczyk and Colcombet (Bojańczyk and Colcombet 2009). Regular cost functions have the specificity that exact values do not really matter, and the functions are considered modulo an equivalence relation $\approx$ that allows some distortion. In exchange for this loss of precision, all the central results concerning regular languages over finite words and trees, as well as infinite words, are recovered (equivalence between logic, algebra and automata, as well as closure and decidability properties).

In this paper we extend the game theoretic framework in a context in which quantitative resource analysis is possible. The object we concretely handle are regular cost functions as well as extensions of the games studied in their resolution, such as domination games.

More precisely, the games that we consider, called games with guess actions, are finite two player zero-sum games of infinite duration, in which the winning condition involves a combination of ω-regular objectives and quantitative objectives. At some moment of such game, one of the players is required to provide a value from $\mathbb{N}$ that she or he is to assign to some register (there are a finite number of such registers). Such an action is called a "guess action". The intention is that the player promises that some quantity (explicit in the winning condition) will never exceed this bound. Both players are subject to such constraints, each of them owning several registers. The registers can also be changed several times (possibly an infinite number of times). The final winner is decided based on a global ω-regular objective that can involve constraints of the form "the promises related to register r were always fulfilled" or "the promises related to register r are fulfilled infinitely often".

We show how one can solve games with guess actions in which the above-mentioned quantity is measured using regular cost functions; we call such games regular games with guess actions.

Such kind of games arise naturally in verification. Imagine for instance some game modeling a printer. The system is played by the existential player, while the users/environment are/is simulated by the universal player. This is still standard. The novelty here is that it is possible using regular games with guess actions, as presented here, to ask the user to declare the number of pages at the time of the printing request, and then check that the system, knowing this number of pages, can guarantee the job to be performed in a bounded time. The essence of such games is the following: Both players may be required to declare quantities (number of pages, time), and then have to play in order to guarantee some measured quantity (number of pages processed/number of time steps) to

* Supported by the European Research Council Seventh Framework Programme (FP7/2007-2013) under grant agreement 259454 (GALE).

† Supported by Labex Digicosme, Univ. Paris-Saclay, project VERICONISS, and by the European Research Council Seventh Framework Programme (FP7/2007-2013) under grant agreement 259454 (GALE).

Permission to make digital or hard copies of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. Copyrights for components of this work owned by others than ACM must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists, contact the Owner/Author. Request permissions from permissions@acm.org or Publications Dept., ACM, Inc., fax +1 (212) 869-0481. Copyright 2016 held by Owner/Author. Publication Rights Licensed to ACM.

LICS '16, July 05-08, 2016, New York, NY, USA
Copyright © 2016 ACM 978-1-4503-4391-6/16/07...$15.00
DOI: http://dx.doi.org/10.1145/2933575.2934502

remain within this limit. The versatility of the model gives rise to much more complicated properties to be checked.

A novelty of this approach is that the objective that the players aim at achieving are not strictly speaking "winning conditions" in the usual sense, *i.e.*, they are not only based on a set of accepting plays. Indeed, virtually, the game is played on an arena in which the values of registers can be chosen, hence the actual arena is infinite. One (technical) difficulty of working in this framework is that one has to solve a game on a given finite arena, though the actual game that has to be solved is in fact played on an implicit infinite arena.

We solve regular games with guess actions by the following reduction steps.

1. In a first step we solve a subclass of games with guess actions on finite arenas in which the measures are given by the combinatorial core of regular cost functions, namely sequences of actions of counters (that can be incremented or reset) and that are not allowed to exceed the previosuly-assigned register value along any play. We call this subclass counter-based games with guess actions. When played on finite arenas we reduce such games to ω-regular games.
2. In a second step, we introduce the more general class of regular games with guess actions in which the measures are given by regular cost functions. We reduce these game to the counter-based case. This is the sole step in which advanced technology of regular cost functions is used by the use of history-deterministic B-automata. We use the existence of such automata as a black box.

Related work

Of course, this work is related to a wide family of games with quantitative objectives that appears in the literature. Not many of these works is, to the knowledge of the authors, close to the present work.

Phrased in our terminology (Fijalkow and Zimmermann 2012) study counter-based games with guess actions, where there is only one register whose value is only assigned in the beginning of the play and only one counter.

Further related work involves domination games that are used in the theory of regular cost functions for deciding the domination preorder for regular cost functions over trees (Colcombet and Löding 2010; Colcombet and Löding 2008; Colcombet 2013a). Such domination games can be understood as the games in this paper in which there are only two registers, and these are successively guessed in the first two rounds of the game, and then never changed anymore along the play.

It happens that alternating guesses of the two players can be understood as nested alternating quantifiers in a formula. In this sense, the games we solve here also encompass the game that should be used for solving magnitude monadic second-order logic over finite trees (a work that has been done over words using algebra (Colcombet 2013b) and that has not been extended to finite trees so far).

Structure of the document

The main definitions are given in the preliminary section Section 2. Section 3 introduces a particular class of winning conditions, namely counter-based winning conditions in which the measure is given by counter actions of B-automata. We give a reduction of counter-based games with guess actions on finite arenas to ω-regular games in Section 4. We introduce regular cost functions in Section 5 and reduce regular games with guess actions to counter-based games with guess actions. We conclude in Section 6.

2. Preliminaries

2.1 Basic definitions

Let X be any non-empty set. For all $k \in \mathbb{N}$ we denote by $X^{\leq k}$ the set of all strings over X of length at most k. For each sequence $\pi = x_0 x_1 \cdots$ over X and each subset $Y \subseteq X$ we denote by $\pi|_Y$ the (finite or infinite) sequence $y_0 y_1 \cdots$, where for each $i \geq 0$ we have $y_i = x_i$ if $x_i \in Y$ and $y_i = \varepsilon$ otherwise. Given a function $f : A \to B$ and $a \in A, b \in B$, we denote by $f[a \mapsto b]$ the mapping $f[a \mapsto b](x) = f(x)$ if $x \neq a$ and $f[a \mapsto b](x) = b$ if $x = a$.

For $i, j \in \mathbb{N}$ we denote by $[i, j]$ the set $\{i, i{+}1, \ldots, j\}$. For each finite word $w = a_1 \cdots a_n$ we denote by $w[i, j]$ the infix $a_i \cdots a_j$. For each set (possibly infinite) X let $\mathbb{B}(X)$ denote the set of infinitary boolean formulae over the set X. Similarly we define the set of positive infinitary Boolean formulae by disallowing formulas of the kind $\neg\varphi$. For each boolean formula φ, we denote by $\textsc{Var}(\varphi)$ the set of variables that appear in φ.

A *substitution* for φ to some set Y is a partial function $s : \textsc{Var}(\varphi) \to Y$. We denote by $\varphi[s]$ the infinitary boolean formula over $X \cup Y$ obtained from φ by replacing each occurrence of a variable $x \in \textsc{Var}(\varphi)$ by $s(x)$. We sometimes also write $x \leftarrow y$ to denote the substitution s with the singleton domain $\{x\}$ such that $s(x) = y$. We view truth assignments of infinitary boolean formulas φ as subsets of $\textsc{Var}(\varphi)$ and write $Y \models \varphi$, where $Y \subseteq \textsc{Var}(\varphi)$, if the truth assignment, where every variable from Y is assigned to `true` and all others to `false`, satisfies φ. If φ is positive, then $\overline{\varphi}$ denotes the dual formula of φ, i.e., the formula one obtains by interchanging $\vee$ and $\wedge$.

2.2 Standard arenas, plays, strategies and games

We begin our description with standard definitions of arenas, winning conditions, games and strategies. The definition of guess actions, specific to this work, will be the subject of the following section.

In any game in this paper, two antagonist players are confronted: Eve (the existential player) and Adam (the universal player).

An *arena* is a tuple $\mathcal{A} = (V, \mathbb{C}, \delta, v_0)$, where V is a set of *vertices*, $\mathbb{C}$ is a set of *actions* and $\delta : V \to \mathbb{B}^+(\mathbb{C} \times V)$ where $\delta(v)$ is either a non-empty disjunction or a non-empty conjunction over $\mathbb{C} \times V$ and $v_0 \in V$ is an initial vertex. By $M = \{(v, a, v') \in V \times \mathbb{C} \times V \mid (a, v') \in \textsc{Var}(\delta(v))\}$ we denote the set of *moves* in $\mathcal{A}$ and by $M(v) = \{(v, a, v') \in M \mid a \in \mathbb{C}, v' \in V\}$ the set of *moves from* v. In case $\delta(v)$ is a non-empty disjunction (resp. conjunction) we also say that v *is controlled by* Eve *(resp. by* Adam*)*.

A *partial play in* $\mathcal{A}$ of length $k \geq 0$ is a sequence of moves of the form

$$\pi = (v_0, a_1, v_1)(v_1, a_2, v_2) \cdots (v_{k-1}, a_k, v_k) \in M^* \ .$$

The *output* of the partial play, $\mathtt{output}(\pi)$ is $a_1 \ldots a_k$. We define $\mathsf{last}(\pi) = v_k$ if $\pi \neq \varepsilon$ and $\mathsf{last}(\varepsilon) = v_0$. A *play* is an infinite sequence from M^ω such that each of its finite prefixes is a partial play in $\mathcal{A}$. The function $\mathtt{output}(\pi)$ is naturally extended to plays.

An Eve-*strategy in* $\mathcal{A}$ is a set σ of partial plays with $\varepsilon \in \sigma$ such that for each $\pi \in \sigma$ with $\mathsf{last}(\pi) = v$ we have $\{(c, v') \in M(v) \mid \pi(v, c, v') \in \sigma\} \models \delta(v)$. We say that a play π is *consistent with an* Eve*-strategy* σ if all finite prefixes of π belong to σ.

The *dual* of an arena $\mathcal{A} = (V, \mathbb{C}, \delta)$ is the arena $\overline{\mathcal{A}} = (V, \mathbb{C}, \overline{\delta}, v_0)$. An Adam*-strategy* in $\mathcal{A}$ is an Eve-strategy in $\overline{\mathcal{A}}$.

A *game* is a pair $\mathcal{G} = (\mathcal{A}, W)$, where $\mathcal{A}(\mathcal{G}) = (V, \mathbb{C}, \delta, v_0)$ is an arena and W is a set of plays called the *winning condition*. A play π is *won by* Eve if $\pi \in W$, otherwise it is *won by* Adam. An Eve-strategy (resp. Adam-strategy) σ is *winning* if every play that is consistent with σ is won by Eve (resp. Adam). We say that

EVE (resp. ADAM) *wins* $\mathcal{G}$ if there exists an EVE-strategy (resp. ADAM-strategy) that is winning. The *dual game* of $\mathcal{G}$ is the game $\overline{\mathcal{G}} = (\overline{\mathcal{A}}, \overline{W})$, where $\overline{W}$ is the set of plays that are not in W. We say $\mathcal{G}$ is *ω-regular* if $\mathbb{C}$ is a finite set and for some ω-regular language $T \subseteq \mathbb{C}^\omega$ the winning condition W consists of all plays π with $\mathtt{output}(\pi) \in T$; recall that $T \subseteq \mathbb{C}^\omega$ is ω-regular if T a finite union of languages of the form $U^* V^\omega$ for regular languages $U, V \subseteq \mathbb{C}^*$.

