Languages vs. ω-Languages in Regular Infinite Games

Namit Chaturvedi*, Jörg Olschewski**, and Wolfgang Thomas

Lehrstuhl Informatik 7, RWTH Aachen University, Germany
{chaturvedi,olschewski,thomas}@automata.rwth-aachen.de

Abstract. Infinite games are studied in a format where two players, called Player 1 and Player 2, generate a play by building up an ω-word as they choose letters in turn. A game is specified by the ω-language which contains the plays won by Player 2. We analyze ω-languages generated from certain classes $\mathcal{K}$ of regular languages of finite words (called $*$-languages), using natural transformations of $*$-languages into ω-languages. Winning strategies for infinite games can be represented again in terms of $*$-languages. Continuing work of Rabinovich et al. and of Selivanov (2007), we analyze how these "strategy $*$-languages" are related to the original language class $\mathcal{K}$. In contrast to that work, we exhibit classes $\mathcal{K}$ where the strategy representations strictly exceed $\mathcal{K}$.

1 Introduction

The theory of regular ω-languages is tied to the theory of regular languages of finite words (regular $*$-languages) in at least two different ways. First, one obtains all regular ω-languages as finite unions of sets $U \cdot V^\omega$ where U, V are regular $*$-languages. This representation is obtained via the model of nondeterministic Büchi automata over infinite words. Second, if one works with deterministic Muller automata, one obtains a representation of all regular ω-languages as Boolean combinations of sets $\lim(U)$ with regular U (cf. [5, 15]), where

$$\lim(U) = \{\alpha \in \Sigma^\omega \mid \text{infinitely many finite } \alpha\text{-prefixes are in } U\}.$$

In this paper we focus on the latter approach as we study the connection between $*$-languages and ω-languages in the context of infinite games. Another canonical transformation of $*$-languages into ω-languages is to consider the *extensions* of words of a $*$-language U:

$$\mathrm{ext}(U) = \{\alpha \in \Sigma^\omega \mid \text{some finite } \alpha\text{-prefix is in } U\}.$$

Boolean combinations of such languages with regular U are recognized by deterministic weak Muller automata (also known as Staiger-Wagner automata [11]).

* Supported by DFG-Graduiertenkolleg AlgoSyn.
** Supported by ESF-Eurocores LogICCC project GASICS.

We write BC(lim(REG)), respectively BC(ext(REG)), for the Boolean combinations of sets $\lim(U)$, respectively $\mathrm{ext}(U)$, with regular U. In each case we refer to a fixed alphabet Σ so that complementation is done with respect to Σ^ω. We prefer the notation $\mathrm{ext}(U)$ over the other popular alternative, $U \cdot \Sigma^\omega$, only for the purpose of emphasizing analogies or differences to $\lim(U)$.

The purpose of this paper is to study the connection between $*$-languages and ω-languages in two dimensions of refinement. First, the class REG is replaced by small subclasses, such as the class of piecewise testable languages or levels within the dot-depth hierarchy of star-free languages. In particular, for such a class $\mathcal{K}$ of $*$-languages, we consider the classes $\mathrm{BC}(\lim(\mathcal{K}))$ and $\mathrm{BC}(\mathrm{ext}(\mathcal{K}))$, defined as above for the case REG. Secondly, we study a natural approach for the reverse direction, from ω-languages back to $*$-languages. Here the concept of infinite games is used, in which ω-languages enter as "winning conditions", and $*$-languages arise as representations of "winning strategies". We shall study the question whether games with a winning condition in classes such as $\mathrm{BC}(\mathrm{ext}(\mathcal{K}))$ or $\mathrm{BC}(\lim(\mathcal{K}))$ can be "solved" with winning strategies that are again representable in $\mathcal{K}$.

Let us recall the framework of infinite games in a little more detail. These games are played between two players, namely Player 1 and Player 2. In each round, first Player 1 picks a letter from an alphabet Σ_1 and then Player 2 a letter from an alphabet Σ_2. An infinite play of the game is thus an ω-word over $\Sigma := \Sigma_1 \times \Sigma_2$. One decides the winner of this play by consulting an ω-language $L \subseteq \Sigma^\omega$, also called the *winning condition*: If the play belongs to L, then Player 2 is the winner, otherwise Player 1 is. Games whose winning conditions belong to the class $\mathrm{BC}(\mathrm{ext}(\mathcal{K}))$ are referred to as *weak games* while those whose winning conditions belong to $\mathrm{BC}(\lim(\mathcal{K}))$ are called *strong games*.

A strategy for either player gives the choice of an appropriate letter $a \in \Sigma_1$, resp. $a \in \Sigma_2$, for each possible play prefix where it is Player 1's, resp. Player 2's, turn. We can capture a strategy for a player by collecting, for each letter a, the set K_a of those finite play prefixes that induce the choice of a. For a strategy of Player 1 we have $K_a \subseteq \Sigma^*$, for Player 2 we have $K_a \subseteq \Sigma^*\Sigma_1$. We say that a strategy is in $\mathcal{K}$ if each language K_a is.

The fundamental Büchi-Landweber Theorem [1] (also see [15, 3]) says that for each regular ω-language $L \in \mathrm{BC}(\lim(\mathrm{REG}))$, one of the two players has a winning strategy, that one can decide who is the winner, and that one can present a *regular* winning strategy (in the sense mentioned above) for the winner. In short, we say that *regular games are determined with regular winning strategies*. In [8, 9] an analogous result for the class SF of star-free languages was shown: Star-free games are determined with star-free winning strategies. We shall focus in this paper on subclasses of SF, where – as it will turn out – the situation is more complicated. For instance, we show that for the class DD_1 of languages of dot-depth 1, games with winning conditions in classes $\mathrm{BC}(\mathrm{ext}(\mathrm{DD}_1))$ and $\mathrm{BC}(\lim(\mathrm{DD}_1))$ are in general determined, not with winning strategies in DD_1, but only with those in classes DD_2 and DD_3 respectively. In contrast to this, we show that for games in the more restricted class $\mathrm{BC}(\mathrm{ext}(\text{pos-}\mathrm{DD}_1))$, we have

determinacy with winning strategies in DD_1. (The class pos-DD_1 is the closure of languages $w_0\Sigma^* w_1 \ldots \Sigma^* w_n$, where $w_i \in \Sigma^*$, under positive Boolean operations, i.e. union and intersection. The Boolean closure of pos-DD_1 is DD_1.)

The paper is structured as follows. In Section 2 we summarize technical preliminaries on infinite games, well known subclasses of the class SF of star-free languages, and the subclasses of infinite languages that we consider in this paper. Subsequently, in Sections 3 to 5 we consider games over these classes of infinite languages and present results pertaining to winning strategies in these games. We conclude with some open questions and perspectives.

2 Technical Preliminaries

2.1 Languages, Automata, Games

We use standard notation [4] regarding languages and automata. As a model of automata with output we use Moore machines, which transform words over an alphabet Σ into words of an alphabet Γ via an output function $\lambda\colon Q \to \Gamma$ over the state set Q.

Over ω-words we use the models of *SW-automata* (Staiger-Wagner automata or weak Muller automata) and Muller automata. These are deterministic automata whose acceptance component is a family $\mathcal{F}$ of state sets. An ω-word α is accepted by an SW-automaton if the set of visited states in the unique run over α belongs to $\mathcal{F}$; for Muller automata one refers to the set of infinitely often visited states instead.

For any alphabet $\Sigma = \Sigma_1 \times \Sigma_2$, an ω-language $L \subseteq \Sigma^\omega$ induces a game with winning condition L; we shall just speak of the "game L". In this game, a *strategy for Player 1* is a mapping $\sigma\colon \Sigma^* \to \Sigma_1$ and a *strategy for Player 2* is a mapping $\tau\colon \Sigma^* \to \Theta$, where $\Theta := {\Sigma_2}^{\Sigma_1}$ is the (finite) set of all mappings from Σ_1 to Σ_2. An infinite word $\alpha = (a_i, b_i)_{i\in\mathbb{N}} \in \Sigma^\omega$ is said to be *consistent with* σ, if for all positions $i \in \mathbb{N}$ we have $\sigma(\alpha[0,i)) = a_i$. Analogously, α is *consistent with* τ, if for all $i \in \mathbb{N}$ we have $\tau(\alpha[0,i))(a_i) = b_i$. For two strategies σ and τ there is a (uniquely determined) word $\alpha(\sigma,\tau)$ that is consistent with both σ and τ.

If $\alpha(\sigma,\tau) \in L$ for every Player 1 strategy σ, then τ is called a *winning strategy* for Player 2. The other way around, if $\alpha(\sigma,\tau) \notin L$ for all Player 2 strategies τ, then σ is a winning strategy for Player 1. We say that a strategy σ for Player 1 is in the class $\mathcal{K}$ if for every $a \in \Sigma_1$ the language $K_a = \{w \in \Sigma^* \mid \sigma(w) = a\}$ is in $\mathcal{K}$. A strategy τ for Player 2 belongs to $\mathcal{K}$ if the language $K_{a\to x} = \{w \mid \tau(w)(a) = x\}$ is in $\mathcal{K}$ for every $(a,x) \in \Sigma$. For the language classes we consider, this definition is consistent with the one presented in the introduction.

In this paper we focus on "finite-state strategies" realized by Moore machines. A Moore machine implementing a strategy τ for Player 2, is given by $M_\tau = (Q, \Sigma, \Theta, q_0, \delta, \lambda)$ with $\lambda\colon Q \to \Theta$ such that for all $w \in \Sigma^*$ it holds $\lambda(\delta(q_0, w)) = \tau(w)$. A Moore machine M_σ for Player 1 is obtained analogously by replacing Θ with Σ_1.

For all games of this paper, winning strategies of this kind suffice. In order to obtain this format of a strategy, the given winning condition L has to be cast

into automata theoretic form. For ω-languages in a class BC(ext($\mathcal{K}$)), where $\mathcal{K} \subseteq$ REG, it is known that SW-automata (over $\Sigma := \Sigma_1 \times \Sigma_2$) can be used; for ω-languages in a class BC(lim($\mathcal{K}$)) we may take Muller automata.