Conventions. It will sometimes be more convenient to generalize in the definition of an arena $\mathcal{A} = (V, \mathbb{C}, \delta, v_0)$ the control function δ such that each $v \in V$ is mapped to a positive boolean combination of atoms of the form $(a_1 \cdots a_n, w)$, where $n \geq 1$, $a_i \in \mathbb{C}$ for each $i \in [1, n]$ and $w \in V$: this is an implicit notation for the introduction of fresh intermediate vertices $w_0, \ldots, w_n$ such that $w_0 = v$, $w_n = w$ and $\delta(w_{i-1}) = (a_i, w_i)$ for each $i \in [1, n]$. As a consequence, partial plays from a vertex v_0 can now be seen sequences of the form

$$(v_0, x_1, v_1)(v_1, x_2, v_2) \cdots (v_{k-1}, x_k, v_k),$$

where $x_1, \ldots, x_k \in \mathbb{C}^+$. A similar remark applies to partial plays, plays, and strategies.

2.3 Arenas and games with guess actions and their semantics

We shall now introduce games where some moves are further labeled with guess actions. When such guess actions are performed, one of the players is required to choose the new value that a register will take. In a second step, we will more precisely use these registers for comparing them with certain quantities.

Let $\text{REG} = \text{REG}_{\text{EVE}} \uplus \text{REG}_{\text{ADAM}}$ denote a finite set of *registers* that is partitioned into the *registers owned* by Eve (REG_{EVE}) and the ones owned by Adam (REG_{ADAM}). Registers range over $\mathbb{N}$. Given a register r, we use the *guess action* letter $[\mathtt{guess}\ r]$ that represents the request to the owner of r to provide a new value for it (semantics defined below). The resulting alphabet $\text{GUESS}[\text{REG}]$ is $\{[\mathtt{guess}\ r] \mid r \in \text{REG}\}$. When the 'real game' is played, the guess actions will be turned into assignments to the registers. The action of *assigning* a value $n \in \mathbb{N}$ to a register r is $[r := n]$, and the resulting alphabet $\text{ASSIGN}[\text{REG}]$ is $\{[r := n] \mid r \in \text{REG}, n \in \mathbb{N}\}$.

An *arena with guess actions* (over actions $\mathbb{C}$ and registers REG) is a tuple $\mathcal{A} = (V, \mathbb{C}, \text{REG}, \delta, v_0)$ such that $(V, \mathbb{C} \uplus \text{GUESS}[\text{REG}], \delta, v_0)$ is an arena. In other words, this is an extension of the notion of arena in which some moves may trigger guess actions. When the set REG is empty this should be understood as a normal arena. We define the semantics of arenas with guess actions by turning them into larger arenas (typically infinite), in which the guess actions are converted into assignments, by which the player owning the guessed register assigns the respective register some value from the non-negative integers.

Formally, given an arena with guess actions $\mathcal{A} = (V, \mathbb{C}, \delta, v_0)$, it induces an *explicit arena* defined as

$$\text{EXP}(\mathcal{A}) \ = \ (V, \mathbb{C} \uplus \text{ASSIGN}[\text{REG}], \delta', v_0) \ ,$$

where for all $v \in V$, $\delta'(v) = \delta(v)[s]$ in which s is the substitution with domain $\text{GUESS}[\text{REG}] \times V$ such that

$$s([\mathtt{guess}\ r], w) \ = \ \begin{cases} \bigvee_{n \in \mathbb{N}} ([r := n], w) & \text{if } r \in \text{REG}_{\text{EVE}}, \\ \bigwedge_{n \in \mathbb{N}} ([r := n], w) & \text{if } r \in \text{REG}_{\text{ADAM}}. \end{cases}$$

Hence, an arena with guess actions is nothing but a compact way to represent the arena (with infinitely many actions) $\text{EXP}(\mathcal{A})$ over $\mathbb{C} \uplus \text{ASSIGN}[\text{REG}]$. This means that a winning condition over this alphabet can turn such an arena into a game.

We introduce now a specific form of such winning conditions. It is parameterized by:

- For all $r \in \text{REG}$ a map f_r from $\mathbb{C}^*$ to $\mathbb{N}$ called a *measure*. We sometimes denote the tuple $(f_r)_{r \in \text{REG}}$ by $\mathbf{f}$.
- a set $T \subseteq (\mathbb{C} \times \{0,1\}^{\text{REG}})^\omega$ which is positive with respect to the $\{0,1\}$-components of EVE and negative with respect to those of ADAM. Formally, we order $\mathbb{C} \times \{0,1\}^{\text{REG}}$ by $(a, \mathbf{b}) \sqsubseteq (a', \mathbf{b'})$ if $b_r \geq b'_r$ for all $r \in \text{REG}_{\text{EVE}}$, and $b'_r \leq b_r$ for all $r \in \text{REG}_{\text{ADAM}}$. The order $\sqsubseteq$ is extended componentwise to $(\mathbb{C} \times \{0,1\}^{\text{REG}})^\omega$ by $u = u_0 u_1 \ldots \sqsubseteq v_0 v_1 \cdots = v$ if $u_i \sqsubseteq v_i$ for all $i \in \mathbb{N}$. Finally, T is defined to be *positive* if whenever $u \in T$ and $u \sqsubseteq v$, then $v \in T$. The language T is called the *long term objective*.

A game with guess actions is $(\mathcal{A}, (f_r)_{r \in \text{REG}}, T)$, in which $\mathcal{A}$ is an arena with guess actions over $\mathbb{C}$ and REG, f_r is a measure function for all $r \in \text{REG}$, and T is a long term objective. The semantics will be as above defined described by converting $(\mathbf{f}, T)$ into an explicit version $\text{EXP}(\mathbf{f}, T)$. The idea is that the measure functions are evaluated on the prefix of the run and its values will be compared with the currently assigned value of the register, and the resulting characteristic bit will enrich the play. The play obtained enriched by all these bits will in turn then be compared with T for deciding the winner of the game.

Let us now formally define $\text{EXP}(\mathbf{f}, T)$. Let us fix ourselves a register r. For all words $u \in (\mathbb{C} \uplus \text{ASSIGN}[\text{REG}])^\omega$ we define $\mathtt{val}_r(u)$ to be the value currently assigned to r after executing u: $\mathtt{val}_r(\varepsilon)$ is undefined, $\mathtt{val}_r(ua) = n$ if $a = [r := n]$ for some $n \in \mathbb{N}$ and $\mathtt{val}_r(ua) = \mathtt{val}_r(u)$ otherwise. We also define $\mathtt{cmp}_r(u) \in \{0,1\}$ for all $r \in \text{REG}$ to be 1 if $f_r(u|_{\mathbb{C}}) > \mathtt{val}_r(u)$ and 0 otherwise. Given a finite word u over $\mathbb{C} \uplus \text{ASSIGN}[\text{REG}]$, let $\mathtt{Cmp}(u)$ be the word over $\mathbb{C} \times \{0,1\}^{\text{REG}}$ defined by $\mathtt{Cmp}(\varepsilon) = \varepsilon$, $\mathtt{Cmp}(v[r := n]) = \mathtt{Cmp}(v)$ for all registers r and all $n \in \mathbb{N}$, and finally $\mathtt{Cmp}(va)$ to be $\mathtt{Cmp}(v)(a, (\mathtt{cmp}_r(va))_{r \in \text{REG}})$ otherwise. This is extended naturally to infinite sequences by limit passing. Note that the result may be a finite word if the original infinite word does only contain finitely many actions from $\mathbb{C}$. Therefore we make the following conventions on arenas with guess actions, whose first point avoids the previously mentioned unwanted finiteness and whose second point gurantees that in $\text{EXP}(\mathcal{G})$ all registers have been assigned a value (i.e $\mathtt{val}_r$ is well-defined) for any partial play π which contains least one move from $V \times \mathbb{C} \times V$ (i.e. $|\mathtt{output}(\pi)|_{\mathbb{C}} \geq 1$).

Conventions. Without loss of generality we make the following assumptions on any arena with guess actions $\mathcal{A} = (V, \mathbb{C}, \text{REG}, \delta, v_0)$:

- There is no play π in the arena $(V, \mathbb{C} \cup \text{GUESS}[\text{REG}], \delta, v_0)$ in which $\mathtt{output}(\pi)|_{\mathbb{C}}$ is finite.
- For all partial plays π in the arena $(V, \mathbb{C} \cup \text{GUESS}[\text{REG}], \delta, v_0)$ with $|\mathtt{output}(\pi)|_{\mathbb{C}}| \geq 1$ we have $|\mathtt{output}(\pi)|_{[\mathtt{guess}\ r]}| \geq 1$ for all $r \in \text{REG}$.

We can finally define the winning condition $\text{EXP}(\mathbf{f}, T)$ as:

$$\text{EXP}(\mathbf{f}, T) = \{u \in (\mathbb{C} \uplus \text{ASSIGN}[\text{REG}])^\omega \mid \mathtt{Cmp}(u) \in T\} \ .$$

Hence, given any game with guess actions $\mathcal{G} = (\mathcal{A}, \mathbf{f}, T)$, one can turn it into the game $\text{EXP}(\mathcal{G}) = (\text{EXP}(\mathcal{A}), \text{EXP}(\mathbf{f}, T))$. We define the *winner of a game with guess action* $\mathcal{G}$ to be the winner of $\text{EXP}(\mathcal{G})$.

In this paper, we consider a specific form of such games, namely *regular games with guess actions*: these are games with guess actions in which measures are regular cost functions (to be defined in Section 5.1), and the long term objective is ω-regular.

The *dual of an arena with guess action* $\mathcal{A}$ is $\overline{\mathcal{A}}$ obtained by (1) exchanging conjunctions and disjunctions like when computing the dual of an arena (2) swapping the owner of registers. The *dual of game with guess action* $\mathcal{G} = (\mathcal{A}, \mathbf{f}, T)$ is $\overline{\mathcal{G}} = (\overline{\mathcal{A}}, \mathbf{f}, \overline{T})$ and it is readily seen that $\text{EXP}(\overline{\mathcal{G}}) = \overline{\text{EXP}(\mathcal{G})}$.

3. Counter-based winning conditions

Let us consider a fixed set of *counters* Γ (whose elements are typically denoted by γ) that can take values ranging over $\mathbb{N}$ (these counters should not be confused with registers). The counter alphabet $\mathbb{C} = \mathtt{ACt}_\Gamma$ is the disjoint union of the alphabets $\mathtt{ACt}_\gamma$ for $\gamma \in \Gamma$, in which $\mathtt{ACt}_\gamma = \{\varepsilon_\gamma, \mathtt{r}_\gamma, \mathtt{ic}_\gamma\}$ for all counters γ in Γ. The letter $\mathtt{r}_\gamma$ is called a *reset* of counter γ, while $\mathtt{ic}_\gamma$ is an *increment* of counter γ. This is formalized through the map count_γ from words to integers that tracks the value of a counter γ along a sequence of actions. Formally, $\mathrm{count}_\gamma(\varepsilon) = \mathrm{count}_\gamma(u\,\mathtt{r}_\gamma) = 0$, $\mathrm{count}_\gamma(u\,\mathtt{ic}_\gamma) = \mathrm{count}_\gamma(u) + 1$, and $\mathrm{count}_\gamma(u\,a) = \mathrm{count}_\gamma(u)$ otherwise.