Usually, classical two-player games are considered in the literature over a graph with two different types of nodes: one type belonging to Player 1, the other to Player 2. Such a game graph can be obtained from the ω-automaton recognizing the winning condition by doubling the state space and splitting the moves by letters of Σ into moves via Σ_1 and Σ_2. However, for our purposes it is more convenient to consider a game graph $G = (Q, \Sigma, q_0, \delta)$ with only one type of nodes, and in every node q we first let Player 1 choose an action from Σ_1 and after that let Player 2 choose from Σ_2. With this "unified" model, the conversions between ω-automata, game graphs, and Moore machines are straightforward.

As is well-known, the nodes (or: states) of the game graph in general do not suffice as states of a Moore machine defining a winning strategy. If the state of the game graph determines the move of the strategy, we speak of a *positional strategy*, which can be represented as a mapping $\sigma\colon Q \to \Sigma_1$ or $\tau\colon Q \to \Theta$, respectively. Positional winning strategies suffice in the case of "reachability games", "Büchi games", and "parity games" (see e.g. [16]). The first two of which correspond to games L of the form ext(K) or lim(K) respectively, for some regular $*$-language K. The last *parity condition* refers to a coloring of game graph vertices by natural numbers, and a play is won by Player 2 iff the maximal color occurring infinitely often in it is even.

The key results about positional determinacy also hold in our unified model of game graphs. This can easily be shown by splitting the nodes of a unified game graph as mentioned above, and copying the color of the original node to the new ones, thereby transforming it to a game on a classical game graph.

The Boolean combinations as they appear in games with the Staiger-Wagner winning condition or Muller winning condition are handled by converting the winning condition into a special form, called parity condition while expanding the game graph by an extra "memory component". For weak games (with Staiger-Wagner winning conditions), one replaces a state q of the game graph by a pair (q, R) where R is the set of those states visited in the play up to the current point. The set R is called the AR ("appearance record"). For strong games (with Muller winning conditions), one refines this information by listing the visited states in the order of most recent visits, and with a pointer to that place in the sequence where the current state was located in the preceding step. In a normalized presentation over a space $\{q_1, \ldots, q_k\}$, we deal with expanded states (q, R) where R is an LAR ("latest appearance record"): a pair consisting of a permutation of $(q_1, \ldots, q_k)$ and a number $h \in \{1, \ldots, n\}$. Over the expanded state-space it suffices to satisfy the mentioned parity condition.

2.2 Classes of regular languages

In the subsequent definitions we recall some basic subclasses of the star-free languages; for more background see [13, 6].

A $*$-language $K \subseteq \Sigma^*$ is *piecewise testable* if it is a Boolean combination of *basic* PT-*sets* $\Sigma^* a_1 \Sigma^* a_2 \cdots \Sigma^* a_n \Sigma^*$ where $a_1, a_2, \ldots, a_n \in \Sigma$. Denote the class of piecewise testable languages by PT. The class of *positive* Boolean combinations of basic PT-sets (in which only $\cup$ and $\cap$ are used) is denoted by pos-PT.

A $*$-language $K \subseteq \Sigma^*$ is *generalized definite* if it is a Boolean combination of sets $w\Sigma^*$ and $\Sigma^* w$ with $w \in \Sigma^*$. We denote the class of generalized definite languages by GDEF.

The dot-depth hierarchy, introduced by Cohen and Brzozowski [2], is a sequence of language classes $\mathrm{DD}_0, \mathrm{DD}_1, \ldots$ where $\mathrm{DD}_0 = \mathrm{GDEF}$ and DD_{n+1} can be obtained as the class of Boolean combinations of languages $K_1 \cdot \ldots \cdot K_\ell$ (over a given alphabet) with $K_1, \ldots, K_\ell \in \mathrm{DD}_n$. As a special case let us mention the languages of *dot-depth 1*; they are the Boolean combinations of *basic* DD_1-*sets* $w_0 \Sigma^* w_1 \Sigma^* \cdots w_{n-1} \Sigma^* w_n$ where $w_0, w_1, \ldots, w_n \in \Sigma^*$. In analogy to the class pos-PT we define pos-DD_1 as the class of positive Boolean combinations of basic DD_1-sets.

The dot-depth hierarchy is strict, and it exhausts the class SF of star-free languages. For later use we also recall a logical characterization of DD_n (see [14]): L (not containing the empty word) is in DD_n iff it can be defined by a first-order sentence that is a Boolean combination of Σ_n sentences[1] where the signature has symbols for the minimal and the maximal position of a word, the predecessor and successor functions, the ordering $<$ of positions, and the predicates Q_a giving, respectively, the positions with letter a.

The study of the language classes above is based on corresponding congruences over a given alphabet. We recall these congruences for the case of languages of dot-depth 1.

For $k, m \in \mathbb{N}$ and an m-tuple $\nu = (w_1, \ldots, w_m)$ of words of length $k+1$, we say that ν *appears* in a word u if u can be written as $u = u_i w_i v_i$ with suitable words u_i, v_i such that $i < j$ implies $|u_i| < |u_j|$. We say that ν *appears* in an ω-word α if α can be written as $u = u_i w_i \alpha_i$ with suitable words u_i, α_i such that $i < j$ implies $|u_i| < |u_j|$. With $\mu_{m,k}(w)$ (resp. $\mu_{m,k}(\alpha)$) we denote the set of all m-tuples of words of length $k+1$ that appear in w (resp. in α).