We define now the function $\mathsf{cost}_\Gamma : \mathtt{ACt}_\Gamma^* \to \mathbb{N}$ as the maximal value assumed by any counter at any moment:

$$\mathsf{cost}_\Gamma(w) = \max_{\gamma\in\Gamma,\ u \text{ prefix of } w} \mathrm{count}_\gamma(u).$$

We define now what *counter-based games with guess actions* are; they are given as $(\mathcal{A}, (\mathsf{cost}_{\Gamma_r})_{r\in\mathrm{REG}}, T)$, in particular $\mathbb{C} = \mathtt{ACt}_\Gamma$. These games form a special case of regular games with guess actions, and are special in two ways. First the measures are of the form $(\mathsf{cost}_{\Gamma_r})_{r\in\mathrm{REG}}$ where Γ_r is a finite set of counters attached to each register $r \in \mathrm{REG}$. Second, it is required that T is ω-regular and also of a special form: as soon as a counter exceeds the value of the corresponding register, then the owner of the register immediately loses. This second point is made formal by requiring that for all $u = (a_0, \mathbf{b^0})(a_1, \mathbf{b^1})\ldots$,

- T is ω-regular,
- if the least $\mathbf{i}$ such that $\mathbf{b^i} \neq \mathbf{0}$ exists, and $b^i_r = 1$ for some $r \in \mathrm{REG}_{\mathrm{ADAM}}$, then $u \in T$, and
- if the least $\mathbf{i}$ such that $\mathbf{b^i} \neq \mathbf{0}$ exists, and $b^i_r = 1$ for some $r \in \mathrm{REG}_{\mathrm{EVE}}$, then $u \notin T$.

Note that the two last two above items may be 'conflicting' in case the vector $\mathbf{b^i}$ is such that $b^i_r = 1$ for both some $r \in \mathrm{REG}_{\mathrm{ADAM}}$ and some $r \in \mathrm{REG}_{\mathrm{EVE}}$, hence the last two items might not necessarily be exclusive in general. In fact, this situation cannot occur. Indeed, each counter is associated to a single register, and only one counter can be modified at a time by an actions in $\mathtt{ACt}_\Gamma$.

4. On finite arenas: From counter-based games with guess actions to ω-regular games

For the rest of this section, we fix a finite game with guess actions with a counter-based winning condition $\mathcal{G} = (\mathcal{A}, (\mathsf{cost}_{\Gamma_r})_{r\in\mathrm{REG}}, T)$, i.e. $\mathcal{A} = (V, \mathbb{C}, \mathrm{REG}, \delta, v_0)$ is an arena with guess actions over $\mathbb{C}$ and REG for some finite set of vertices V. We will translate $\mathcal{G}$ into an ω-regular game $\mathrm{Imp}(\mathcal{G})$ such that EVE wins $\mathrm{EXP}(\mathcal{G})$ if, and only if, EVE wins $\mathrm{Imp}(\mathcal{G})$ (Theorem 4.2). In Section 4.1 we formally define the game $\mathrm{Imp}(\mathcal{G})$. Finite-memory strategies, whose existence ω-regular games enjoy, are recalled in Section 4.2. In Section 4.3 we translate winning strategies in $\mathrm{Imp}(\mathcal{G})$ to winning strategies in $\mathrm{EXP}(\mathcal{G})$. Finally, we prove Theorem 4.2 in Section 4.4.
Notational convention. Recall that the functions cost_{Γ_r} map words from $\mathtt{ACt}_{\Gamma_r}$ to $\mathbb{N}$. However, given a play (resp. partial play) π it turns out to be more convenient to write $\mathsf{cost}_{\Gamma_r}(\pi)$ instead of $\mathsf{cost}_{\Gamma_r}(\mathtt{output}(\pi)|_{\Gamma_r})$. Without risk of confusion, similar remarks apply to other functions that we would like to apply to plays (resp. partial plays) rather than to a restriction of their output.

4.1 Definition of $\mathrm{Imp}(\mathcal{G})$

The ω-regular game $\mathrm{Imp}(\mathcal{G})$ is defined as

$$\mathrm{Imp}(\mathcal{G}) = (\mathrm{Imp}(\mathcal{A}), W((\mathsf{cost}_{\Gamma_r})_{r\in\mathrm{REG}}, T)),$$

where

- $\mathrm{Imp}(\mathcal{A}) = (V, \mathbb{C} \cup \mathrm{GUESS}[\mathrm{REG}], \delta, v_0)$, i.e. the symbols $\mathrm{GUESS}[\mathrm{REG}]$ appear as additional usual action symbols.

For defining $W((\mathsf{cost}_{\Gamma_r})_{r\in\mathrm{REG}}, T)$ we assign to all plays π in $\mathrm{Imp}(\mathcal{G})$ the set of *exceeding registers* $\mathrm{EXCEEDREG}(\pi)$, formally defined to be the set of registers $r \in \mathrm{REG}$ such that

- letter $[\mathtt{guess}\ r]$ appears only finitely often in $\mathtt{output}(\pi)$ and
- there exists some $\gamma \in \Gamma_r$ such that
  - $\mathtt{ic}_\gamma$ appears infinitely often in $\mathtt{output}(\pi)$ and
  - $\mathtt{r}_\gamma$ appears finitely often in $\mathtt{output}(\pi)$.

The winning condition $W =^{\mathrm{def}} W((\mathsf{cost}_{\Gamma_r})_{r\in\mathrm{REG}}, T)$ consists of all plays π in $\mathrm{Imp}(\mathcal{G})$ that satisfy at least one of the following two properties, either

- $\mathrm{EXCEEDREG}(\pi) \neq \emptyset$ and there exists some register $r \in \mathrm{EXCEEDREG}(\pi) \cap \mathrm{REG}_{\mathrm{ADAM}}$ such that in $\mathtt{output}(\pi)$ the last occurrence of $[\mathtt{guess}\ r]$ is before the last occurrence of $[\mathtt{guess}\ r']$ for all $r' \in \mathrm{EXCEEDREG}(\pi) \setminus \{r\}$, or
- $\mathrm{EXCEEDREG}(\pi) = \emptyset$ and $\mathtt{output}(\pi)|_{\mathbb{C}} \otimes \{0\}^{\mathrm{REG}}$ is in $\mathrm{EXP}((\mathsf{cost}_{\Gamma_r})_{r\in\mathrm{REG}}, T)$, where for any word $u \in \mathbb{C}^\omega$ we denote by $u \otimes \{0\}^{\mathrm{REG}}$ the unique word from $(\mathbb{C} \times \{0\}^{\mathrm{REG}})^\omega$ whose projection onto $\mathbb{C}$ yields u.

It is easy to verify that W is indeed ω-regular. The proof of the following lemma follows easily from the definition of $\mathrm{Imp}(\mathcal{G})$.

Lemma 4.1. $\mathrm{Imp}(\overline{\mathcal{G}}) = \overline{\mathrm{Imp}(\mathcal{G})}$.

For the rest of this section we concern ourselves with the proof of the following theorem.

Theorem 4.1. *If* EVE *wins* $\mathrm{Imp}(\mathcal{G})$*, then* EVE *wins* $\mathrm{EXP}(\mathcal{G})$.

Our main theorem of this section is a consequence of Theorem 4.2 and Lemma 4.1.

Theorem 4.2. EVE *wins* $\mathrm{Imp}(\mathcal{G})$ *if, and only if,* EVE *wins* $\mathrm{EXP}(\mathcal{G})$.

Proof. Since ω-regular games are determined (i.e. every ω-regular game is won by some player) it suffices to prove that if ADAM wins $\mathrm{Imp}(\mathcal{G})$, then ADAM wins $\mathrm{EXP}(\mathcal{G})$:

$$\begin{array}{lcl}
\text{ADAM wins } \mathrm{Imp}(\mathcal{G}) & \Longrightarrow & \text{EVE wins } \overline{\mathrm{Imp}(\mathcal{G})} \\
& \overset{\text{Lemma 4.1}}{\Longrightarrow} & \text{EVE wins } \mathrm{Imp}(\overline{\mathcal{G}}) \\
& \overset{\text{Theorem 4.1}}{\Longrightarrow} & \text{EVE wins } \mathrm{EXP}(\overline{\mathcal{G}}) \\
& \Longrightarrow & \text{EVE wins } \overline{\mathrm{EXP}(\mathcal{G})} \\
& \Longrightarrow & \text{ADAM wins } \mathrm{EXP}(\mathcal{G}) \quad \square
\end{array}$$

In order to prove Theorem 4.1, we need to introduce finite-memory strategies in the ω-regular game $\mathrm{Imp}(\mathcal{G})$.

4.2 Finite memory strategies

Recall $\mathrm{Imp}(\mathcal{A}) = (V, \mathbb{C} \cup \mathrm{GUESS}[\mathrm{REG}], \delta, v_0)$. Let M denote the set of moves of $\mathrm{Imp}(\mathcal{A})$.

A strategy σ for $\mathrm{Imp}(\mathcal{G})$ is called *finite-memory* if there is a tuple (Z, z_0, ξ, ℓ), where

- Z is a finite set (called the *memory*) and $z_0 \in Z$,
- $\xi : V \times Z \to 2^M$ assigns to all pairs of vertices and memory elements a set of moves,
- $\ell : Z \times M \to Z$ is a function that updates the memory that is inductively extended to finite sequences over M as follows: $\ell(z, \varepsilon) = z$ and $\ell(z, \pi m) = \ell(\ell(z, \pi), m)$ for all $\pi \in M^*$, $m \in M$ and $z \in Z$, and

- $\xi(\mathsf{last}(\pi), \ell(z_0, \pi)) = \{m \in M \mid \pi m \in \sigma\}$, hence in particular $\xi(\mathsf{last}(\pi), \ell(z_0, \pi)) \models \delta(\mathsf{last}(\pi))$ for all partial plays π in $\mathrm{Imp}(\mathcal{G})$.

Theorem 4.3. *(Gurevich and Harrington 1982) The winner of any ω-regular game has a winning finite-memory strategy.*

Thus, we can assume that if EVE has a winning strategy in $\mathrm{Imp}(\mathcal{G})$ she has a winning finite-memory strategy. The following Lemma will be particularly useful and follows immediately from the definition of finite-memory strategies.

Lemma 4.2. *Let σ be a finite-memory strategy witnessed by the tuple (Z, z_0, ξ, ℓ), let $\pi = (v_0, a_1, v_1)(v_1, a_2, v_2)\cdots$ be a play that is consistent with σ and assume there exist $0 \leq i < j$ such that $v_i = v_j$ and $\ell(z_0, \pi[1,i]) = \ell(z_0, \pi[1,j])$. Then the play*

$$\pi[1,i](\pi[i+1,j])^\omega$$

is also consistent with σ.

4.3 From a winning EVE-strategy σ in $\mathrm{Imp}(\mathcal{G})$ to a winning EVE-strategy TRANS(σ) in EXP$(\mathcal{G})$

Let us fix an arbitrary winning strategy σ for EVE in $\mathrm{Imp}(\mathcal{G})$. By Theorem 4.3 we may assume that σ is finite-memory, given by the tuple (Z, z_0, ξ, ℓ), say.

First, let us define the following function SIMPLY from moves in EXP$(\mathcal{A})$ to moves in $\mathrm{Imp}(\mathcal{A})$ that acts as the identity on moves labeled by $\mathbb{C}$ and makes moves with an action from ASSIGN[REG] "implicit" by replacing it with the move with action corresponding symbol in GUESS[REG]. Formally we define

$$\textsc{Simply}(m) = \begin{cases} (v, [\mathtt{guess}\ r], v') & \text{if } m = (v, [r := n], v') \\ & \in V \times \textsc{Assign}[\textsc{Reg}] \times V \\ m & \text{otherwise.} \end{cases}$$

The function SIMPLY is naturally extended to a morphism from plays (resp. partial plays) in EXP$(\mathcal{A})$ to plays (resp. partial plays) in $\mathrm{Imp}(\mathcal{A})$. The following fact follows immediately from definition of SIMPLY.