Two words u, v are (m,k)-equivalent ($u \sim_{m,k} v$) if

1. u and v have the same k first letters,
2. the same m-tuples of words of length $k+1$ appear in u and v, and
3. u and v have the same k last letters.

Then we have: $(*)$ *A $*$-language $K \subseteq \Sigma^*$ is of dot-depth one iff it is a union of $\Sigma^*/\!\sim_{m,k}$ equivalence classes for some $m, k \in \mathbb{N}$.* (In the definition of $\sim_{m,k}$ we refer to possibly overlappig infixes; this dos not affect the statement $(*)$.)

Let us proceed to ω-languages. For two ω-words α, β we write $\alpha \sim_{m,k} \beta$ if α and β have the same k first letters, and the same m-tuples of words of length $k+1$ appear in α and β. Then we clearly have: $(**)$ *An ω-language $L \subseteq \Sigma^\omega$ is in* $\mathrm{BC}(\mathrm{ext}(\text{pos-}\mathrm{DD}_1))$ *iff it can be written as a union of $\Sigma^\omega/\!\sim_{m,k}$ equivalence classes for some $m, k \in \mathbb{N}$.*

[1] First-order sentences in prenex normal form with n alternating quantifier blocks starting with an existential block.

3 Winning strategies in restricted weak games

We start with games in BC(ext(pos-DD_1)) which coincides with the class of Boolean combinations of sets ext(K) where K is a basic DD_1-set, or in other words: Boolean combinations of sets $w_0\Sigma^* w_1\Sigma^* \cdots w_{n-1}\Sigma^* w_n\Sigma^\omega$.

Theorem 1. *Games in the class* BC(ext(pos-DD_1)) *are determined with winning strategies in* DD_1.

Proof. By the characterization $(**)$ at the end of the preceding section, we can write an ω-language L in BC(ext(pos-DD_1)) as a union of $\Sigma^\omega/{\sim_{m,k}}$ equivalence classes $L = \bigcup_{i=1}^{n}[\alpha_i]$ where each $\alpha_i \in \Sigma^\omega$. We show how to obtain a game graph with a parity winning condition that captures the game with winning condition L.

In the graph, the play prefix w will lead to the $\sim_{m,k}$-class $[w]$ of w. The game graph consists of the set of nodes $Q = \Sigma^*/{\sim_{m,k}}$. For every every $(a,x) \in \Sigma$, we have edges from $[w]$ to $([w(a,x)])$. Note that this definition is well-defined, as from the set of m-tuples of length $k+1$ occurring in w, the suffix of length k of w, and the new letter (a,x), one can determine the set of m-tuples of length $k+1$ occurring in $w(a,x)$. We designate $q_0 = [\epsilon]$ as the start node of a play. For the winning condition, we assign a color $\chi(q)$ to every node q, namely $\chi([w]) = 2 \cdot |\mu_{m,k}(w)|$ if there is an $\alpha \in L$ such that the prefix of α of length k equals the length k prefix of w and $\mu_{m,k}(\alpha) = \mu_{m,k}(w)$, and $\chi([w]) = 2 \cdot |\mu_{m,k}(w)| - 1$ otherwise. Note that χ is increasing since for $w \in \Sigma^*$, and $(a,x) \in \Sigma$ we have $\chi([w]) \leq \chi([w(a,x)])$. A play is won by Player 2 in the game for L iff the corresponding play in the graph game reaches ultimately an even color (and stays there), giving again a win for Player 2.

By a well-known result on parity games, the parity game is determined, and the winning player has a uniform positional winning strategy. This means in particular, that in the parity game the winning player has a positional winning strategy from q_0. We show that she also has a DD_1 winning strategy in the original game.

Let $\lambda\colon Q \to \Sigma_1$ be a positional winning strategy of Player 1 in the parity game. Define $\sigma\colon \Sigma^* \to \Sigma_1$ to be $\sigma(w) = a$ where a is the letter such that $\lambda([w]) = a$. The strategy σ is in DD_1, because for each $a \in \Sigma_1$ we know that $\sigma^{-1}(a) = \bigcup_{\lambda(w)=a}[w]$ is in DD_1. We still have to show that σ is winning for Player 1 in the game with winning condition L. For this purpose, let $\alpha = (a_0,x_0)(a_1,x_1)(a_2,x_2)\cdots \in \Sigma^\omega$ be consistent with σ. We have to show that $\alpha \notin L$. Then

$$\rho = [\epsilon], [(a_0,x_0)], [(a_0,x_0)(a_1,x_1)], \ldots$$

is a play in the parity game that is consistent with λ. So Player 1 wins ρ and thus the maximal color p that occurs infinitely often in ρ is odd. Let $i \in \mathbb{N}$ such that $\chi(\rho(i)) = p$. Then all following positions must have the same priority $p = \chi(\rho(i)) = \chi(\rho(i+1)) = \ldots$, because χ is increasing. This means the set $\mu_{m,k}(w)$ of m-tuples appearing in a word w from $\rho(i)$ does not change from i onwards. So the set of m-tuples of α is $\mu_{m,k}(\alpha) = \mu_{m,k}(w)$ for any $w \in \rho(i)$. Furthermore

the prefix of α of length k is equal to the length k prefix of w for any $w \in \rho(i)$. Since p is odd, and by the definition of χ there does not exist such a word $\alpha \in L$, so $\alpha \notin L$. This proves that σ is winning for Player 1.