Fact 4.1. *The following holds for all plays (resp. partial plays) π in* EXP$(\mathcal{A})$*:*

(i) SIMPLY(π) *is a play (resp. partial play) in* $\mathrm{Imp}(\mathcal{A})$ *and*

(ii) $\mathtt{output}(\pi)$ *and* $\mathtt{output}(\textsc{Simply}(\pi))$ *are the same words if we replace in* $\mathtt{output}(\pi)$ *letters of the form* $[r := n]$ *by* $[\mathtt{guess}\ r]$.

Note that for every play (resp. partial play) π in EXP$(\mathcal{G})$ there is a corresponding play (resp. partial play) in $\mathrm{Imp}(\mathcal{G})$, namely SIMPLY$(\pi)$. Conversely, note that for every play π in $\mathrm{Imp}(\mathcal{A})$ there is a *set of (resp. partial) plays* in EXP$(\mathcal{G})$ that corresponds to π, nameley SIMPLY$^{-1}(\pi)$.

Furthermore, one can easily prove that SIMPLY$^{-1}(\sigma)$ is indeed an EVE-strategy in EXP$(\mathcal{A})$, immediately by the definition of $\mathrm{Imp}(\mathcal{G})$ and of SIMPLY. However, SIMPLY$^{-1}(\sigma)$ is not necessarily a winning strategy since SIMPLY$^{-1}(\sigma)$ contains (possibly often) the assignment $[r := n]$ for all possible values $n \in \mathbb{N}$ for all $r \in \textsc{Reg}_{\textsc{Eve}}$, possibly also assignments that are not sufficiently large in order to satisfy the counter-based winning condition EXP$((\mathsf{cost}_{\Gamma_r})_{r\in\textsc{Reg}}, T)$.

Therefore, in order to define a winning strategy for EVE in EXP$(\mathcal{G})$ we must assign to EVE-registers sufficiently large values such that she does not eventually violate the counter-based winning condition. To provide EVE with a sufficiently large such bound we prove the existence of a function R that assigns to every partial play π in EXP$(\mathcal{A})$ a sufficiently large value $R(\pi) \in \mathbb{N}$. The choice of R involves a Ramsey-like argument that we make explicit later in the proof and not already when defining the strategy in EXP$(\mathcal{G})$ we claim to be winning for EVE.

To this end we define a function TRANS that assigns to every partial play π in $\mathrm{Imp}(\mathcal{G})$ a set of partial plays TRANS(π) in EXP$(\mathcal{G})$. We define TRANS inductively as follows,

- TRANS$(\varepsilon) = \{\varepsilon\}$ and
- TRANS$(\pi m) =$ TRANS$(\pi) \cdot X_m$, where

$$X_m = \begin{cases} \{(v, a, v')\} & \text{if } a \in \mathbb{C} \\ \{(v, [r := n], v') \mid n \in \mathbb{N}\} & \text{if } a = [\mathtt{guess}\ r] \\ & \text{for some } r \in \textsc{Reg}_{\textsc{Adam}} \\ \{(v, [r := R(\pi)], v')\} & \text{if } a = [\mathtt{guess}\ r] \\ & \text{for some } r \in \textsc{Reg}_{\textsc{Eve}} \end{cases}$$

for all $\pi m \in \sigma$ with $m = (v, a, v') \in M$.

The following lemma is a simple consequence of the definition of EXP$(\mathcal{A})$, of SIMPLY and of TRANS.

Lemma 4.3. *We have $\sigma =$ SIMPLY(TRANS(σ)) and hence* TRANS(σ) *is an* EVE*-strategy in* EXP$(\mathcal{A})$.

4.4 The strategy TRANS(σ) is winning.

Towards a contradiction let us assume that TRANS(σ) is not a winning strategy for EVE in the game EXP$(\mathcal{G})$. Hence there exists a play $\pi \notin$ EXP$(\mathcal{A})$ that is compatible with TRANS(π) that is winning for ADAM. We make a case distinction.

Case A: $\mathtt{Cmp}(\pi) \in (\mathbb{C} \times \{0\}^{\textsc{Reg}})^\omega$ i.e. that none of the players ever exceeds any of its measures along the play π.

We first claim that EXCEEDREG(SIMPLY(π)) $= \emptyset$. By contradiction assume some $r \in$ EXCEEDREG(SIMPLY(π)). Then

(1) letter $[\mathtt{guess}\ r]$ appears only finitely often in the sequence $\mathtt{output}(\textsc{Simply}(\pi))$ and

(2) there exists some counter $\gamma \in \Gamma_r$ such that

   (a) $\mathtt{ic}_\gamma$ appears infinitely often in $\mathtt{output}(\textsc{Simply}(\pi))$ and

   (b) $\mathtt{r}_\gamma$ appears only finitely often in $\mathtt{output}(\textsc{Simply}(\pi))$.

Firstly, by Point (ii) of Fact 4.1 the same properties (1) and (2) hold for $\mathtt{output}(\pi)$. This implies that for every $m \in \mathbb{N}$ there exists a finite prefix π_m of π such that $\mathsf{cost}_{\Gamma_r}(\pi_m) \geq m$, immediately from definition of cost_{Γ_r}. Secondly, since we have that $\mathtt{output}(\textsc{Simply}(\pi))$ contains only finitely many occurrences of $[\mathtt{guess}\ r]$ it follows that $\mathtt{output}(\pi)$ contains only finitely many occurrences of letters from $\{[r := n] \mid n \in \mathbb{N}\}$ by Point (ii) of Fact 4.1. Taking these former two properties together we conclude $\mathtt{Cmp}(\pi) \notin (\mathbb{C} \times \{0\}^{\textsc{Reg}})^\omega$, contradicting our assumption $\mathtt{Cmp}(\pi) \in (\mathbb{C} \times \{0\}^{\textsc{Reg}})^\omega$.

Hence we have shown that EXCEEDREG(SIMPLY(π)) $= \emptyset$. Since π is consistent with TRANS(σ) recall that the play SIMPLY(π) is consistent with σ by Lemma 4.3. By our assumption σ is winning. Hence due to EXCEEDREG(SIMPLY(π)) $= \emptyset$ we must have that $\mathtt{output}(\textsc{Simply}(\pi))|_{\mathbb{C}} \otimes \{0\}^{\textsc{Reg}}$ is in the set EXP$((\mathsf{cost}_{\Gamma_r})_{r\in\textsc{Reg}}, T)$ by definition of W. Hence

$$\begin{aligned} \mathtt{Cmp}(\pi) &\overset{\text{Assumption}}{=} \mathtt{output}(\pi)|_{\mathbb{C}} \otimes \{0\}^{\textsc{Reg}} \\ &\overset{\text{Fact 4.1,(ii)}}{=} \mathtt{output}(\textsc{Simply}(\pi))|_{\mathbb{C}} \otimes \{0\}^{\textsc{Reg}} \end{aligned}$$

and therefore $\mathtt{Cmp}(\pi) \in$ EXP$((\mathsf{cost}_{\Gamma_r})_{r\in\textsc{Reg}}, T)$ implying that π is a play in EXP$(\mathcal{G})$ that is won by EVE, a contradiction.

Case B: $\mathtt{Cmp}(\pi) \notin (\mathbb{C} \times \{0\}^{\textsc{Reg}})^\omega$, i.e. there is at least one moment along the play π in which some player exceeds one of his/her measures.

Since by assumption π is won by ADAM the first such an exceed must involve the measure of a register of EVE. Hence there must exist a finite prefix

$$\pi[1,k] = (v_0, a_1, v_1) \cdots (v_{k-1}, a_k, v_k)$$

of π such that

B1 $\texttt{Cmp}(\pi[1,k-1]) \in (\mathbb{C} \times \{0\}^{\text{REG}})^*$ and

B2 $\texttt{Cmp}(\pi[1,k]) \in (\mathbb{C} \times \{0\}^{\text{REG}})^*(a_k, (b_r)_{r\in\text{REG}})$, where we have $b_{r_0} = 1$ for some $r_0 \in \text{REG}_{\text{EVE}}$.

Point B2 implies that there exists a last moment $h \in [1,k]$ in which the register r_0 was assigned some value by EVE that is strictly less than $\texttt{cost}_{\Gamma_{r_0}}(\pi[1,k])$. Formally there exists some $h \in [1,k]$ such that

B3 $a_h = [r_0 := n_0]$, with $n_0 < \texttt{cost}_{\Gamma_{r_0}}(\pi[1,k])$ and $n_0 = R(\pi[1,h])$, and

B4 $a_i \neq [r_0 := N]$ for all $N \in \mathbb{N}$ and all $i \in [h+1,k]$.

We make the value $R(\pi[1,h])$ explicit in the proof later. For simplicity, let $\widehat{\pi}$ be a shortcut for $\text{SIMPLY}(\pi)$. In particular,

$$\widehat{\pi}[1,k] = (v_0, \widehat{a_1}, v_1) \cdots (v_{k-1}, \widehat{a_k}, v_k),$$

$(v_{i-1}, \widehat{a_i}, v_i) = \text{SIMPLY}(v_{i-1}, a_i, v_i)$ for all $i \in [1,k]$. Recall that the EVE-strategy σ is finite-memory witnessed by the tuple (Z, z_0, ξ, ℓ). Moreover, let

$$z_i = \ell(z_0, \widehat{\pi}[1,i])$$

for each $i \in [1,k]$ be the memory information that is assigned to all the prefixes of $\widehat{\pi}[1,k]$. We recall that if there exist $h < i < j < k$ with $v_i = v_j$ and $z_i = z_j$, then

$$\widehat{\pi}[1,i](\widehat{\pi}[i+1,j])^\omega$$

is a play that is consistent with σ by Lemma 4.2.

To establish the desired contradiction to the assumption that the strategy $\text{TRANS}(\sigma)$ is winning for ADAM we show that we could have defined the value $n_0 = R(\pi[1,h])$ so large that there are indices $h < i_0 < j_0 < k$ such that $\widehat{\pi}[1,i_0](\widehat{\pi}[i_0+1,j_0])^\omega$ is actually a play in $\text{Imp}(\mathcal{G})$ that is compatible with σ that is actually winning for ADAM, hence contradicting our assumption that σ is a winning strategy. More precisely, we shall prove the following claim.

Claim (★). There exist two indices $h < i_0 < j_0 < k$ such that

C1 $v_{i_0} = v_{j_0}$ and $z_{i_0} = z_{j_0}$, thus

$$\rho =^{\text{def}} \widehat{\pi}[1,i_0](\widehat{\pi}[i_0+1,j_0])^\omega$$

is consistent with σ,

C2 $r_0 \in \text{EXCEEDREG}(\rho)$, and

C3 for all $r \in \text{EXCEEDREG}(\rho) \cap \text{REG}_{\text{ADAM}}$ we have that $[\texttt{guess}\ r]$ appears in $\texttt{output}(\rho)$ after position h.

Before proving the above claim, let us show that it indeed contradicts our assumption that σ is a winning strategy. By C1 we have that ρ is a play that is consistent with σ. By C2 it follows that $\text{EXCEEDREG}(\rho) \neq \emptyset$. Condition C3 implies $\rho \notin W$, contradicting that σ is indeed a winning strategy.

Thus it remains to prove Claim (★). Let us give some intuition why Claim (★) holds once we have chosen $n_0 = R(\pi[1,h])$ sufficiently large. Clearly, the larger we have chosen n_0 the bigger k has to be, since it is at position k at which $\texttt{cost}_{\Gamma_{r_0}}(\pi[1,k])$ exceeds the value n_0 of register r_0 assigned at position h. This means that $\texttt{output}(\pi[1,k])$ and still $\texttt{output}(\pi[h+1,k-1])$ must contain a huge number of occurrences of some letter $\texttt{ic}_{\gamma_0}$, where $\gamma_0 \in \Gamma_{r_0}$. Moreover, in the long word $\texttt{output}(\pi[h+1,k-1])$ every register $r \in \text{REG}_{\text{ADAM}}$ and every counter $\gamma \in \Gamma_r$ either has the property that (i) $\texttt{r}_\gamma$ appears very frequently, or (ii) $\texttt{r}_\gamma$ appears rarely, but then there cannot be any long $\texttt{r}_\gamma$-free infix $\pi[i,j]$ in which $\texttt{ic}_\gamma$ appears often unless r is guessed a new value in the meanwhile, meaning that some action $[x := n]$ has to appear in $\texttt{output}(\pi[h+1,j])$ or equivalently $[\texttt{guess}\ r]$ has to appear in $\texttt{output}(\widehat{\pi}[h+1,j])$, for otherwise the measure $\texttt{cost}_{\Gamma_r}$ would have already exceeded the current value of register r earlier than at position k, thus contradicting B1. We will prove this combinatorial intuition by applying Ramsey's Theorem.

Counters summaries. Recall that $\texttt{ACt}_\gamma = \{\varepsilon_\gamma, \texttt{r}_\gamma, \texttt{ic}_\gamma\}$ and $\Gamma = \bigcup\{\Gamma_r \mid r \in \text{REG}\}$. For defining R we color every non-empty infix $\pi[i,j]$ of $\widehat{\pi}[h+1,k]$ by an information that summarizes by one symbol from $\texttt{ACt}_\gamma$ the sequence of counter actions seen in $\texttt{output}(\pi[i,j])|_{\text{Act}_\gamma}$ for each counter $\gamma \in \Gamma$. This information should summarize what the "dominating" counter action is when assuming that $\widehat{\pi}[1,i](\widehat{\pi}[i+1,j])^\omega$ is a play compatible with σ. For instance if $\texttt{output}(\pi[i+1,j])|_{\text{Act}_\gamma} = \texttt{ic}_\gamma\varepsilon_\gamma\texttt{ic}_\gamma$, then its summary would just be $\texttt{ic}_\gamma$; if $\texttt{output}(\pi[i,j])|_{\text{Act}_\gamma}$ contains at least one $\texttt{r}_\gamma$, then its summary would be $\texttt{r}_\gamma$; and if $\texttt{output}(\pi[i,j])|_{\text{Act}_\gamma}$ contains no occurrence of $\texttt{ic}_\gamma$ nor of $\texttt{r}_\gamma$, its summary would just be ε_γ. Let us make this intuition formal.

For all $\gamma \in \Gamma$, let $\mathbb{M}_\gamma = (\texttt{ACt}_\gamma, \odot_\gamma, \varepsilon_\gamma)$ denote the unique finite commutative monoid in which all elements are idempotents and where $\texttt{ic}_\gamma \odot_\gamma \texttt{r}_\gamma = \texttt{r}_\gamma \odot_\gamma \texttt{ic}_\gamma = \texttt{r}_\gamma$. Let $h_\gamma : (\text{Act}_\gamma^*, \cdot, \varepsilon) \to \mathbb{M}_\gamma$ denote the unique monoid morphism that satisfies $h_\gamma(a) = a$ for all $a \in \texttt{ACt}_\gamma$. The following lemma follows immediately from definition of $\mathbb{M}_\gamma$.

Lemma 4.4. *Let $\gamma \in \Gamma$ and let $u \in \texttt{ACt}_\gamma^*$. Then*

$$h_\gamma(u) = \begin{cases} \texttt{r}_\gamma & \text{if, and only if, } |u|_{\texttt{r}_\gamma} \geq 1, \\ \texttt{ic}_\gamma & \text{if, and only if, } |u|_{\texttt{r}_\gamma} = 0 \text{ and } |u|_{\texttt{ic}_\gamma} \geq 1, \text{ and} \\ \varepsilon_\gamma & \text{if, and only if, } |u|_{\texttt{r}_\gamma} = |u|_{\texttt{ic}_\gamma} = 0 \end{cases}$$

In fact, we will color certain infixes of $\widehat{\pi}[h+1,k-1]$ with some element from the larger monoid $\mathbb{M} = (\prod_{\gamma\in\Gamma} \mathbb{M}_\gamma, \odot, \boldsymbol{\varepsilon})$, where $\boldsymbol{\varepsilon} = (\varepsilon_\gamma)_{\gamma\in\Gamma}$ and $\odot$ is defined componentwise. We define the monoid morphism $h_{\mathbb{M}} : (\texttt{ACt}_\Gamma^*, \cdot, \varepsilon) \to \mathbb{M}$ as $h_{\mathbb{M}}(u) = (h_\gamma(u|_{\texttt{ACt}_\gamma}))_{\gamma\in\Gamma}$.

Still, before finally defining the function R we need to recall (a special case of) Ramsey's Theorem that we state in terms of our purposes.

Theorem 4.4 (Ramsey's Theorem)**.** *Let $n \geq 1$ and let $\mathbb{D}$ be a finite set (of colors). There exists a natural number $R_{\mathbb{D}}(n)$ such that for all sets I with $|I| \geq R_{\mathbb{D}}(n)$ and all colorings $\chi : \binom{I}{2} \to \mathbb{D}$ of two-element subsets of I there exists a subset $J \subseteq I$ and some color $d \in \mathbb{D}$ such that $|J| = n$ and $\chi\binom{J}{2} = \{d\}$.*

We are now ready to define our function R. For every partial play ψ we define (recall that $\texttt{val}_r(\psi)$ denotes the current value of register r after playing ψ)

$$\begin{aligned} R(\psi) \quad = \quad & |\psi| + 2 + (|V|+1)\cdot(|Z|+1) \\ & \cdot R_{\mathbb{M}}(\max\{\texttt{val}_r(\psi) \mid r \in \text{REG}\} + 1)\,. \end{aligned}$$

Thus, by B3 we have

$$\begin{aligned} n_0 \quad = \quad & h + 2 + (|V|+1)\cdot(|Z|+1) \\ & \cdot R_{\mathbb{M}}(\max\{\texttt{val}_r(\pi[1,h]) \mid r \in \text{REG}\} + 1). \end{aligned}$$

Let us prepare the proof of Claim (★). Recall that by B3 we have $a_h = [r_0 := n_0]$ with $n_0 < \texttt{cost}_{\Gamma_{r_0}}(\pi[1,k])$. It follows from the definition of $\texttt{cost}_{\Gamma_{r_0}}$ that there exists some $\gamma_0 \in \Gamma_{r_0}$ and some $1 \leq m' \leq n' \leq k$ such that

- $|\texttt{output}(\pi[m',n'])|_{\texttt{r}_{\gamma_0}} = 0$ and
- $|\texttt{output}(\pi[m',n'])|_{\texttt{ic}_{\gamma_0}} = \texttt{cost}_{\Gamma_{r_0}}(\pi[1,k]) > n_0$.

Thus, there exists some some $h < m < n < k$ such that

B5 $|\texttt{output}(\pi[m,n])|_{\mathtt{r}_{\gamma_0}} = 0$ and

B6 $|\texttt{output}(\pi[m,n])|_{\mathtt{ic}_{\gamma_0}} > n_0 - (h+2)$.

Let $O = \{j \in [m,n] \mid a_j = \mathtt{ic}_{\gamma_0}\}$ and thus

$$\begin{aligned}|O| &> n_0 - (h+2)\\ &= (|V|+1)\cdot(|Z|+1)\\ &\qquad \cdot R_{\mathbb{M}}(\max\{\mathtt{val}_r(\pi[1,h]) \mid r \in \textsc{Reg}\}+1).\end{aligned}$$

By the pigeonhole principle there exists some subset $I \subseteq O$, some $v \in V$, and some $z \in Z$ such that

B7 $v_i = v$ and $z_i = z$ for all $i \in I$ and

B8 $|I| > R_{\mathbb{M}}(\max\{\mathtt{val}_r(\pi[1,h]) \mid r \in \textsc{Reg}\}+1)$.

Let $\chi : \binom{I}{2} \to \mathbb{M}$ denote the coloring where

$$\chi\{i,j\} = h_{\mathbb{M}}(\pi[i,j]) \text{ for all } i,j \in I \text{ with } i < j.$$

By Ramsey's Theorem there exists a subset $J \subseteq I$ and some $\boldsymbol{m} \in \mathbb{M}$ such that

B9 $\chi\binom{J}{2} = \{\boldsymbol{m}\}$, i.e. $h_{\mathbb{M}}(\pi[i,j]) = \boldsymbol{m}$ for all $i,j \in J$ s.t. $i<j$,

B10 $|J| = \max\{\mathtt{val}_r(\pi[1,h]) \mid r \in \textsc{Reg}\}+1$.

We are now ready to prove to prove Claim (★). We choose $i_0 = \min J$ and $j_0 = \max J$ and note that we have $h < i_0 < j_0 < k$.

We define

$$\rho = \widehat{\pi}[1,i_0](\widehat{\pi}[i_0+1,j_0])^{\omega}\ .$$

Let us show that conditions C1, C2 and C3 are all satisfied.

Condition C1 is obviously fullfilled by B7 since $i_0, j_0 \in J \subseteq I$.

Let us next establish condition C2 by showing that $r_0 \in \textsc{ExceedReg}(\rho)$. Since $a_i \neq [r_0 := N]$ for all $N \in \mathbb{N}$ and for all $i \in [h+1,k]$ by B4, it follows that $\texttt{output}(\widehat{\pi}[h+1,k])$ does not contain any occurrence of $[\texttt{guess}\ r_0]$ by Point (ii) of Fact 4.1, in particular $[\texttt{guess}\ r_0]$ appears only finitely often in $\texttt{output}(\rho)$. We have $|\texttt{output}(\pi[m,n])|_{\mathtt{r}_{\gamma_0}} = 0$ by B5 and thus

$$|\texttt{output}(\widehat{\pi}[m,n])|_{\mathtt{r}_{\gamma_0}} \overset{\text{Fact 4.1,(ii)}}{=} |\texttt{output}(\pi[m,n])|_{\mathtt{r}_{\gamma_0}} = 0$$

and hence

$$|\texttt{output}(\widehat{\pi}[i_0+1,j_0])|_{\mathtt{r}_{\gamma_0}} = 0$$

since $J \subseteq I \subseteq [m,n]$. It follows that $\texttt{output}(\rho)$ contains only finitely many occurrences of $\mathtt{r}_{\gamma_0}$. Similarly one shows from B6 that $\texttt{output}(\rho)$ contains infinitely many occurrences of $\mathtt{ic}_{\gamma_0}$. This shows $r_0 \in \textsc{ExceedReg}(\rho)$ and hence we have established (C2).