In the analogous way it is shown that if Player 2 has a positional winning strategy in the parity game from q_0, then Player 2 has a DD_1 winning strategy in the game L.

□

Next we turn to pos-PT, the class of positive combinations of basic piecewise testable languages; they are of the form $\Sigma^* a_1 \Sigma^* a_2 \dots a_n \Sigma^*$. We show that in this case we can proceed with a much simpler approach that avoids the formation of equivalence classes.

As a preparation we recall a result of I. Simon [10] about the transition structure of automata that accept piecewise testable languages. For a DFA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ and any $\Gamma \subseteq \Sigma$, let $G(\mathcal{A}, \Gamma)$ denote the directed graph underlying the automaton $\mathcal{A}$, such that G only has edges labeled with elements of Γ.

Proposition 2 ([10]). *Let $\mathcal{A}$ be the minimal DFA accepting the $*$-language K. Then K is piecewise testable iff*

1. *$G(\mathcal{A}, \Sigma)$ is acyclic, and*
2. *for every $\Gamma \subseteq \Sigma$, each component of $G(\mathcal{A}, \Gamma)$ has a unique maximal state.*

Theorem 3. *Games in* BC(ext(pos-PT)) *are determined with winning strategies in* PT.

Proof. For every ω-language $L \in \mathrm{BC}(\mathrm{ext}(\text{pos-PT}))$, there exists a regular language $K \in \mathrm{PT}$ such that $L = \lim(K)$. This is shown easily by induction over Boolean combinations (cf. [7]). The minimal DFA $\mathcal{A} = (Q, \Sigma, q_0, \delta, F)$ accepting K can be considered as an ω-automaton accepting K. We thus obtain a game graph with a Büchi winning condition. Since Büchi games are determined with positional winning strategies (see e.g. [3]), the strategy of the winning player only depends on the current state in the play. Assume, without loss of generality, that Player 2 has a winning strategy. Then for every mapping $\theta\colon \Sigma_1 \to \Sigma_2$, let $F_\theta \subseteq Q$ be set the states that induce a choice of θ. Consider the automaton $\mathcal{A}_\theta = (Q, \Sigma, q_0, \delta, F_\theta)$. Since $G(\mathcal{A}_\theta, \Sigma) = G(\mathcal{A}, \Sigma)$, we conclude from the proposition above that the language accepted by $\mathcal{A}_\theta$ is a piecewise testable language.
□

It is worth noting that the game graphs for games in $\mathrm{BC}(\mathrm{ext}(\text{pos-PT}))$ are obtained directly from the finite automata that recognize the piecewise testable languages in question, and that piecewise testable winning strategies are obtained by observing a certain form of the associated transition graphs. For games in $\mathrm{BC}(\mathrm{ext}(\text{pos-DD}_1))$ we had to resort to the domain of congruences or, in algebraic terms, to the concept of syntactic monoids. This results in exponentially larger game graphs. In order to avoid this blow-up one might try to apply a result of Stern [12] that gives a property of transition graphs of (minimal) automata which characterizes the languages in DD_1; however, it seems that the necessary step towards obtaining parity games (as done in Theorem 1) spoils this property – which prevents a direct approach as for pos-PT.

4 Winning strategies in weak games

Theorem 4. *There are games in* $\mathrm{BC}(\mathrm{ext}(\mathrm{PT}))$*, and therefore in* $\mathrm{BC}(\mathrm{ext}(\mathrm{DD}_1))$*, that do not admit* DD_1 *winning strategies.*

Proof. Let $\Sigma \coloneqq \{a,b,c,d\} \times \{0,1\}$, and define $*$-languages $K_1 \coloneqq (a,0)^*(b,0)$, $K_2 \coloneqq \Sigma^*(d,1)$, $K_d \coloneqq \Sigma^*(d,0)\Sigma^* \cup \Sigma^*(d,1)\Sigma^*$, and for every letter $x \in \{a,b,c\}$ define $K_x \coloneqq \Sigma^*(x,1)\Sigma^*$. Let L be the ω-language over Σ, that contains all ω-words α such that

$$\alpha \in \mathrm{ext}(K_d) \wedge \big(\alpha \in \mathrm{ext}(K_1) \Leftrightarrow \alpha \in \mathrm{ext}(K_2)\big) \wedge \bigwedge_{x \in \{a,b,c\}} \alpha \in \overline{\mathrm{ext}(K_x)}.$$

All the $*$-languages above are in PT, so L is in $\mathrm{BC}(\lim(\mathrm{PT}))$ and a fortiori in $\mathrm{BC}(\lim(\mathrm{DD}_1))$. We see that L is won by Player 2: she remembers whether a prefix in K_1 has occurred; if so, then she responds to a later occurrence of d with 1, otherwise with 0. We claim there is no DD_1 winning strategy (and a fortiori no PT winning strategy) for Player 2. Assume there is such a winning strategy $\tau\colon \Sigma^* \to \Theta$, which is implemented by a DD_1 Moore machine. Then there are $*$-languages $K_{\theta_1}, \ldots, K_{\theta_n}$ of dot-depth one, implementing this strategy. In particular, $K_{d1} \coloneqq \{w \mid \tau(w)(d) = 1\}$ is a dot-depth one language as a finite union of K_{θ_i} languages. So it is a finite union of equivalence classes $[w_i]_{\sim_{m,k}}$.