Finally let us prove condition (C3). For the sake of contradiction assume there exists some register $r_1 \in \textsc{ExceedReg}(\rho) \cap \textsc{Reg}_{\textsc{Adam}}$ such that $[\texttt{guess}\ r_1]$ does not appear in $\texttt{output}(\rho)$ after position h. Since $r_1 \in \textsc{ExceedReg}(\rho)$ there exists a counter $\gamma_1 \in \Gamma_{r_1}$ such that $\mathtt{ic}_{\gamma_1}$ appears infinitely often in $\texttt{output}(\rho)$ but $\mathtt{r}_{\gamma_1}$ appears only finitely often in $\texttt{output}(\rho)$. Hence

$$\begin{aligned}|\texttt{output}(\widehat{\pi}[i_0+1,j_0])|_{\mathtt{ic}_{\gamma_1}} &\geq 1 \qquad \text{and}\\ |\texttt{output}(\widehat{\pi}[i_0+1,j_0])|_{\mathtt{r}_{\gamma_1}} &= 0\end{aligned}$$

by definition of ρ and therefore

$$\begin{aligned}|\texttt{output}(\pi[i_0+1,j_0])|_{\mathtt{ic}_{\gamma_1}} &\geq 1 \qquad \text{and}\\ |\texttt{output}(\pi[i_0+1,j_0])|_{\mathtt{r}_{\gamma_1}} &= 0\end{aligned}$$

by Point (ii) of Fact 4.1. Since $a_{i_0} = \mathtt{ic}_{\gamma_0}$ it follows

$$\begin{aligned}|\texttt{output}(\pi[i_0,j_0])|_{\mathtt{ic}_{\gamma_1}} &\geq 1 \qquad \text{and}\\ |\texttt{output}(\pi[i_0,j_0])|_{\mathtt{r}_{\gamma_1}} &= 0 \qquad .\end{aligned}$$

Therefore for all $i,j \in J$ we have

$$\boldsymbol{m}_{\gamma_1} \overset{\text{B9}}{=} h_{\gamma_1}(\pi[i,j]) \overset{i_0,j_0 \in J}{=} h_{\gamma_1}(\pi[i_0,j_0]) \overset{\text{Lemma 4.4}}{=} \mathtt{ic}_{\gamma_1} \quad .$$

Hence,

B11

$$\begin{aligned}&|\texttt{output}(\pi[i_0,j_0])|_{\mathtt{ic}_{\gamma_1}}\\ \geq\ & |J|\\ \overset{\text{B10}}{=}\ & \max\{\mathtt{val}_r(\pi[1,h]) \mid r \in \textsc{Reg}\}+1\end{aligned}$$

and

B12 $\texttt{output}(\pi[i_0,j_0])_{\mathtt{r}_{\gamma_1}} = 0$.

But then we obtain the following lower bound on $\mathsf{cost}_{\Gamma_{\gamma_1}}(\pi[1,j_0])$:

$$\begin{aligned}&\mathsf{cost}_{\Gamma_{\gamma_1}}(\pi[1,j_0])\\ \geq\ & \mathsf{cost}_{\Gamma_{\gamma_1}}(\pi[i_0,j_0])\\ \overset{\text{B11,B12}}{\geq}\ & \max\{\mathtt{val}_r(\pi[1,h]) \mid r \in \textsc{Reg}\}+1\end{aligned}$$

But since by assumption $[\texttt{guess}\ r_1]$ does not appear in $\texttt{output}(\rho)$ after position h, we conclude by Point (ii) of Fact 4.1 that for all $N \in \mathbb{N}$ the assignment $[r_1 := N]$ does not appear in $\texttt{output}(\pi[h,k])$ and therefore

$$\mathtt{val}_{r_1}(\pi[1,j_0]) = \mathtt{val}_{r_1}(\pi[1,h]).$$

Altogether we obtain

$$\begin{aligned}\mathtt{val}_{r_1}(\pi[1,j_0]) &= \mathtt{val}_{r_1}(\pi[1,h])\\ &\leq \max\{\mathtt{val}_r(\pi[1,h]) \mid r \in \textsc{Reg}\}\\ &< \mathsf{cost}_{\Gamma_{r_1}}(\pi[1,j_0]).\end{aligned}$$

Hence $\mathtt{cmp}_{r_1}(\pi[1,j_0]) = 1$ and therefore we have that $\mathtt{Cmp}(\pi[1,j_0])$ is not in $(\mathbb{C} \times \{0\}^{\textsc{Reg}})^*$, contradicting B1.

5. Reduction to the counter-based case

We have seen above how to solve counter-based games with guess actions. In this section, we show how to reduce regular games with guess actions to the action-based case.

5.1 Regular cost functions

It is finally time to introduce more formally what cost functions are. In this paper, we assume the measure functions used in the game to be defined in terms of regular cost functions. Exactly as a regular language can be defined by several means (automata, monoids, monadic second-order logic, regular expressions, ...), regular cost functions can be defined in numerous equivalent ways (B-automata, S-automata, stabilisation monoids, cost monadic second-order logic, B-expressions, ...). As far as the complexity of the decision procedures is not concerned, all this formalisms are equivalent.

Le us briefly describe cost monadic second-order logic, which is the most concise among these formalisms (see (Colcombet 2013b)). We assume the reader familiar with monadic second-order logic (see for instance (Thomas 1997)). In *cost monadic second-order logic*, the full syntax of monadic second-order logic is available, and is augmented with the construct $|X| \leq n$, in which X is a monadic variable and n is a unique non-negative integer variable. This construct has furthermore to appear positively in the formula (*i.e.* below an even number of negations). The semantics is as expected. For instance, the formula $\forall X.(|X| \leq n \to \forall x \in X.a(x))$ is a formula that holds on a given word u and for a given n if and only if $|u|_a \leq n$. The semantics of a formula is in fact to compute, for each input u, the least n such that the formula holds. Hence, the above formula computes the number of occurrences of the letter a in the input word. Examples of this logic over graphs shows that it can for instance express quantities like 'the diameter of the graph'.

The specificity of cost functions is to only consider the functions up to the following equivalence: $f, g : \mathbb{A}^* \to \mathbb{N} \cup \{\infty\}$ are $\approx$-*equivalent* if for all sets $X \subseteq \mathbb{N} \cup \{\infty\}$, f is bounded over X if and only if g is bounded over X. Indeed, a *cost functions* is an equivalence class for $\approx$. An equivalent way for defining the $\approx$-equivalence relation is as follows. A *correction function* is a map $\alpha : \mathbb{N} \to \mathbb{N}$ strictly increasing, extended with $\alpha(\infty) = \infty$. Given two maps $f, g : \mathbb{A}^*$, $f \preccurlyeq_\alpha g$ if $f \leq \alpha \circ g$, and $f \approx_\alpha g$ is $f \preccurlyeq_\alpha g$ and $g \preccurlyeq_\alpha f$. It happens that $f \approx g$ if and only if $f \approx_\alpha g$ for some α (see for instance (Colcombet 2013b)).

In fact, the winner of a game with guess actions does not change if we replace the measures by $\approx$-equvalent one. We will not establish this point (this is not very complicated), but rather prove that the winner is decidable for measures that are regular cost functions, and these are defined up to $\approx$.

5.2 History-deterministic B-automata

It happens that the actions introduced in the above section are exactly the one used in the automata model used for recognizing regular cost functions, namely B-automata. We shall introduce them now, paying a specific attention to a semantically restricted form of such automata, namely history-deterministic B-automata. These are specially designed for behaving well in the alternating context of games.

Formally, a *B-automaton* $\mathcal{A} = (Q, \mathbb{A}, q_0, F, \Gamma, \Delta)$ has a finite set of *states*, an alphabet $\mathbb{A}$, an *initial state*[1] q_0, a set of *accepting states* F, a finite set of counters Γ, and a *transition relation* $\Delta \subseteq Q \times \mathbb{A} \times \mathtt{ACt}_\Gamma \times Q$. A *run* of $\mathcal{A}$ over a word $u = a_1 \dots a_n \in \mathbb{A}^*$ is a sequence of the form $\rho = (p_0, a_1, h_1, p_1)(p_1, a_2, h_2, p_2) \dots (p_{n-1}, a_n, h_n, p_n) \in \Delta^*$ such that $p_0 = q_0$. It is *accepting* if furthermore $p_n \in F$. Its *cost* is $\mathsf{cost}(\rho) = \mathsf{cost}_\Gamma(h_1 \dots h_n)$. Given such a B-automaton, is computes a function:

$$\begin{aligned} [\![\mathcal{A}]\!] &: \mathbb{A}^* \to \mathbb{N} \cup \{\infty\} \\ &u \mapsto \inf\{\mathsf{cost}(\rho) \mid \rho \text{ accepting run of } \mathcal{A} \text{ over } u\}\ . \end{aligned}$$

A *regular cost function* is an equivalence class under

A *translation strategy* is a family $(\tau^n)_{n\in\mathbb{N}}$ of maps $\tau^n : Q \times \mathbb{A} \to \Delta$ that have the property that for all $n \in \mathbb{N}$:

- $\tau^n(u)$ is a run of $\mathcal{A}$ for all $u \in \mathbb{A}^*$, in which τ^n has been extended into a map $\tau^n : \mathbb{A}^* \to \Delta^*$ by $\tau^n(\varepsilon) = q_0$, and $\tau^n(ua) = \tau^n(\tau^n(u), a)$,
- for all words $u \in \mathbb{A}^*$, $\mathsf{cost}(\tau^n(u)) \leq n$.

A B-automaton is *history-deterministic for the map* $f : \mathbb{A}^* \to \mathbb{N} \cup \{\infty\}$ if there exist a translation strategy τ and a correction function α such that for all words $u \in \mathbb{A}^*$:

- $\alpha([\![\mathcal{A}]\!](u)) \geq f(u)$, *i.e.* all accepting runs ρ of $\mathcal{A}$ for u are such that $\alpha(\mathsf{cost}(\rho)) \geq f(u)$, and
- if $[\![\mathcal{A}]\!]_{\mathcal{A}}(u) \leq n$ then $\tau^m(u)$ ends in an accepting state for all $m \geq \alpha(n)$.

An B-automaton is *history-deterministic* if is is history-deterministic for the map $[\![\mathcal{A}]\!]$. In short, the meaning of this definition is that, though the B-automaton is not deterministic, there is a deterministic way (namely the translation strategy) to construct of run (first condition) which yields the correct result up to a $\approx$.

[1] Usually there is a set of initial states, but it is convenient in this paper, for history-deterministic B-automata to restrict to one. The models are equivalent.

Theorem 5.1. *Given a f in a regular cost function, there exists effectively an history-determinisitic B-automaton for f.*

5.3 Reduction to the counter-based case

The goal of this section is to show how a regular game with guess actions can be turned into a counter-based one. We fix a regular game with guess actions $\mathcal{G} = (\mathcal{A}, \mathbf{f}, T)$ with $\mathcal{A} = (V, \mathbb{C}, \textsc{Reg}, \delta, v_0)$. For all registers $r \in \textsc{Reg}$, f_r belongs to a regular cost function. This means that there is an history-deterministic B-automaton $\mathcal{A}_r = (Q_r, \mathbb{C}, i_r, F_r, \Gamma_r, \Delta_r)$ for f_r. Hence, there exists a correction function α a translation strategy τ_r such that for all input words $u \in \mathbb{C}^*$, $\alpha([\![\mathcal{A}]\!]) \geq f(u)$ and $\mathsf{cost}(\tau_r^m(u)) \leq \alpha(f(u))$ for all $m \geq \alpha(n)$. Our objective is to construct a new game $\mathcal{G}^\dagger$ that has same winner, and is counter-based.

Ideas about the construction. The essential idea behind this reduction is to execute during the play at the same time the original game $\mathcal{G}$ and the history-deterministic B-automaton $\mathcal{A}_r$ for each register r, its non-determinism being controlled by the owner of the register. The tuple of states obtained in this way is denoted $\mathbf{p} \in \prod_{r\in\textsc{Reg}} Q_r$. By looking whether these states are accepting or not, vectors of bits indexed by $\textsc{Reg}$ are produced at each step. These bits decorate the play, producing a word in $\mathbb{C} \times \{0,1\}^{\textsc{Reg}}$. The winner is decided by testing whether this word in $\mathbb{C} \times \{0,1\}^{\textsc{Reg}}$ belongs to the long term objective T.