Note that in the word $w \coloneqq (a^k b^k c^k a^k c^k b^k a^k)^m$ all possible m-tuples of length $k+1$ over the alphabet $\{a,b,c\}$ appear. Let Player 1 play a strategy σ_1 that chooses $w \cdot d \cdot a^\omega$. Consider the unique word α_1 that is consistent with both σ_1 and τ. Since τ is a winning strategy for Player 2 and σ_1 plays $a^k b$ in the beginning, we have $(d,1)$ occurring in α. So the word $w \times \{0\}$ is in K_{d1}.

Now let Player 1 play the strategy σ_2 which chooses $a^k c^k \cdot w \cdot d \cdot a^\omega$. The word $w_2 \coloneqq a^k c^k \cdot w$ contains all possible m-tuples of length $k+1$ over the alphabet $\{a,b,c\}$, as well. Then we have $w_1 \sim_{m,k} w_2$ and thus $w_2 \times \{0\} \in K_{d1}$. Then the unique word α_2 that is consistent with both σ_2 and τ contains $(d,1)$ as an infix. This contradicts that τ is a winning strategy for Player 2. ◻

Theorem 5. *A game* $L \in \mathrm{BC}(\mathrm{ext}(\mathrm{DD}_i))$ *is determined with winning strategies in* DD_{i+1}.

Proof. Given the language L as a Boolean combination of ω-languages $\mathrm{ext}(K)$ with $K \in \mathrm{DD}_i$, we first proceed to a game graph where every node is a DD_i equivalence class (and hence a $*$-language in DD_i). The game graph is equipped with a Staiger-Wagner winning condition.

As explained in Section 2.1, we transform the game graph via the AR construction to a new game graph with the parity winning condition. A state in the new game graph is a pair (q,R) consisting of a state $q \in \Sigma^*/{\sim_{\mathrm{DD}_i}}$ of the original graph and an AR R. Over this game graph (with the parity winning condition), the winner has a positional winning strategy. We have to show that each node (q,R) corresponds to a DD_{i+1} language in the sense that the play

prefixes leading to (q, R) form a language $K_{(q,R)}$ in DD_{i+1}. (Then the play prefixes that cause the winner to choose a fixed letter a are obtained as a union of languages $K_{(q,R)}$ and hence in DD_{i+1} as desired.)

For this purpose, it is convenient to apply the logical characterization of DD_i-languages as mentioned in Section 2.2. Each vertex q corresponds to the language K_q consisting of play prefixes leading to q. Each K_q is a language in DD_i, defined by a Boolean combination ψ_q of Σ_i-sentences. For clarity we write $\psi_q(\max)$ emphasizing the last position max of the current play prefix; so $\psi_q(x)$ expresses that the play prefix up to position x belongs to q. We have to express the restriction that such a play prefix leads to the AR $R \subseteq Q$ for the state space Q. This is formalized by the sentence

$$\varphi_{(q,R)} = \psi_q(\max) \wedge \bigwedge_{r \in R} \exists x \psi_r(x) \wedge \bigwedge_{r \notin R} \neg \exists x \psi_r(x).$$

Since ψ_r is a Boolean combination of Σ_i-sentences, we obtain (in prenex form) a Boolean combination of Σ_{i+1}-sentences. In this way we obtain the membership of $K_{(q,R)}$ in DD_{i+1}. □

5 Winning strategies in strong games

Theorem 6. *Games in* $\mathrm{BC}(\lim(\mathrm{PT}))$ *are determined with winning strategies in* PT.

Proof. For every ω-language $L \in \mathrm{BC}(\lim(\mathrm{PT}))$, there exists a regular language $K \in \mathrm{PT}$ such that $L = \lim(K)$. (see [7]). The remainder of this proof is identical to that of Theorem 3. □

Theorem 7. *There are games in the class* $\mathrm{BC}(\lim(\mathrm{DD}_1))$ *that do not admit* DD_1 *winning strategies.*