This first description works well, but it is not sufficient. In fact, it would be perfect if all the registers would be guessed at the beginning of the game. The problem is that an assignment to a register makes the run of the $\mathcal{A}_r$ automata constructed so far irrelevant. Indeed proving the correctness of the above construction involves 'executing' a winning strategy for Eve in $\textsc{Exp}(\mathcal{G}^\dagger)$ 'fighting' against the translation strategies τ_r for the registers owned by Adam. However, if one inspects the definition of a translation strategy, it is parameterized by a bound n, and assigning a new value to the register amounts to change this bound in the middle of the run. This would be meaningless. To circumvent this problem, the game also maintains a set of reachable states for each $\mathcal{A}_r$ (denoted R, R' in what follows). A state is drawn from it whenever a register is assigned a value (once more chosen by the register owner). Virtually these sets R, R' account for possible runs of $\mathcal{A}_r$, but for which we do not not have any idea of the counter values seen so far. We use these pieces of run as prefixes before beginning to follow the translation strategy when an assignment is met. Since we do not control counter values along these pieces of run, the technique introduces some uncertainness. We resolve this problem in the correctness proof: by changing the register values chosen by Eve during each assignment. These changed register values allow to 'absorb' this uncertainness.

The construction. Let us make these ideas more concrete. Let Q be the disjoint union of the Q_r sets, Δ be the disjoint union of the Δ_r sets, and Γ be the disjoint union of the Γ_r sets. Let us also fix ourselves an order on the registers $r_1, \dots, r_{|\textsc{Reg}|}$.

To simplify the description of $\mathcal{G}^\dagger$, we extend the syntax of the move map expressions with the construction '$a; \phi$' with a an action, and ϕ a subformula. This is a shorthand for denoting a move (a, v)

to a fresh vertex v, with $\delta(v) = \phi$. This notation can be used recursively.

We construct the new game with guess actions $\mathcal{G}^\dagger$ as follows:

$$\begin{aligned} &\mathcal{G}^\dagger = (\mathcal{A}', (\mathsf{cost}_{\Gamma_r})_{r\in\text{REG}}, T') \\ \text{with}\quad &\mathcal{A}' = (V', \mathbb{C}', \text{REG}, \delta', v_0')\ , \\ &V' = V \times \{0,1\}^Q \times \prod_{r\in\text{REG}} Q_r\ , \\ &v_0' = (v_0, \bigcup_{r\in\text{REG}} I_r, (i_r)_{r\in\text{REG}})\ , \\ &\mathbb{C}' = (\mathbb{C}\times\{0,1\}^{\text{REG}}) \uplus \mathtt{ACt}_\Gamma\ , \end{aligned}$$

and the move map δ' is defined as follows. For all $(v, R, \mathbf{p}) \in V'$, $\delta'(v, R, \mathbf{p}) = \delta(v)[\eta]$ in which η is the substitution defined for all $a \in \mathbb{C}$ and all $v' \in V$ as $\eta(a, v') = t_{r_{|\text{REG}|}}(\mathbf{p})$ with :

$$t_0(\mathbf{p}) = ((a, \mathbf{b}), (v', \{q' \in Q \mid q \in R,\ (q, a, *, q') \in \Delta\}, \mathbf{p})) \tag{1}$$

$$t_i(\mathbf{p}) = \bigvee_{(p_{r_i}, a, h, p')\in\Delta_{r_i}} h; (t_{i-1}(\mathbf{p}[r \leftarrow p'])) \quad \text{if } r_i \in \text{REG}_{\text{EVE}} \tag{2}$$

$$t_i(\mathbf{p}) = \bigwedge_{(p_{r_i}, a, h, p')\in\Delta_{r_i}} h; (t_{i-1}(\mathbf{p}[r \leftarrow p'])) \quad \text{if } r_i \in \text{REG}_{\text{ADAM}} \tag{3}$$

where $b_r = 0$ if $p_r \in F_r$, 1 otherwise, and

$$\eta([\mathtt{guess}\ r], v') = \bigvee_{p'\in R\cap Q_r} ([\mathtt{guess}\ r], (v', R, \mathbf{p}[r \leftarrow p'])) \quad \text{for all } r \in \text{REG}_{\text{EVE}} \tag{4}$$

$$\eta([\mathtt{guess}\ r], v') = \bigwedge_{p'\in R\cap Q_r} ([\mathtt{guess}\ r], (v', R, \mathbf{p}[r \leftarrow p'])) \quad \text{for all } r \in \text{REG}_{\text{ADAM}}. \tag{5}$$

Hence, Eq. 1 copies the actions of the original arena $\mathcal{A}$, and updates the set of reachable states R. Eq. 2 and 3 give control of the B-automaton $\mathcal{A}_r$ to the owner of the register r. Finally Eq. 4 and 5 copy the guess actions, and give the right to the owner of the guessed register to jump to another (reachable) state of the B-automaton $\mathcal{A}_r$.

Finally, one defines $T' = \{u \in (\mathbb{C}')^\omega \mid u|_{(\mathbb{C}\times\{0,1\}^{\text{REG}})} \in T\}$. Note in particular in this construction that the bits occurring in $(\mathbb{C}\times\{0,1\}^{\text{REG}})$ are not the ones associated to the counters Γ but are obtained by testing whether the states of $\mathbf{p}$ are final. The bit issued from the cost_{Γ_r} functions are implicitly dealt with by the definition of a counter-based winning condition.

The goal in the remaining of this section is to prove the correctness of the construction, as stated as follows.

Lemma 5.1. *$\mathcal{G}$ is determined, and $\mathcal{G}^\dagger$ has the same winner.*

For proving this statement, we will begin with some considerations and definitions concerning the structure of the game $\mathcal{G}^\dagger$, that will lead to the main part of the proof, which consists in transforming a winning strategy for Eve in $\text{EXP}(\mathcal{G}^\dagger)$ into a winning strategy for Eve in $\text{EXP}(\mathcal{G})$ (Lemma 5.3). We will finally appeal to symmetry considerations for obtaining the same result for strategies for Adam. The result then immediately follows.

Definitions and first remarks concerning the structure of $\mathcal{G}^\dagger$. In order to be able to state the properties of the construction, it is convenient to define a certain number of projections of plays in $\text{EXP}(\mathcal{G}^\dagger)$. Recall first that the set of actions labelling this game belong to $\mathbb{C}' = (\mathbb{C}\times\{0,1\}^{\text{REG}}) \uplus \mathtt{ACt}_\Gamma$ to which have to be added the assignment actions from $\text{ASSIGN}[\text{REG}]$. Thus, given a (partial) play ψ in the game $\text{EXP}(\mathcal{A}')$, it is natural to consider:

- Its projection on $\mathbb{C}$, written $\mathtt{proj}_{\mathbb{C}}(\psi) \in \mathbb{C}^\omega$ (or *), on $\mathbb{C}\times\{0,1\}^{\text{REG}}$, written $\mathtt{proj}_{\mathbb{C}\times\{0,1\}^{\text{REG}}}(\psi)$, on $\mathtt{ACt}_{\Gamma_r}$ denoted $\mathtt{proj}_{\mathtt{ACt}_{\Gamma_r}}(\psi)$ or on $\mathbb{C}'$ denoted $\mathtt{proj}_{\mathbb{C}'}(\psi)$.
- Its projection $\mathtt{proj}_{\Delta_r}(\psi)$ to Δ_r^ω (or *), which can be reconstructed from the structure of the game (formally, when $r = r_i$, the move $(p, r, h, p') \in \Delta_r$ in definitions 2 and 3). Note that, though it is a sequence of transitions, it is not in general a run of the B-automaton $\mathcal{A}_r$, since when $[\mathtt{guess}\ r]$ is encountered, the state may jump to another one.
- $\mathtt{lgt}_r(\psi)$ which is the length of the longest prefix of ψ that ends with an assignment of r.

Note that all these projections yield infinite words when applied to infinite words.

Lemma 5.2. *For all partial plays ψ in $\text{EXP}(\mathcal{A}')$ and all register $r \in \text{REG}$ ending in $\mathsf{last}(\psi) = (v, R, \mathbf{p})$, a state $q \in Q_r$ belongs to R if and only if there is a run of $\mathcal{A}_r$ over $\mathtt{proj}_{\mathbb{C}}(\psi)$ starting in an initial state and ending in q, and $p_r \in R$.*

Furthermore, let $\psi\psi'$ be a partial play in $\text{EXP}(\mathcal{A}')$ that starts in $(v, R, \mathbf{p})$ and ends in $(v', R', \mathbf{p}')$, and such that ψ' contains no assignment of register r, then $\mathtt{proj}_{\Delta_r}(\psi)$ is a run of $\mathcal{A}_r$ from p_r to p_r' over $\mathtt{proj}_{\mathbb{C}}(\psi)$.

Translation of strategies. We are entering now the heart of the proof. Our goal is now to establish the following.

Lemma 5.3. *If Eve wins $\mathcal{G}^\dagger$, she also wins $\mathcal{G}$.*

Thus, let us fix a winning strategy σ' for Eve in $\text{EXP}(\mathcal{G}^\dagger)$. We aim at constructing a winning strategy σ for Eve in $\text{EXP}(\mathcal{G})$.

The principle of the construction of the strategy σ is essentially to 'project' σ' on $\text{EXP}(\mathcal{A})$. However, for this to be meaningful, we need to explain what to do with the choices of Adam that did not exist in $\text{EXP}(\mathcal{A})$. The way we describe this translation is as a partially defined map ρ from partial plays of $\text{EXP}(\mathcal{A}')$ to partial plays of $\text{EXP}(\mathcal{A})$. This partial map ρ is defined by induction on the length of the input partial play, in such a way that the image under ρ of σ' is a winning strategy in $\text{EXP}(\mathcal{G})$.

At the same time we define ρ, we establish a certain number of (induction) properties that we list now. Let ψ be some partial play in σ' such that $\rho(\psi)$ is defined and that ends in a vertex of the form $(v, R, \mathbf{p})$, we shall prove that:

P1 If v is owned by Eve, then ψ is prefix of $\psi' \in \sigma'$ such that $\rho(\psi')$ is defined, and $\rho(\psi') = \rho(\psi)m$ for some move from v in $\text{EXP}(\mathcal{A})$; if v is owned by Adam, and m is some move from v in $\text{EXP}(\mathcal{A})$, there exists $\psi' \in \sigma'$ such that $\rho(\psi')$ is defined, and $\rho(\psi') = \rho(\psi)m$.

P2 If $\rho(\psi)$ is a strict prefix of $\rho(\psi')$, then ψ is a prefix of ψ'. Furthermore, $\mathtt{proj}_{\mathbb{C}}(\psi) = \mathtt{proj}_{\mathbb{C}}(\rho(\psi))$.