Proof. Let L be the ω-language over $\{a, b\} \times \{0, 1\}$ where $(a, 1)$ does not occur and where $(b, 0)$ occurs infinitely often iff $(b, 1)$ occurs infinitely often. The language L is in $\mathrm{BC}(\lim(\mathrm{DD}_1))$. We can easily see that L is won by Player 2. We claim there is no DD_1 winning strategy (and a fortiori no LT winning strategy) for Player 2. Assume there is such a winning strategy $\tau\colon \Sigma^* \to \Theta$, which is implemented by a DD_1 Moore machine. Then there are $*$-languages K_{a0}, K_{a1}, K_{b0}, and K_{b1} of dot-depth one, implementing this strategy (cf. the proof of Theorem 4). In particular K_{b0} and K_{b1} are dot-depth one languages, and we have that K_{b0} is a finite union of equivalence classes $[w]_{\sim_{m,k}}$:

$$K_{b0} = \bigcup_{i=1}^{n} [w_i]_{\sim_{m,k}}$$

Let Player 1 play a strategy σ which chooses $(ba^k)^\omega$. Consider the unique word α that is consistent with both σ and τ. Since τ is a winning strategy for

Player 2 and σ plays infinitely many letters b, we have both $(b,0)$ and $(b,1)$ occurring infinitely often in α.

For any prefix $u_i \coloneqq \alpha[0,i)$ of α, the set of m-tuples of words of length $k+1$ that appear in u_i is a subset of the set of m-tuples that appear in u_{i+1}. This means that there exists a $j \in \mathbb{N}$ such that the set of m-tuples is the same for all u_{j+i}, $i \in \mathbb{N}$. We choose $\hat{\jmath} \geq j$ such that $k+1$ divides $\hat{\jmath}$, which yields that the suffix of $u_{\hat{\jmath}}$ of length k is equal to $(a,0)^k$. But then we have $[u_{\hat{\jmath}}]_{\sim_{m,k}} = [u_{\hat{\jmath}+i(k+1)}]_{\sim_{m,k}}$, $i \in \mathbb{N}$, since $u_{\hat{\jmath}}$ and $u_{\hat{\jmath}+i(k+1)}$ have the same m-tuples, the same prefix $(b,0)(a,0)^{k-1}$ resp. $(b,1)(a,0)^{k-1}$, and the same suffix $(a,0)^k$ of length k.

We conclude that either $u_{\hat{\jmath}+i(k+1)} \in K_{b0}$ for all $i \in \mathbb{N}$, or $u_{\hat{\jmath}+i(k+1)} \in K_{b1}$ for all $i \in \mathbb{N}$. In the first case, we have only finitely many occurrences of $(b,1)$ in α, whereas in the second case we have only finitely many letters $(b,0)$ in α. This contradicts that τ is a winning strategy for Player 2. □

While staying at the same level of the dot-depth hierarchy does not yield winning strategies for strong games, the final result shows that there are winning strategies at most two levels higher in the hierarchy. Whether winning strategies are also located in the level between remains open.

Theorem 8. *A game $L \in \mathrm{BC}(\lim(\mathrm{DD}_i))$ is determined with winning strategies in DD_{i+2}.*

Proof. We proceed as in the proof of Theorem 5. We first construct a graph where every node is a DD_i equivalence class – a $*$-language in DD_i. Now, for languages $K \in \mathrm{DD}_i$, we are given a game over the ω-language $L \in \mathrm{BC}(\lim(K))$. We obtain the game graph for this game when we equip the graph constructed above with a Muller winning condition. As explained in Section 2.1, we then transform this game graph via the LAR construction to a new game graph with a parity winning condition. A state in the new game graph is a pair (q,R) consisting of a state $q \in \Sigma^*/\!\sim_{\mathrm{DD}_i}$ and an LAR R. Over this party game graph, the winner has a positional winning strategy. To show that the winning strategy belongs to the class DD_{i+2}, it suffices to show that each node (q,R) corresponds to a DD_{i+2} language.

We proceed to apply the logical characterization of DD_i-languages (cf. Section 2.2). Each vertex q collects the play prefixes that belong to a language $K_q \in \mathrm{DD}_i$, defined by a Boolean combination ψ_q of Σ_i-sentences. So $\psi_q(x)$ expresses that the play prefix up to position x belongs to q, with $\psi_q(\max)$ qualifying the last position of the current play prefix. With the help of these formulae, we now express the fact that each play prefix leading to any state (q,R) in the parity game graph forms a language $K_{(q,R)} \in \mathrm{DD}_{i+2}$.

Given a permutation perm of the state space of the original Muller game, and an index h, an LAR can be defined as $R = (\mathrm{perm}, h)$. Let the sentence φ_R expresses the fact that a play prefix has lead to R, then it is evident that $K_{(q,R)} = \psi_q(\max) \wedge \varphi_R$. In order to avoid overloaded notation, we only provide a description for an example: the most recent prefix types in perm are q,r,s, in that order; the index value is $h = 3$. With φ_R, we express that the most recent prefix types are q,r,s in this order and that for the previous prefix this

sequence is r, s, q: (1) the current play prefix (at position max) is q, at the previous position is r, and any preceding position that is not occupied by r is occupied by s, and (2) for the play prefix at position $\max - 1$ the most recent play prefixes are in r, s, q in this order. This can be formally described as:

$$\begin{aligned}\varphi_R = \psi_q(\max) \wedge \psi_r(\max - 1) \\ \wedge \exists x, y, z\big(\max > x > y > z \wedge \psi_r(x) \wedge \psi_s(y) \wedge \psi_q(z) \\ \wedge \forall x'(\max > x' > x \to \psi_r(x')) \\ \wedge \forall y'(x > y' > y \to \psi_s(y'))\big)\end{aligned}$$