P3 $\alpha(\mathtt{val}_r(\psi) + \mathtt{lgt}_r(\psi)) = \mathtt{val}_r(\rho(\psi))$ for all $r \in \text{REG}_{\text{EVE}}$,

P4 $\mathtt{val}_r(\psi) + \mathtt{lgt}_r(\psi) = \alpha(\mathtt{val}_r(\rho(\psi)))$ for all $r \in \text{REG}_{\text{ADAM}}$,

P5 if $\rho(\psi)$ is defined, $\mathsf{cost}_{\Gamma_r}(\mathtt{proj}_{\mathtt{ACt}_{\Gamma_r}}(\psi)) \le \mathtt{val}_r(\psi)$ for all registers $r \in \text{REG}_{\text{ADAM}}$,

P6 for all registers $r \in \text{REG}_{\text{ADAM}}$ and all partial plays $\psi\psi' \in \sigma'$ such that ψ' does not contain an assignment to r then $\tau_r^n(\mathtt{proj}_{\mathbb{C}}(\psi\psi')) = \tau_r^n(\mathtt{proj}_{\mathbb{C}}(\psi))\mathtt{proj}_{\Delta_r}(\psi')$ where n is $\mathtt{val}_r(\psi) - \mathtt{lgt}_r(\psi)$, and

P7 $\mathtt{proj}_{\mathbb{C}\times\{0,1\}^{\text{REG}}}(\psi) \sqsubseteq \mathtt{Cmp}(\rho(\psi))$.

Before proceeding, let us show that, assuming P1, P2, P5 and P7, we can prove Lemma 5.3. The property P1 implies directly that $\sigma = \rho(\sigma')$ is indeed a strategy for Eve in $\textsc{Exp}(\mathcal{A})$. We have to prove it winning. Consider a play ϕ consistent with σ. By P2, there is a play ψ consistent with σ' such that $\rho(\psi) = \phi$ (at the limit). Since σ' is winning for Eve, $\mathtt{proj}_{\mathbb{C}'}(\psi)$ belongs to $\textsc{Exp}(\mathsf{cost}, T')$. There are two cases: (1) either Eve wins because one of Adam's counter has exceeded its value, i.e. $\mathtt{cost}_{\Gamma_r}(\mathtt{proj}_{\mathtt{ACt}_{\Gamma_r}}(\psi)) > \mathtt{val}_r(\psi)$ for some $r \in \textsc{Reg}_{\textsc{Adam}}$, but this is not possible, since this contradicts P5, or (2) $\mathtt{proj}_{\mathbb{C}'}(\psi) \in T'$, or equivalently $\mathtt{proj}_{\mathbb{C}\times\{0,1\}^{\textsc{Reg}}}(\psi) \in T$; but, since by P7 $\mathtt{proj}_{\mathbb{C}\times\{0,1\}^{\textsc{Reg}}}(\psi) \sqsubseteq \mathtt{Cmp}(\rho(\psi)) = \mathtt{Cmp}(\phi)$, it follows that $\mathtt{Cmp}(\phi) \in T$, which means that ϕ is winning for Eve. Hence σ is winning, and Lemma 5.3 is proved.

Let us finally describe the ρ mapping. We establish at the same time the above invariants P1-P7. For readability, we concentrate on the meaningful parts of the proofs. In particular, we leave to the reader the verification of P1 and P2, and we do not mention along the construction the properties that are immediately preserved.

Let us fix ourselves $\psi \in \sigma'$.

Case 1: m was produced in Eq. 1. Then m is of the form $((v,*,*),(a,\mathbf{b}),(v',*,*))$ with $a \in \mathbb{C}$, $\mathbf{b} \in \{0,1\}^{\textsc{Reg}}$. In this case, $\eta(\psi m) = \eta(\psi)(v,a,v')$. The only property that is touched is P7, and more precisely what we have to prove is that the vector of bits $\mathbf{b}$ is correct. Formally, we have to prove that (a) for all $r \in \textsc{Reg}_{\textsc{Eve}}$, if $b_r = 0$ then $f_r(\mathtt{proj}_{\mathbb{C}}(\psi)) \leq \mathtt{val}_r(\rho(\psi))$ and (b) for all $r \in \textsc{Reg}_{\textsc{Adam}}$, if $b_r = 1$ then $f_r(\mathtt{proj}_{\mathbb{C}}(\psi)) > \mathtt{val}_r(\rho(\psi))$.

For (a), by definition, $b_r = 0$ means $p_r \in F_r$. It follows, by Lemma 5.2, that there is an accepting run of $\mathcal{A}_r$ over $\mathtt{proj}_{\mathbb{C}}(\psi)$. Furthermore, this run has maximal cost $\mathtt{lgt}_r(\psi)$ up to the last time an assignment of r has been performed, and coincides with $\mathtt{proj}_{\Delta_r}(\psi)$ from that point on, which means that it has cost at most $\mathtt{val}_r(\psi)$ from that point on. Hence, this run witnesses that $[\![\mathtt{proj}_{\mathbb{C}}(\psi)]\!] \leq \mathtt{val}_r(\psi) + \mathtt{lgt}_r(\psi)$. It follows, by assumption, that $f(\mathtt{proj}_{\mathbb{C}}(\psi)) \leq \alpha(\mathtt{val}_r(\psi) + \mathtt{lgt}_r(\psi))$, which in turn, by P3, is at most $\mathtt{val}_r(\rho(\psi))$. It follows the expected $f_r(\mathtt{proj}_{\mathbb{C}}(\psi)) \leq \mathtt{val}_r(\rho(\psi))$.

For (b), by definition, $b_r = 1$ means that $p_r \notin F_r$. But by P6, p_r is the last state of the of $\mathcal{A}_r$ over $\mathtt{proj}_{\mathbb{C}}(\psi)$ when followig the translation strategy τ_r^n for $n = \mathtt{val}_r(\psi)$. By the definition of history-determinism, since this state is not accepting, and by P4 $n \geq \alpha(\mathtt{val}_r(\rho(\psi)))$, we obtain $f_r(\mathtt{proj}_{\mathbb{C}}(\psi)) > \mathtt{val}_r(\rho(\psi))$.

This settles the first case.

Case 2: m was produced in Eq. 2. Then m is of the form $((v,*,*),h,(v',*,*))$ with $h \in \mathtt{ACt}_r$ for some $r \in \textsc{Reg}_{\textsc{Eve}}$. We set $\eta(\psi m) = \eta(\psi)$. Nothing special has to be proved in this case.

Case 3: m was produced in Eq. 3. Then m is of the form $((v,*,p),h,(v',*,p'))$ with $h \in \mathtt{ACt}_r$ for some $r \in \textsc{Reg}_{\textsc{Adam}}$. In this case, because the control is given to Adam, there is one such move for all possible transitions of $\mathcal{A}_r$. In fact here, we want only to follow that transition offered by the translation strategy for $\mathcal{A}_r$. Hence, if $(p_r,a,h,p'_r) = \tau_r^{\mathtt{val}(r)-\mathtt{lgt}_r(\psi)}(p_r)$, then $\rho(\psi m) = \rho(\psi)$, and is undefined otherwise. It is easy to check that the property P6 is maintained in this case for register r.

Case 4: m was produced in Eq. 4. Then m is of the form $((v,*,p),[r := n],(v',*,p'))$ with $r \in \textsc{Reg}_{\textsc{Eve}}$ and $n \in \mathbb{N}$. In this case, one sets $\rho(\psi m) = \rho(\psi)(v,[r := n'],v')$ where n' is set to be $\alpha(\mathtt{val}_r(\psi) + \mathtt{lgt}_r(\psi))$. In this case, one has to check that property P3 is preserved. And this is obvious.

Case 5: m was produced in Eq. 5. Then m is of the form $((v,R,\mathbf{p}),[r := n],(v',R',\mathbf{p}'))$ with $r \in \textsc{Reg}_{\textsc{Adam}}$ and $n \in \mathbb{N}$. Then $\rho(\psi m)$ is defined to be $\rho(\psi)(v,[r := k],v')$ for some $k \in \mathbb{N}$ if and only if

- $n = \alpha(k) - \mathtt{lgt}_r(\psi)$, and
- $\mathbf{p}' = \mathbf{p}[r \leftarrow q]$ for q the last state of the run $\tau_r^{\alpha(k)}(\mathtt{proj}_{\mathbb{C}}(\psi))$.

It is easy to check that in this case P4 and P6 are preserved.

The duality argument We have seen Lemma 5.3, which states that a winning strategy for Eve in $\textsc{Exp}(\mathcal{G}^\dagger)$ can be turned into a winning strategy for Eve in $\textsc{Exp}(\mathcal{G})$. As we did before, we use a symmetry argument for the Adam counterpart. It is sufficient to note that syntactically,

$$\overline{\mathcal{G}}^\dagger = \overline{\mathcal{G}^\dagger}\ .$$

It follows that if Adam wins $\mathcal{G}^\dagger$, by definition he wins $\textsc{Exp}(\mathcal{G}^\dagger)$. Hence, Eve wins $\overline{\textsc{Exp}(\mathcal{G}^\dagger)} = \textsc{Exp}(\overline{\mathcal{G}^\dagger}) = \textsc{Exp}(\overline{\mathcal{G}}^\dagger)$. Thus, by Lemma 5.3, Eve wins $\textsc{Exp}(\overline{\mathcal{G}}) = \overline{\textsc{Exp}(\mathcal{G})}$. Hence Adam wins $\textsc{Exp}(\mathcal{G})$. Since furthermore $\textsc{Exp}(\mathcal{G}^\dagger)$ is determined by Martin's determinacy theorem, it follows that $\mathcal{G}$ is determined. This completes the proof of Lemma 5.1.

6. Conclusion

In this paper, we have introduced games with guess actions as a natural way to model behaviors involving infinite quantities in finite games. We show that for non-trivial cases of such games, namely regular games with guess actions, the winner can be decided. A natural continuation of this work is to study the case of similar games played on pushdown arenas. In fact, the main obstacle in such a generalization is of a technical nature: it is extremely complicated with the classical tools to write proofs on these models. That is why we believe that the next step should be the development of a 'game oriented' mathematical framework in which most of the arguments involved in this work would be natural.

References

M. Bojańczyk and T. Colcombet. Bounds in ω-regularity. *Logical Methods in Compter Science*, page 54, 2009.

T. Colcombet. Safra-like constructions for regular cost functions over finite words. Unpublished, 2011.

T. Colcombet. Cost functions with several orders of magnitudes and the use of relative internal set theory. In *LICS 2013*, page 123, 2013a.

T. Colcombet. Regular cost functions, part I: logic and algebra over words. *Logical Methods in Compter Science*, page 47, 2013b.

T. Colcombet and C. Löding. The nesting-depth of disjunctive μ-calculus for tree languages and the limitedness problem. In *CSL 2008*, volume 5213, pages 416–430, 2008.

T. Colcombet and C. Löding. Regular cost functions over finite trees. In *LICS 2010*, pages 70–79. IEEE Computer Society, 2010.

N. Fijalkow and M. Zimmermann. Cost-parity and cost-streett games. In *FSTTCS 2012*, volume 18 of *LIPIcs*, pages 124–135. Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, 2012.

Y. Gurevich and L. Harrington. Trees, automata, and games. In *STOC 1982*, pages 60–65. ACM, 1982.

K. Hashiguchi. Relative star height, star height and finite automata with distance functions. In *Formal Properties of Finite Automata and Applications*, volume 386, pages 74–88, 1988.

D. Kirsten. Distance desert automata and the star height problem. *RAIRO - Theor. Inf. Appl.*, 3(39):455–509, 2005.

H. Leung. On the topological structure of a finitely generated semigroup of matrices. *Semigroup Forum*, 37:273–287, 1988.

I. Simon. On semigroups of matrices over the tropical semiring. *RAIRO - Theor. Inf. Appl.*, 28(3-4):277–294, 1994.

W. Thomas. Languages, automata and logic. In *Handbook of Formal Languages*, volume 3, chapter 7, pages 389–455. Springer, 1997.