Since ψ_q, ψ_r, and ψ_s are Boolean combinations of Σ_i-sentences, we obtain (in prenex form) a Σ_{i+2}-sentence. Thus, we obtain languages $K_{(q,R)} \in \mathrm{DD}_{i+2}$. □

6 Conclusion

The present paper continues the study of a question that was raised already by Büchi and Landweber in their pioneering paper [1, Sect.3]: to analyze "how simple winning strategies do exist" for a given class of games. Complementing the results of [1, 8, 9] where solvability of regular and star-free infinite games was established with corresponding winning strategies (again regular and star-free strategies, respectively), we showed in this paper that for games of lower complexity three levels need to be distinguished.

1. When we take the basic (pattern-) languages K underlying the piecewise testable languages and the languages of dot-depth 1 and work with Boolean combinations of sets $\mathrm{ext}(K)$, then determinacy with piecewise testable winning strategies, respectively of dot-depth 1, holds.
2. Games with winning conditions in $\mathrm{BC}(\mathrm{ext}(\mathcal{K}))$, where $\mathcal{K}$ is now the full class of piecewise testable languages or languages of dot-depth 1, are determined only with winning strategies beyond $\mathcal{K}$, namely in DD_2.
3. This situation is no better when games in $\mathrm{BC}(\lim(\mathrm{DD}_1))$ are considered; there are winning strategies in DD_3 but not in DD_1. For $\mathrm{BC}(\lim(\mathrm{PT}))$ we fall back to case 1, and obtain winning strategies again in PT.

Finally, there remain some open problems. First, it is left open here whether the bound $i+2$ of Theorem 8 can be improved to $i+1$. A more general problem is to study complexity issues, e.g. how the sizes of automata for game presentations and strategy presentations can diverge. Finally, the results of this paper motivate setting up an abstract framework of passing from $*$-language classes to corresponding ω-language classes (as winning conditions of games) and back (by considering winning strategies), so that classes beyond the special cases of the present paper are covered as well.

References

1. J. R. Büchi and L. H. Landweber. Solving sequential conditions by finite-state strategies. *Transactions of the American Mathematical Society*, 138:295–311, 1969.

2. R. S. Cohen and J. A. Brzozowski. Dot-depth of star-free events. *Journal of Computer and System Sciences*, 5:1–16, February 1971.
3. E. Grädel, W. Thomas, and T. Wilke, editors. *Automata, Logics, and Infinite Games*, volume 2500 of *Lecture Notes in Computer Science*. Springer-Verlag, 2002.
4. J. E. Hopcroft, R. Motwani, and J. D. Ullman. *Introduction to automata theory, languages, and computation - (2. ed.)*. Addison-Wesley series in computer science. Addison-Wesley-Longman, 2001.
5. D. Perrin and J.-É. Pin. *Infinite Words*. Elsevier, Amsterdam, 2004.
6. J.-É. Pin. *Varieties of Formal Languages*. North Oxford, London and Plenum, New-York, 1986.
7. J.-É. Pin. Positive varieties and infinite words. In C. L. Lucchesi and A. V. Moura, editors, *LATIN*, volume 1380 of *Lecture Notes in Computer Science*, pages 76–87. Springer-Verlag, 1998.
8. A. Rabinovich and W. Thomas. Logical refinements of Church's problem. In J. Duparc and T. Henzinger, editors, *Computer Science Logic*, volume 4646 of *Lecture Notes in Computer Science*, pages 69–83. Springer-Verlag, 2007.
9. V. L. Selivanov. Fine hierarchy of regular aperiodic ω-languages. In *Proceedings of the 11th international conference on Developments in language theory*, DLT'07, pages 399–410, Berlin, Heidelberg, 2007. Springer-Verlag.
10. I. Simon. Piecewise testable events. In H. Brakhage, editor, *Automata Theory and Formal Languages*, volume 33 of *Lecture Notes in Computer Science*, pages 214–222. Springer-Verlag, 1975.
11. L. Staiger and K. W. Wagner. Automatentheoretische und automatenfreie Charakterisierungen topologischer Klassen regulärer Folgenmengen. *Elektronische Informationsverarbeitung und Kybernetik*, 10(7):379–392, 1974.
12. J. Stern. Characterizations of some classes of regular events. *Theoretical Computer Science*, 35:17–42, 1985.
13. H. Straubing. *Finite Automata, Formal Logic, and Circuit Complexity*. Birkhäuser Verlag, Basel, Switzerland, 1994.
14. W. Thomas. Classifying regular events in symbolic logic. *Journal of Computer and System Sciences*, 25(3):360–376, 1982.
15. W. Thomas. *Languages, automata, and logic*, pages 389–455. Springer-Verlag, New York, NY, USA, 1997.
16. W. Thomas. Church's problem and a tour through automata theory. In A. Avron, N. Dershowitz, and A. Rabinovich, editors, *Pillars of Computer Science*, volume 4800 of *Lecture Notes in Computer Science*, pages 635–655. Springer-Verlag, 2008.